Universidad de Valladolid

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

DEPARTAMENTO DE TEORÍA DE LA SEÑAL, COMUNICACIONES E INGENIERÍA TELEMÁTICA

TESIS DOCTORAL:

DESARROLLO Y EVALUACIÓN DE UNA APLICACIÓN WEB ESTANDARIZADA PARA EL ALMACENAMIENTO E INTERCAMBIO DE HISTORIALES CLÍNICOS ELECTRÓNICOS (HCEs) EN OFTALMOLOGÍA: TELEOFTALWEB

Presentada por Isabel de la Torre Díez para optar al grado de doctora por la Universidad de Valladolid

Dirigida por:

Francisco Javier Díaz Pernas

María Isabel López Gálvez

2010

Universidad de Valladolid

Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Escuela Técnica Superior de Ingenieros de Telecomunicación

TESIS DOCTORAL

DESARROLLO Y EVALUACIÓN DE UNA APLICACIÓN WEB ESTANDARIZADA PARA EL ALMACENAMIENTO E INTERCAMBIO DE HISTORIALES CLÍNICOS ELECTRÓNICOS (HCES) EN OFTALMOLOGÍA: TELEOFTALWEB

DOCTORANDO: ISABEL DE LA TORRE DÍEZ
DIRECTORES: FRANCISCO JAVIER DÍAZ PERNAS
MARÍA ISABEL LÓPEZ GÁLVEZ

TRIBUNAL

PRESIDENTE:	DR. D. MIGUEL LÓPEZ CORONADO
VOCALES:	DR. D. JOSE JOAQUÍN RIETA IBÁÑEZ
	DR. D. CARLOS DE CASTRO LOZANO
	DR. D. FRANCISCO GÓMEZ- ULLA DE IRAZAZÁBAL
SECRETARIO:	DR. D. ROBERTO HORNERO SÁNCHEZ
SUPLENTES:	DR. D. JESÚS MARÍA HERREROS GONZÁLEZ
	DR. D. RAFAEL MOMPÓ GÓMEZ

FECHA DE LECTURA:

CALIFICACIÓN:

A mis padres

Agradecimientos

En primer lugar, quiero agradecer a mis padres (Antonio y Flori) todo su apoyo, ánimos y ayuda durante todos estos años y siempre. Ellos siempre han estado a mi lado. Muchas gracias. Todo os lo debo a vosotros. A mi abuela Clara, que aunque ella ya no esté con nosotros, le haría mucha ilusión ver culminado este Trabajo.

A mis directores de Tesis (Paco y Maribel), a los Catedráticos de Teoría de la Señal, Miguel López Coronado y Roberto Hornero Sánchez, por su inestimable ayuda a la hora de realizar este Trabajo. Gracias por todo.

Al personal de la Unidad de Retinopatía Diabética del Instituto Universitario de Oftalmobiología Aplicada (IOBA) por colaborar estrechamente con nosotros.

A todas aquellas personas que me han acompañado durante estos años, entre ellos a Pablo por estar a mi lado este último año, muchas gracias a tod@s.

"Porque para alcanzar las metas es siempre necesario una mano amiga que te acompañe y guíe ... Porque aunque los inicios fueron duros ... ***el trabajo y el esfuerzo*** *al final tuvieron su* ***recompensa****. Gracias a tod@s."*

Resumen

Uno de los campos de aplicación de las telecomunicaciones más antiguo y al mismo tiempo con mayor potencial es el de la **Telemedicina**. La posibilidad de aumentar la capacidad de diagnóstico y curación de enfermedades a larga distancia con la ayuda de la tecnología ha atraído siempre a mucha gente, pero únicamente se ha podido mostrar de forma plena el potencial ofrecido una vez alcanzado el desarrollo actual de las comunicaciones de banda ancha y móviles. Así, se abre un gran abanico de posibilidades de uso de estos dispositivos en la Telemedicina, como, entre otros, el telediagnóstico, la teleconsulta, la teleasistencia, la telecirugía, la **Historia Clínica Electrónica (HCE)**, etc.

A pesar de los avances en los sistemas telemédicos actuales, surgen ciertas complicaciones en su implementación completa: la necesidad de interoperabilidad de dispositivos, el correcto manejo y procesado de la información generada por ellos es una necesidad crítica, que debe ser solventada de forma que el uso de tecnologías ya existentes puedan mantenerse pese a la actualización de los sistemas completos. Por ello, se requiere que la implementación final permita cierta escalabilidad, así como se debe dejar lugar a la evolución de dispositivos y formatos de información para poder vislumbrar una cierta garantía de éxito en esta interoperabilidad.

En lo relativo al tratamiento y almacenamiento de información, resulta obvio reconocer la necesidad de las **bases de datos (BDs)**. Además, de ser éstas el núcleo fundamental de sistemas como la **HCE.** En la actualidad, existen numerosas aplicaciones Web relativas a HCEs en diferentes especialidades médicas como pediatría, cardiología, urgencias, oftalmología, etc.

A lo largo de esta Tesis, se explicará con detalle el diseño, desarrollo, puesta en

marcha y posterior evaluación de la aplicación Web (**TeleOftalWeb**) en sus cuatro versiones, cuyo objetivo es el almacenamiento e intercambio de HCEs y retinografías en el campo de la oftalmología. Cada una de las versiones está desarrollada con diferentes modelos de BDs: dbXML 2.0 más MySQL 5.0, eXist 1.1.1 más MySQL, Xindice 1.2. más MySQL 5.0 y Oracle 10g. Para explicar las diferentes **BDs** se empleará un capítulo de esta Tesis.

Se estudiarán los **estándares de HCE** más relevantes como son: *Health Level 7* (HL7), *Digital Imaging Communications in Medicine* (DICOM), *International Organization for Standardization / Technical Committee* (ISO/TC) 215, *European Committee for Standardization / Technical Committee* (CEN/TC) 251; y las iniciativas: *Integrating the Healthcare Enterprise* (IHE), *Common Object Request Broker Architecture* (CORBA), *Good European Health Record* (GEHR) y OpenEHR.

Se analizarán las tecnologías empleadas para el desarrollo de **TeleOftalWeb**, tales como: *eXtensible Markup Language* (XML), Java, *Java Server Pages* (JSP), *Servlets*, y la programación de los módulos de **TeleOftalWeb.** Se explicará detalladamente su arquitectura y los diferentes modelados de datos de las distintas versiones de la misma en el Capítulo de Métodos.

Como resultados del presente trabajo se muestran los dos módulos de que consta **TeleOftalWeb**: módulo gestor y módulo de gestión de HCEs y retinografías, la experiencia realizada por los especialistas de la Unidad de Retinopatía Diabética del Instituto de Oftalmobiología Aplicada de la Universidad de Valladolid (IOBA), y los tiempos de respuesta obtenidos con las diferentes versiones de la aplicación (empleando modelos de colas vs. pruebas propias). El objetivo final del presente Trabajo será la implantación y posterior evaluación de la versión de **TeleOftalWeb** que emplee la/s BD/s, que genere/n un menor tiempo de respuesta a nuestra aplicación. Como bien sabemos, la implantación de todo sistema está condicionada por la rapidez en los tiempos de respuesta de su uso.

Mediante el empleo de **TeleOftalWeb** por parte de médicos especialistas de la Unidad de Retinopatía Diabética del IOBA se comprueba su correcto funcionamiento, con la inserción de más de 1000 HCs y más de 2000 retinografías. Su posterior evaluación constata la usabilidad de la misma. A su vez, las diferentes pruebas de

medición llevadas a cabo en las distintas versiones han permitido seleccionar el modelado de datos más eficiente, y de esta manera llevar a cabo la elección de la versión más efectiva para su empleo en la práctica médica diaria.

Abstract

One of the oldest fields of application of telecommunications and at the same time with the greatest potential is that of Telemedicine. The possibility of increasing the diagnostic capacity and the curing of illness at long distance with the help of technology has always attracted many people, but the full potential has only been wholly demonstrated with the development of broadband and mobile communications. In this way, a broad range of possibilities for the use of these devices in Telemedicine has opened up, such as, amongst other things, telediagnosis, teleconsultations, teleassistance, telesurgery, Electronic Health Record (EHR), etc.

Despite the advances in the current telemedicine systems, certain complications arise with full implementation: the need for interoperability of devices and the correct handling and processing of the information generated by them is an absolute necessity, which must be solved in such a way that the use of already existing technologies can be maintained, despite the updating of complete systems. To achieve this, the final implementation must allow for scaling up, whilst at the same time leaving room for the evolution of devices and information formats in order to come close to a certain guarantee of success in interoperability.

Regarding the treatment and storage of information, recognizing the need for databases is obvious. Additionally, these are the fundamental nucleus of systems such as EHR. Presently, there are numerous Web applications which exist pertaining to EHRs in different medical specializations, such as paediatrics, cardiology, emergencies, ophthalmology, etc.

Throughout this thesis, we will explain in detail the design, development, set up and post evaluation of the Web application (**TeleOftalWeb**) in its four different versions, the objective of which is the storage and exchange of EHRs and fundus

photographs in the field of ophthalmology. Each one of the versions is developed with different database models: dbXML 2.0, plus MySQL 5.0, eXist 1.1.1. plus MySQL, Xindice 1.2, plus MySQL 5.0 and Oracle 10g. A chapter in this thesis will be dedicated to explaining the different databases.

The most relevant standards of EHR will be studied, such as: Health Level 7 (HL7), Digital Imaging and Communications in Medicine (DICOM), International Organization for Standardization / Technical Committee (ISO/TC) 215, European Committee for Standardization / Technical Committee (CEN/TC) 251; and the initiatives: Integrating the Healthcare Enterprise (IHE), Common Object Request Broker Architecture (CORBA), Good European Health Record (GEHR) and OpenEHR.

The technologies used for the development of **TeleOftalWeb** will be analyzed, such as: eXtensible Markup Language (XML), Java, Java Server Pages (JSP), Servlets, and the programming of **TeleOftalWeb** modules. The architecture and the different data models of different versions of the same will be explained in detail in the chapter on Methods.

The results of the work currently being carried out highlight two modules of which **TeleOftalWeb** consists of: the administrator module and the module of administration of EHRs and fundus photographs, the experiment being carried out by the specialists of the Diabetic Retinopathy Unit of the Institute of Applied Ophthalmology of the University of Valladolid (IOBA), and the response times obtained with the different versions of the applications (using queuing models vs. benchmarking). The final objective of the current project is the putting in place and the post evaluation of the version of **TeleOftalWeb** used by the database(s), which generate the shortest response time to our application. As we well know, the implementation of any system is conditioned by the speed of reply time. With specialized doctors of the Diabetic Retinopathy Unit of the IOBA using **TeleOftalWeb** its correct operation can be verified, with the inclusion of more than 1000 records and more than 2000 fundus photographs. Post evaluation shows its usability. At the same time, the different measurements carried out in the various versions have allowed us to select the most efficient data modeling, which leads to making the choice of the most effective version in practical daily medical use.

Índice General

RESUMEN ____ I

ABSTRACT ____ IV

ÍNDICE GENERAL ____ VII

ÍNDICE DE FIGURAS ____ XI

ÍNDICE DE TABLAS ____ XVI

1. INTRODUCCIÓN ____ 1

1.1. ANTECEDENTES ____ 2

1.2. OBJETIVOS DE LA INVESTIGACIÓN ____ 3

1.3. METODOLOGÍA EMPLEADA ____ 4

1.4. DESCRIPCIÓN DE LA MEMORIA ____ 11

2. TELEMEDICINA. APLICACIONES. TELEOFTALMOLOGÍA ____ 15

2.1. TELEMEDICINA ____ 16

2.2. TELEOFTALMOLOGÍA ____ 18

2.2.1 INTRODUCCIÓN ____ 18

2.2.2. RETINOPATÍA DIABÉTICA Y TELEMEDICINA ____ 19

2.2.3 EXPERIENCIAS DE TELEOFTALMOLOGÍA Y RETINOPATÍA DIABÉTICA ____ 21

2.2.4 BIBLIOGRAFIA ____ 24

3. HISTORIALES CLÍNICOS ELECTRÓNICOS ____ 27

3.1. INTRODUCCIÓN ____ 28

3.2. EXPERIENCIAS DE HCE ____ 39

3.2.1. EXPERIENCIAS DE HCE EN ESPAÑA ____ 41

3.2.1.1. LA SANIDAD ELECTRÓNICA EN EL TERRITORIO ESPAÑOL ____ 42

3.2.1.2. EXPERIENCIAS DE HCE EN ANDALUCÍA ____ 43

3.2.1.3. EXPERIENCIAS DE HCE EN CASTILLA Y LEÓN ____ 47

3.2.1.4. EXPERIENCIAS DE HCE EN NAVARRA ____ 48

3.2.1.5. EXPERIENCIAS DE HCE EN EXTREMADURA ____ 49

3.2.1.6. EXPERIENCIAS DE HCE EN LA COMUNIDAD VALENCIANA ____ 50

3.2.1.7. EXPERIENCIAS DE HCE EN EL PAÍS VASCO ____ 50

3.2.2. EXPERIENCIAS EN OTROS PAÍSES ____ 51

3.2.2.1. EXPERIENCIAS DE HCE EN EL REINO UNIDO ____ 51

3.2.2.2. EXPERIENCIAS DE HCE EN IRLANDA ____ 52

3.2.2.3. EXPERIENCIAS DE HCE EN EE.UU ____ 53

3.3. PRIVACIDAD Y CONFIDENCIALIDAD DE DATOS ____ 54

EN ESPAÑA, LA REGULACIÓN LEGAL ES LA SIGUIENTE: ____ 56

4. ESTÁNDARES DE HISTORIALES CLÍNICOS ELECTRÓNICOS ____ 59

4.1. Introducción 61
4.2. Objetivos de los estándares de los historiales clínicos electrónicos 61
4.3. Algunos problemas de los sistemas de información sanitarios 62
4.4. Estándares y normas 62
4.5. Health Level 7 (HL7) 66
4.5.1. Introducción 66
4.5.2. Antecedentes de HL7 68
4.5.3. Objetivos y misión del HL7 69
4.5.4. Formato de HL7 70
4.5.5. Estándares 71
4.5.6. Modelo básico de transacciones HL7 71
4.5.7. Modelo funcional de HL7 72
4.5.8. Clinical Document Architecture (CDA) 75
4.5.8.1. Metas y Objetivos del CDA 76
4.5.8.2. Niveles del CDA 76
4.5.8.3. Identificadores de un CDA 79
4.5.8.4. Diseño de un documento CDA 80
4.5.8.4.1. CDA Header 81
4.5.8.4.2. CDA Level One Body 86
4.5.8.5. HL7 v3.0 Data Types 92
4.6. Digital Imaging and Communications in Medicine (DICOM) 93
4.6.1. Introducción general 93
4.6.2. Objetivos de DICOM 99
4.6.3. DICOM y otros estándares 100
4.6.4. Relación entre las partes del estándar DICOM 102
4.6.5. Partes del Estándar DICOM 105
4.6.6. Suplemento 91: Clases SOP de imágenes fotográficas oftalmológicas 111
4.7. Internacional Organization for Standardization (ISO)/TC 215 113
4.8. European Committee for Standardization / Technical Committee (CEN/TC) 251 114
4.9. Common Object Request Broker Architecture (CORBA) 119
4.10. Good European Health Record (GEHR) 121
4.10.1. Los requisitos del paciente 121
4.10.2. Descripción de la arquitectura de GEHR 122
4.11. Integrating the Healthcare Enterprise (IHE) 122
4.12. OpenEHR 124
4.13. Estándares de Seguridad en HCE 125

5. Bases de datos 129

5.1. Introducción 130
5.2. Conceptos básicos 131
5.3. ¿Por qué usar bases de datos? 135
5.4. Componentes principales de una base de datos 136
5.5. Ventajas en el uso de bases de datos y SGBD 138
5.6. Tipos de bases de datos 140
5.6.1. Según la variabilidad de los datos almacenados 140

5.6.2. Según el contenido *140*
5.7. Modelos de bases de datos 141
5.7.1. Bases de datos jerárquicas *141*
5.7.2. Bases de datos en red *142*
5.7.3. Bases de datos relacionales *142*
5.7.3.1. El modelo Entidad-Relación *143*
5.7.4. Bases de datos orientadas a objetos *145*
5.8. Bases de datos en telemedicina 146
5.9. Bases de datos de la aplicación TeleOftalWeb 147
5.9.1. MySQL *148*
5.9.2. Bases de datos nativas XML *152*
5.9.2.1. dbXML *154*
5.9.2.2. Exist *156*
5.9.2.3. Xindice *159*
5.9.3. Oracle *162*
5.9.3.1. Historia de Oracle Corporation *163*
5.9.3.2. Características generales del SGBD Oracle *165*
5.9.3.3. Oracle Database 10g *167*

6. Métodos 175

6.1. Introducción 176
6.2. Arquitectura de la aplicación 176
6.3. Tecnologías empleadas 177
6.3.1. Módulos de TeleOftalWeb empleando las BDs dbXML 2.0 y MySQL 5.0 *178*
6.3.2. Módulos de TeleOftalWeb empleando las BDs Xindice 1.2 y MySQL 5.0 *191*
6.3.3. Módulos de TeleOftalWeb empleando las BDs eXist 1.1.1 y MySQL 5.0 *198*
6.3.4. Módulos de TeleOftalWeb empleando Oracle 10g *205*
6.4. Modelado de datos 238
6.4.1. Base de datos MySQL *240*
6.4.2. Bases de datos nativas XML *244*
6.4.3. Base de datos Oracle 10g *252*
6.5. Modelos de colas 253
6.5.1. Modelado de la aplicación y estimaciones *257*
6.6. Programas de medición del tiempo de carga 262
6.6.1. Alternativas para realizar las pruebas *262*
6.6.2. La aplicación Firebug *263*

7. Resultados 265

7.1. Introducción 266
7.2. Módulo gestor 266
7.3. Módulo usuario 268
7.4. Experiencia y evaluación de la aplicación 273
7.5. Comparativa de diferentes versiones de la aplicación 278
7.5.1. Tiempos de respuesta de la aplicación empleando modelos de colas *278*
7.5.2. Tiempos de respuesta de la aplicación. Pruebas propias *280*

8. Discusión 337

8.1. Introducción 338
8.2. Aplicación TeleOftalWeb vs. otras aplicaciones de HCE 338
8.3. Tiempos de respuesta obtenidos 350

9. Conclusiones y líneas futuras 351

9.1. Introducción 352
9.2. Contribuciones originales 353
9.3. Líneas futuras de investigación 356

Apéndice A. Publicaciones a las que ha dado lugar la realización de la Tesis Doctoral 357

A.1. Revistas internacionales indexadas en el Journal Citation Report 358
A.2. Revistas nacionales 358
A.3. Capítulos de libro 359
A.4. Congresos internacionales 359
A.5. Congresos nacionales 360

Apéndice B. Manual de usuario 361

B.1. Modulo de administración de usuarios 362
B.2. Módulo de gestión de historiales 375

Apéndice C. Glosario de abreviaturas y acrónimos 413

Bibliografía 425

Índice de Figuras

FIGURA 1. DIAGRAMA DE FLUJO DE LA METODOLOGÍA DE DISEÑO EN LAS TRES PRIMERAS VERSIONES DE LA APLICACIÓN. FUENTE: PROPIA. 6
FIGURA 2. DIAGRAMA DE FUNCIONAMIENTO SIMPLIFICADO DE LA APLICACIÓN. FUENTE: PROPIA. 9
FIGURA 3. DIAGRAMA DE FUNCIONAMIENTO SIMPLIFICADO DEL MÓDULO DE ADMINISTRACIÓN. FUENTE: PROPIA. 10
FIGURA 4. (A) FONDO NORMAL, (B) FONDO CON LESIONES DE RD. 20
FIGURA 5. RETINÓGRAFO. 20
FIGURA 6. MODELO DE HC EMPLEADO POR LA UNIDAD DE RETINOPATÍA DIABÉTICA DEL IOBA (A) ANAMNESIS, (B) Y (C) EXPLORACIÓN. FUENTE: PROPIA. 31
FIGURA 7. LA SANIDAD EN LA SOCIEDAD DE LA INFORMACIÓN (SEIS 2003). 35
FIGURA 8. SISTEMA DE INFORMACIÓN SANITARIO (JUNTA ANDALUCÍA 2008). 45
FIGURA 9. EVOLUCIÓN TÍPICA DEL CICLO DE VIDA DE UN ESTÁNDAR DESDE SU PROPUESTA HASTA SU ESTABLECIMIENTO CONSOLIDADO (MONTEAGUDO 2003). 65
FIGURA 10. INTEROPERABILIDAD CON HL7 (MONTEAGUDO 2003). 67
FIGURA 11. ARQUITECTURA ISO-OSI (HL7 2008). 69
FIGURA 12. MODELO BÁSICO DE TRANSACCIONES HL7 (ANSI/HL7 2007). 71
FIGURA 13. MODELO FUNCIONAL DE HL7: RESUMEN FUNCIONAL (ANSI/HL7 2007). 73
FIGURA 14. MODELO FUNCIONAL DE HL7: SUBDIVISIONES DEL RESUMEN FUNCIONAL (ANSI/HL7 2007). 73
FIGURA 15. ELEMENTOS DE LAS FUNCIONES EN EL MODELO FUNCIONAL DE HL7 (ANSI/HL7 2007). 74
FIGURA 16. CDA NIVEL 1 (DOLIN *ET ÁL.* 2000). 77
FIGURA 17. CDA NIVEL 2 (DOLIN *ET ÁL.* 2000). 78
FIGURA 18. CDA NIVEL 3 (DOLIN *ET ÁL.* 2000). 78
FIGURA 19. ESTRUCTURA DE UN DOCUMENTO CDA (DOLIN *ET ÁL.* 2000). 79
FIGURA 20. ESQUEMA DE LA CABECERA CDA (TREINS *ET ÁL.* 2006). 82
FIGURA 21. ESQUEMA DE LOS DOCUMENTOS CDA (TREINS *ET ÁL.* 2006). 87
FIGURA 22. CUERPO DEL HISTORIAL XML. FUENTE: PROPIA. 89
FIGURA 23. REVISIONES DENTRO DE UN HISTORIAL XML. FUENTE: PROPIA. 89
FIGURA 24. REVISIÓN DE UN HISTORIAL XML. FUENTE: PROPIA. 90
FIGURA 25. PARTE DE UNA REVISIÓN DE UN HISTORIAL XML. FUENTE: PROPIA. 91
FIGURA 26. CÓDIGO EN EL QUE LA IMAGEN SAGITAL NO HA SIDO MODIFICADA. FUENTE: PROPIA. 91
FIGURA 27. CÓDIGO EN EL QUE LA IMAGEN FRONTAL HA SIDO MODIFICADA. FUENTE: PROPIA. 91
FIGURA 28. RELACIÓN MÓDULOS IMÁGENES (OTECH 2008). 96
FIGURA 29. PROPUESTAS DE ESTÁNDAR (OTECH 2008). 101
FIGURA 30. RELACIÓN ENTRE PARTES DEL ESTÁNDAR DICOM (OTECH 2008). 103
FIGURA 31. NIVELES DE LA ARQUITECTURA DE UN SGBD (CABRERA 2001). 133
FIGURA 32. DIAGRAMA DE FLUJO DE LA INFORMACIÓN AL UTILIZAR UNA BD (BURBANO 2006). 135
FIGURA 33. DIAGRAMA DE INTERACCIÓN DEL DBA CON LOS USUARIOS (BURBANO 2006). 137
FIGURA 34. EJEMPLO DE ESTRUCTURA JERÁRQUICA DE UNA BD. FUENTE: PROPIA. 141
FIGURA 35. EJEMPLO DE MODELO RELACIONAL DE UN SISTEMA DE INFORMACIÓN HOSPITALARIO. FUENTE: PROPIA. 143
FIGURA 36. EJEMPLO DE DIAGRAMA E-R. FUENTE: PROPIA. 144
FIGURA 37. LOGO DE MYSQL (MYSQL 2008). 148
FIGURA 38. LOGO DE DBXML (DBXML 2008). 155
FIGURA 39. VENTANA IDENTIFICACIÓN USUARIO, EXIST. FUENTE: PROPIA. 157
FIGURA 40. ADMINISTRADOR DE LA BASE DE DATOS EXIST. FUENTE: PROPIA. 158
FIGURA 41. ESTABLECIMIENTO DE PERMISOS, EXIST. FUENTE: PROPIA. 158
FIGURA 42. INICIALIZACIÓN DE LA BASE DE DATOS EXIST. FUENTE: PROPIA. 159

FIGURA 43. LOGO DE EXIST (EXIST 2008). 159
FIGURA 44. INTERFAZ DE LA BASE DE DATOS XINDICE, COLECCIONES. FUENTE: PROPIA 161
FIGURA 45. INTERFAZ DE LA BASE DE DATOS XINDICE, DOCUMENTOS. FUENTE: PROPIA. 162
FIGURA 46. LOGO DE XINDICE (XINDICE 2008). 162
FIGURA 47. LOGO DE ORACLE (ORACLE 2008). 162
FIGURA 48. LOGO DE ORACLE DATABASE 10G (ORACLE 2008). 168
FIGURA 49. INTERFAZ DE ADMINISTRACIÓN BASADO EN WEB DE LA BD. FUENTE: PROPIA 172
FIGURA 50. ARQUITECTURA DE LA APLICACIÓN TELEOFTALWEB CON LA BDS NATIVAS XML. FUENTE: PROPIA. 176
FIGURA 51. ARQUITECTURA DE LA APLICACIÓN TELEOFTALWEB CON LA BD ORACLE 10G. FUENTE: PROPIA. 177
FIGURA 52.ESQUEMA DE COMUNICACIÓN ENTRE MÓDULOS. FUENTE: PROPIA. 179
FIGURA 53. CLASE MYSQL. FUENTE: PROPIA. 180
FIGURA 54. CLASE DBXML. FUENTE: PROPIA. 182
FIGURA 55. CLASE PARSEXML. FUENTE: PROPIA. 187
FIGURA 56. ESTRUCTURA XML PÁRRAFO. FUENTE: PROPIA. 188
FIGURA 57. ESTRUCTURA XML MÚLTIPLES PÁRRAFOS. FUENTE: PROPIA. 188
FIGURA 58. ESTRUCTURA XML MÚLTIPLES PÁRRAFOS (SUBSECCIONES). FUENTE: PROPIA. 189
FIGURA 59. ESTRUCTURA XML ESQUEMAS DE OJO. FUENTE: PROPIA. 189
FIGURA 60. CLASE BDXML PARA XINDICE. FUENTE: PROPIA. 194
FIGURA 61. CLASE BDXML PARA EXIST. FUENTE: PROPIA. 202
FIGURA 62. PANTALLA DE LA HERRAMIENTA (PASO I). FUENTE: PROPIA. 206
FIGURA 63. PANTALLA DE LA HERRAMIENTA (PASO II-1). FUENTE: PROPIA. 207
FIGURA 64. PANTALLA DE LA HERRAMIENTA (PASO II-2). FUENTE: PROPIA. 208
FIGURA 65. PANTALLA DE LA HERRAMIENTA (PASO II-3). FUENTE: PROPIA. 209
FIGURA 66. PANTALLA DE LA HERRAMIENTA (PASO II-4). FUENTE: PROPIA. 213
FIGURA 67. PANTALLA DE LA HERRAMIENTA (PASO II-5). FUENTE: PROPIA. 214
FIGURA 68. PÁGINA QUE MUESTRA ALGUNAS FUNCIONES DE ORACLE BAJO EL ESQUEMA SYS. FUENTE: PROPIA. 232
FIGURA 69. TABLA *IMAGENES*. FUENTE: PROPIA. 235
FIGURA 70. ESQUEMA DE COMUNICACIÓN DE LA APLICACIÓN EMPLEANDO LA BD DBXML 2.0. FUENTE: PROPIA. 240
FIGURA 71. MODELO DE DATOS: TABLA DE USUARIOS. FUENTE: PROPIA. 243
FIGURA 72. MODELO DE DATOS: TABLA DE PERMISOS. FUENTE: PROPIA. 243
FIGURA 73. MODELO DE DATOS: RELACIÓN ENTRE AMBAS TABLAS. FUENTE: PROPIA. 244
FIGURA 74. INTERFAZ DE LA BASE DE DATOS DBXML. FUENTE: PROPIA. 245
FIGURA 75. CAMPOS DE LA CABECERA CDA. FUENTE: PROPIA. 246
FIGURA 76. CAMPO DE LA CABECERA CDA. FUENTE: PROPIA. 247
FIGURA 77. CAMPOS DE LA CABECERA CDA (*LOCAL_HEADER*). FUENTE: PROPIA. 248
FIGURA 78. ESTRUCTURA DEL CUERPO DEL HISTORIAL XML. FUENTE: PROPIA. 249
FIGURA 79. ESQUEMA DE REVISIONES DENTRO DE UN HISTORIAL XML. FUENTE: PROPIA. 249
FIGURA 80. ESTRUCTURA DE UNA REVISIÓN DE UN HISTORIAL XML. FUENTE: PROPIA. 250
FIGURA 81. ESTRUCTURA XML DE UN DATO DEL HISTORIAL. FUENTE: PROPIA. 251
FIGURA 82. ESTRUCTURA XML DE UNA IMAGEN DE UN HISTORIAL. FUENTE: PROPIA. 251
FIGURA 83. MODELADO DE DATOS EN ORACLE 10G. FUENTE: PROPIA. 253
FIGURA 84. MODELADO DE LA ARQUITECTURA DE TELEOFTALWEB. FUENTE: PROPIA. 258
FIGURA 85. MODELO DE COLAS DE TELEOFTALWEB. FUENTE: PROPIA. 259
FIGURA 86. INTERFAZ FIREBUG, CARGA DE PÁGINAS. FUENTE: PROPIA. 263
FIGURA 87. INTERFAZ FIREBUG, TIEMPOS DE CARGA. FUENTE: PROPIA. 264
FIGURA 88. PANTALLA INICIAL DEL MÓDULO GESTOR DE LA APLICACIÓN. FUENTE: PROPIA. 267
FIGURA 89. USUARIOS DE LA APLICACIÓN TELEOFTALWEB. FUENTE: PROPIA. 268
FIGURA 90. PANTALLA INICIAL DEL MÓDULO USUARIO DE LA APLICACIÓN. FUENTE: PROPIA. 269
FIGURA 91. PANTALLA DE BIENVENIDA DE LA APLICACIÓN. FUENTE: PROPIA. 270
FIGURA 92. HISTORIALES DE UN MÉDICO QUE EMPLEA LA APLICACIÓN. FUENTE: PROPIA. 271
FIGURA 93. EDITOR DE IMAGEN. FUENTE: PROPIA. 272

FIGURA 94. BUSCADOR DE IMÁGENES DE LA APLICACIÓN. FUENTE: PROPIA. 273
FIGURA 95. HISTORIAL DE UN PACIENTE EN FORMATO PDF. FUENTE: PROPIA. 274
FIGURA 96. HISTORIAL EN PDF CON IMÁGENES DICOM. FUENTE: PROPIA. 275
FIGURA 97. ENCUESTA DE SATISFACCIÓN DE LA APLICACIÓN. FUENTE: PROPIA. 276
FIGURA 98. TIEMPOS DE RESPUESTA PARA 1 MB CON DIFERENTES MODELOS DE BDS EN MS. FUENTE: PROPIA. 280
FIGURA 99. TIEMPOS DE RESPUESTA PARA 100 KB CON DIFERENTES MODELOS DE BDS. EN MS. FUENTE: PROPIA. 280
FIGURA 100. TIEMPO MEDIO DE CARGA DE LA PÁGINA PRINCIPAL EN IE. FUENTE: PROPIA. 307
FIGURA 101. TIEMPO MEDIO DE CARGA DE LA PÁGINA PRINCIPAL EN MF. FUENTE: PROPIA. 307
FIGURA 102. TIEMPO MEDIO DE CARGA DE LA PÁGINA PRINCIPAL EN IE CON 5 USUARIOS. FUENTE: PROPIA. 308
FIGURA 103. TIEMPO MEDIO DE CARGA DE LA PÁGINA PRINCIPAL EN IE CON 10 USUARIOS. FUENTE: PROPIA. 308
FIGURA 104. TIEMPO MEDIO DE CARGA DE LA PÁGINA PRINCIPAL EN IE CON 20 USUARIOS. FUENTE: PROPIA. 309
FIGURA 105. TIEMPO MEDIO DE CARGA DE LA PÁGINA PRINCIPAL EN IE DENTRO DE LA RED. FUENTE: PROPIA. 309
FIGURA 106. TIEMPO MEDIO DE CARGA DE LA PÁGINA PRINCIPAL EN IE FUERA DE LA RED. FUENTE: PROPIA. 310
FIGURA 107. TIEMPO MEDIO DE CARGA EN BUSCAR HCE EN IE. FUENTE: PROPIA. 310
FIGURA 108. TIEMPO MEDIO DE CARGA EN BUSCAR HCE EN MF. FUENTE: PROPIA. 311
FIGURA 109. TIEMPO MEDIO DE CARGA EN BUSCAR HCE EN IE CON 5 USUARIOS. FUENTE: PROPIA. 311
FIGURA 110. TIEMPO MEDIO DE CARGA EN BUSCAR HCE EN IE CON 10 USUARIOS. FUENTE: PROPIA. 312
FIGURA 111. TIEMPO MEDIO DE CARGA EN BUSCAR HCE EN IE CON 20 USUARIOS. FUENTE: PROPIA. 312
FIGURA 112. TIEMPO MEDIO DE CARGA EN BUSCAR HCE DENTRO DE LA RED. FUENTE: PROPIA. 313
FIGURA 113. TIEMPO MEDIO DE CARGA EN BUSCAR HCE FUERA DE LA RED. FUENTE: PROPIA. 313
FIGURA 114. TIEMPO MEDIO EN INTRODUCIR 1 HCE CON IMÁGENES EN IE. FUENTE: PROPIA. 314
FIGURA 115. TIEMPO MEDIO EN INTRODUCIR 1 HCE CON IMÁGENES EN MF. FUENTE: PROPIA. 314
FIGURA 116. TIEMPO MEDIO EN INTRODUCIR 1 HCE CON IMÁGENES EN IE CON 5 USUARIOS. FUENTE: PROPIA. 315
FIGURA 117. TIEMPO MEDIO EN INTRODUCIR 1 HCE CON IMÁGENES EN IE CON 10 USUARIOS. FUENTE: PROPIA. 315
FIGURA 118. TIEMPO MEDIO EN INTRODUCIR 1 HCE CON IMÁGENES EN IE CON 20 USUARIOS. FUENTE: PROPIA. 316
FIGURA 119. TIEMPO MEDIO EN INTRODUCIR 1 HCE CON IMÁGENES EN IE DENTRO DE LA RED. FUENTE: PROPIA. 316
FIGURA 120. TIEMPO MEDIO EN INTRODUCIR 1 HCE CON IMÁGENES EN IE FUERA DE LA RED. FUENTE: PROPIA. 317
FIGURA 121. TIEMPO MEDIO EN AÑADIR 1 IMAGEN JPEG EN IE. FUENTE: PROPIA. 317
FIGURA 122. TIEMPO MEDIO EN AÑADIR 1 IMAGEN JPEG EN MF. FUENTE: PROPIA. 318
FIGURA 123. TIEMPO MEDIO EN AÑADIR 1 IMAGEN JPEG EN IE CON 5 USUARIOS. FUENTE: PROPIA. 318
FIGURA 124. TIEMPO MEDIO EN AÑADIR 1 IMAGEN JPEG EN IE CON 10 USUARIOS. FUENTE: PROPIA. 319
FIGURA 125. TIEMPO MEDIO EN AÑADIR 1 IMAGEN JPEG EN IE CON 20 USUARIOS. FUENTE: PROPIA. 319
FIGURA 126. TIEMPO MEDIO EN AÑADIR 1 IMAGEN JPEG EN IE DENTRO DE LA RED. FUENTE: PROPIA. 320
FIGURA 127. TIEMPO MEDIO EN AÑADIR 1 IMAGEN JPEG EN IE FUERA DE LA RED. FUENTE: PROPIA. 320
FIGURA 128. TIEMPO MEDIO EN AÑADIR 1 IMAGEN JPEG EN IE. FUENTE: PROPIA. 321
FIGURA 129. TIEMPO MEDIO EN AÑADIR 1 IMAGEN DICOM EN MF. FUENTE: PROPIA. 321
FIGURA 130. TIEMPO MEDIO EN AÑADIR 1 IMAGEN DICOM EN IE CON 5 USUARIOS. FUENTE: PROPIA. 322
FIGURA 131. TIEMPO MEDIO EN AÑADIR 1 IMAGEN DICOM EN IE CON 10 USUARIOS. FUENTE: PROPIA. 322
FIGURA 132. TIEMPO MEDIO EN AÑADIR 1 IMAGEN DICOM EN IE CON 20 USUARIOS. FUENTE: PROPIA. 323
FIGURA 133. TIEMPO MEDIO EN AÑADIR 1 IMAGEN DICOM EN IE DENTRO DE LA RED. FUENTE: PROPIA. 323
FIGURA 134. TIEMPO MEDIO EN AÑADIR 1 IMAGEN DICOM EN IE FUERA DE LA RED. FUENTE: PROPIA. 324
FIGURA 135. TIEMPO MEDIO DE CARGA EN BUSCAR IMÁGENES EN IE. FUENTE: PROPIA. 324
FIGURA 136. TIEMPO MEDIO DE CARGA EN BUSCAR IMÁGENES EN MF. FUENTE: PROPIA. 325
FIGURA 137. TIEMPO MEDIO DE CARGA EN BUSCAR IMÁGENES EN IE CON 5 USUARIOS. FUENTE: PROPIA. 325
FIGURA 138. TIEMPO MEDIO DE CARGA EN BUSCAR IMÁGENES EN IE CON 10 USUARIOS. FUENTE: PROPIA. 326
FIGURA 139. TIEMPO MEDIO DE CARGA EN BUSCAR IMÁGENES EN IE CON 20 USUARIOS. FUENTE: PROPIA. 326
FIGURA 140. TIEMPO MEDIO DE CARGA EN BUSCAR IMÁGENES EN IE DENTRO DE LA RED. FUENTE: PROPIA. 327
FIGURA 141. TIEMPO MEDIO DE CARGA EN BUSCAR IMÁGENES EN IE FUERA DE LA RED. FUENTE: PROPIA. 327
FIGURA 142. TIEMPO MEDIO EN CREAR 1 HCE EN PDF EN IE. FUENTE: PROPIA. 328
FIGURA 143. TIEMPO MEDIO EN CREAR 1 HCE EN PDF EN MF. FUENTE: PROPIA. 328
FIGURA 144. TIEMPO MEDIO EN CREAR 1 HCE EN PDF EN IE CON 5 USUARIOS. FUENTE: PROPIA. 329

FIGURA 145. TIEMPO MEDIO EN CREAR 1 HCE EN PDF EN IE CON 10 USUARIOS. FUENTE: PROPIA. 329
FIGURA 146. TIEMPO MEDIO EN CREAR 1 HCE EN PDF EN IE CON 20 USUARIOS. FUENTE: PROPIA. 330
FIGURA 147. TIEMPO MEDIO EN CREAR 1 HCE EN PDF EN IE DENTRO DE LA RED. FUENTE: PROPIA. 330
FIGURA 148. TIEMPO MEDIO EN CREAR 1 HCE EN PDF EN IE FUERA DE LA RED. FUENTE: PROPIA. 331
FIGURA 149. TIEMPO MEDIO EN BORRAR 1 HCE EN IE. FUENTE: PROPIA. 331
FIGURA 150. TIEMPO MEDIO EN BORRAR 1 HCE EN MF. FUENTE: PROPIA. 332
FIGURA 151. TIEMPO MEDIO EN BORRAR 1 HCE EN IE CON 5 USUARIOS. FUENTE: PROPIA. 332
FIGURA 152. TIEMPO MEDIO EN BORRAR 1 HCE EN IE CON 10 USUARIOS. FUENTE: PROPIA. 333
FIGURA 153. TIEMPO MEDIO EN BORRAR 1 HCE EN IE CON 20 USUARIOS. FUENTE: PROPIA. 333
FIGURA 154. TIEMPO MEDIO EN BORRAR 1 HCE EN IE DENTRO DE LA RED. FUENTE: PROPIA. 334
FIGURA 155. TIEMPO MEDIO EN BORRAR 1 HCE EN IE FUERA DE LA RED. FUENTE: PROPIA. 334
FIGURA 156. INTERFAZ DEL SISTEMA YKONOS (WITSA GLOBAL ICT EXCELLENCE AWARDS 2009). 345
FIGURA 157. INTERFAZ DE LA HERRAMIENTA OFTALCLINIC (DISA S.L. 2009). 347
FIGURA 158. PANTALLA DE ACCESO AL MÓDULO DE ADMINISTRACIÓN. FUENTE: PROPIA. 363
FIGURA 159. PANTALLA INICIAL DE LA HERRAMIENTA DE ADMINISTRACIÓN. FUENTE: PROPIA. 364
FIGURA 160. LISTADO COMPLETO DE USUARIOS. FUENTE: PROPIA. 366
FIGURA 161. BÚSQUEDA DE USUARIOS POR APELLIDOS Y TIPO DE USUARIO. FUENTE: PROPIA. 366
FIGURA 162. ABRIR DATOS DE USUARIO. FUENTE: PROPIA. 367
FIGURA 163. CONFIRMACIÓN DE ELIMINACIÓN DE USUARIO. FUENTE: PROPIA. 368
FIGURA 164. MODIFICACIÓN DE DATOS DE USUARIO. FUENTE: PROPIA. 369
FIGURA 165. VENTANA DE CREACIÓN DE USUARIOS. FUENTE: PROPIA. 370
FIGURA 166. LISTADO DE HISTORIALES DE UN USUARIO. FUENTE: PROPIA. 372
FIGURA 167. VENTANA DE CONFIRMACIÓN PARA ELIMINAR UN HISTORIAL. FUENTE: PROPIA. 372
FIGURA 168. LISTADO DE PARTICIPANTES DE UN HISTORIAL. FUENTE: PROPIA. 373
FIGURA 169. ESTADÍSTICAS DE USUARIO. FUENTE: PROPIA. 374
FIGURA 170. VENTANA DE ACCESO A LA HERRAMIENTA TELEOFTALWEB. FUENTE: PROPIA. 376
FIGURA 171. VENTANA DE AUTENTIFICACIÓN. FUENTE: PROPIA. 376
FIGURA 172. VENTANA INICIAL DE PRESENTACIÓN DE ESTADÍSTICAS. FUENTE: PROPIA. 377
FIGURA 173. VENTANA CON LOS DATOS DE USUARIO. FUENTE: PROPIA. 378
FIGURA 174. VENTANA CON LOS DATOS DE FILIACIÓN DE LA REVISIÓN DE UN PACIENTE. FUENTE: PROPIA. 379
FIGURA 175. VENTANA CON EL MOTIVO DE CONSULTA DE UNA REVISIÓN DE UN PACIENTE. FUENTE: PROPIA. 379
FIGURA 176. VENTANA CON LOS ANTECEDENTES PERSONALES OFTALMOLÓGICOS DE LA REVISIÓN DE UN PACIENTE. FUENTE: PROPIA. 380
FIGURA 177. VENTANA CON LA PRIMERA PARTE DE LOS DATOS DE LOS ANTECEDENTES PERSONALES GENERALES DE LA REVISIÓN DE UN PACIENTE. FUENTE: PROPIA. 380
FIGURA 178. VENTANA CON LA SEGUNDA PARTE DE LOS ANTECEDENTES PERSONALES DE LA REVISIÓN DE UN PACIENTE. FUENTE: PROPIA. 381
FIGURA 179. VENTANA CON LA TERCERA PARTE DE LOS ANTECEDENTES PERSONALES GENERALES DE LA REVISIÓN DE UN PACIENTE. FUENTE: PROPIA. 381
FIGURA 180. VENTANA CON LA TERCERA PARTE DE LOS ANTECEDENTES PERSONALES GENERALES DE LA REVISIÓN DE UN PACIENTE. FUENTE: PROPIA. 382
FIGURA 181. VENTANA CON LOS ANTECEDENTES FAMILIARES DE LA REVISIÓN DE UN PACIENTE. FUENTE: PROPIA. 382
FIGURA 182. VENTANA DE INDICACIÓN DE QUE LOS DATOS DE FILIACIÓN Y LA ANAMNESIS HAN SIDO GRABADOS EN LA BASE DE DATOS DE FORMA CORRECTA. FUENTE: PROPIA. 383
FIGURA 183. VENTANA CON LA PRIMERA PARTE DE LA EXPLORACIÓN DE UNA REVISIÓN DEL PACIENTE. FUENTE: PROPIA. 384
FIGURA 184. VENTANA CON LA SEGUNDA PARTE DE LA EXPLORACIÓN. FUENTE: PROPIA. 384
FIGURA 185. VENTANA DE EDICIÓN DE LOS ESQUEMAS DE OJO DE UNA REVISIÓN DEL PACIENTE. FUENTE: PROPIA. 385
FIGURA 186. VENTANA CON LA TERCERA PARTE DE LA EXPLORACIÓN EN LA REVISIÓN DE UN PACIENTE. FUENTE: PROPIA. 385
FIGURA 187. MENSAJE DE INDICACIÓN ACERCA DE QUE LOS DATOS DE EXPLORACIÓN HAN SIDO GRABADOS EN LA BASE DE DATOS DE FORMA CORRECTA. FUENTE: PROPIA. 386
FIGURA 188. VENTANA CON LOS DATOS DE LAS PRUEBAS ADICIONALES REALIZADAS. FUENTE: PROPIA. 387

FIGURA 189. VENTANA DE DIAGNÓSTICOS TRAS INSERTAR UN NUEVO DIAGNÓSTICO. FUENTE: PROPIA. 387
FIGURA 190. VENTANA CON EL TRATAMIENTO Y LAS RECOMENDACIONES PERTINENTES. FUENTE: PROPIA. 388
FIGURA 191. VENTANA DE BÚSQUEDA DE UN HISTORIAL. FUENTE: PROPIA. 389
FIGURA 192. VENTANA CON LAS REVISIONES DE UN HISTORIAL. FUENTE: PROPIA. 390
FIGURA 193. VENTANA CON LOS DATOS PERSONALES DE UNA REVISIÓN DE UN HISTORIAL PROPIO. FUENTE: PROPIA. 391
FIGURA 194. VENTANA CON LOS DATOS PERSONALES DE UNA REVISIÓN DE UN HISTORIAL COMPARTIDO. FUENTE: PROPIA. 391
FIGURA 195. VENTANA CON LOS PARTICIPANTES DE UN HISTORIAL. FUENTE: PROPIA. 392
FIGURA 196. ABRIR HISTORIAL DEL PACIENTE. FUENTE: PROPIA. 393
FIGURA 197. ACCEDER A LA HERRAMIENTA DE IMPRESIÓN. FUENTE: PROPIA. 394
FIGURA 198. VENTANA PARA LA CREACIÓN DEL PDF. FUENTE: PROPIA. 395
FIGURA 199. CREACIÓN DEL PDF EN ESPAÑOL. FUENTE: PROPIA. 396
FIGURA 200. CREACIÓN DEL PDF EN INGLÉS. FUENTE: PROPIA. 396
FIGURA 201. MENSAJE DE ALERTA AL NO PODER ABRIR EL PDF. FUENTE: PROPIA. 397
FIGURA 202. MENSAJE DE ALERTA. FUENTE: PROPIA. 397
FIGURA 203. PARTE DE UN DOCUMENTO PDF EN ESPAÑOL. FUENTE: PROPIA. 398
FIGURA 204. PARTE DE UN DOCUMENTO PDF EN INGLÉS. FUENTE: PROPIA. 399
FIGURA 205. PARTES DE LA SECCIÓN AÑADIR IMÁGENES A EL DOCUMENTO PDF. FUENTE: PROPIA. 400
FIGURA 206. NO HAY IMÁGENES DISPONIBLES PARA AÑADIR AL PDF. FUENTE: PROPIA. 401
FIGURA 207. SELECCIÓN DE IMÁGENES PARA EL HISTORIAL. FUENTE: PROPIA. 402
FIGURA 208. IMÁGENES SELECCIONADAS PARA LA IMPRESIÓN. FUENTE: PROPIA. 402
FIGURA 209. IMÁGENES QUE APARECEN EN EL PDF. FUENTE: PROPIA. 403
FIGURA 210. SELECCIÓN DE IMAGEN. FUENTE: PROPIA. 404
FIGURA 211. IMAGEN ORIGINAL. FUENTE: PROPIA. 404
FIGURA 212. IMAGEN MODIFICADA. FUENTE: PROPIA. 405
FIGURA 213. PRIMERA PÁGINA DE IMÁGENES. FUENTE: PROPIA. 406
FIGURA 214. SEGUNDA PÁGINA DE IMÁGENES. FUENTE: PROPIA. 406
FIGURA 215. VENTANA PARA AÑADIR IMÁGENES AL HISTORIAL DE UN PACIENTE. FUENTE: PROPIA. 407
FIGURA 216. PANTALLA CON UNA IMAGEN DEL HISTORIAL AMPLIADA. FUENTE: PROPIA. 408
FIGURA 217. SELECCIÓN DE COLOR. FUENTE: PROPIA. 409
FIGURA 218. VENTANA CON LOS RESULTADOS DE UNA BÚSQUEDA DE LAS IMÁGENES DE UN PACIENTE. FUENTE: PROPIA. FUENTE: PROPIA. 411

Índice de Tablas

TABLA 1. SOLUCIONES DE BASES DE DATOS XML. FUENTE: PROPIA. 154
TABLA 2. LA INFORMACIÓN EN LA HC. FUENTE: PROPIA. 38
TABLA 3. ANÁLISIS DAFO DE HCE. FUENTE: PROPIA. 39
TABLA 4. ESTÁNDARES DE TELEMEDICINA (MONTEAGUADO 2003). 64
TABLA 5. MODALIDADES DE IMAGEN Y EN PACS. FUENTE: PROPIA. 123
TABLA 6. OBJETIVOS Y MEDIDAS DE SEGURIDAD (GARBAYO ET ÁL. 2003). 126
TABLA 7. COLAS DEL SISTEMA VS. TIEMPOS DE SERVICIO. FUENTE: PROPIA. 259
TABLA 8. TIEMPO MEDIO DE CONSULTA CON DIFERENTES BDS. FUENTE: PROPIA. 260
TABLA 9. RESULTADOS DE LA ENCUESTA DE USABILIDAD DE LA APLICACIÓN. FUENTE: PROPIA. 277
TABLA 10. TIEMPOS DE RESPUESTA CON LAS DIFERENTES BDS. FUENTE: PROPIA. 278
TABLA 11. COMPARATIVA DE BASES DE DATOS NATIVAS XML CON ORACLE (ZHANG 2003). 281
TABLA 12. TIEMPO MEDIO DE CARGA DE LA PÁGINA PRINCIPAL DESDE UN ORDENADOR DE LA RED ETSIT Y 5 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 282
TABLA 13. TIEMPO MEDIO DE CARGA DE LA PÁGINA PRINCIPAL DESDE UN ORDENADOR DE LA RED ETSIT Y 10 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 283
TABLA 14. TIEMPO MEDIO DE CARGA DE LA PÁGINA PRINCIPAL DESDE UN ORDENADOR DE LA RED ETSIT Y 20 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 284
TABLA 15. TIEMPO MEDIO DE CARGA DE LA PÁGINA PRINCIPAL DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 5 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 284
TABLA 16. TIEMPO MEDIO DE CARGA DE LA PÁGINA PRINCIPAL DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 10 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 285
TABLA 17. TIEMPO MEDIO DE CARGA DE LA PÁGINA PRINCIPAL DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 20 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 285
TABLA 18. TIEMPO MEDIO DE CARGA BÚSQUEDA DE HISTORIALES DESDE UN ORDENADOR DE LA RED ETSIT Y 5 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 286
TABLA 19. TIEMPO MEDIO DE CARGA BÚSQUEDA DE HISTORIALES DESDE UN ORDENADOR DE LA RED ETSIT Y 10 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 286
TABLA 20. TIEMPO MEDIO DE CARGA BÚSQUEDA DE HISTORIALES DESDE UN ORDENADOR DE LA RED ETSIT Y 20 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 287
TABLA 21. TIEMPO MEDIO DE CARGA BÚSQUEDA DE HISTORIALES DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 5 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 287
TABLA 22. TIEMPO MEDIO DE CARGA BÚSQUEDA DE HISTORIALES DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 10 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 288
TABLA 23. TIEMPO MEDIO DE CARGA BÚSQUEDA DE HISTORIALES DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 20 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 288
TABLA 24. TIEMPO MEDIO DE INTRODUCCIÓN DE UN HISTORIAL CON IMÁGENES DE EXPLORACIÓN DESDE UN ORDENADOR DE LA RED ETSIT Y 5 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 289
TABLA 25. TIEMPO MEDIO DE INTRODUCCIÓN DE UN HISTORIAL CON IMÁGENES DE EXPLORACIÓN DESDE UN ORDENADOR DE LA RED ETSIT Y 10 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 289
TABLA 26. TIEMPO MEDIO DE INTRODUCCIÓN DE UN HISTORIAL CON IMÁGENES DE EXPLORACIÓN DESDE UN ORDENADOR DE LA RED ETSIT Y 20 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 290
TABLA 27. TIEMPO MEDIO DE INTRODUCCIÓN DE UN HISTORIAL CON IMÁGENES DE EXPLORACIÓN DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 5 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 290
TABLA 28. TIEMPO MEDIO DE INTRODUCCIÓN DE UN HISTORIAL CON IMÁGENES DE EXPLORACIÓN DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 10 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 291

TABLA 29. TIEMPO MEDIO DE INTRODUCCIÓN DE UN HISTORIAL CON IMÁGENES DE EXPLORACIÓN DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 20 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 291
TABLA 30. TIEMPO MEDIO EN AÑADIR IMAGEN JPEG DESDE UN ORDENADOR DE LA RED ETSIT Y 5 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 292
TABLA 31. TIEMPO MEDIO EN AÑADIR IMAGEN JPEG DESDE UN ORDENADOR DE LA RED ETSIT Y 10 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 292
TABLA 32. TIEMPO MEDIO EN AÑADIR IMAGEN JPEG DESDE UN ORDENADOR DE LA RED ETSIT Y 20 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 293
TABLA 33. TIEMPO MEDIO EN AÑADIR IMAGEN JPEG DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 5 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 293
TABLA 34. TIEMPO MEDIO EN AÑADIR IMAGEN JPEG DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 10 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 294
TABLA 35. TIEMPO MEDIO EN AÑADIR IMAGEN JPEG DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 20 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 294
TABLA 36. TIEMPO MEDIO EN AÑADIR IMAGEN DICOM DESDE UN ORDENADOR DE LA RED ETSIT Y 5 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 295
TABLA 37. TIEMPO MEDIO EN AÑADIR IMAGEN DICOM DESDE UN ORDENADOR DE LA RED ETSIT Y 10 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 295
TABLA 38. TIEMPO MEDIO EN AÑADIR IMAGEN DICOM DESDE UN ORDENADOR DE LA RED ETSIT Y 20 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 296
TABLA 39. TIEMPO MEDIO EN AÑADIR IMAGEN DICOM DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 5 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 296
TABLA 40. TIEMPO MEDIO EN AÑADIR IMAGEN DICOM DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 10 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 297
TABLA 41. TIEMPO MEDIO EN AÑADIR IMAGEN DICOM DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 20 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 297
TABLA 42. TIEMPO MEDIO DE CARGA BÚSQUEDA DE IMÁGENES DESDE UN ORDENADOR DE LA RED ETSIT Y 5 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 298
TABLA 43. TIEMPO MEDIO DE CARGA BÚSQUEDA DE IMÁGENES DESDE UN ORDENADOR DE LA RED ETSIT Y 10 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 298
TABLA 44. TIEMPO MEDIO DE CARGA BÚSQUEDA DE IMÁGENES DESDE UN ORDENADOR DE LA RED ETSIT Y 20 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 299
TABLA 45. TIEMPO MEDIO DE CARGA BÚSQUEDA DE IMÁGENES DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 5 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 299
TABLA 46. TIEMPO MEDIO DE CARGA BÚSQUEDA DE IMÁGENES DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 10 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 300
TABLA 47. TIEMPO MEDIO DE CARGA BÚSQUEDA DE IMÁGENES DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 20 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 300
TABLA 48. TIEMPO MEDIO DE CARGA CREACIÓN PDF DESDE UN ORDENADOR DE LA RED ETSIT Y 5 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 301
TABLA 49. TIEMPO MEDIO DE CARGA CREACIÓN PDF DESDE UN ORDENADOR DE LA RED ETSIT Y 10 USUARIOS EN EL SISTEMA. 301
TABLA 50. TIEMPO MEDIO DE CARGA CREACIÓN PDF DESDE UN ORDENADOR DE LA RED ETSIT Y 20 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 302
TABLA 51. TIEMPO MEDIO DE CARGA CREACIÓN PDF DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 5 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 302
TABLA 52. TIEMPO MEDIO DE CARGA CREACIÓN PDF DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 10 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 303
TABLA 53. TIEMPO MEDIO DE CARGA CREACIÓN PDF DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 20 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 303
TABLA 54. TIEMPO MEDIO EN ELIMINAR HISTORIAL DESDE UN ORDENADOR DE LA RED ETSIT Y 5 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 304

TABLA 55. TIEMPO MEDIO EN ELIMINAR HISTORIAL DESDE UN ORDENADOR DE LA RED ETSIT Y 10 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 304
TABLA 56. TIEMPO MEDIO EN ELIMINAR HISTORIAL DESDE UN ORDENADOR DE LA RED ETSIT Y 20 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 305
TABLA 57. TIEMPO MEDIO EN ELIMINAR HISTORIAL DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 5 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 305
TABLA 58. TIEMPO MEDIO EN ELIMINAR HISTORIAL DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 10 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 306
TABLA 59. TIEMPO MEDIO EN ELIMINAR HISTORIAL DESDE UN ORDENADOR DE FUERA DE LA RED ETSIT Y 20 USUARIOS EN EL SISTEMA. FUENTE: PROPIA. 306
TABLA 60. COMPARATIVA ENTRE DIFERENTES SISTEMAS DE HCE. FUENTE: PROPIA. 341
TABLA 61. SISTEMAS DE INFORMACIÓN MÉDICA RELACIONADOS CON HCE PUBLICADOS DESDE 1995 HASTA 2005. FUENTE: (CRUZ-CORREIA ET ÁL. 2007) 342
TABLA 62. SOLUCIONES DE HCE COMERCIALES EN OFTALMOLOGÍA. FUENTE: PROPIA. 343
TABLA 63. APLICACIONES DE HCE DE SOFTWARE LIBRE. FUENTE: PROPIA. 348
TABLA 64. COMPARATIVA DE SISTEMAS DE HCE. FUENTE: PROPIA. 349
TABLA 65. PUBLICACIONES DE LA IEEE ANALIZADAS. FUENTE: PROPIA. 354
TABLA 66. ARTÍCULOS ANALIZADOS EN REVISTAS DE IMPACTO DE TELEMEDICINA. FUENTE: PROPIA. 355

1

Introducción

1.1. ANTECEDENTES 2
1.2. OBJETIVOS DE LA INVESTIGACIÓN 3
1.3. METODOLOGÍA EMPLEADA 4
1.4. DESCRIPCIÓN DE LA MEMORIA 11

1.1. Antecedentes

La **Diabetes Mellitus** (DM) constituye una enfermedad crónica cuya prevalencia estimada a nivel mundial va en aumento. Según (Wild *et ál*. 2004), en el año 2030 el número de diabéticos en el mundo será de unos 366,212 millones de personas. Según la Organización Mundial de la Salud (OMS), esta prevalencia se incrementará en los próximos años, especialmente en la diabetes tipo 2 que es la que padece el adulto.

La **Retinopatía Diabética** (RD) es la complicación ocular más grave de la DM y la primera causa de ceguera en los países desarrollados. La RD causa de 12.000 a 24.000 nuevos casos de ceguera todos los años en los Estados Unidos. La OMS estima que después de 15 años de padecer DM, aproximadamente el 2% de la población diabética llegará a ser ciega, mientras que el 10% desarrollarán severos problemas visuales. Entre los años 1992 y 1999, la Consejería de Sanidad de la Junta de Castilla y León promovió la primera Campaña de Prevención de la Ceguera por RD, campaña realizada por el Instituto Universitario de Oftalmobiología Aplicada (IOBA) en el área rural de Valladolid como experiencia piloto. Dicha experiencia ha incluido diversas acciones encaminadas a mejorar la formación de los médicos rurales de Valladolid. La campaña contó con una unidad móvil que se desplazó a todos los centros de salud de Valladolid, formando a los médicos de Atención Primaria (AP) en las técnicas de exploración oftalmológicas básicas. Se atendieron a 3.544 pacientes distintos, de los que 1.874 participaron en las dos fases del programa. Como era de esperar, la mayoría de los diabéticos (76%) era del tipo 2 y de todos los atendidos el 22% presentaba RD. En 1999, se inició una nueva experiencia piloto de Teleoftalmología con fines similares: mejorar la cualificación de los médicos de AP. Mediante un sistema digital capaz de tomar imágenes del fondo de ojo de los pacientes y transmitirlos al IOBA, los médicos de pueblos castellano-leoneses de Piedrahita y el Barco de Ávila remitieron dichas imágenes y un resumen de la **Historia Clínica (HC)** al Instituto, donde se interpretaron y se hicieron una serie de recomendaciones a seguir. Se revisaron 123 pacientes, detectándose RD en 44 ojos de 23 pacientes. Se recomendó la remisión a su oftalmólogo a los pacientes con retinopatías más graves y, además, se detectaron lesiones de fondo de ojo no asociadas a la diabetes.

Una vez que se dispone de la HC en papel y de los fondos de ojo en diferentes formatos de imagen en un ordenador, surge la necesidad de que dichas HCs puedan ser

accesibles desde cualquier lugar y compartidas por los diferentes médicos involucrados (médicos de familia, endocrinos y oftalmólogos). Para ello se ha diseñado, desarrollado y evaluado una aplicación Web (**TeleOftalWeb**) con el fin de almacenar en formato electrónico siguiendo unos estándares esas HCs y las retinografías de los pacientes y que, a su vez, pudieran ser compartidas por los diferentes profesionales. La aplicación tiene por objeto el almacenamiento e intercambio de los **Historiales Clínicos Electrónicos (HCEs)** y retinografías en cualquier formato digital por parte de los médicos. Además se desarrollarán y analizarán diferentes versiones de la misma empleando distintos modelos de Bases de Datos (BDs), con el fin de elegir la más eficiente de todas ellas.

1.2. Objetivos de la investigación

1. **Estudio de sistemas HCEs.** Búsqueda bibliográfica de las aplicaciones de HCEs publicadas en revistas con alto índice de impacto en telemedicina como son: *IEEE Transactions on Information Technology in Biomedicine (IEEE-ITB), Journal of the American Medical Informatics Association (JAMIA), Telemedicine and e-Health Journal, Journal of Telemedicine and Telecare, Methods of Information in Medicine, Medical Informatics and the Internet in Medicine, Journal of Biomedical Informatics, International Journal of Medical Informatics,* etc. con el fin de poder realizar una comparación entre ellas y **TeleOftalWeb**.

2. **Estudio de los estándares de HCE.** Se estudiarán los estándares: *Health Level 7* (HL7), *Digital Imaging and Communications in Medicine* (DICOM), *International Organization for Standardization* (ISO) 18308 del Comité Técnico (TC) 215, ENV 13606 de Comité Europeo de Estandarización TC 251, las iniciativas OpenEHR, *Good European Health Record* (GEHR), *Common Object Request Broker Architecture* (CORBA), *Integrating the Healthcare Enterprise* (IHE), y su vinculación a la especialidad de Oftalmología.

3. **Mejora técnica y funcional de la aplicación TeleOftalWeb**. Posteriormente, se llevará a cabo el desarrollo de la aplicación empleando diferentes modelados de datos distintos al inicial. Se utilizarán para ello BDs nativas *eXtensible Markup Language* (XML) y relacionales como Oracle.

4. **Medición de los tiempos de respuesta de las diferentes versiones de TeleOftalWeb** con el fin de comprobar cuál de ellas es la más efectiva.

5. **Puesta en marcha de la versión más eficiente de TeleOftalWeb** para un correcto funcionamiento de la misma de manera segura y su empleo por parte de los médicos y pacientes.

6. **Introducción de las HCs y de los fondos de ojo disponibles en la Unidad de RD del IOBA.**

7. **Evaluación de TeleOftalWeb**, a través de una encuesta vía Web, con el fin de comprobar su nivel de usabilidad.

8. En la discusión, se llevará a cabo una **comparativa de TeleOftalWeb** con otros sistemas de HCE publicados en diferentes revistas de impacto, comerciales y de *software* libre en relación a diferentes aspectos técnicos y funcionales.

9. **Publicación de los resultados obtenidos**, preferiblemente en revistas internacionales con alto índice de impacto.

1.3. Metodología empleada

Se ha desarrollado y evaluado una herramienta Web que permite gestionar y compartir HCs oftalmológicas y retinografías. La aplicación hace uso de tecnologías como XML para el intercambio de la información. Ha sido desarrollada utilizando el lenguaje de programación Java y las *Application Programming Interfaces* (APIs) que se derivan del mismo (Java Servlets, *Java Server Pages* (JSP), Swing, etc.) asegurando de esta manera la independencia respecto a la plataforma. Como entorno de desarrollo se ha utilizado el *Integrated Development Environment* (IDE) de Sun NetBeans 4.1. También se ha hecho uso de recursos adicionales que complementan el núcleo programado en Java, como por ejemplo el lenguaje *HyperText Markup Language* (HTML) y *Dynamic HyperText Markup Language* (DHTML), dentro del cual va embebida la aplicación desarrollada, permitiendo dar dinamismo al interfaz de usuario. En cuanto a los servidores y BDs utilizados en la implantación de la herramienta, se ha hecho uso de versiones de libre distribución y gratuitas como Tomcat, *MySQL Server*,

dbXML 2.0, Xindice 1.2, eXist 1.1.1, y de la BD comercial Oracle 10g en las diferentes versiones de la aplicación. El objetivo final de este trabajo es elegir la BD que ofrezca los mejores tiempos de respuesta a la aplicación **TeleOftalWeb**.

A la hora de enfrentarse al problema de desarrollar la herramienta *software* es importante una correcta división de tareas del problema, puesto que esto nos simplificará enormemente la implementación de la aplicación. Para ello se ha seguido el procedimiento que puede resumirse en el diagrama de flujo de la figura 1. En la cuarta versión de la aplicación **TeleOftalWeb**, se han cambiado los módulos consultas MySQL, XML y parseo XML por el módulo de consultas de la BD Oracle 10g. Para el resto de las versiones que emplean BD nativas XML el diagrama de flujo es similar al de la figura 1. Los diferentes módulos de **TeleOftalWeb** se explicarán con detalle en el Capítulo 6.

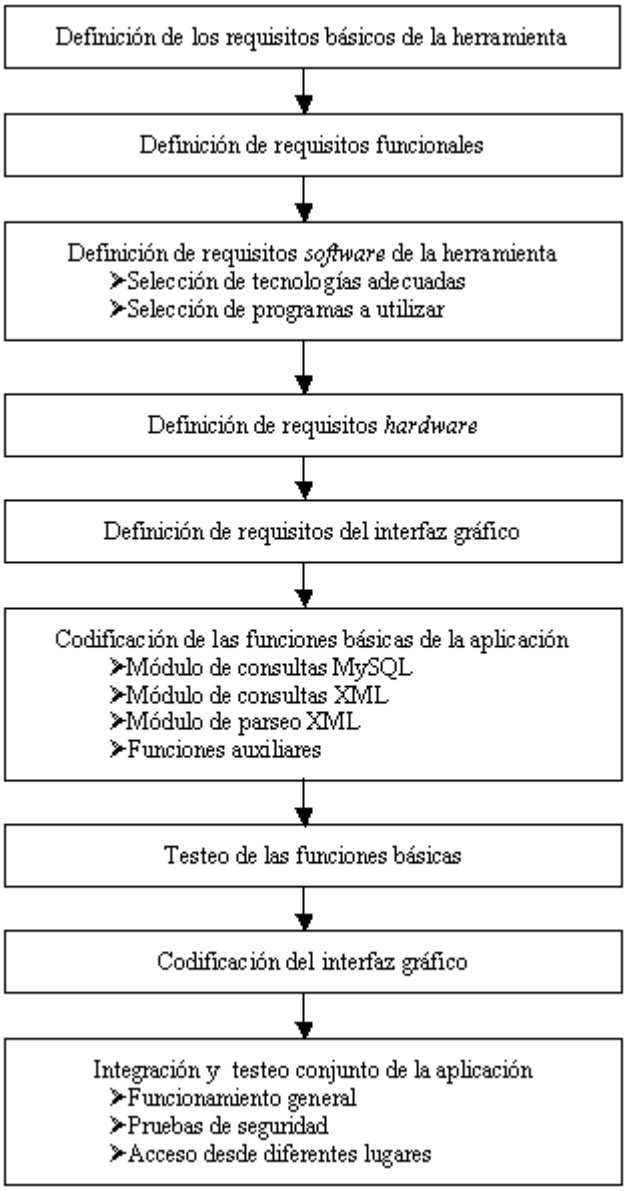


Figura 1. Diagrama de flujo de la metodología de diseño en las tres primeras versiones de la aplicación. Fuente: propia.

Los requerimientos básicos que debe de cumplir la aplicación son las siguientes:

- Se debe distinguir dos modos de funcionamiento, en apariencia independientes: acceso y manipulación de historiales médicos, y tratamiento de imágenes.

- El acceso a las HCs de un paciente sólo podrá realizarse por el oftalmólogo que el paciente tenga asignado, o por los usuarios cuyo acceso haya sido autorizado por dicho especialista.

- La aplicación debe permitir compartir HCs, la manera de implementarlo sería permitiendo que el oftalmólogo asignado a un paciente mantuviera una lista en

la que aparecieran los nombres de los usuarios. Cuando una ficha sea compartida, los datos personales del paciente, a excepción del número de identificación no deberían ser accesibles al usuario autorizado, sino que éste debería poder acceder al historial del paciente (sin poder modificarlo) y a las imágenes asociadas al mismo.

- Las revisiones se deben presentar de forma secuencial.

- La aplicación se debe poder utilizar en cualquier punto geográfico, así como desde cualquier equipo, de manera correcta. Para ello además de los conceptos de aplicación Web deben introducirse las tecnologías XML y Java, para que exista una accesibilidad universal y una independencia respecto a la plataforma y los datos.

- La aplicación debe ser segura a todos los niveles, pues la privacidad en la información manejada es algo prioritario. En este sentido, sería conveniente que el acceso a la misma estuviera autentificado, y las sesiones establecidas fueran seguras, por ejemplo a través del uso de protocolos como *Secure Socket Layer* (SSL) y *Hypertext Transfer Protocol over Secure Socket Layer* (HTTPS).

Para poder implementar los requerimientos de la aplicación, serán necesarios una serie de elementos básicos, a nivel *software*:

- La manera de asegurar la independencia respecto a la plataforma ha sido utilizando el lenguaje de programación Java, como base de la aplicación.

- Para el intercambio y almacenamiento de extractos de HC entre diversas aplicaciones e instituciones, el estándar XML y las tecnologías que lo acompañan son los más adecuados desde el punto de vista técnico. Esto se debe a que los HCs suelen tener una estructura jerárquica, al igual que XML. Otra de las razones que justifica la utilización de XML para la estandarización de la HCE es su independencia del *software* y hardware utilizado, así como la posibilidad de separar el contenido (documento XML) de las reglas de presentación, lo cual permite una gran flexibilidad a la hora de desarrollar sistemas de comunicación de HCE (SEIS 2003; Alschuler & Boyer 2000).

- Se ha hecho uso de recursos adicionales que complementan el núcleo programado en Java, como por ejemplo el lenguaje HTML, dentro del cual va embebida la aplicación desarrollada. Además se han utilizado otro tipo de tecnologías como DHTML para homogeneizar y dar dinamismo al interfaz gráfico.

En cuanto a los servidores y BDs que se han utilizado para implementar la aplicación lo siguiente:

- Para el módulo de gestión de usuarios se ha mantenido la BD relacional en la que se almacenan los datos personales de los mismos. El servidor de BD utilizado ha sido MySQL Server 5.0 junto con el Conector-J 3.1.11 en las tres primeras versiones de la aplicación y la cuarta versión, se ha optado por la BD Oracle 10g.

- Para el módulo de gestión y almacenamiento de historiales se han utilizado diferentes BDs dependiendo de la versión de la aplicación. Para la primera versión de la misma, se ha utilizado una BD nativa XML de código abierto, dbXML 2.0. Para la segunda y tercera versiones, se ha optado por otras dos BD nativas XML de código abierto: eXist 1.1.1 y Xindice 1.2. En la cuarta versión, se ha empleado la BD comercial Oracle 10g.

- Para el procesamiento de peticiones JSP+Servlets y despliegue de la aplicación se ha utilizado el servidor Tomcat 5.5.9.

- Para la simplificar la gestión de la aplicación se ha desarrollado un módulo exclusivo para tal fin que se ha denominado **TeleOftalWeb Admin**.

- La aplicación Web ha sido optimizada para el navegador *Mozilla Firefox* (MF) 1.5.0.3 debido a su grado de seguridad y robustez frente a *spyware* así como su mayor velocidad y funcionalidad.

A continuación, en la figura 2 se puede observar el funcionamiento de forma simplificada de la aplicación.

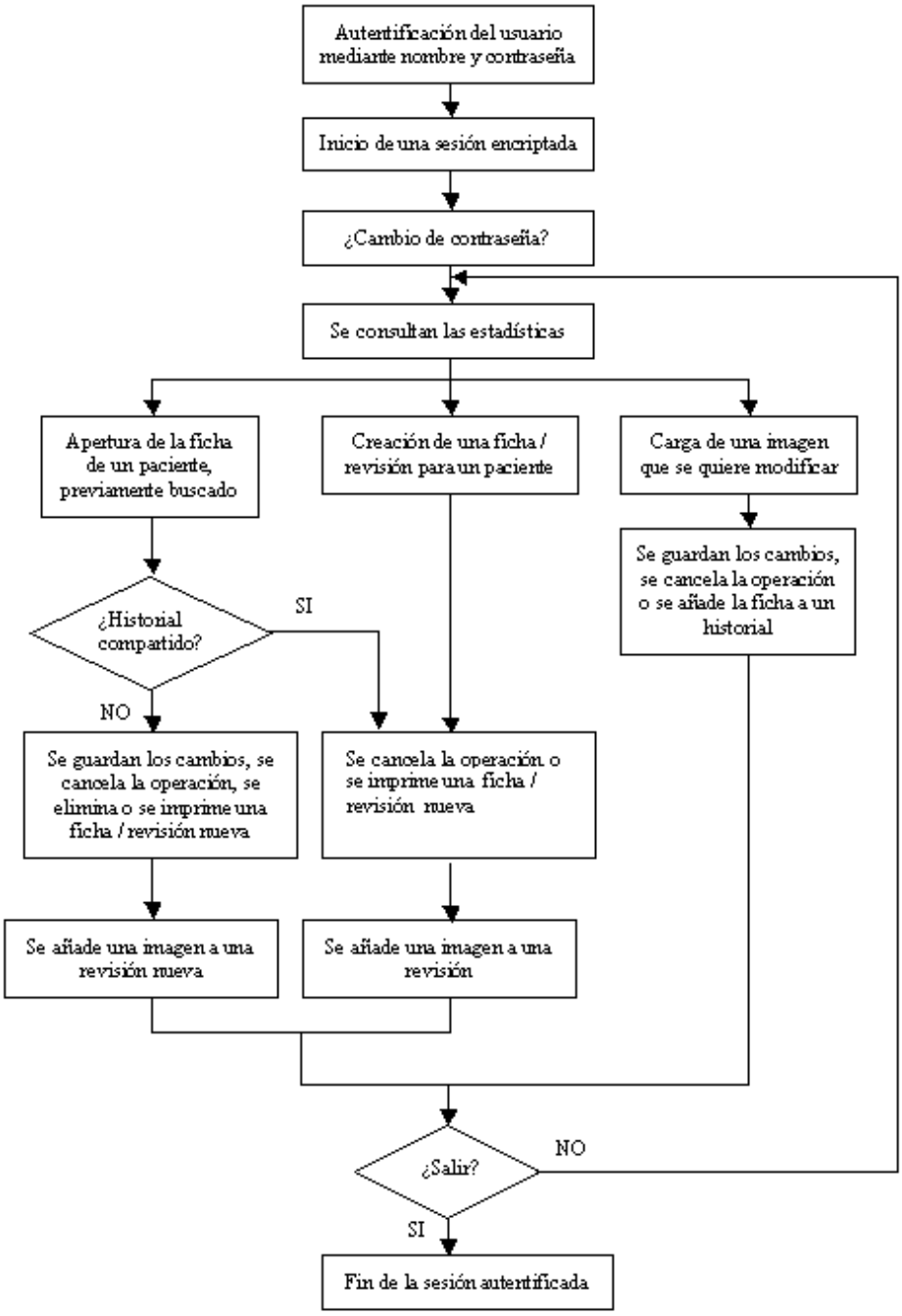


Figura 2. Diagrama de funcionamiento simplificado de la aplicación. Fuente: propia.

En la figura 3 se puede observar un esquema de funcionamiento simplificado del módulo de administración. Una vez iniciada la sesión autentificada puede realizarse cualquiera de las acciones permitidas en dicho módulo, eliminar usuarios, consultar los datos de usuarios existentes, dar de alta nuevos usuarios, etc. Algunas de las acciones anteriores requieren una confirmación adicional por parte del administrador, como es el caso de la eliminación de historiales o usuarios. Una vez finalizada una de las acciones anteriores se volvería a la situación inicial en la que se listan los usuarios del sistema.

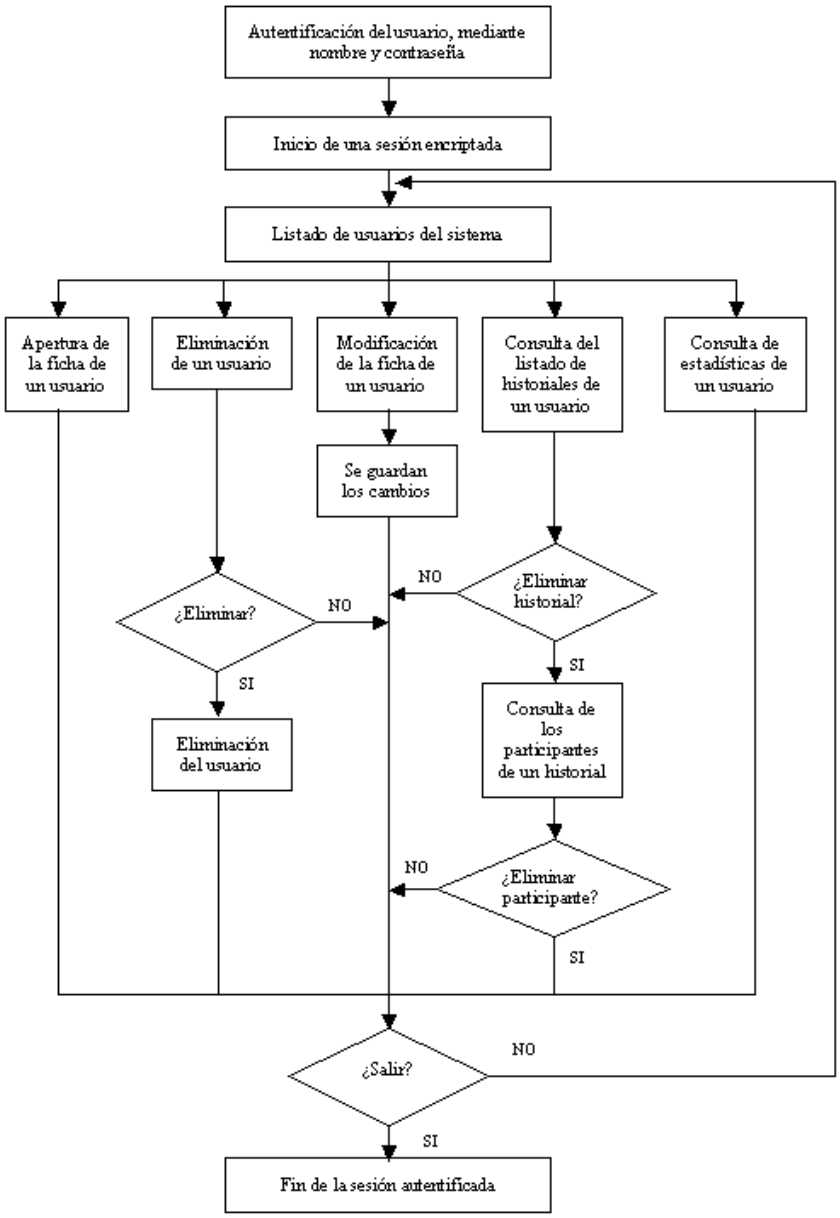


Figura 3. Diagrama de funcionamiento simplificado del módulo de administración. Fuente: propia.

Otras de las tareas fundamentales en la presente Tesis Doctoral ha sido la estandarización de la aplicación de acuerdo a los estándares *HL7 Clinical Document Architecture* (CDA), DICOM e ISO 18308 de ISO/TC 215. Para ello se han realizado los siguientes pasos:

- Estudio de los estándares y del estado del arte de los mismos.
- Aplicación del estándar HL7/CDA Release 1.0 2000 en la aplicación y posteriormente HL7/CDA Release 2.0.

- Cumplimiento del modelo funcional HL7-S.
- Aplicación del estándar DICOM 3.0 en el módulo imágenes de **TeleOftalWeb**.
- Cumplimiento del estándar ISO/TC 215 en la aplicación.

1.4. Descripción de la memoria

Finalmente, en lo que se refiere a la organización de esta memoria, ésta se va a estructurar en nueve capítulos y tres apéndices. Su contenido, aparte del incluido en el presente capítulo de Introducción, se detalla a continuación:

- **Capítulo 2: Telemedicina. Aplicaciones. Teleoftalmología.** En este capítulo, se llevará a cabo una descripción del estado del arte de la telemedicina y sus aplicaciones. Se analizaron 218 artículos de las revistas: *IEEE-ITB, JAMIA, Telemedicine and e-Health Journal, Journal of Telemedicine and Telecare, Methods of Information in Medicine, Medical Informatics and the Internet in Medicine, Journal of Biomedical Informatics, Journal of Medical Systems, International Journal of Medical Informatics,* desde su origen hasta la fecha actual, relacionados con la telemedicina en general. También se analiza el estado de la Teleoftamología.

- **Capítulo 3: Historiales Clínicos Electrónicos (HCEs).** En el capítulo cuatro se analiza el estado de los sistemas HCEs revisando revistas con índice de impacto. Se analizaron 530 artículos de las revistas *IEEE-ITB, JAMIA, Telemedicine and e-Health Journal, Journal of Telemedicine and Telecare, Methods of Information in Medicine, Medical Informatics and the Internet in Medicine, Journal of Biomedical Informatics, Journal of Medical Systems, International Journal of Medical Informatics*, desde su origen hasta la fecha actual.

- **Capítulo 4: Estándares de HCE.** En este capítulo se analiza el estado de los estándares de HCE (HL7, DICOM, ISO 18308, ENV 13606, las iniciativas OpenEHR, GEHR, Corba e IHE), analizando revistas con alto índice de impacto. Se analizaron 274 artículos de las revistas *IEEE-ITB, JAMIA, Telemedicine and e-Health Journal, Journal of Telemedicine and Telecare, Methods of*

Information in Medicine, Medical Informatics and the Internet in Medicine, Journal of Biomedical Informatics, Journal of Medical Systems, International Journal of Medical Informatics, desde su origen hasta la fecha actual.

- **Capítulo 5: Bases de datos (BDs).** Se analizan conceptos sobre BDs, tipos, modelado de datos y BDs empleadas en algunos sistemas de telemedicina y en nuestra aplicación **TeleOftalWeb**.

- **Capítulo 6: Métodos.** En el capítulo seis se explica la arquitectura de la aplicación, las tecnologías empleadas en su desarrollo, el modelado de datos de utilizado en cada una de las versiones, el modelo de colas empleado para medir los tiempos de respuesta y diferentes aplicaciones informáticas empleadas para calcular los tiempos de carga de las diferentes versiones de la aplicación.

- **Capítulo 7: Resultados.** En este capítulo se recogen los resultados del desarrollo de la aplicación Web; Módulo gestor y usuario, así como la experiencia práctica llevada a cabo con **TeleOftalWeb** y el análisis de los tiempos de respuesta obtenidos en las diferentes versiones de la misma.

- **Capítulo 8: Discusión.** En base a los resultados obtenidos en el capítulo anterior, en el capítulo octavo se analiza una comparativa de aplicaciones de HCE publicadas en revistas y conferencias con la aplicación **TeleOftalWeb**, así como una discusión sobre los tiempos de respuesta obtenidos con las diferentes versiones.

- **Capítulo 9: Conclusiones y líneas futuras.** En este capítulo final se extraen las conclusiones más relevantes de la investigación llevada a cabo. Asimismo, se detallan las principales aportaciones de esta Tesis Doctoral y se comentan las futuras líneas de investigación.

- **Apéndice A: Publicaciones generadas.** En este apéndice se enumeran las publicaciones más relevantes a las que ha dado lugar la presente Tesis Doctoral, agrupadas en revistas internacionales, congresos internacionales, congresos nacionales y capítulos de libros.

- **Apéndice B: Manual de usuario.** En este apéndice se explica detalladamente el

funcionamiento de la aplicación (tanto del módulo gestor como del módulo usuario de la misma).

- **Apéndice C: Siglas de abreviaturas y acrónimos.** En este apéndice se enumeran las siglas y acrónimos que aparecen en la memoria de la Tesis Doctoral.

2

Telemedicina. Aplicaciones. Teleoftalmología

2.1. TELEMEDICINA 16
2.2. TELEOFTALMOLOGÍA 18
2.2.1 INTRODUCCIÓN 18
2.2.2. RETINOPATÍA DIABÉTICA Y TELEMEDICINA 19
2.2.3. EXPERIENCIAS DE TELEOFTALMOLOGÍA Y RETINOPATÍA DIABÉTICA 21
2.2.4. BIBLIOGRAFIA 24

2.1. Telemedicina

El Sector Sanitario es un dominio muy adecuado para la utilización de aplicaciones telemáticas, dadas sus peculiaridades. Existen muchas definiciones para el concepto de Telemedicina. Según (Perís 1974), no hay que entender la Telemedicina como un conjunto de tecnologías de última generación capaces de resolver todos los problemas, sino que no es más que una herramienta para mejorar el trabajo de los médicos generales y especialistas, y sobretodo el servicio dado al paciente. Yu-Hua *et ál.* (2000) y Bai *et ál.* (1998) definen la telemedicina como un concepto general que incluye diagnósticos, exámenes, reuniones médicas, operaciones colaborativas y seminarios. A su vez, telemedicina es un término genérico que se emplea para definir varios aspectos relacionados con el cuidado sanitario a distancia. Un aspecto clave de la telemedicina es el empleo de señales electrónicas con el fin de transferir la información de un lugar a otros. Horsch & Balbach (1999) definen diferentes Sistemas de Telemedicina, entre los que se encuentran los HCEs, fundamentales en el desarrollo de la presente investigación. Los servicios de telemedicina que integran HCE decrementan el número de casos donde los médicos no pueden acceder a dichos historiales (Zach 1996).

Según la Organización Mundial de la Salud (OMS), la telemedicina es 'el suministro de servicios de atención sanitaria, en los que la distancia constituye un factor crítico, por profesionales que apelan a las Tecnologías de la Información y de las Comunicaciones (TIC) con objeto de intercambiar datos para hacer diagnósticos, preconizar tratamientos y prevenir enfermedades y heridas, así como para la formación permanente de los profesionales de atención de salud y en actividades de investigación y evaluación, con el fin de mejorar la salud de las personas y de las comunidades en que viven'. Según (Ferrer-Roca 2001), la telemedicina es la provisión de medicina a distancia. Se trata de un servicio a los ciudadanos. Es una práctica a distancia en la que las Nuevas Tecnologías garantizan los mismos niveles de seguridad y calidad que en la práctica médica de tipo presencial.

Desde hace décadas, la medicina ha demostrado una gran capacidad para adaptarse a los avances tecnológicos. La aplicación de la cibernética y la robótica han hecho posible algunos logros como el almacenamiento y procesamiento de información accesible a varios profesionales simultáneamente, la monitorización de parámetros

biológicos a distancia, etc. Internet, la televisión o la telefonía móvil son el mejor vehículo para llegar hasta el ciudadano, esté donde esté, y, en consecuencia serán un medio fundamental para garantizar su acceso a la asistencia sanitaria (Ferrer-Roca 2001).

El registro más antiguo de telemedicina se remonta a una demostración en Nebraska en la década de 1950 que usaba un circuito cerrado de televisión para proporcionar servicios psiquiátricos desde un centro médico universitario hacia un hospital a casi 200 Km. de distancia. Hace cuarenta años dio inicio el programa de telemedicina en vuelos espaciales de la *Nacional Aeronautics and Space Administration* (NASA), de tal manera que el personal en tierra podía monitorear las respuestas biomédicas de los astronautas al vuelo espacial y proporcionarles el apoyo médico necesario. El proyecto de la NASA "Puente Espacial de Telemedicina a Armenia" proporcionó asistencia médica en respuesta al gran terremoto de Armenia en 1988. Usando un enlace satelital en vivo y bidireccional, personal médico en hospitales de *Salt Lake City, Houston Texas* y *Maryland* participó en muchas consultas en conjunto con los médicos de Armenia. Debido al enorme coste de estos esfuerzos pioneros sólo hasta hace una década la práctica de la telemedicina se trasladó de proyectos piloto a la disponibilidad pública. También es necesaria una evaluación posterior del sistema de telemedicina una vez implantado y usado los profesionales de la salud y los propios pacientes (Holle & Zahlmann 1999).

En este trabajo, se ha llevado a cabo la **revisión bibliográfica de 218 artículos que tratan de sistemas de información en telemedicina** en revistas con alto índice de impacto *(IEEE-ITB, JAMIA, Telemedicine and e-Health Journal, Journal of Telemedicine and Telecare, Methods of Information in Medicine, Medical Informatics and the Internet in Medicine, Journal of Biomedical Informatics, Journal of Medical Systems, International Journal of Medical Informatics).* De todos ellos, existen numerosas publicaciones que tratan de estudios sobre la adopción de los sistemas de telemedicina por parte de médicos que viven en las zonas rurales y urbanas. Según (Croteau & Vieru 2002), en uno de estos estudios se basa en un modelo de adopción de la tecnología por parte de los médicos. Se les pasó un cuestionario a todos ellos y los resultados mostraron la utilidad percibida en relación al empleo de la telemedicina por todos los médicos en su práctica diaria.

Segura *et ál.* (2008) realizaron un estudio para evaluar la intención de los médicos españoles para aceptar y emplear la telemedicina. Participaron 985 médicos de zonas rurales y urbanas (internistas, endocrinos, médicos de familia, cardiólogos y nefrólogos). Respecto a los beneficios de la telemedicina, fueron los cardiólogos los profesionales que la percibieron como más útil en sus trabajos, seguidos de los médicos de familia y pediatras. Como conclusión final se extrajo, que **la mayoría de los médicos españoles no ha implementado la telemedicina en su práctica clínica**, sin embargo estarían interesados en aplicaciones como son la formación *on-line* o el control de enfermedades. También señalar que prefieren el contacto presencial paciente-médico.

Otros de los artículos analizados trataban sobre la evaluación de sistemas de telemedicina a través de dispositivos móviles o vía Web en diferentes especialidades médicas o para el tratamiento de enfermedades crónicas y graves (Thiyagarajan & Clarke 2006). En (Paul *et ál. 1999*), se pone de manifiesto que la calidad de los equipos de telemedicina es frecuentemente una de las barreras en la implantación de dichos sistemas.

2.2. Teleoftalmología

2.2.1 Introducción

La oftalmología es una especialidad médica en la que el diagnóstico por imagen es fundamental y en consecuencia una buena candidata para los nuevos sistemas de telemedicina (Hijar et al 1998). Permite cubrir un amplio rango de actividades como proporcionar cuidados médicos en zonas remotas, realizar programas de despistaje de enfermedades causantes de ceguera en la población general o ayudar en el diagnóstico y tratamiento de determinados procesos. Es útil, además, para compartir recursos médicos (Tang et al 2005), colaborar en ensayos clínicos y llevar a cabo programas de formación continuada.

Buenos ejemplos de ello son, entre otros, los *programas de apoyo* de teleoftalmología entre el Reino Unido (Moorfields Eye Hospital -Londres-) y Sudáfrica, la red de teleoftalmologia de Oriente Medio (MEON) o la red que conecta la Republica

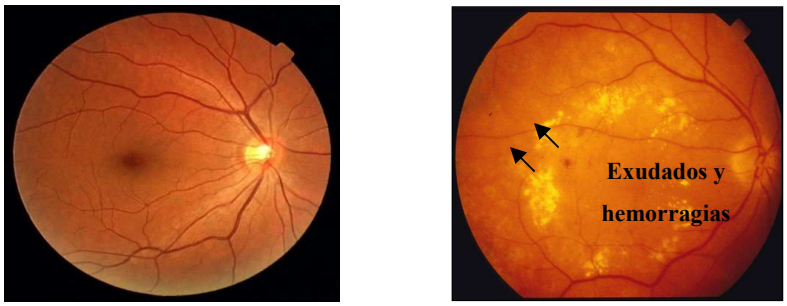
de Azerbayan y Moscu. En la actualidad, existen programas de teleoftalmología de alta resolución prácticamente en todo el mundo occidental gracias a los nuevos dispositivos de cáptura de imágenes (retinógrafos, tomógrafos de coherencia óptica-OCT-, ecógrafos, etc) y a los avances de las tecnologías de las comunicaciones (Yogesan et al 2006).

Los sistemas de teleoftalmología pueden funcionar a tiempo real, en diferido (*store and foreward*) o con una combinación de ambas técnicas dependiendo de la finalidad perseguida. Su utilidad ya ha sido demostrada y su éxito va a depender de factores tales como la motivación, la financiación, la planificación y el grado de aceptación (Yogesan 2006, Paul 2006).

2.2.2. Retinopatía Diabética y Telemedicina

La Diabetes Mellitus y sus complicaciones crónicas constituyen en la actualidad un problema de enorme trascendencia sociosanitaria, al tratarse de una enfermedad de prevalencia elevada y creciente de la que, de acuerdo con las estimaciones de la Organización Mundial de la Salud, se verán afectados más de 300 millones de personas superado el año 2025 (Wild et al 2004).

De todas las complicaciones oculares que puede producir esta enfermedad, la Retinopatía Diabética (RD) es la más temida al estar considerada como la causa más frecuente de ceguera en edad laboral activa en el mundo occidental. Sin embargo, varios ensayos clínicos prospectivos, randomizados y controlados han demostrado que con el seguimiento y tratamiento a tiempo de las formas de riesgo es posible reducir la progresión hacia la ceguera en un número significativo de pacientes (Early Treatment Diabetic Retinopathy Study Clinical Trials. Reports number 9). La Retinopatía Diabética es una enfermedad que se caracteriza por el desarrollo en el fondo de ojo de una serie de lesiones que determinan diferentes estadios de gravedad creciente que comprometen la función visual del paciente y que pueden ser fácilmente identificados al explorar el fondo de ojo por oftalmoscopía o en las fotografías de fondo –retinografías- (figura 4 a y b).



(a) (b)

Figura 4. (a) Fondo normal, (b) Fondo con lesiones de RD.

El seguimiento periódico desde el punto de vista oftalmológico es, por lo tanto, la medida preventiva más eficaz y rentable. El problema es que, pese a ello, incluso en los países más desarrollados solo un 50% de los pacientes diabéticos acuden periódicamente al oftalmólogo, siendo las razones fundamentales que justifican este hecho la ausencia de síntomas visuales en los estadios iniciales de la enfermedad (Vijan et al 2000), factores socioeconómicos y geográficos o la escasa educación sanitaria..

En un intento de mejorar estas cifras se han puesto en marcha en los últimos años, aprovechando el desarrollo de los nuevos retinógrafos no midriáticos (figura 5) y la alta resolución de la fotografia digital, varios programas de teleoftalmología para detectar a tiempo las formas de alto riesgo de progresión hacia la ceguera.

Figura 5. Retinógrafo.

De hecho, la Academia Americana de Telemedicina (ATA), el grupo de trabajo en salud ocular (OT-SIG) y el Instituto Nacional de Tecnología y desarrollo de estándares (NIST) están trabajando en los requerimientos que deben reunir estos

programas para poder ser implantados y utilizados para el cribado de la Retinopatía Diabética (Cavallerano et al 2004).

2.2.3 Experiencias de Teleoftalmología y Retinopatía Diabética

Hasta el momento presente se han llevado a cabo experiencias de teleoftalmología en pacientes diabéticos (Tennant et al 2000, Hall et al 2005) prácticamente en todos los países y se han diseñado e implantado varios sistemas de telemedicina Web para el "*screening*" de la RD empleando retinógrafos no midriáticos y haciendo uso de software libre.

Australia:

Existe, desde hace ya más de una década, un programa de cribado de Retinopatía Diabética en áreas remotas en la población indígena cuyos resultados iniciales fueron publicados en 1998 (Diamond et al JP 1998). Además, en el año 2003, el Departamento de Salud de la zona Oeste de Australia puso en marcha un proyecto de teleoftalmología que conectaba el Hospital Regional de Carnavon y el Instituto de Oftalmología de Lions de la ciudad de Perth (940 Km) que permitía la realización de interconsultas de oftalmología. Su finalidad era evaluar el impacto de este sistema en el diagnóstico, tratamiento, grado de satisfacción y coste, siendo el control de la diabetes y sus complicaciones oculares el objetivo fundamental. Los resultados fueron publicados por Kumar et al en 2005. Durante los doce meses que duró el proyecto se realizaron 118 consultas de teleoftalmología, el 94% de ellas en pacientes diabéticos o con glaucoma. Sólo un 3% de los pacientes tuvieron que ser derivados al hospital de referencia y un 36% debían seguir siendo revisados mediante este sistema de teleoftalmología. En cuanto al grado de satisfacción, el 98% de los pacientes estaban satisfechos con este sistema y en los que respecta a los costes el sistema se hace rentable cuando se revisan al menos 126 pacientes por año.

Canadá:

En Canadá y para el despistaje de la RD se han planteado dos soluciones de teleoftalmología una vez que Boucher et al demostrasen en 2003 que con el uso de cámaras no midriáticas y la cáptura de dos imágenes por paciente era posible llevar a

cabo de manera eficaz el despistaje de esta complicación de la diabetes. La solución de Québec es un sistema fijo de teleoftalmología que conecta centros de atención primaria con una consulta especializada de oftalmología (Boucher et al 2003) y la solución remota para áreas rurales alejadas que implica el desplazamiento de unidades móviles por las zonas rurales con la finalidad de cubrir el mayor número posible de pacientes. En un trabajo pulicado por Nathoo N en 2010 se pone de manifiesto la utilidad de estos sistemas para el seguimiento de la RD en las áreas rurales de Alberta (Canadá).

Recientemente se ha propuesto utilizar este sistema también para llegar a las zonas urbanas marginales una vez que Boucher et al, en el año 2008, publicaran los resultados de un programa de despistaje de RD llevado a cabo con unidades móviles de teleoftalmología instaladas en las farmacias que pretendía llegar a la población urbana que carecía de recursos para acceder a los servicios médicos. Revisaron de este modo a 3505 diabéticos conocidos de las zonas de Quebec, British Columbia, Alberta, Manitoba y Saskatchewan. El 85% de los pacientes pudo ser evaluado eficazmente de este modo. Sólo un 30% de los pacientes había sido visto en los dos años anteriores por un oftalmólogo y un 38% no habia sido visto nunca pese a ser diabético. Un 2% de los pacientes tuvo que ser remitido para tratamiento.

Estados Unidos (EEUU)

En Estado Unidos se han puesto en marcha varios programas de teleoftalmología para el despistaje de la Retinopatía diabética, como el programa VOIC (Vanderbilt Ophthalmic Imaging Center) o el programa de la Clínica Joslin de Boston (Joslin Vision Network) entre otros, que han demostrado que además de eficaces, estos sistemas son económicamente rentables (James et al 2000).

Ambos programas y sobre todo el de la clínica Joslin han permitido establecer los requisitos necesarios para su validación y posterior uso en el diagnóstico y seguimiento de la RD por teleoftalmología (Cavallerano et al 2003). De hecho se dispone en estos momentos ya de un protocolo que incluye quién puede realizar la cáptura de las imágenes y cómo debe llevarse a cabo y cuáles son los requisitos del *hardware y software* de teleconsulta, así como las necesidades de ancho de banda, servidor, sistemas de transmisión, almacenamiento y archivo de datos e imágenes de acuerdo a los estandares existentes. La iniciativa de la clínica Joslin es una plataforma tecnológica que facilita a día de hoy el manejo y tratamiento de los pacientes con retinopatía diabética y que ha sido implantada en numerosas localidades de EEUU.

Europa

También en Europa se han llevado a cabo experiencias piloto de teleoftalmología en casi todos los países en la última década.

En España y aunque actualmente se está intentando implantar en todas las Comunidades Autónomas, las experiencias pioneras son las que se llevaron a cabo en Galicia, Las Islas Canarias y Castilla y León.

Castilla-León

En 1999, el Instituto de Oftalmobiología Aplicada de la Universidad de Valladolid, con el apoyo de la Junta de Castilla y León, puso en marcha un programa piloto de teleoftalmología al servicio del médico de atención primaria de las áreas rurales periféricas que ha demostrado ser muy útil para el despistaje de la Retinopatía Diabética en esas zonas. Sus resultados fueron presentados en el 8th Annual Meeting de la Academia Americana de Telemedicina (López-Gálvez et al 2003) y han favorecido la implantación de este sistema en la provincia de Avila y su incorporación a la cartera actual de servicios del Sacyl en esta provincia de la Comunidad Autónoma.

Galicia

Existe desde el año 1999 un servicio de teleoftalmología entre el Complejo Hospitalario Universitario de Santiago de Compostela y varios centros de atención primaria que garantizan a día de hoy la cobertura de más de 16500 personas del área rural y que está permitiendo aumentar el grado de cumplimiento con los protocolos de despistaje de la RD en esas áreas. Además, en este mismo Complejo Hospitalario se llevó a cabo la validación, en el año 2002, de este sistema de teleoftalmología para el cribado de la RD (Gómez Ulla et al 2002). Actualmente esta Comunidad Autónoma dispone, además, de otro servicio de teleoftalmología que conecta al Instituto Gallego de Oftalmología con el norte de Galicia.

Islas Canarias

En 1999 se publicó la primera experiencia de teleconsulta en oftalmología en las Islas Canarias (Abreu et al 1999) y también desde ese año existe un programa de teleconsulta que conecta el Servicio de Oftalmología del Hospital de la Candelaria con los centros de salud periféricos para la detección de la RD, siendo uno de los más eficaces y rentables de nuestro país en estos momentos.

Muchas de estas experiencias han servido para poner de manifiesto la utilidad de las nuevas tecnologías en la prevención de la ceguera por RD. Sin embargo, para implementarlos en la práctica clínica es preciso disponer no sólo de un protocolo de captura, almacenamiento y transmisión sino que además es necesario que estos programas estén integrados en la red sanitaria y que los profesionales involucrados en el seguimiento de estos pacientes puedan acceder a los datos de manera eficaz y segura.

En el caso concreto de la RD, disponer además de las imágenes de fondo de ojo, de la historia clínica completa es de gran ayuda pues su progresión está influenciada por la presencia de determinados factores de riesgo y por el grado de control metabólico, Es una enfermedad que habitualmente requiere un abordaje multidisciplinar.

2.2.4 Bibliografia

Abreu González R, Abreu Reyes JA, Ferrer Roca O 1999, Ophthalmic advice on internet. Archivos de la Sociedad Canaria de Oftalmología vol 10, pp 125-130

Boucher MC, Gresset JA, Angioi K, Olivier S 2003. Effectiveness and safety of screening for diabetic retinopathy with two nonmydriatic digital images compared with the seven standard stereoscopic photographic fields. Can J Ophthalmol. 38(7):557-68

Boucher MC, Desroches G, Garcia-Salinas R, Kherani A, Maberley D, Olivier S, Oh M, Stockl F 2008. Teleophthalmology screening for diabetic retinopathy through mobile imaging units within Canada. Can J Ophthalmol.;43(6):658-68

Cavallerano J, Lawerence MG, Zimmer-Galler I et al, American Telemedicine Association Ocular Telehealth Special Interest Group, National Standards and Technology Working Group 2004 Telehealth practice recommendations for diabetic retinopathy. Telemedicine and e-Health Journal, vol 10, pp: 469-482

Cavallerano A, Cavallerano, J, Katalinic P; Tolson A, Aiello, LLP; Aiello LLM; and the Joslin Vision Network Clinical Team 2003. Use of Joslin Vision Network Digital-Video Nonmydriatic Retinal Imaging To Assess Diabetic Retinopathy in A Clinical Program. Retina, 23(2); 215-223

Diamond JP, McKinnon M, Barry C, Gerry D, McAllister I, House P, Constable IJ, 1998, Non Mydriatic fundus photography a viable alternative for identification of diabetic retinopathy in Aboriginal population in rural Western Australia. Ant NZJ Ophthalmol

26: 102-115

Early Treatment Diabetic Retinopathy Research Group. 1990. Early photocoagulation for diabetic retinopathy: ETDRS report number 9. Ophthalmology 98. pp 766-785

GómezUlla F, Fernandez MI, González F, Rey P, Rodriguez M, Rodriguez-Cid MJ, Casanueva FF, Tomé MA, García Toribio J, Gaude F 2002, Digital retinal images and teleophthalmology for detecting and grading diabetic retinopathy. Diabetes Care vol 25, pp: 1384-1389

Hall G, Hennesy M, Bartin J & Coroneo M 2005, Teleophthalmology-Assisted Corneal Foreing Body removal in a rural hospital, Telemedicine and e-Health Journal vol 11, pp: 79-83

Hijar CA, Moreno J & Jiménez L. 1998, Prevalencia y riesgo de pérdida de visión por retinopatía diabética en Soria. Archivos de la Sociedad Española de Oftalmología. pp. 103-107

James M, Turner DA, Broadvent DM, Vora J, Harding SP 2000, Cost-effectiveness analysis of screening for sight threatening diabetic eye disease. British Medical Journal vol 320, pp 1627-1631

Kumar S, Tay-Kearney ML, Constable IJ, et al. Internet based ophthalmology service: impact assessment. Br J Ophthalmol 2005 89: 1382-1383

Lopez-Gálvez MI, Hornero R, Acebes M, Calonge T. 2003, Teleophthalmology for diabetic retinopathy screening in a rural area of Spain. In American Telemedicine Association 8th Annual Meeting, Abstract 13.

Nathoo N, Ng M, Rudnisky CJ, Tennant MT 2010. The prevalence of diabetic retinopathy as identified by teleophthalmology in rural Alberta. Can J Ophthalmol. 45(1):28-32.

Paul P, Raman R, Rani P, Deshmukh H & Sharma T 2006, Patient Satisfaction Levels During Teleophthalmology Consultation in Rural South India. Telemedicine and e-Health Journal, vol 12, nº 5, pp. 571-578

Tang R, Morales M, Giselle R & Schiffman J, 2005, Telemedicine for eye care, Journal of Telemedicina and Telecare, vol. 11, nº8, pp. 391-19

Tennant M, Rudinski C, Hinz B, MacDonald I & Greve M 2000, Tel-Ophthalmology via Stereoscopic Digital Imaging: A pilot project. Telemedicine and e-Health Journal, vol 2 nº 4 pp 583-587

Vijan S, Hofer TP & Hayward RA 2000, Cost utility analysis of screening intervals for diabetic retinopathy in patients with type 2 diabetes mellitus, JAMA, vol 283, pp 889-896

Wild S, Roglie C, Green A, Sicree, R & King H 2004, Global Prevelence of Diabetes. Estimates for the year 2000 and projection for 2030, Diabetes care, vol 27,nº 5 pp, 1047-1053

Yogesan K, Kumar S, Goldschmith L & Cuadros J, 2006, Teleophthalmology, Springer.

3

Historiales Clínicos Electrónicos

3.1. INTRODUCCIÓN 28
3.2. EXPERIENCIAS DE HCE 39
3.2.1. EXPERIENCIAS DE HCE EN ESPAÑA 41
3.2.2. EXPERIENCIAS EN OTROS PAÍSES 51
3.3. PRIVACIDAD Y CONFIDENCIALIDAD DE DATOS 53

3.1. Introducción

En relación a este capítulo, se ha llevado a cabo la **revisión bibliográfica de 530 artículos** que tratan sobre sistema de HCE en revistas indexadas como son: *IEEE-ITB, JAMIA, Telemedicine and e-Health Journal, Journal of Telemedicine and Telecare, Methods of Information in Medicine, Medical Informatics and the Internet in Medicine, Journal of Biomedical Informatics, Journal of Medical Systems e International Journal of Medical Informatics.* Se han realizado estudios sobre la situación de los HCE en diferentes países como Holanda (Spil & Katsma 2007), Canadá y China (Noseworthy 2004), etc. En el Capítulo 8, se llevará a cabo una comparativa entre el sistema **TeleOftalWeb** y otros sistemas de HCE publicados en conferencias y revistas de investigación indexadas.

Cada vez más los registros sanitarios acumulados a lo largo de la vida de las personas involucran centros y especialistas de diferentes ciudades y países, debido a la movilidad de la población. Por esta y otras razones se debe pensar en la HC como un conjunto de información que debe poder ser consultada en cualquier momento y lugar por los especialistas médicos que lo necesiten. A lo largo de este capítulo se analizarán las razones de la evolución de la HC en papel hasta la HCE.

El incremento en el número de publicaciones sobre HCE indica que su importancia va incrementando en todo el mundo. Wen *et ál.* (2007) realizaron un estudio sobre la producción científica sobre HCE entre los años 1991 y 2005. De todas las publicaciones, 1455 (80.7%) de un total de 1803 eran artículos, seguidos de *abstracts* para reuniones. La mayoría de los artículos estaban escritos en lengua inglesa (98%) y eran de origen americano (57%). Por lo tanto, América domina la producción de HCE (57%), seguida de Europa (36%) y de Asia (4.7%). Los autores de este estudio llegaron a la conclusión de que había una relación cerrada entre el estado económico de un país y su capacidad para generar publicaciones de HCE. También hay una relación bastante importante entre la producción científica y la Investigación y Desarrollo (I+D) de un país (Wen *et ál.* 2007).

Uslu & Stausberg (2008) realizaron una revisión sistemática de la literatura científica basada en estudios publicados sobre los beneficios y los costes de los sistemas de HCE. Revisaron artículos sobre implantaciones de HCE encontrados en MEDLINE desde los años 1966 a 2004. El número de estudios analizados con detalle fue de 20. En

ellos, se observó el número de usuarios, el periodo de evaluación del sistema, una evaluación estadística del mismo, etc. Según los autores, se consigue una reducción de costes empleando sistemas de HCE (Uslu & Stausberg 2008).

En este contexto, se puede definir la HC como el conjunto de documentos que contienen los datos, valoraciones e informaciones de cualquier índole sobre la situación y la evolución clínica de un paciente a lo largo del proceso asistencial (SEIS 2003). La HC está constituida por el conjunto de documentos, tanto escritos como gráficos, que hacen referencia a los episodios de salud y enfermedad de una persona, y a la actividad sanitaria que se genera con motivo de esos episodios.

La situación actual y realidad de implantación de los HCEs en España arroja los siguientes datos (Calidad y Riesgo 2009):

- 150.000 médicos con práctica clínica.
- 800 Hospitales.
- 1000 Centros de Atención Primaria.

Solamente 50.000 de los profesionales sanitarios (aproximadamente el 30%) trabajan con HCE, el resto con HC "clásica" (en papel). Los médicos utilizan un gran número de diferentes modelos de software de HCE.

En relación a la Sanidad pública, España cuenta con 17 Comunidades Autónomas, con:

- 17 Consejerías de Sanidad diferentes.
- 17 modelos de Tarjeta Sanitaria diferentes.
- 17 Sistemas de Información Sanitaria diferentes.

En la mayoría de las Comunidades Autónomas existen diferentes modelos de HCE para Hospitales y para Atención Primaria sin conectividad entre ellos.

En cuanto a la Sanidad privada, existe un gran número de:

- Consultas médicas, Clínicas.
- Grupos de Hospitales Privados.
- Compañías de Seguros de Salud Privada.

En el mercado conviven un buen número de diferentes modelos de HCE en el Sistema de Salud Privada. No existe ninguna conectividad entre ellos y además no hay conectividad con los Sistemas de Información Sanitaria Públicos. Esto si bien se plantea como un gran problema presenta un precedente positivo para la instauración de un sistema universal de HCE a nivel global.

En la figura 6 se muestra un ejemplo de HC de la Unidad de Retinopatía Diabética del IOBA empleado en la presente Tesis. Las partes de la que consta este modelo de HC son las siguientes:

- **Anamnesis.** Parte del examen clínico que reúne todos los datos personales, hereditarios y familiares del enfermo, anteriores a la enfermedad (consiste en hacer memoria de los antecedentes). Se trata pues, de la acción previa a cualquier estudio clínico o psicosocial que trata de recoger todos los datos personales, hereditarios, familiares y del entorno del enfermo o de la persona con deficiencia, anteriores a la enfermedad o a la situación de deficiencia.
- **Exploración.** Los pasos a seguir en la exploración oftalmológica son los siguientes (Ferreiro 2003):
    - Agudeza visual (AV): siempre se tomará al menos la AV de lejos con corrección óptica.
    - Reflejos pupilares: buscar sobre todo un posible defecto pupilar aferente.
    - Paralelismo de los ejes oculares y motilidad ocular extrínseca.
    - Inspección de párpados y globo ocular: con luz ambiental o una linterna y si es posible con lupa.
    - El examen del fondo de ojo es una exploración fundamental no sólo para el enfermo oftalmológico sino también en el contexto de otras patologías, especialmente neurológicas y cardiovasculares. Para su realización disponemos de varios instrumentos: oftalmoscopio directo, oftalmoscopio indirecto, observación con lámpara de hendidura a través de lentes de contacto como la de Goldmann o de no contacto como la de Hubry, + 78 d., + 90 d. etc.

- **Otras pruebas.** Se realizan otras pruebas complementarias oportunas como son una *Optical Coherence Tomography* (OCT), ecografías, radiografías, etc. que el médico considere oportunas.
- **Diagnóstico y tratamiento.** En esta parte de la HC, el médico especificará el diagnóstico del paciente y el tratamiento a ser por el mismo.

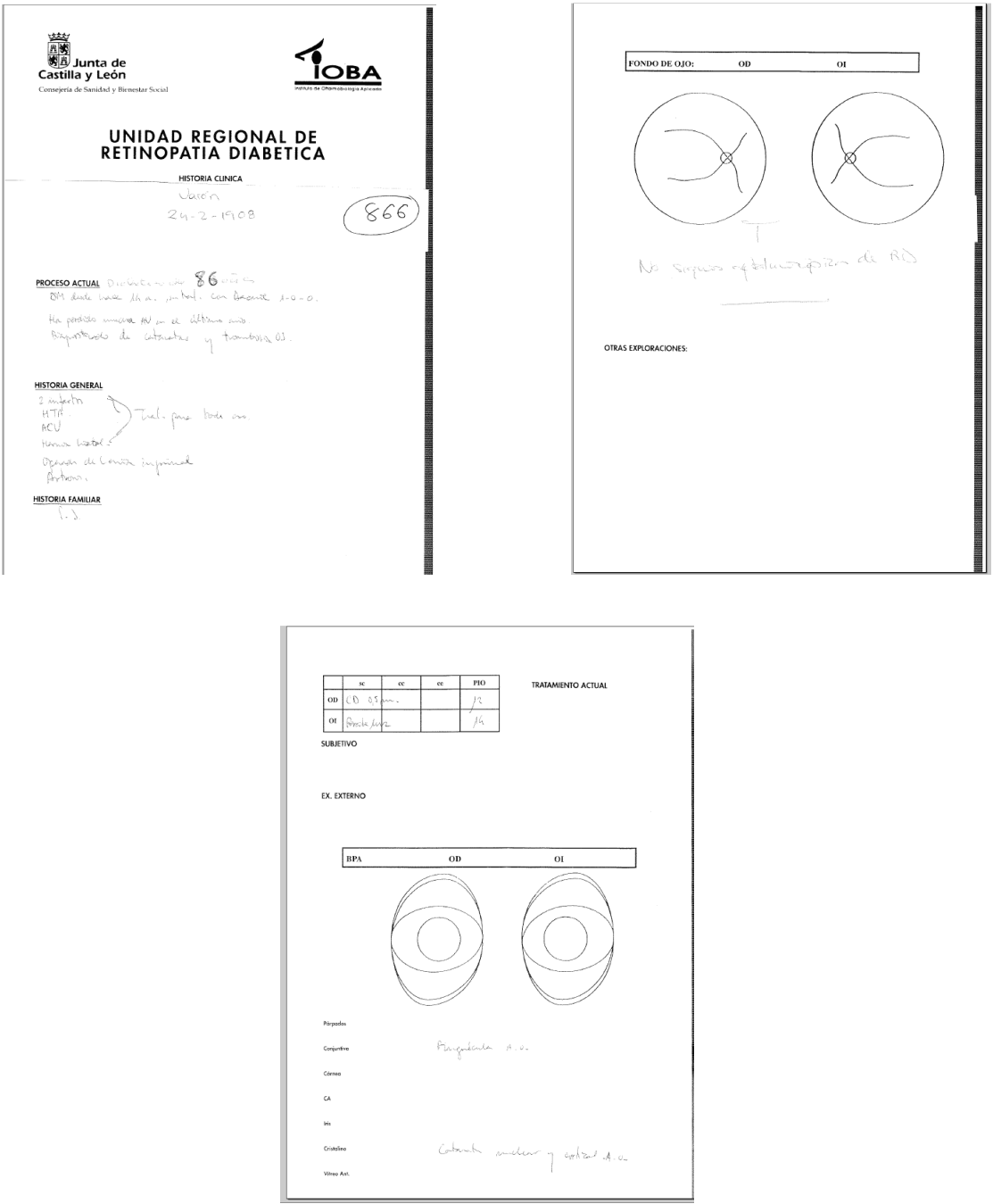


Figura 6. Modelo de HC empleado por la Unidad de Retinopatía Diabética del IOBA (a) anamnesis, (b) y (c) exploración. Fuente: propia.

A su vez, la HC se considera el documento clínico por excelencia, al ser el soporte de la información generada por el equipo sanitario y actuar como vehículo de transmisión entre los diferentes miembros que intervienen en la atención, o para otros equipos que puedan prestar atención sanitaria al ciudadano en otro lugar o tiempo. Las funciones clásicas de la HC son la asistencia, la docencia y la investigación. Desde éstas, se desarrollan otras, que estando íntimamente ligadas a ellas, tienen la suficiente trascendencia como para ser destacadas (Perís 1974):

- Asistencial. Es un documento básicamente asistencial, siendo su misión principal recoger toda la información patográfica relevante, con objeto de poder prestarle al ciudadano la atención más adecuada a su caso.
- Docente. Cuando en cada historia clínica se refleja exactamente cuál es el modo correcto de tratar cada caso clínico, explicando razonadamente las decisiones exploratorias y terapéuticas que se toman.
- Investigación clínica. Estableciendo los mecanismos precisos para localizar las historias clínicas que pertenecen a una determinada patología, o a un determinado tratamiento y como fuente de conocimiento de la propia actividad clínica.
- Investigación epidemiológica. Cuando además de conocer lo anterior, se conocen los denominadores poblacionales adecuados.
- Gestión clínica y planificación de recursos asistenciales. Sirve para la gestión clínica, la evaluación de la utilización de los recursos sanitarios disponibles y la planificación de futuras inversiones.
- Jurídico-legal. Se trata de un testimonio documental de la asistencia prestada.
- Controles de calidad asistencial. Las HC sirven para la evaluación de los objetivos científicos-técnicos.

La HC debe reunir las características de ser única para cada persona, acumular toda su información clínica y ser integrada, de forma que contenga la información de todos los contactos y episodios del paciente.

El actual soporte de la HC es susceptible de plantear una serie de problemas sobre la información en ella contenida, entre los que podemos destacar (SEIS 2003):

- El desorden habitual de los documentos que soportan la información, que conlleva a una desorganización e información fragmentada, con resultados insatisfactorios a

la hora de tratar de recuperar de una forma racional y lógica la información contenida en estos documentos.

- La falta de uniformidad de los documentos que provoca inseguridad sobre el contenido de los mismos.
- La ilegibilidad de determinadas informaciones contenidas en la historia clínica, al soportarse fundamentalmente en documentos manuscritos.
- La alterabilidad de la información, dado que con su actual soporte no se dispone de mecanismos que permitan garantizar que su contenido no sea alterado por los distintos usuarios.
- La cuestionable disponibilidad y por lo tanto inaccesibilidad de la información contenida. El soporte es único y el acceso también. No es posible acceder simultáneamente a la misma historia clínica por dos o más personas desde diferentes lugares.
- Los errores de archivado parciales (determinada información), o totales, que provocan la pérdida del valor de la misma al no estar disponible la información en ella contenida.
- El problema de espacio que origina su almacenamiento y el personal necesario para su manipulación.
- Las dificultades técnicas para garantizar la separación de los datos de identificación del paciente de los de carácter clínico-asistencial en los supuestos de acceso diferentes a los asistenciales y judiciales, tal y como exige la Legislación vigente (Falagán 2001).
- La dudosa garantía de confidencialidad, al no poder existir un riguroso control de accesos a la historia clínica fuera de su lugar de custodia.
- El deterioro del soporte documental y el riesgo de pérdida del mismo por accidentes ocasionados por agua o fuego, entre otros.

Estas dificultades son más fáciles de resolver en el caso de la HCE, cuya implantación no debe suponer una distorsión en la actividad clínica. La informatización de la historia además de facilitar la solución a los problemas anteriores, es una oportunidad para llevar a cabo la integración de la información clínica, y para revisar la organización de los servicios y de los profesionales.

Uno de los mayores retos durante décadas para los profesionales de la asistencia sanitaria ha sido el de poder compartir información entre distintas fuentes. Con la

creciente tendencia hacia la implantación de HCE, la necesidad de lograr esta meta se ha convertido en un objetivo de carácter urgente (Monteagudo 2001).

La informática y la aparición de las nuevas tecnologías son herramientas que nos ayudan a mejorar en el campo de las aplicaciones clínicas. De esta forma, el uso de un formato de almacenamiento de documentos sencillo, fiable, portable y que ocupe poco espacio es fundamental para permitir la comunicación entre distintas plataformas.

La estructura y formato de estos informes electrónicos permitirá a los médicos la opción no sólo de imprimir estos documentos en papel sino que también tendrán la capacidad de acceder al HC (en este caso centrado en la rama de la oftalmología) del paciente de una forma rápida y sencilla.

La información de salud de un ciudadano debe incluir, al menos, información procedente de los siguientes sistemas (SEIS 2003):

- BD de la tarjeta sanitaria.
- HC actuales, cualquiera que sea el lugar en que se hayan generado.
- Sistemas clínicos departamentales, como los de los laboratorios y servicios de diagnóstico por imagen.
- Programas de promoción para la salud y de prevención de la enfermedad.
- Centros sanitarios concertados o de otros servicios de salud.
- Contingencias de salud laboral.
- Sistema de receta electrónica.
- Prestaciones sanitarias complementarias.
- Sistemas de ayuda a la toma de decisiones clínicas.

La HCE incluye toda la información de salud de un ciudadano, con independencia de dónde y cuándo haya sido generada. Además, este sistema clínico debe formar parte del sistema de información del servicio de salud correspondiente, relacionándose con los sistemas de gestión económica financiera, planificación estratégica y control de gestión. Uno de los objetivos de los HCE es poder interconectar a los diferentes agentes (pacientes, médicos especialistas) en un punto común donde se reúne toda la información de salud (Kwak 2005). En la figura 7, se muestra la sanidad en la SI.

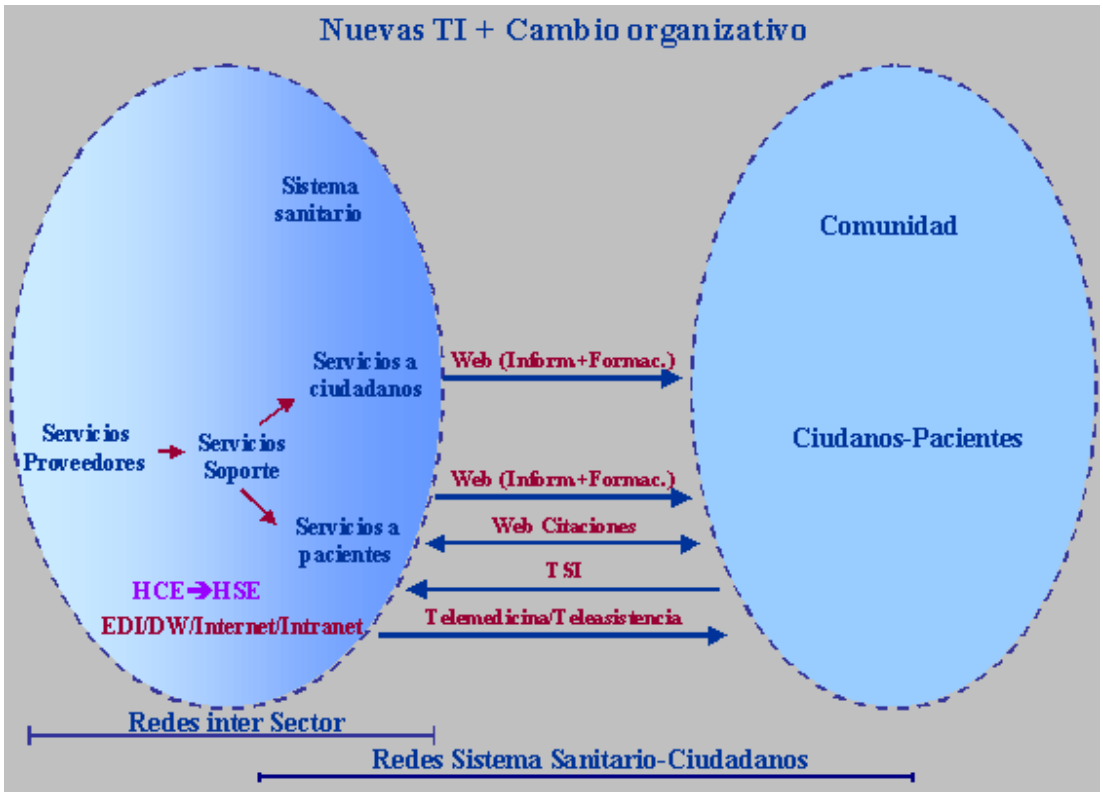


Figura 7. La sanidad en la Sociedad de la Información (SEIS 2003).

Para que todos estos sistemas se relacionen e intercambien entre sí información referida a las personas, resulta indispensable que esas personas estén identificadas de forma unívoca. Por otro lado la HCE debe contemplar unas premisas mínimas que se enumeran a continuación:

- La información relevante de interés general y que permanece, debe estar siempre presente, ser fácilmente consultada y a su vez extraída de la información general. Este es el caso por ejemplo de los antecedentes tanto familiares como personales, que perduran en el tiempo y a lo más, aumentan su contenido y son de aplicación clínica en cualquier acto médico, independientemente de donde éste tenga lugar.
- La información temporal limitada, generada en hechos aislados, puede perfectamente soportarse en documentos electrónicos cerrados. Estos documentos pueden contener las particularidades propias de cada especialidad o servicio proporcionado en cuanto a su diseño y funcionalidad, formando parte de cada episodio asistencial.
- La información generada con discontinuidad temporal, pero que en su conjunto forman una unidad en cuanto a su contenido (curso clínico, notas evolutivas, etc.),

deberá soportarse en documentos electrónicos que hagan inalterables las anotaciones previas, pero que a su vez, permitan la adhesión de nuevos comentarios o notas derivadas de los actos asistenciales.

- La información que generan las órdenes médicas, tanto de tratamientos, como de exploraciones complementarias, debe soportarse en documentos electrónicos que permitan su enlace con los servicios que deben cumplimentar esas órdenes, sin más requerimientos que su anotación en el documento correspondiente.

Las posibilidades de tener registros sanitarios localmente distribuidos, a lo largo de la vida de las personas, cada vez con más movilidad afecta a centros sanitarios de todos los países. Es por ello que la informatización de la HC se ha convertido durante los últimos años en objetivo de la mayor parte de los trabajos sobre Telemedicina (SEIS 2003). La sustitución de la HC tradicional, en soporte papel, por una HCE y la posibilidad de acceder a ella en cualquier momento y lugar responde a las siguientes necesidades:

- La existencia de un crecimiento continuo del volumen almacenado, creando graves problemas de espacio físico.
- Permitir la transferencia rápida de la información sanitaria existente de un paciente a puntos lejanos, unificando así, la HC por encima de los límites de las instituciones asistenciales.
- La posibilidad de riesgo de pérdida y deterioro de los HCs convencionales.
- Poner a disposición de los investigadores y de los planificadores sanitarios esta información, en forma fácilmente accesible y tratable.

La incipiente demanda de información adecuadamente estructurada, en combinación con el marcado desarrollo de la ciencia computacional, ha permitido el desarrollo de la HCE.

Los primeros pasos de las HCE se dieron en ambientes hospitalarios y estuvieron dirigidos hacia aquellas esferas fáciles de estructurar en aquellos instantes: diagnósticos, exámenes de laboratorio y tratamientos medicamentosos.

La narrativa clínica (antecedentes, examen físico) ha resultado por el contrario, mucho más difícil de recoger en forma estructurada, por lo que la mayoría de los esfuerzos de los grupos que actualmente trabajan en ese campo están dirigidos a perfeccionar la concepción y el procesamiento de la misma. En toda institución médica

la HC es el archivo más importante, contiene información vital para la gestión médica, administrativa y legal. Es el módulo más importante del sistema, ya que todos los demás hacen referencia a los registros que este administra. Puede o no estar presente en el sistema, pero sus bases tienen que estar instaladas para que los demás módulos puedan funcionar.

A la HC básica se pueden integrar módulos de diferentes especialidades como el de Laboratorio y estudios complementarios al cual se le pueden añadir modelos de historias de diferentes especialidades y estudios específicos según las necesidades de cada usuario. Es un sistema para el almacenamiento de datos de pacientes en el consultorio (clínicas, hospitales), de fácil manejo. Permite la realización de la HC en forma ordenada, la codificación de las patologías de acuerdo a la clasificación de la OMS y la realización de estadísticas.

Es inimaginable una institución médica actual carente de un archivo de información centralizado como la HC tal cual la conocemos, donde la información sobre cada paciente se encuentra distribuida en diversos informes separados generados por cada servicio y departamento. El acceso al total de la información acumulada por parte de un médico recién incorporado al grupo constituiría una tarea complicada. La superación de este problema sin duda constituye el justificado y definitivo triunfo del concepto de la HC como herramienta fundamental de la atención médica, sin considerar otros aspectos igualmente importantes como son su valor como documento legal e instrumento de auditoría de la práctica médica.

La acumulación progresiva de conocimiento determina el ingreso de un número creciente de técnicos y especialistas que, a veces sin verse la cara en momento alguno, colaboran en la atención de un mismo paciente. En estos casos, la HC se transforma en el único medio de comunicación entre cada uno de los integrantes de este equipo virtual. Si bien la informatización de las HC no resuelve, por sí sola, la totalidad de la problemática del manejo de la información hospitalaria, en muchos casos ofrece alternativas que permiten reducirlos en su expresión o consecuencias, lo cual torna deseable su implementación.

De entre los muchos beneficios que se pueden obtener de los HCE destacamos los siguientes:

- Continuidad en tu atención médica.
- Intercambio y disponibilidad de información en línea las 24 horas.
- Control del perfil de salud de la población atendida.

- Legibilidad en las recetas.
- Se evita la pérdida de los expedientes en papel.
- La posibilidad de almacenar un gran volumen de información, permitiendo a su vez realizar búsquedas avanzadas atendiendo a muy diversos parámetros, como pueden ser apellidos, nombre, patologías, etc.
- El proceso de atención médica será más seguro.
- Permite programar citas.

La Tabla 1 muestra las principales ventajas de la HCE respecto a la HC en papel.

HC	HCE
Desorden y desorganización	Orden y organización
Falta de uniformidad	Uniformidad
Información fragmentada	Unidad de información
Ilegilibilidad	Legilibilidad
Alterabilidad	Inalterabilidad
Dudosa confidencialidad	Control de accesos
Cuestionable disponibilidad	Accesibilidad
Errores de archivado	Minimización del error
Deterioro del soporte	Garantía del soporte
Almacenamiento	Otro soporte
Difícil separación de datos	Fácil separación de datos

Tabla 1. La información en la HC. Fuente: propia.

Como en muchos campos de la Tecnologías de la Información (TI), la visión teórica técnica ha dado lugar a considerables expectativas, pero la realización de estas expectativas está llevando más tiempo de lo que se pensaba. Es por consiguiente fundamental analizar las ventajas, inconvenientes, oportunidades y amenazas (el llamado análisis DAFO) de la situación actual. Para la HCE, un análisis de su situación en España es el que se muestra en la Tabla 2.

Fortalezas	Debilidades
Hay muchos proyectos internacionales y europeos cuyo objetivo es implantar las HCE. Hay consenso entre los Ministerios de Sanidad europeos de que ésta es una cuestión de alta prioridad. Los profesionales de la sanidad comprenden los beneficios: que una HC de pacientes intercambiable puede facilitar la cooperación entre los profesionales, al tiempo que evita tareas y costes repetitivos.	La cooperación entre diferentes instituciones asistenciales continúa siendo difícil. La implantación de los sistemas necesarios se ve a menudo como una actividad que cae fuera del presupuesto normal de una institución. La disposición y la capacidad del personal sanitario para aceptar y gestionar el cambio se considera un obstáculo importante para implantar las HCE.
Oportunidades	**Amenazas**
Como "gestores" de su información personal, los ciudadanos podrán gestionar su propia asistencia sanitaria. No duplicar datos cuando un paciente se traslada de un centro sanitario a otro reduce el riesgo de errores en los datos y permite que las instituciones se centren en su tarea primordial de asistencia. Los intermediarios de la información y las soluciones informáticas intermedias se convertirán en una importante área de desarrollo. La diversidad de soluciones ya disponibles presenta una rica oportunidad de aprendizaje. Hay una oportunidad de cooperación estrecha en la definición de normas. La proliferación continua de aplicaciones de la sanidad electrónica probablemente provocará un movimiento hacia las estructuras de acceso compartido.	Se necesita un cambio de actitud de los profesionales: debería ser un proceso orientado al cliente. Esto exige personas que cooperen más allá de los límites de sus instituciones, para describir la HC de los pacientes de modo que convenga a todos los usuarios de la información. Si no se aborda la cuestión de la protección y la certificación de los datos, la implantación de los servicios puede encontrar resistencia.

Tabla 2. Análisis DAFO de HCE. Fuente: propia.

3.2. Experiencias de HCE

Se ha investigado acerca de las experiencias en el desarrollo e implantación de un sistema de HCEs y otros sistemas de Telemedicina similares a la tratada en esta Tesis Doctoral. Éstas no abarcan sólo el campo de la Teleoftalmología, sino que se pueden aplicar a todos los campos de la medicina.

En todas ellas los resultados son similares: hay una gran aceptación por parte de pacientes y profesionales médicos, una reducción muy importante de los desplazamientos de los pacientes y de las listas de espera, detección precoz de enfermedades, reducción de los costes por paciente, etc.

En función de sus características podemos realizar una rápida clasificación, realizando tres distinciones siguientes:

- **Sistemas *Computer-based Patient Record* (CPR).** Son los primeros sistemas de telemedicina. Aunque la idea surgió en los años 80, los CPR empezaron a adquirir importancia en el momento que el *Institute of Medicine* (IOM) publicó: *The Computer Based Patient Record: An Essential Technology for Health Care*, en 1991. Lo que el IOM proponía eran unos sistemas que capturaran directamente el historial médico en papel en formato de una imagen, para posteriores consultas.
- **Sistemas *Electronic Medical Record* (EMR).** Representan la actual generación. Los sistemas EMR engloban a los sistemas CPR, extendiendo su definición. Han sido típicamente pensados como sistemas propietarios diseñados por una organización para prestar servicios específicos de creación y gestión de historiales médicos, en formato electrónico. De esta manera, los sistemas EMR proporcionan una cierta interacción y acceso en tiempo real a los datos, en la organización en los que están implantados.
- **Sistemas *Electronic Health Record* (EHR).** Implican la superación de los anteriores sistemas, en el sentido de que operan más allá del nivel de un sistema propietario de información. En este caso, tanto los datos, como su intercambio, y los módulos de la arquitectura estarán estandarizados, siguiendo unas directrices generales. En esta categoría se encuadraría el anteriormente citado, borrador de estándar ISO/TC 215, en el cual se sientan las bases de este tipo de sistemas. De esta manera, un comité del *Institute of Medicine of the National Academies* ha identificado un conjunto de ocho funciones fundamentales que los sistemas EHR deberían ser capaces de realizar, de manera que se alcanzara una mayor seguridad, calidad y eficiencia en la práctica clínica (IOM 2008). Esta lista ha ayudado al organismo HL7 a diseñar el mencionado borrador de estándar ISO/TC 215. Estas ocho funcionalidades básicas serían: definición de la información, o datos médicos, gestión de resultados, gestión de instrucciones, soporte de

decisiones, definición de la comunicación y conectividad electrónica, especificación del apoyo a los pacientes, y definición de los procesos administrativos y de información.

Algunas experiencias de este tipo se han llevado a cabo en centros como el *Baylor College of Medicine* en el cual se ha estimado que un uso de tan sólo 100 horas anuales justificaría la inversión en equipos de 36.000 €, en cada centro. En el hospital Beaumont (Dublín), por ejemplo, se ha desarrollado un repositorio de HCE basado en XML y el estándar CDA del HL7 que permite almacenar todas las consultas con los pacientes (Beaumont 2008). Además todos los participantes en el proceso sanitario comparten en tiempo real toda la información relativa a los pacientes, lo cual ha permitido al hospital mejorar su respuesta y adaptarse más fácilmente a las necesidades del cuidado sanitario. De forma similar el hospital de la ciudad de Poole ha implementado un sistema de HCE con un repositorio XML que actualmente asciende a 12 millones de documentos entre resultados de pruebas realizadas a pacientes, informes de atención primaria e informes de facturación. Otra experiencia que cabe destacar es la realizada en el hospital de Alberta, donde se ha realizado una aproximación a un sistema de HCE basado en un interfaz HL7 y la arquitectura CDA para la historia clínica electrónica (Alberta 2008). El sistema está constituido por una red de información que enlaza físicos, farmacéuticos, hospitales, y otros profesionales de la atención sanitaria, a lo largo de esta provincia. Permite a estos profesionales ver y actualizar la información sanitaria, tal y como la referente a alergias, prescripciones, y pruebas de laboratorio. El sistema de HCE también implementa herramientas como alertas de interacción entre medicamentos para prevenir prescripciones inadecuadas, una base de datos con un listado de todos los medicamentos y sus dosis normales, o un repositorio de enlaces a la información de soporte adicional.

3.2.1. Experiencias de HCE en España

El número de iniciativas de globalización de los HCEs son abundantes en todo el mundo. En España el **Plan de Calidad de la Sociedad Nacional de la Salud** merece especial atención desde su objetivo de implementación de los HCE a nivel nacional (Sistema Nacional de Salud 2009). Así se espera que para 2011 la integración del sistema de HCE sea una realidad y se encuentre funcionando en las 17 Comunidades Autónomas (IDG Communications 2009).

La implantación de un sistema global de este calibre requiere de unas pruebas previas en busca de la posterior adaptación y una mayor aceptación entre los futuros usuarios. Por esa razón, en los últimos años el número de aplicaciones para el intercambio de HCEs ha proliferado entre las distintas especialidades (Serrano et ál. 2007).

3.2.1.1. La Sanidad Electrónica en el territorio español

La Federación Española de Empresas de Tecnología Sanitaria (Fenin), en colaboración con la Fundación Observatorio de Prospectiva Tecnológica Industrial (OPTI), ha llevado a cabo un estudio de prospectiva **"e-Salud 2020"** en el que se dan a conocer los avances que, con la incorporación de las TIC, transformarán en los próximos 15 años el sistema sanitario español. La sanidad electrónica (e-Salud) será la revolución más importante en el campo de la salud desde la implantación de la medicina moderna y una oportunidad para el cambio y la mejora de la gestión sanitaria.

En el actual modelo sanitario español, los profesionales, hospitales, clínicos, compañías farmacéuticas, así como las empresas de tecnología sanitaria operan de forma prácticamente aislada. La provisión sanitaria está muy segmentada y, como resultado, la asistencia sanitaria al paciente (consumidor final) es lenta y costosa y el sistema sanitario en su conjunto, inefectivo e ineficiente.

Según datos de Naciones Unidas, la proporción de personas mayores de 60 años se duplicará entre 2000 y 2050, y la proporción de niños se reducirá un tercio. Asimismo, las enfermedades crónicas representarán más del 60% de todas la enfermedades mundiales para el 2020. Ante este panorama, la e-Salud se convierte en un elemento clave y necesario para establecer en un futuro una buena gestión sanitaria.

Las conclusiones más importantes son:

- **Hacia un nuevo modelo sanitario.** La incorporación de las TIC al sistema sanitario producirá una transformación progresiva del actual modelo, hasta evolucionar en un modelo completamente integrado y centrado en el paciente, que verá simplificada y mejorada su relación con el amplio abanico de profesionales al cuidado de su salud. El paciente, cada vez más y mejor informado, demandará nuevas y más avanzadas aplicaciones de la tecnología, lo que acelerará la implantación de los sistemas de e-Salud que funcionarán por y para él. El ciudadano tendrá a su alcance información imparcial y contrastada

sobre los servicios prestados por los diferentes centros, podrá decidir cual le interesa en función de su criterio de calidad.

- **Ayuda al diagnóstico.** La incorporación de las TICs a la medicina dará paso a importantes avances en los métodos de tratamiento y diagnóstico. La medicina dejará de estar centrada casi exclusivamente en el tratamiento de enfermedades, para pasar a un modelo orientado a la prevención. Será habitual el uso de sistemas de diagnóstico por imagen sin soporte físico (imágenes digitales), y los centros sanitarios dispondrán de sistemas de transmisión de imágenes de alta resolución que permita el intercambio seguro de las mismas.
- **Telemedicina.** Aparecerán en el mercado una oferta de servicios y productos que mejorarán considerablemente la vida del ciudadano tales como dispositivos para el diagnóstico y análisis (*biochips* y biosensores), que incrementarán el grado de autonomía de pacientes dependientes o crónicos. En el área de los servicios, la implantación de la sanidad electrónica supondrá la creación de centros sanitarios virtuales de atención al ciudadano, desde donde, de manera remota, se analizarán, administrarán y distribuirán los servicios y prestaciones demandados por el paciente.
- **Informatización de los servicios al paciente.** La posibilidad de informatizar todos los procesos administrativos y burocráticos del sistema sanitario es una de las ventajas que aportará la e-Salud. Se generalizará el uso de la receta electrónica desde la prescripción hasta la dispensación del medicamento al paciente. Todos los historiales clínicos serán electrónicos y accesibles a todos los equipos médicos del sistema sanitario. También se generalizará la utilización de tarjetas inteligentes que contengan el HCE del paciente o la información más relevante del mismo.

3.2.1.2. Experiencias de HCE en Andalucía

Gracias a la Consejería de Salud de la Junta de Andalucía nace en 1999 el Sistema Integral de Telemedicina de Andalucía, con dos objetivos: la universalidad y la calidad de su atención. Se trata de satisfacer adecuadamente las necesidades sanitarias de todos los ciudadanos, en cualquier lugar de la Comunidad donde se produzcan. La telemedicina incide en este doble objetivo y evita que la dispersión geográfica impida a cualquier ciudadano el acceso a una atención sanitaria adecuada. Se ha creado una red integral que comunica audiovisualmente y permite la transferencia de imágenes entre Centros de Salud, Hospitales Comarcales y Hospitales Regionales de referencia

(pertenecientes al Sistema Sanitario Público de Andalucía). Todo ello a través de un Centro Integral de Comunicaciones de Telemedicina, eje central del sistema (González 2002).

Los objetivos de este sistema integral de telemedicina son: garantizar el acceso equitativo de cualquier paciente a los servicios sanitarios de los Centros Coordinadores de Urgencias y Emergencias y de Hospitales, en el momento en que lo necesite sea cual sea su ubicación geográfica, adecuar al máximo el envío de recursos (helicópteros, ambulancias, etc.) en situaciones de urgencia o de emergencia y evitar desplazamientos innecesarios del paciente al hospital.

Los resultados esperados de este proyecto son los siguientes: aumentar la capacidad de resolución de los Centros de Salud, evitar desplazamientos innecesarios de recursos y pacientes a hospitales, alcanzar un alto nivel de satisfacción del usuario, fomentar la cultura de atención integral y continuada, entre los profesionales de los diferentes niveles que actúan en el proceso asistencial y potenciar la colaboración entre los distintos profesionales sanitarios.

En la actualidad la HCE se complementa con un módulo de explotación de datos a través del cual se obtiene información pormenorizada sobre oferta de servicios en cada centro, presión asistencial, distintos tipos de consulta según profesionales, pruebas diagnósticas, interconsultas con especialistas, la utilización de servicios de urgencias y las exploraciones y derivaciones que se hacen. Del mismo modo, permite conocer de forma exacta las personas incluidas en cada servicio de Cartera y el grado de cumplimiento de las Normas Técnicas Mínimas y su comparación en diversos centros de atención (Ledesma & Garavís 2004).

Este conjunto de elementos dan como resultado un Sistema de Información Sanitario que es una red única y no la suma de varias redes locales y en el cual, la historia de salud de cada ciudadano es única sea cual sea el lugar donde contacta con el sistema sanitario, tiene una parte común para la atención primaria y la especializada y otros elementos específicos para cada unidad, pero accesibles para todos los profesionales dependiendo de los permisos de acceso. Este sistema de Información (véase figura 8) es la Historia de Salud Digital del Ciudadano (HSDC) y lleva aparejado además lo siguiente (Junta de Andalucía 2008):

- Una BD única de usuarios (BDU).
- Un sistema de comunicaciones fuerte para conectar una Intranet.
- Corporativa en configuración de Entorno Virtual de Trabajo.

- Una red corporativa de comunicaciones de la Junta de Andalucía.
- Los cambios que se introducen en ella son centralizados y generalizados. Las tablas maestras también lo son y las salidas de los indicadores son estándares para todas las unidades asistenciales para poder hacer comparaciones.

Figura 8. Sistema de Información Sanitario (Junta Andalucía 2008).

La HSDC comprende:

- Una unidad de atención al usuario que permite resolver "in situ" las demandas administrativas tales como cambio de médico, desplazamientos temporales y citaciones, entre otras opciones.
- Una historia orientada por paciente en lugar de con el concepto clásico de orientación por problemas y en sucesión cronológica. Está estructurada a modo de navegador a través del paciente, lo que facilita la gestión clínica de los profesionales y favorece el planteamiento de planes de futuro sobre el paciente que incluyen la imputación de costes.
- El módulo de explotación de datos es común a otras aplicaciones, funciona como sistema independiente y se alimenta de los datos de la HSDC y de otros sistemas, como farmacia, cuyas informaciones pueden cruzarse entre sí.
- El proyecto "Receta XXI" permite la gestión de las prestaciones de fármacos a los usuarios, pone en conexión la BDU, la historia clínica y la oficina de farmacia

usando como llave de dispensación la Tarjeta Sanitaria Individual mediante la cual se accede a los datos centralizados de los usuarios.

- Una oficina virtual que permite a los ciudadanos realizar gestiones sanitarias a través de Internet.

Todo lo anterior es lo que compone el **proyecto DIRAYA** y se encuadra dentro de una red regional de información sanitaria, el llamado entorno virtual EVISAND. Las redes de información sanitaria regionales implican el enlace de los centros sanitarios vía telemedicina y vía servicios basados en Web para conectar a profesionales y pacientes diseminados por un área geográfica y que no podrían ser atendidos sin la tecnología.

EVISAND está operativa desde el año 2000 en tres provincias de Andalucía (Almería, Jaén y Málaga) que agrupan a 2,5 millones de habitantes. Conecta entre sí a 21 centros de salud, 13 hospitales, 3 centros coordinadores de emergencias y 4 residencias de tercera edad.

Un ejemplo de telemedicina en Andalucía es el del El Hospital Virgen Macarena, de Sevilla que cuenta con una "*Teleunidad de Lesiones Pigmentarias*". En este hospital los dermatólogos pueden realizar, a distancia, el *screening* de las lesiones pigmentarias que les envían los médicos de Atención Primaria, a partir de imágenes captadas, con cámaras digitales, en el centro de A. Primaria y enviadas, en formato digital y por "correo electrónico". Este sistema de telediagnóstico utiliza una arquitectura técnica que funciona en modo asíncrono, en él que la información, datos e imagen, es recogida y almacenada en un primer momento, y, posteriormente en forma diferida, enviada al dermatólogo para su estudio. Tras el diagnóstico, el especialista elabora un informe que es remitido al médico que pidió la teleconsulta.

Este método, a diferencia de los sistemas de telemedicina en tiempo real, no precisa de videoconferencia, ni hace necesario que coincidan paciente, médico y especialista, por lo que supone menor complejidad técnica y organizativa, permitiendo al especialista una mejor gestión de su tiempo y analizar más imágenes por sesión. Según sus responsables, la experiencia en los primeros meses de funcionamiento de la unidad es muy satisfactoria.

Algunos de los resultados que se pueden obtener son los siguientes:

- Las consultas en línea representan el 80% de la actividad frente a la asistencia en emergencias que supone el 20%.
- Las consultas en línea tienen una duración media de 17 minutos y el 76% de los pacientes dijo que ya no necesitó trasladarse al hospital de referencia.
- Del 24% restante, que sí se trasladó, el 82% precisó procedimientos diagnósticos adicionales que no pudieron ser suministrados en el lugar de la consulta.
- El grado de satisfacción con el servicio ha sido muy alto tanto entre los médicos como entre los pacientes.

3.2.1.3. Experiencias de HCE en Castilla y León

El IOBA de la Universidad de Valladolid lleva varios años realizando una experiencia pionera en el campo de la oftalmología en Castilla y León. La idea surgió con el propósito de acercar los recursos sanitarios que ofrece el IOBA a poblaciones que tienen difícil acceso a los mismos, dentro de una campaña para la prevención de la ceguera por RD. Dicha enfermedad está considerada como la causa más frecuente de ceguera en los países industrializados.

La experiencia llevada a cabo consiste en un sistema de transmisión de imágenes digitalizadas instalado en los centros de atención primaria de dos poblaciones rurales de la provincia de Ávila. Dicho sistema es manejado por los médicos de atención primaria, que envían imágenes de fondo de ojo al IOBA. Allí son interpretadas por un oftalmólogo y éste remite sus recomendaciones por correo electrónico.

Debido al éxito de la experiencia, aparece la necesidad de utilizar un sistema que organice tanto las imágenes como la información relativa a éstas, a fin de facilitar su gestión y consulta. Así surge la herramienta **TeleOftalWeb** de la cual trata la presente Tesis, cuyo objetivo fundamental es el almacenamiento e intercambio de HCEs e imágenes médicas entre los profesionales médicos.

El IOBA decidió iniciar una experiencia pionera en este campo y utilizar las nuevas tecnologías para acercar sus recursos sanitarios a las poblaciones que tenían un difícil acceso a los mismos. Para llevarla a cabo se instaló un sistema de transmisión de imágenes digitalizadas montado en una cámara instalada en un retinógrafo en los centros de salud de *Piedrahita y El Barco de Ávila*. Dicha cámara es manejada por médicos de atención primaria que han recibido la capacitación necesaria por parte de los miembros del IOBA. Estos dos centros fueron seleccionados tanto por sus características orográficas como de población, que plantean serios problemas de

desplazamiento y de asistencia médica especializada a sus habitantes. Las imágenes del fondo de ojo son enviadas al IOBA, donde un oftalmólogo se encarga de leerlas y remitir por correo electrónico el informe y las recomendaciones correspondientes.

En los años que el proyecto ha estado en marcha se ha comprobado que el porcentaje de pacientes que se someten a revisiones periódicas al acercarse los servicios de atención especializada es de más de un 80% donde antes era sólo un 20%. A esto se le añade la ventaja de evitar una gran parte de los desplazamientos de la población, puesto que la exploración se realiza en su localidad de origen y los resultados se reciben también allí. Sólo los casos que requieran de una atención especializada para la que no se dispone de medios en los centros de atención primaria, serán remitidos al oftalmólogo. Esto influye directamente en la reducción de las listas de espera para la consulta oftalmológica.

3.2.1.4. Experiencias de HCE en Navarra

En Navarra se ha puesto en marcha un Proyecto de prevención de la RD por teleconsulta. Se trata de un modelo de teleconsulta que será ampliado a otros procesos sanitarios para que en muchos casos ni siquiera será necesario el desplazamiento del paciente al Centro de Consultas.

Actualmente, más de 400 pacientes se han beneficiado ya del proyecto de prevención de la RD, puesto en marcha durante el mes de octubre del pasado año por el Servicio Navarro de Salud-Osasunbidea, en el marco de una decidida apuesta por la Telemedicina.

Este sistema que ayuda a la mejora de circuitos y listas de espera en las revisiones de Oftalmología, en el que participan de forma coordinada los dos servicios de Oftalmología de los hospitales Virgen del Camino y Hospital de Navarra, el Centro Príncipe de Viana y los centros de Atención Primaria. El nuevo sistema también reduce los tiempos de atención y garantiza las revisiones necesarias a las personas afectadas.

Por este sistema los pacientes son citados desde los centros de salud de Atención Primaria y Endocrinología, en el contexto de los programas de revisión (prevención) de la RD. Con este fin se genera una agenda de 20 citaciones diarias, localizada en el Centro Príncipe de Viana, para pacientes diabéticos que requieren revisión de fondo de ojo. Para este fin se utiliza el retinógrafo ubicado en este Centro de Consultas. Este

programa, que comenzaba a introducirse en octubre de 2005 con la atención de 20 pacientes por día y cuatro jornadas asistenciales por semana, permitirá atender a más de 2000 pacientes/año.

Los beneficios que se han conseguido por este sistema son los siguientes:

- Se realiza la revisión de los pacientes diabéticos por un procedimiento de imagen digital de alta calidad (valorada y reconocida por los oftalmólogos) y fiabilidad.
- Se reducen visitas al oftalmólogo por este motivo.
- Se implanta un nuevo modelo de coordinación entre Atención Primaria y Especializada con un nuevo modelo de interconsulta acortando circuitos clínicos.
- Se pueden seleccionar mejor los pacientes que realizarán un *screening* fiable para la detección precoz de RD.
- Se utiliza una tecnología que permitirá en un futuro no desplazar al paciente para realizarse esta prueba, dado que el oftalmólogo sólo requiere ver la imagen y realizar un informe.
- Se mejora la captación de pacientes en riesgo y garantizar la revisión periódica de los mismos. Actualmente se estima, que sólo alrededor de un 30% de diabéticos acude al oftalmólogo para revisión de fondo de ojo.
- Se ha realizado el proyecto de forma coordinada entre los dos servicios de ambos hospitales con la participación de 6 profesionales médicos.
- El sistema ha sido valorado muy positivamente por el resto de oftalmólogos dado que este modelo asistencial permite gran fiabilidad y reduce consultas innecesarias de revisión.
- Este modelo tecnológico está avalado por la literatura médica como método fiable de *screening* de RD.

3.2.1.5. Experiencias de HCE en Extremadura

En Extremadura se ha iniciado proyecto Jara que afectará a 14 hospitales públicos de la región, 107 centros de salud, más de 300 consultorios, y casi 15.000 profesionales, y supone una iniciativa pionera en cuanto a su alcance y a su carácter global y de transformación integral que cubrirá todos los procesos de actividad del Servicio Extremeño de Salud.

Como resultado del proyecto, la sanidad extremeña dispondrá de:

- La HCE única por paciente, accesible por todos los profesionales del Servicio, según los niveles de autorización que se establezcan para cada uno de los perfiles de usuarios.
- Un sistema para la gestión y administración de pacientes único e integrado en todos los niveles de prestación.
- La estación de trabajo para el profesional, desde donde podrá cursar órdenes y peticiones así como recibir, consultar e incorporar en la HC los resultados e informes correspondientes (Laboratorios, Radiología, Anatomía Patológica, Farmacia y Dietética).
- Un sistema de control de gestión que permitirá a los profesionales (gestores y clínicos) conocer la evolución de los indicadores de asistencia, calidad y gestión.
- Un sistema para la Gestión de Recursos Humanos, Gestión de Compras y Aprovisionamiento y Gestión Económico-Financiera, integrados con la gestión asistencial.

3.2.1.6. Experiencias de HCE en la Comunidad Valenciana

En la Comunidad Valenciana se usa el sistema Abucasis II de informatización de los centros sanitarios de toda la región. Ya más de 200 los centros de salud que cuentan con este sistema que está permitiendo implantar la HCE única por paciente en el ámbito ambulatorio, así como la prescripción, también electrónica, de medicamentos y prestaciones farmacéuticas.

Esto significa que 6.452 profesionales sanitarios (el 72% del total) ya están utilizando esta nueva herramienta de trabajo y que la población cubierta por el nuevo sistema asciende a 2.404.687 personas, lo que supone cerca del 50% de toda la población de la comunidad autónoma. Este sistema también supone la integración de los servicios de citación de Atención Primaria y Especializada, así como la creación de un registro nominal de vacunas, para saber las vacunas que se han suministrado a cada paciente en cualquier momento.

3.2.1.7. Experiencias de HCE en el País Vasco

El proyecto puesto en marcha en el País Vasco tiene también como elementos básicos:

- La integración de los Sistemas de Información.
- La HCE Única por paciente: el proyecto bautizado *e-Osabide* cubre el diseño, desarrollo e implantación del Sistema de Gestión Asistencial Global.

- La Tarjeta Sanitaria Electrónica como llave.
- La Receta Electrónica.
- El Portal de Salud para ciudadanos, trabajadores y terceros, sin previsión de servicios transaccionales sino solo de información.

El proyecto *e-Osabide* desplegado por el Servicio Vasco de Salud no difiere en esencia del proyecto Andaluz.

3.2.2. Experiencias en otros países

3.2.2.1. Experiencias de HCE en el Reino Unido

En el Reino Unido se encuentra uno de los proyectos más avanzados del mundo. Con un profundo compromiso político por parte del gobierno británico, este ha iniciado un plan de informatización de la salud para que todos sus habitantes dispongan de su HCE dentro de los próximos. La prioridad del *National Health Service* (NHS) Británico es un ambicioso plan de informatización de toda su red estatal de salud a desarrollar en los próximos años (Burns 1998).

Han dividido el país en regiones, licitando las áreas entre distintos prestadores, habiendo asignado a *British Telecom* el área de Londres y la construcción y operación de la red que interconecta a todo el país para que, en tiempo real, médicos, instituciones medicas y pacientes, accedan y dispongan de la información a través de la Web.

BT Health, rama de Consultoría en Salud de *British Telecom*, está implementando el proyecto de HCE Única a nivel nacional. Este programa, denominado *National Programme for Information Technology* (NPfIT), es la responsabilidad de *Connecting for Health*, una agencia ejecutiva del Ministerio de Salud británico, creada en el año 2002 para modernizar el sistema de salud ingles. NPfIT crea una infraestructura que mejora el cuidado de los pacientes, incrementando la eficiencia y la efectividad de los médicos y el *staff* del *National Health Service* (NHS).

Los pilares del proyecto son:

- Creación del NHS *Care Records Service* (Servicio de Registros de Salud del NHS) para poder compartir HC a nivel nacional de manera segura con médicos y profesionales de la salud dependiendo de su autoridad. Esta BD Transaccional servirá a 50 millones de pacientes.

- Reserva de turnos en cualquier hospital del país por Internet.
- Transmisión Electrónica de Prescripciones entre Farmacias, Hospitales y consultorios.
- *New National Network* (N3): nueva infraestructura informática y de comunicaciones que pueda cubrir las necesidades corrientes y futuras a través de una nueva red nacional segura de salud.
- *Picture Archiving and Communication Systems* (PACS): almacenamiento y Transmisión de todo tipo de imágenes digitales y videos entre todas las organizaciones de salud.

Los pacientes no necesitaran volver a repetir la información acerca de su HC en diferentes etapas del proceso de cuidado de la salud (Atención Primaria, Secundaria, etc.) y esto mejorará la calidad de atención a los pacientes.

Esto revolucionará el cuidado de la salud y transformará el NHS en una de las organizaciones más avanzadas del mundo, estandarizando de la información de salud a nivel nacional.

3.2.2.2. Experiencias de HCE en Irlanda

El hospital de Beaumont en Irlanda se ha embarcado en la tecnología para tener un HCE para almacenar la información de los pacientes, basado en plataforma de BEA Web*Logic 8.1*. Gracias a esto el hospital puede registrar todas las consultas realizadas con el paciente. Esta arquitectura se basa en una *Service Oriented Architecture* (SOA) que permite que el hospital se adapte a las necesidades sanitarias de hoy día. Gracias a este sistema se puede ahorrar hasta un 50% de tiempo, que se pierde mediante el uso de las herramientas tradicionales.

El Departamento de Epilepsia del hospital identificó la necesidad de crear una BD de los registros de los pacientes que serían utilizados para propósitos de investigación. Los requisitos claves fueron: crear un registro electrónico de la salud para pacientes de epilepsia, el registro de todos los encuentros con un paciente y proporcionar información consolidada para el Departamento de Epilepsia. También se propuso mantener los principales procesos en el Departamento de Epilepsia, y continuar proporcionando las visitas de los pacientes, además de incluir visitas vía remoto (Beaumont Hospital 2004).

Buena parte de la tecnología usada en el proyecto es *Java 2 Enterprise Edition* (J2EE) dando como resultado una infraestructura que facilita la sencilla implementación

de arquitecturas complementarias de aplicación e integración, tal como una arquitectura de Servicio orientado.

La solución tiene dos sistemas emparejados: un servidor de HCE y un sistema de administración para la enfermedad de la Epilepsia. Este servicio será usado en Web usando los estándares de HCE tales como HL7, CEN, y openEHR.

3.2.2.3. Experiencias de HCE en EE.UU

Cada vez son más los hospitales y hogares de Estados Unidos que utilizan las tecnologías de la medicina a distancia (Telemedicina) y los HCE.

La *American Telemedicine Association* (ATA) define esta práctica como "el intercambio de información médica de un sitio a otro a través de comunicaciones electrónicas" en el que se encuentran las fotografías digitales, los HCE, las videoconferencias, etc.

Así, las radiografías, electrocardiogramas o signos vitales de un paciente en Albany, capital del estado de Nueva York, pueden ser leídos e interpretados por un cardiólogo o radiólogo en otra ciudad, de la misma manera que si tuviera al paciente en su consultorio. Incluso las heridas, dermatitis, otitis o dolores de garganta, que requieren del diagnóstico en persona, son ahora chequeados con cámaras digitales en un servicio por Internet que ofrece acceso a médicos que están de turno las 24 horas del día.

Hay que indicar que donde más se emplea la telemedicina en EE.UU. es en zonas rurales, asilos de ancianos, cruceros, prisiones, el Ejército y las misiones de la NASA. Estudios de varias agencias del departamento de Salud y de la ATA estiman que unos 200 programas de telemedicina han sido instaurados en unos dos mil centros médicos de EE.UU., número mayor aún si se consideran las escuelas y otras instituciones.

La telemedicina tiene especialidades como la "*telepediatría*" y la "*telesiquiatría*", que se practican a través de videoconferencias, y explora nuevos campos, como el diagnóstico de cáncer de mama, diabetes y cuidado dental. Los hospitales que la utilizan sostienen que el sistema es efectivo para reducir los costos de atención al paciente y agilizar la respuesta médica.

El Hospital de Niños de Washington aumentó sus ingresos un 55% y se ahorró 250.000 dólares en un año en costos de consultas directas por eco-cardiología gracias a la telemedicina, de acuerdo con el informe anual 2005 de la ATA. Con las conexiones de las telecomunicaciones cada vez más ágiles es posible enviar y recibir fotos y rayos

X mucho más rápido. Un dermatólogo puede consultar un caso a través de la Web y dar un diagnóstico sin tener que hacer grandes desplazamientos.

Un estudio de la Universidad de Arkansas indica que el 41% de los adscritos al Proyecto de Telesalud en Hospitales Rurales no hubiese recibido atención médica de no haber sido por la telemedicina, debido a los altos costos de viajar a los centros de salud de las ciudades y a la escasez de médicos en sus comunidades.

También la telemedicina ha llegado a las Unidades de Cuidados Intensivos. Diversas organizaciones proveedoras de asistencia hospitalaria, en los Estados Unidos, han comenzado a implantar en sus hospitales, sistemas de monitorización y control, a distancia, de los pacientes críticos; el objetivo es proporcionar ayuda complementaria a los centros que no cuentan con especialistas "in situ", durante las 24 horas. Para ello, se utilizan plataformas tecnológicas que permiten la monitorización y visualización de los pacientes, el acceso a la información clínica y la toma de decisiones. Éstas iniciativas, están motivadas en factores relacionados con la seguridad de los pacientes, con la disponibilidad de especialistas, y en otros de carácter laboral y económico. *Leapfrog*, la coalición de organizaciones proveedoras de asistencia sanitarias en los Estados Unidos, creada para estimular iniciativas de mejora de la seguridad en la asistencia y de reducción de errores prevenibles, ha introducido la existencia *de "Tele-intensivistas*" entre sus Estándares de Prácticas Seguras para las Unidades de Cuidados Intensivos.

3.3. Privacidad y confidencialidad de datos

La HCE implica un riesgo de acceso remoto a la información del paciente que no existe con la historia en papel, al igual que riesgos de violación sistémica de la confidencialidad, es decir, de toda una población de pacientes. Sin embargo, también es importante destacar que la confidencialidad de la información en la HC tradicional es fácilmente violable, no quedando además rastros de dicho acceso: Muchas veces alcanza con tener puesta una "bata blanca" y solicitar la HC para que ésta sea entregada. Podemos definir la confidencialidad como la condición en la cual la información es compartida o entregada de una manera controlada, desde una perspectiva del personal de la Salud (Margolis 2002). La privacidad, a su vez, puede ser definida como el deseo individual de limitar la divulgación de información personal, desde la perspectiva del paciente.

Con el fin de mejorar las prácticas de seguridad y privacidad de la información clínica, es necesario tomar medidas técnicas y organizacionales. Estas medidas tienen que ser consideradas en el contexto del marco legal existente. Entre ellas están los procedimientos técnicos (identificación individual de los usuarios de los sistemas de HCE, controles de acceso al sistema de acuerdo a un perfil de usuario, seguridad física y recuperación ante desastres, protección de las comunicaciones externas, controles de auditoría, etc.) y prácticas organizacionales (comités de seguridad y confidencialidad, programas de entrenamiento para los usuarios, sanciones definidas, políticas definidas en materia de seguridad y de confidencialidad por parte de las Instituciones públicas, etc.).

Por ejemplo, en los Estados Unidos no hay legislación federal sobre este tema, excepto respecto a la información en poder gubernamental. Las leyes estatales son diversas, proveyendo protección inconsistente y se aplican a tipos limitados de información médica. Sin embargo, hay iniciativas federales en curso (Bennett, 1998).

En España, se reconoce el derecho del paciente al respeto a su privacidad. Esta privacidad incluye tanto los datos y las sustancias biológicas que permitan la identificación personal de un paciente como los datos referentes a su enfermedad (diagnóstico, pronóstico y tratamiento).

De esta forma todo el personal que trata con los datos de carácter personal de los pacientes (ficheros de cualquier tipo) o que mantiene relación laboral con ellos, y por tanto tiene acceso a información confidencial está obligado mantener el secreto de la información conocida.

No sólo está obligado por el Código Deontológico de su profesión (en el caso de los médicos y enfermeras); sino también por la legislación en materia de protección de datos y por la legislación penal.

El secreto profesional alcanza a los facultativos, enfermería, fisioterapeutas, terapeutas ocupacionales, auxiliares de enfermería, podólogos, celadores y personal administrativo, así como al personal auxiliar del Hospital.

El mantenimiento de la confidencialidad y privacidad de los pacientes implica primeramente a la HC, que debe estar custodiada de forma adecuada, permaneciendo accesible únicamente al personal autorizado.

Sin embargo, los preceptos de privacidad deben ser observados en todos los campos de la vida hospitalaria: la privacidad en el momento de la realización de la anamnesis y de la exploración física, la privacidad en el momento de la información a los familiares, las conversaciones entre sanitarios en los pasillos, el mantenimiento de la reserva adecuada de los datos de los pacientes en los controles de enfermería de las plantas de hospitalización (tablones, pizarras), las conversaciones telefónicas, los interfonos abiertos, etc.

En España, la regulación legal es la siguiente:

- Ley 14/1986, de 25 de Abril, General de Sanidad :

El paciente tiene derecho a que quede constancia por escrito de todo su proceso, (en una historia clínica), así como a recibir un Informe de Alta al finalizar su estancia hospitalaria y al informe de consulta externa.

La HC se identifica con un número único por cada paciente y debe ser almacenada de forma centralizada en un único lugar.

- *Real Decreto 63/1995, de 20 de enero, sobre Ordenación de prestaciones sanitarias del Sistema Nacional de Salud: El paciente tiene derecho a la comunicación o entrega, a petición del interesado, de un ejemplar de su historia clínica o de determinados datos contenidos en la misma, sin perjuicio de la obligación de su conservación en el centro sanitario.*

  El Sistema Público debe preservar y garantizar la confidencialidad de los datos contenidos en la HC.

  "El paciente tiene derecho a la confidencialidad de toda la información relacionada con su proceso y con su estancia en instituciones sanitarias públicas o privadas que colaboren con el sistema público".

El acceso a la HC sin autorización, en perjuicio de un tercero, está tipificado como delito grave y está castigado con penas de prisión. Igualmente el profesional que revele o divulgue datos de la historia clínica será castigado con las mismas penas.

La HC debe ser un reflejo veraz del curso de la enfermedad. Los datos incluidos entre sus documentos no pueden ser alterados, falseados ni simulados, lo cual constituiría un delito de falsedad documental que está penado por el mismo Cuerpo Legal.

- *Ley Orgánica 15/1999, de 13 de diciembre de Protección de Datos de carácter personal(LOPD).*
- *Ley 41/2002, de 14 de noviembre, básica reguladora de la autonomía del paciente y de derechos y obligaciones en materia de información y documentación clínica.*

La Ley 41/2002 de 15 de noviembre de la autonomía del paciente y los derechos y obligaciones en materia de información y documentación clínica (España), no especifica a quién pertenece la historia clínica, por el contrario sí que indica cuales son las instituciones asistenciales las encargadas de custodiar, vigilar, regular y facilitar el acceso a ellas.

No es, tampoco, propiedad del equipo facultativo, pese a ser artífices materiales de ella. Sin embargo, al igual que el paciente, tendrán el derecho a ser conservada su intimidad en caso de revisión de la misma por parte del paciente.

Tampoco, se indica en la ley, que **la HC sea propiedad del paciente**, pese a ser el agente causal de ella.

4

Estándares de Historiales Clínicos Electrónicos

4.1. INTRODUCCIÓN 61
4.2. OBJETIVOS DE LOS ESTÁNDARES DE LOS HISTORIALES CLÍNICOS ELECTRÓNICOS 61
4.3. ALGUNOS PROBLEMAS DE LOS SISTEMAS DE INFORMACIÓN SANITARIOS 62
4.4. ESTÁNDARES Y NORMAS 62
4.5. HEALTH LEVEL 7 (HL7) 66
4.5.1. INTRODUCCIÓN 66
4.5.2. ANTECEDENTES DE HL7 68
4.5.3. OBJETIVOS Y MISIÓN DEL HL7 69
4.5.4. FORMATO DE HL7 70
4.5.5. ESTÁNDARES 71
4.5.6. MODELO BÁSICO DE TRANSACCIONES HL7 71
4.5.7. MODELO FUNCIONAL DE HL7 72
4.5.8. CLINICAL DOCUMENT ARCHITECTURE (CDA) 75
4.5.8.4. DISEÑO DE UN DOCUMENTO CDA 80
4.6. DIGITAL IMAGING AND COMMUNICATIONS IN MEDICINE (DICOM) 93
4.6.1. INTRODUCCIÓN GENERAL 93
4.6.2. OBJETIVOS DE DICOM 99
4.6.3. DICOM Y OTROS ESTÁNDARES 100
4.6.4. RELACIÓN ENTRE LAS PARTES DEL ESTÁNDAR DICOM 102
4.6.5. PARTES DEL ESTÁNDAR DICOM 105
4.6.6 . SUPLEMENTO 91: CLASES SOP DE IMÁGENES FOTOGRÁFICAS OFTALMOLÓGICAS 111
4.7. INTERNACIONAL ORGANIZATION FOR STANDARDIZATION (ISO)/TC 215 113
4.8. EUROPEAN COMMITTEE FOR STANDARDIZATION / TECHNICAL COMMITTEE (CEN/TC) 251 114
4.9. COMMON OBJECT REQUEST BROKER ARCHITECTURE (CORBA) 119
4.10. GOOD EUROPEAN HEALTH RECORD (GEHR) 121
4.10.1. LOS REQUISITOS DEL PACIENTE 121

5.10.2. DESCRIPCIÓN DE LA ARQUITECTURA DE GEHR 122
5.11. INTEGRATING THE HEALTHCARE ENTERPRISE (IHE) 122
5.12. OPENEHR 124
5.13. ESTÁNDARES DE SEGURIDAD EN HCE 125

4.1. Introducción

En sanidad como en otros campos de aplicaciones telemáticas existe una corriente general de demanda de los usuarios hacia sistemas abiertos, distribuidos, interconectados e interoperables, con un alto grado de fiabilidad y requisitos de seguridad cada vez más exigentes. La gestión integrada de los servicios sanitarios y la continuidad en los cuidados médicos requiere la adopción de mensajes, formatos, codificación y estructura de registros médicos aceptados de forma generalizada (De Moore 1993).

Es bien conocido el papel que han jugado las normas en los sectores tradicionales de la industria, aumentando la seguridad, disminuyendo los costos y favoreciendo el desarrollo de los mercados. Existen argumentos fundados a favor de considerar efectos similares en el campo de la Informática y Telemática aplicada a la Salud. Este sector está caracterizado por la fragmentación del mercado, la proliferación de aplicaciones incompatibles, los costos de desarrollo de soluciones particulares, su corto ciclo de vida, los problemas de mantenimiento, y las barreras para conseguir la integración operativa de sistemas diferentes y aislados (Monteagudo 2003).

Es necesaria la estandarización de la interfaz entre el equipamiento de telemedicina y los sistemas de telecomunicaciones (Komiya 2005).

En la presente Tesis se ha llevado a cabo la **revisión bibliográfica de 274 artículos** que tratan de estándares e iniciativas de HCE (HL7, DICOM, ISO/TC 215, CEN/TC 251, GEHR, OpenEHR, IHE, CORBA), en revistas sobre telemedicina de alto índice de impacto entre las que están: *IEEE-ITB, JAMIA, Telemedicine and e-Health Journal, Journal of Telemedicine and Telecare, Methods of Information in Medicine, Medical Informatics and the Internet in Medicine, Journal of Biomedical Informatics, Journal of Medical Systems e International Journal of Medical Informatics.*

4.2. Objetivos de los estándares de los historiales clínicos electrónicos

Actualmente nos movemos hacia la heterogeneidad de los distintos dispositivos en el mundo sanitario. Existen diferentes terminales de usuarios, sistemas de transporte, servicios y aplicaciones (Monteagudo 2003). Lo que se quiere conseguir en un sistema de información es lo siguiente:

- Extensibilidad.

- Interoperabilidad.
- Portabilidad.
- Escalabilidad.
- Disponibilidad.
- Fiabilidad.
- Seguridad.
- Costes asumibles.

4.3. Algunos problemas de los sistemas de información sanitarios

Existen una serie de problemas actualmente y que hacen que no se cumplan los objetivos anteriormente citados (SEIS 2003):

- Fragmentación del mercado.
- Florecimiento de aplicaciones diferentes.
- Corto ciclo de vida de los sistemas.
- Integración con el resto de sistemas y aplicaciones.
- Falta de compatibilidad.
- Alto coste de las soluciones particulares.

Todos los sistemas propietarios suelen ser cerrados donde las especificaciones técnicas están sujetas a secreto o bajo patentes propiedad del fabricante. Existen sistemas abiertos los cuales hacen pública toda la información y normalmente éstos están basados en estándares públicos. Gracias a que hacen pública la información, permiten el diseño por terceras partes de componentes conectables e interoperables, consiguiendo uno de los objetivos para el usuario.

4.4. Estándares y normas

En el campo de la telemedicina, es necesario adoptar por parte de todos los fabricantes una serie de normas:

- Interoperabilidad e integración de sistemas.
- Independencia de suministradores.
- Disminución de costes de operación y mantenimiento.
- Simplificación de tareas.

- Formación y calificación del personal.
- Aumento de la seguridad.
- Instrumento para la gestión de la calidad.
- Reingeniería de procesos.

Con esto se crean los estándares abiertos, que siguiendo la filosofía de compartir, se publica y todo el mundo tiene acceso a él, se adopta siguiendo un proceso de decisión abierto abaratando costes. Los derechos de la propiedad intelectual del estándar están depositados en una organización sin ánimo de lucro, que funciona con una política de acceso libre.

Siguiendo una arquitectura por capas podemos distinguir 3 niveles: **comunicación, servicios y aplicación**. Estos niveles pueden ser distintos para cada usuario y lo que busca el estándar es una interfaz que haga que se puedan entender los usuarios. En el caso de un sistema de información se tienen que establecer una serie de normas para medios de conexión y servicios genéricos. Para ello surgen una serie de normas, que consisten en documentos respaldados por una autoridad reconocida al efecto, producidos por un procedimiento público bien establecido orientado a facilitar el consenso de todos los actores involucrados en el tema. Existen normas a diferentes niveles: internacional (ISO, *International Telecommunication Union*, ITU), europeo (*European Committee for Standardization*, CEN) y nacionales como la Asociación Española de Normalización y Certificación (AENOR). En España, la estandarización de la HCE está regulada por el Comité Técnico AEN CTN 139 (1994) de AENOR.

Pero para el caso de un sistema de información destinada para la Telemedicina tiene que cumplir otra serie de normas, que dependiendo de cada término tiene un estándar diferente (Monteagudo 2003):

- Distribución de componentes (CORBA, .NET, etc.).
- Bus de comunicación interna (IP, *Firewire*, etc.).
- Buses de dispositivo (IrDA, USB, *Bluetooth*, etc.).
- Medios de comunicación interna (*Integrated Services Digital Network* (ISDN), *Digital Subscriber Line* (xDSL), etc.).
- Interfaces de usuarios.
- Dispositivos médicos (IEEE 1073, *Point of Care Testing* (POCT), DICOM, etc.).
- Directorio de historial de pacientes (ENV 13606, openEHR, etc.).
- Comunicaciones de historial de pacientes (HL7, OMG, etc.).

- Imágenes (DICOM, CIAS, etc.).
- Videoconferencia (*Session Initiation Protocol* (SIP), H.323, etc.).
- Seguridad y terminología (SNOMED, etc.).

Podemos considerar en la Tabla 3 un resumen de los estándares para la telemedicina.

VOCABULARIO	ICD 9.0 & ICD 10.O				
Tipos de Datos	Imagen médica DICOM 3.0	Imagen no médica JPEG/MPEG, M-JPEG	Texto SGML *HyperText Markup Language* (HTML) XML Texto	Audio MPEG2	Videoconferencia (Estándares de uso común, según la calidad de la línea)
Transmisión	DICOM 3.0, HL7, *File Transfer Protocol* (FTP), *Simple Mail Transfer Protocol* (SMTP), *HyperText Transfer Protocol* (HTTP), *Transmission Control Protocol* (TCP)				
Seguridad	Autentificación, Encriptación				
Infraestructura	Los de uso común en el entorno de red donde se presta el servicio				

Tabla 3. Estándares de Telemedicina (Monteaguado 2003).

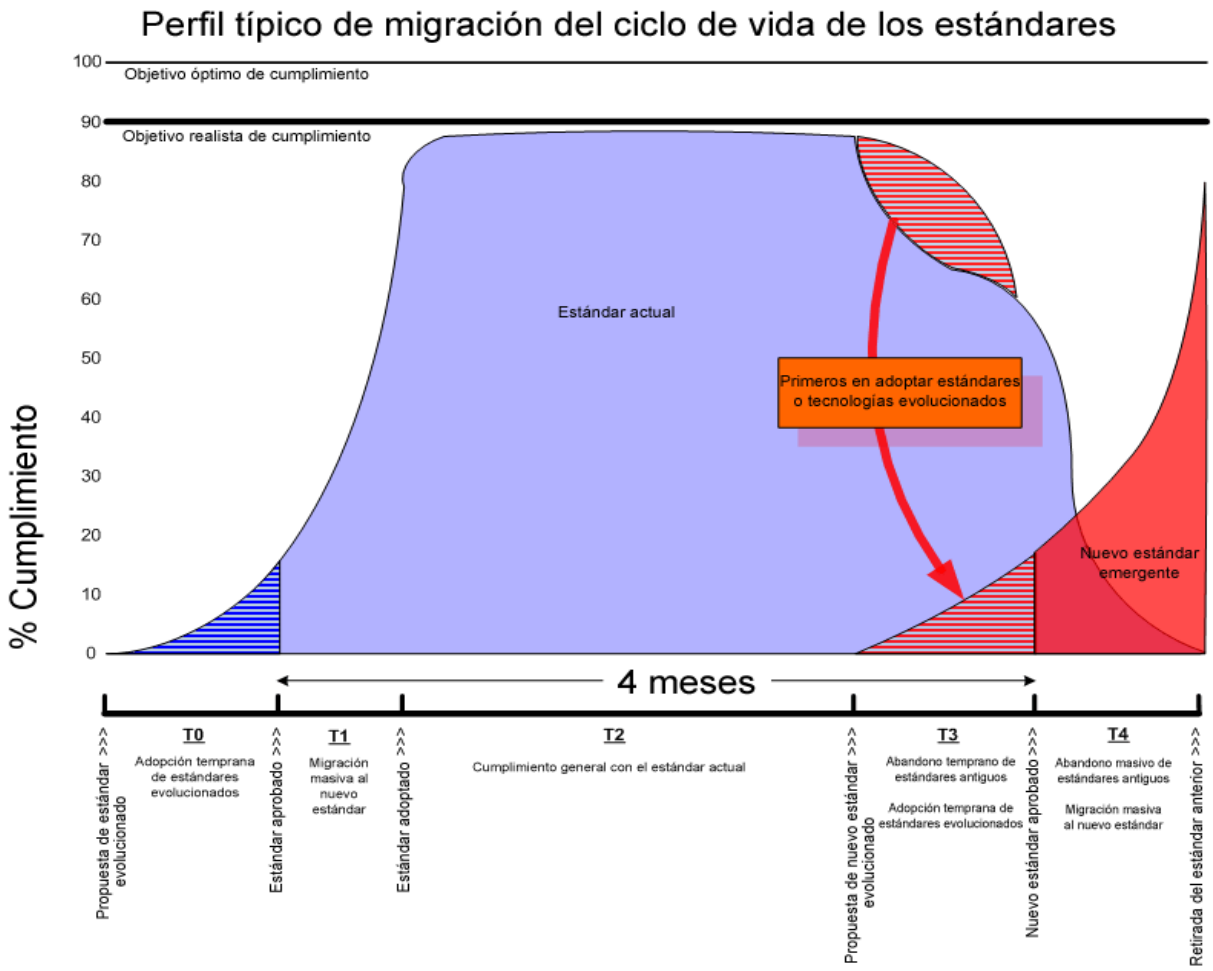


Figura 9. Evolución típica del ciclo de vida de un estándar desde su propuesta hasta su establecimiento consolidado (Monteagudo 2003).

En la figura 9 se puede observar la evolución típica en el ciclo de vida de un estándar. En la primera etapa se lleva a cabo la adopción del mismo, posteriormente se realiza la migración masiva al nuevo estándar una vez que éste ha sido adoptado. El periodo entre que un estándar es adaptado y un nuevo estándar es aprobado es de aproximadamente 4 meses.

La adopción de estándares (acuerdos para formatos, mensajes, codificaciones, estructuras de datos, contenidos, etc.) es vital para el éxito de la coordinación de actividades (aumenta la seguridad, disminuyen los costes, favorecen el desarrollo de nuevos servicios, etc.).

Los estándares estudiados en el presente trabajo son: HL7, DICOM, ISO/TC 215, CEN/TC 251, de seguridad y las iniciativas CORBA, GEHR, IHE y OpenEHR. A continuación, se explicada cada uno de ellos.

4.5. Health Level 7 (HL7)

Se ha llevado a cabo la revisión bibliográfica de más de 80 artículos en revistas sobre telemedicina de alto índice de impacto. La mayoría de los artículos analizados trataban sobre aplicaciones Web en especialidades como la cardiología y la radiología.

4.5.1. Introducción

HL7 es una especificación para un estándar de intercambio de datos electrónicos en el ambiente de la atención de la salud, con especial énfasis en las comunicaciones intra-hospitalarias. Se emplea para el intercambio de información clínica, financiera y administrativa entre sistemas informáticos (Chronaki *et ál.* 2001). Es el resultado del trabajo de un Comité de proveedores de usuarios, vendedores y consultores de sistemas de aplicación del área de salud. El hospital promedio de la actualidad posee programas instalados que se ocupan del registro de los procesos de admisión y egreso de pacientes, de registro y producción de información de laboratorio clínico, de informes de radiología y patología, de facturación y administración general, y otros (HL7 2008).

A menudo estas aplicaciones han sido desarrolladas por diferentes proveedores o grupos propios, poseyendo cada producto formatos de datos altamente específicos. A medida que los hospitales van expandiéndose, las operaciones de procesamiento de información se expanden en forma concomitante, y la necesidad de compartir los datos que encierran esa información se torna crítica. El desarrollo y disponibilidad de sistemas globales de informatización hospitalaria, que solo algunos proveedores muy selectos han desarrollado hasta la fecha, mitigarían la necesidad de estándares para la transmisión externa de datos del tipo del HL7. No obstante, estos programas son todavía escasos y su implementación requiere fuertes inversiones iniciales tanto de *hardware* como de *software*, lo cual hace que sigan existiendo y desarrollándose las aplicaciones específicas de bajo coste. En la figura 10 se muestra la interoperabilidad de diferentes sistemas clínicos con HL7.

Figura 10. Interoperabilidad con HL7 (Monteagudo 2003).

HL7 es una asociación sin ánimo de lucro, abierta a cualquiera que desee participar y con representantes elegidos democráticamente. El objetivo es el de favorecer la extensión del uso del estándar entre los distintos actores del sector a nivel internacional, lo que redundará en beneficio de todos al facilitar los proyectos de sistemas de información que impliquen la integración de datos, aplicaciones, procesos y personas en ámbitos heterogéneos (HL7 2008).

La necesidad del desarrollo de estándares que sirvan de base a interfaces entre sistemas surge a partir de la aparición de la tecnología de redes para la integración de programas de aplicación al área de salud que residen en estaciones de trabajo funcional y técnicamente diferentes. Lograr su integración suele requerir enorme cantidad de horas de programación específicas al sitio y al ambiente de redes. Esto ocurre a expensas del comprador y/o el vendedor, e impide al personal involucrado dedicarse a iniciativas más productivas como el desarrollo de nuevos módulos o productos. La cantidad de sistemas a vincular aumenta en forma exponencial la cantidad de tiempo invertido en el desarrollo de interfaces, en tanto que el apego a un estándar solo requiere del esfuerzo de su desarrollo por única vez.

Las características que aporta este estándar son:

- Un estándar independiente de la tecnología y de la plataforma.

- Posibilidad de intercambio de información entre aplicaciones desarrolladas por diferentes proveedores.
- Reducción de los costes de programación en el desarrollo y mantenimiento de interfaces.
- Flexibilidad, porque es posible desarrollar aplicaciones en diferentes entornos tecnológicos y conectarlas entre sí.

4.5.2. Antecedentes de HL7

El Comité de HL7, comenzó su actividad en Marzo de 1987 a raíz de una conferencia organizada por el Hospital de la Universidad de Pennsylvania, a propósito de la problemática de los estándares en salud. Su objetivo es la estandarización del formato y del protocolo para el intercambio de ciertos conjuntos de datos entre sistemas de aplicación al área de salud. El mecanismo de trabajo del comité consta básicamente de la definición de grupos de datos relevantes a determinados procesos que forman parte de un sistema de atención de la salud, de la determinación de un formato específico para los mismos, y de la producción de un documento final del estándar, que es votado y aprobado por los miembros del grupo de trabajo.

HL7 ha experimentado un gran crecimiento como organización en los últimos años. Actualmente existen más de 170 organizaciones vinculadas a la provisión de servicios de salud que han implementado sus interfaces entre sistemas sobre la base del estándar HL7. Además, existen numerosas instituciones que lo utilizan sin ser miembros oficiales del Grupo.

El término "Nivel 7" se refiere al más alto de los niveles del modelo *Open Systems Interconnection* (OSI) de *International Standards Organization* (ISO). Esto no implica que el HL7 conforme específicamente a determinados elementos constitutivos del séptimo nivel del OSI. De hecho, el HL7 no especifica un conjunto de especificaciones propias del OSI para los niveles 1 al 6. En cambio, si conforma específicamente a la definición conceptual de un enlace de aplicación en lo que hace estrictamente al nivel 7 del modelo OSI. La arquitectura ISO-OSI se puede observar en la figura 11.

Figura 11. Arquitectura ISO-OSI (HL7 2008).

En el modelo OSI, las funciones que hacen a la comunicación entre sistemas, tanto en materia de *hardware* como de *software*, se dividen en siete niveles. En el nivel 7 (siete), a los cuales se refiere específicamente el HL7, se definen los datos a ser intercambiados, la finalidad de dichos intercambios y la comunicación de determinados mensajes de error específicos de las aplicaciones que participan del intercambio.

4.5.3. Objetivos y misión del HL7

El HL7 se ocupa de las interfaces entre sistemas que emiten o reciben mensajes de registro, admisión, transferencia y alta de pacientes, pedidos de información al sistema, órdenes, resultados, observaciones clínicas, facturación, y actualización de información de archivos maestros. El HL7 no asume ninguna arquitectura en particular con respecto a la ubicación de los datos dentro de la aplicación, aunque está diseñado para dar soporte tanto a un sistema central de atención de pacientes, como a un ambiente más distribuido donde las aplicaciones departamentales son los repositorios de los datos.

Considerando la enorme cantidad de aplicaciones que actualmente existen en el campo de la salud, así como la variedad de ambientes en los cuales se efectúan procesos de atención e la salud, resulta evidente que exciten muchas interfaces adicionales que se beneficiarían con el desarrollo de estándares.

Otras áreas incluidas en el HL7 son las siguientes:

- Apoyo a la toma de decisiones.
- Aplicaciones de enfermería.
- Aplicaciones de departamentos de servicios auxiliares.
- HC computarizadas.
- Necesidades de información externas al ámbito hospitalario.

Además HL7 tiene las siguientes metas:

- Desarrollar y publicar las normas para su aprobación por ANSI.
- Promover el uso de HL7 en el ámbito sanitario.
- Promover la educación y divulgación del estándar.
- Promover los servicios de certificación de conformidad.
- Definir especificaciones metodológicas para crear extensiones de la norma.
- Aceptación y uso de HL7 en el mundo a través de la creación de afiliados internacionales.

Para lograr alcanzar las anteriores metas se van a realizar una serie de funciones: para cada escenario el HL7 define la estructura y codificación de los mensajes que se han de intercambiar, aprovechar las experiencias existentes en el desarrollo de protocolos y estándares, evolucionar permanentemente a medida que evolucionan los productos y la tecnología y no favorecer a los intereses de los propietarios de ninguna compañía, organización o empresa.

4.5.4. Formato de HL7

El HL7 es una especificación de formato de mensajes entre aplicaciones dirigidas al soporte informático de los procesos de atención de la salud. El formato general de los mensajes consiste de campos de datos de longitud variable, separados por caracteres especiales, según reglas específicas de codificación. Los campos de datos se combinan para formar agrupamientos lógicos denominados segmentos, los cuales a su vez están separados entre sí por caracteres específicos. Es decir, un mensaje se compone de diversas líneas, cada una de las cuales representa un segmento. En su versión actual, el HL7 codifica 27 tipos de mensaje, cada uno de ellos referido a un proceso especifico del conjunto de los que forman el proceso general de la atención de la salud.

4.5.5. Estándares

- **Versión 2.4:** esta versión fue acreditada por ANSI en Octubre del año 2000. Incluye nuevos mensajes para automatización de laboratorio, gestión de personal y gestión de aplicaciones.
- **Versión 2.5:** es la última versión liberada. Contiene nuevos mensajes respecto a versiones anteriores y ofrece multitud de opciones el contenido de los mensajes y la posibilidad de definir mensajes y segmentos Ad Hoc. Ésto proporciona una gran flexibilidad, pero en contrapartida, causa pérdida de interoperabilidad.
- **Versión 3:** esta versión adopta una rigurosa metodología y desarrollo de mensajes, acopla el Modelo de Referencia (RIM) con los mensajes.
- ***Clinical Document Architecture* (CDA):** fue reconocida como estándar ANSI en Noviembre del 2000 propone una estructura de documentos en formato XML, que gracias al uso del RIM y vocabularios codificados, el CDA convierte a los documentos clínicos, en objetos interpretables por multitud de aplicaciones y trasferibles a través de cualquier medio electrónico.
- ***Clinical Context Object Workgroup* (CCOW):** es gestionado por un grupo de trabajo que promociona la creación de estándares para la integración visual, desde el punto de vista del usuario final, de las aplicaciones de ámbito sanitario.

4.5.6. Modelo básico de transacciones HL7

El modelo básico de transacciones de HL7 se puede ver en la figura 12.

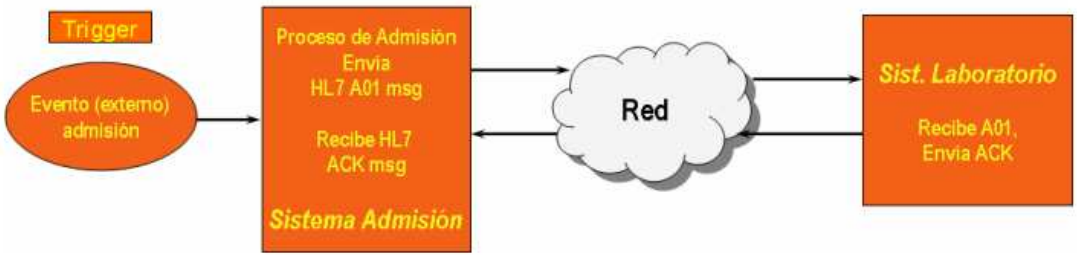


Figura 12. Modelo básico de transacciones HL7 (ANSI/HL7 2007).

A continuación se explica el intercambio de mensajes en el modelo básico de transacciones:

Paso 1. El sistema emisor construye un mensaje HL7 basado en datos de la aplicación y lo envía al sistema receptor.

Paso 2. El sistema receptor recibe el mensaje y pueden ocurrir varias opciones:

- Valida sintácticamente el mensaje. Si falla, envía un mensaje de rechazo al emisor. Si no continúa y pasa a b.
- Pasa el mensaje a la aplicación, la cual puede realizar una de las siguientes opciones:
    - Crea un mensaje de respuesta.
    - Crea un mensaje de error.
    - Crea un mensaje de rechazo.
- Envía el mensaje de respuesta, error o rechazo.

4.5.7. Modelo funcional de HL7

ANSI ha aprobado el primer modelo funcional para HCE. Este modelo funcional elaborado por HL7 define los rangos más importantes y las funciones que debe de contener un sistema de HCE.

El modelo funcional contiene aproximadamente unos 1000 criterios de conformidad con unas 130 funciones definidas donde quedan reflejadas desde funcionalidades para la prescripción asistida, antecedentes, sistemas de apoyo a la decisión clínica, órdenes y modelos de seguridad. Este nuevo modelo marca las pautas para el diseño de sistemas de HCE con los que se podrá dar conformidad. El modelo permite la creación de "perfiles". Estos perfiles son agrupaciones de funcionalidades definidas por administraciones, asociaciones, grupos de interés, etc., que recogen un conjunto deseable de características a cumplir por un sistema. Estos conjuntos son certificables.

El Modelo Funcional de HL7 (EHR-S) está compuesto por una estructura que cuenta con varias partes: el *resumen funcional* (que está dividido en tres secciones, el cuidado, el soporte y la infraestructura de la información*), los perfiles funcionales* y las *prioridades* asignadas a las funciones en cada perfil. El resumen funcional debe de

contener todas las funciones de HCE y a los perfiles hay que destinarles el uso de las anteriores funciones (ANSI/HL7 2007).

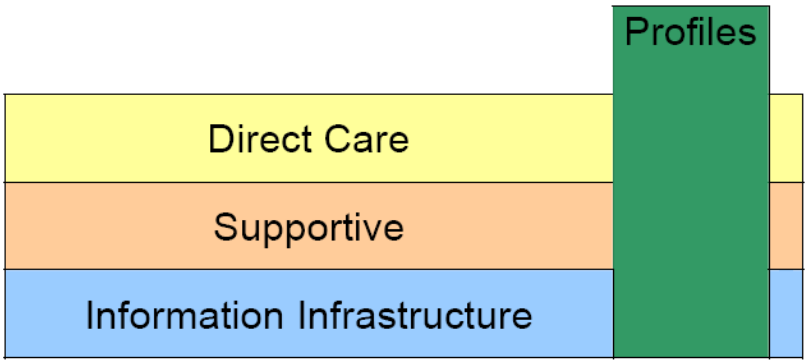


Figura 13. Modelo funcional de Hl7: resumen funcional (ANSI/HL7 2007).

Dentro de las tres secciones principales (en el resumen funcional, véase figura 13) hay trece subdivisiones (figura 14). Hay más de 140 funciones individuales en el resumen, cada una incluye un nombre, su declaración y los criterios de conformidad, así como otra información como la descripción de la función. El modelo funcional está pensado para incluir funciones dentro de cada conjunto de funciones, es decir que el propio usuario puede crear su subconjunto de funciones.

Direct Care	DC.1	Care Management
	DC.2	Clinical Decision Support
	DC.3	Operations Management and Communication
Supportive	S.1	Clinical Support
	S.2	Measurement, Analysis, Research and Reports
	S.3	Administrative and Financial
Information Infrastructure	IN.1	Security
	IN.2	Health Record Information and Management
	IN.3	Registry and Directory Services
	IN.4	Standard Terminologies & Terminology Services
	IN.5	Standards-based Interoperability
	IN.6	Business Rules Management
	IN.7	Workflow Management

Figura 14. Modelo funcional de HL7: subdivisiones del resumen funcional (ANSI/HL7 2007).

La cláusula de conformidad explica los conceptos críticos para la comprensión y la implementación del modelo funcional de HL7, tal como: definición del perfil, criterios de conformidad, y cómo saber que es obligatorio u opcional. Una cláusula de conformidad puede proporcionar también la comunicación entre las empresas y los usuarios (compradores) en cuanto a lo que es requerido. Sirve como la base para las actividades de certificación que pueden ser realizadas por organizaciones externas a HL7 (ANSI/HL7 2007).

El resumen funcional proporciona una lista de funciones organizadas en distintas secciones (figura 15). Las funciones describen como tiene que ser un sistema para que el usuario pueda reconocer los elementos claves de un HCE (ANSI/HL7 2007).

Las funciones de HCE pueden ser utilizadas para:

- Facilitar al usuario final los resultados como son la seguridad del paciente, la calidad del servicio y el coste en términos de HCE.
- Promover una visión general de las funciones HCE sobre los vendedores, los usuarios y otras personas interesadas, para poder evaluar las funciones de HCE.
- Proporcionar el *framework* necesario para manejar los requisitos y las aplicaciones de próximos estándares.
- Establecer un método para que cada país pueda aplicar estas funciones de HCE para sus escenarios, usos y prioridades.

Cada función en el modelo funcional de HL7 utilizando un conjunto de elementos o componentes (ANSI/HL7 2007):

ID	Type	Name	Statement /Description	See Also	Conformance Criteria	Row #
		Normative	Normative/Reference	Referenc e	Normative	

Figura 15. Elementos de las funciones en el modelo funcional de HL7 (ANSI/HL7 2007).

- *Elemento ID*

Este elemento es identificación una función en el resumen. Las funciones directas del cuidado son identificadas por "DC" seguido por un número (DC.1.1.3.1 o DC.1.1.3.2), las funciones de soporte son identificadas con "S" seguido por un número (S.2.1 o S.2.1.1) y las funciones de infraestructura de la información son identificadas con "EN" seguido por un número (IN.1.1 o IN.1.2).

- *Elemento Type*

  Dos posibles valores H (para indicar cabecera) o F (para indicar simplemente función).

- *Elemento Name*

  Este elemento indica el nombre de la función.

- *Elemento Statement/Description*

  Statement es una breve declaración del propósito de esta función.

  Description es la descripción detallada de la función, incluye ejemplos si es necesario.

- *Elemento See Also*

  Este elemento está pensado para identificar las relaciones entre las distintas funciones.

- *Elemento Conformance Criteria*

  Pensado para verificar la conformidad de la función en cuestión.

4.5.8. *Clinical Document Architecture* (CDA)

CDA es una arquitectura clínica de documentos de HL7. Es un estándar de marcaje para definir la estructura y la semántica de documentos clínicos que se requiere intercambiar en un entorno de interoperabilidad, conocido anteriormente como *Patient Record Architecture* (PRA). Un documento CDA es un elemento de información completamente definido que puede incluir texto, imágenes, sonidos, o cualquier tipo de contenido multimedia, y que puede ser transportado dentro de un mensaje HL7. Los documentos CDA derivan su significado del modelo de información de HL7 RIM (Schadow 2000).

Un documento clínico debe de tener las siguientes las características:

- Persistencia: un documento clínico debe mantenerse en un período de tiempo definido por requisitos locales y regulativos.
- Gerencia: un documento clínico es mantenido por una persona o la organización que confió en su cuidado.
- Autenticación: un documento clínico es una colección de datos creada para ser legal.
- Integridad: la autenticación de un documento clínico tiene que ser total.

- Legible: un documento clínico tiene que ser humanamente legible.

4.5.8.1. Metas y Objetivos del CDA

Las metas del CDA son las siguientes (Alschuler & Dolin 2000):

1. Prioridad en el cuidado de los enfermos.
2. Uso de los estándares.
3. Intercambio de documentos legibles entre usuarios, incluidos aquellos con diferentes niveles de sofisticación técnica.
4. Promover la longevidad de toda información codificada.
5. Permitir una gran variedad de aplicaciones de procesamiento económico.
6. Compatible con una gran variedad de aplicaciones.
7. Permitir controlar sus propios requisitos de información sin la extensión a esta especificación.

Los objetivos de diseño de CDA son los siguientes (Kim *et ál.* 2006; Treins *et ál.* 2006):

1. Esta arquitectura debe ser compatible con XML.
2. Las barreras técnicas de la arquitectura deben de mínimas.
3. La arquitectura debe imponer limitaciones o requisitos mínimos en la estructura y contenido de un documento.
4. La arquitectura debe ser escalable.
5. Documentar las especificaciones basadas en esta arquitectura.
6. Documentar las especificaciones para la creación y procesamiento de documentos.
7. Los documentos de CDA deben ser legibles.

4.5.8.2. Niveles del CDA

CDA es una *Document Type Definition* (DTD) jerárquica de tres niveles en cada uno de los cuales se va añadiendo más detalle en el marcado del documento. Actualmente sólo el nivel uno está completamente definido. El nivel uno es la base de todos y es la DTD más general. A pesar de ser el más general, es posible diferenciar entre diferentes tipos de documentos, como notas clínicas, altas a pacientes, etc. A lo largo de los tres niveles el contenido del documento no varía, sino que sólo varía la

granularidad y el detalle semántico con el que se puede describir el documento. Esto permite una gran flexibilidad dependiendo de los requerimientos de la aplicación. Por lo tanto, un documento marcado de acuerdo al nivel tres puede ser procesado por una aplicación que sólo necesita un marcado de nivel uno (Dolin *et ál.* 2000). Vamos a ver las características de cada uno de los niveles:

- **Nivel 1:** es la base de la jerarquía, especifica la semántica de la cabecera del documento (Dolin *et ál.* 2000), que se basa en el modelo de información de HL7 RIM. Es un nivel muy estructural adecuado para documentos clínicos muy narrativos ya que en el cuerpo del documento no contiene semántica. En la figura 16, se puede apreciar el nivel 1 de CDA.

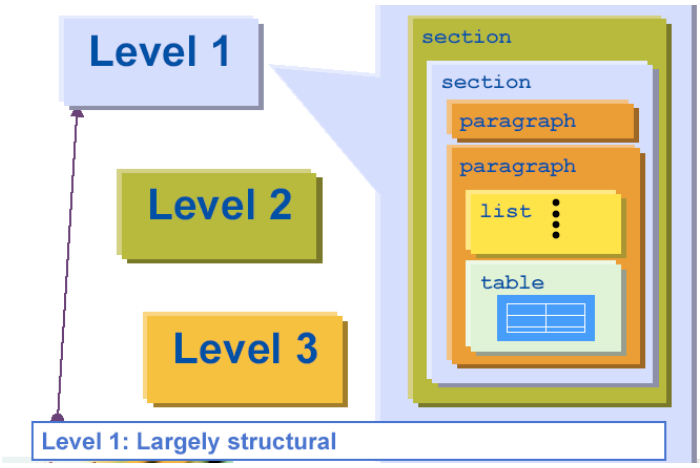


Figura 16. CDA nivel 1 (Dolin *et ál.* 2000).

- **Nivel 2:** el nivel dos utiliza la misma codificación para la cabecera y las secciones que el nivel uno, pero permite establecer más restricciones. Consiste en una serie de estructuras y semánticas que dependen del tipo de documento clínico que especifiquen (Dolin *et ál.* 2000). En la figura 17, se puede observar el nivel 2 de CDA.

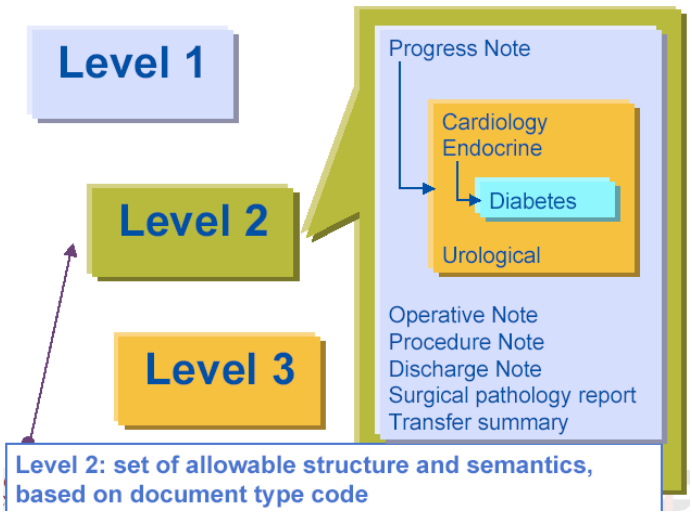


Figura 17. CDA nivel 2 (Dolin *et ál.* 2000).

- **Nivel 3:** este nivel permite definir observaciones y servicios dentro del cuerpo del documento. Añade etiquetas derivadas del modelo RIM mucho más específicas dentro del documento (Dolin *et ál.* 2000). En la figura 18, se puede apreciar el nivel 3 de CDA.

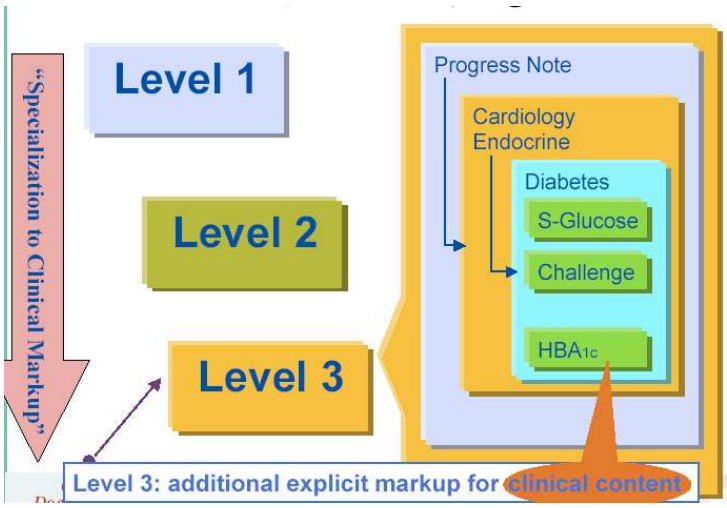


Figura 18. CDA nivel 3 (Dolin *et ál.* 2000).

Actualmente los niveles dos y tres no están definidos totalmente. El nivel uno consiste en tres especificaciones técnicas:

- CDA *Header*.
- CDA *Level One Body*.

- HL7 v3.0 *data types*.

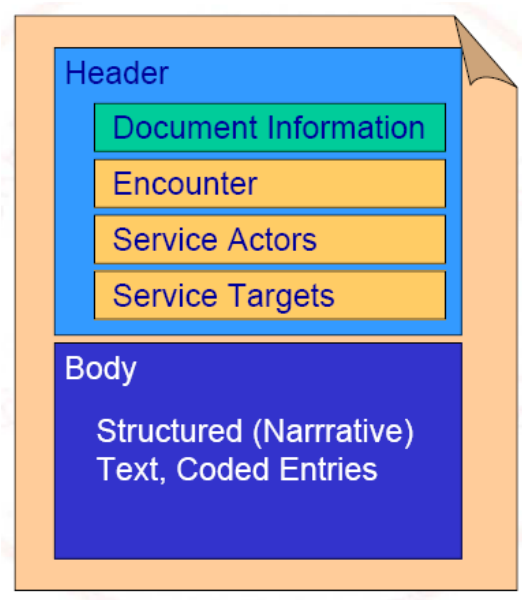


Figura 19. Estructura de un documento CDA (Dolin *et ál.* 2000).

La estructura de un documento CDA se puede ver en la figura 19. El CDA extiende el uso de los códigos para identificar los tipos del documento, las secciones en las que se divide, los procedimientos clínicos que se mencionan y los resultados clínicos. Puede ser utilizado cualquier código usado en el RIM, incluyendo vocabularios internos del RIM y códigos externos tales como *Current Procedural Terminology* (CPT), ICD, MEDCIN o *Systematized Nomenclature of Medicine* (SNOMED). Se recomienda el uso de los códigos *Logical Observations Identifiers, Names, Codes* (LOINC) para la clasificación de los tipos del documento (por ejemplo: informe de alta, informe de consulta, etc.). Los códigos de LOINC están disponibles para el uso comercial, conforme a los términos de una licencia que asegure la integridad y la propiedad de los códigos.

4.5.8.3. Identificadores de un CDA

Los identificadores globales y únicos necesarios para construir un CDA dependen de cada documento clínico. Hay algunos que siempre van a aparecer (ID), y otros que aparecerán con mucha frecuencia (identificadores de personas y roles que ejercen, lugares, observaciones médicas, etc.). A continuación se muestran algunos de los identificadores que hay que definir en un CDA:

- Identificador único para cada instancia CDA (requerido siempre): es una estructura fija que referencia al esquema del CDA normativo del HL7.
- Identificación de las personas: pacientes (*recordTarget*), médicos (como autores, personas que atienden a los pacientes, etc.), y otros posibles agentes como autentificador legal, persona que acompaña al paciente, etc.
- Identificaciones de organizaciones: a la que pertenezca el médico, la que sea responsable de la custodia del documento, etc.
- Otras identificaciones: confidencialidad, secciones de un documento, documentos, actos clínicos, observaciones médicas, diagnósticos, procedimientos y tratamientos, medicaciones y otras observaciones, etc.

4.5.8.4. Diseño de un documento CDA

Un documento CDA está compuesto, como mínimo, por una cabecera que contiene elementos de información que son obligatorios (*author*, *recordTarget*, etc.) y de un cuerpo que puede ser un bloque estructurado o no. El cuerpo no requiere de casi ninguna clase obligatoria (*text*, *component y section*). En un cuerpo estructurado, la información clínica se estructura y define a través de clases tipo *entry* (opcional), y, aunque no es obligatorio, es muy interesante su uso.

Para el diseño de un CDA es importante conocer su R-MIM, las reglas básicas para construir un archivo XML a partir de éste y los requerimientos propios del documento clínico en cuestión. A continuación, se va a mostrar una metodología de trabajo para el diseño y desarrollo de un CDA. Cuando el documento clínico es muy sencillo y se tiene cierto conocimiento de HL7, del R-MIM del CDA y de XML, se puede afrontar de forma directa su desarrollo, pero la idea es dar unas pautas generales para desarrollar cualquier CDA, por muy complejo que sea.

La metodología a seguir sería la siguiente:

- Estudio básico del estándar HL7 v.3.
- Comprobar que el dominio más adecuado para registrar la información es *Health and Clinical Management* (y en particular, como CDA). Para ello se debe consultar en el estándar las características que debe cumplir un documento clínico para ser especificado como un CDA, y compararlo con las características del documento que se pretende estandarizar. Por ejemplo, el R-MIM del CDA no

contempla incluir información relativa al dominio Administrative Management, y por lo tanto, para definir esta información no debe utilizarse el CDA.

- Análisis del modelo de referencia R-MIM de HL7 para CDA (R-MIM POCD_RM000040) y de todas sus clases.
- Analizar los requisitos y características del documento clínico: información que aparece, BDs implicadas, etc.
- Identificación de las clases, atributos y relaciones del R-MIM del CDA necesarias. En este punto se debe volver a comprobar, con el conocimiento más profundo del estándar y del documento, que es correcto el uso de este dominio.
- Diseño del R-MIM del CDA restringido para el documento.

4.5.8.4.1. CDA Header

El propósito de la cabecera CDA es permitir el cambio clínico de documentos a través de instituciones, facilitar la administración clínica de documentos y facilitar la compilación de documentos clínicos individuales de los pacientes en el registro electrónico.

La cabecera CDA contiene los metadatos que describen el documento. Se puede dividir en cuatro componentes lógicos (Alschuler & Dolin 2000):

- *Document Information*: información del documento, incluyendo la relación con otros documentos.
- *Encounter data*: contiene información relativa al servicio que se le está prestando a un paciente.
- *Service actors*: información relativa a los participantes en el servicio clínico.
- *Service targes*: información relativa a los receptores del servicio, como por ejemplo el paciente.

En la figura 20 se puede observar un esquema de la estructura de la cabecera CDA:

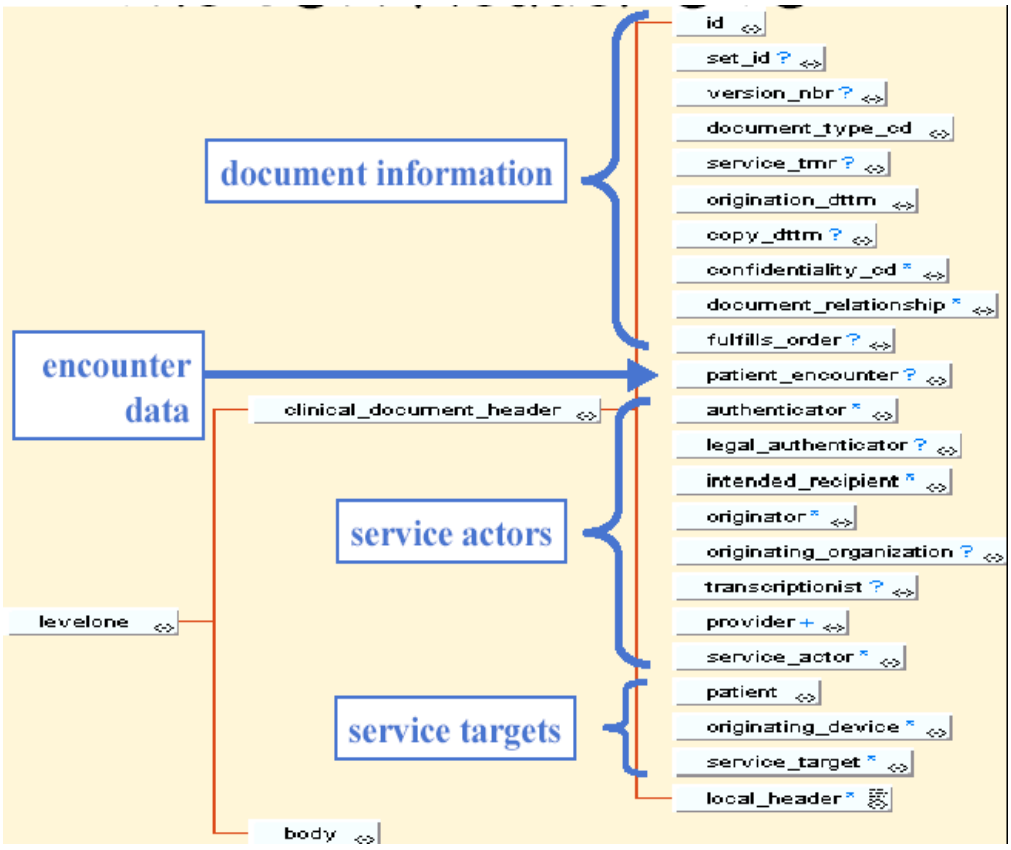


Figura 20. Esquema de la cabecera CDA (Treins *et ál.* 2006).

A continuación se van a mostrar los elementos más utilizados en esta Tesis y otros que aparecen con frecuencia en la cabecera de un documento CDA, después se mostrará un ejemplo obtenido de un HC de la aplicación **TeleOftalWeb**.

- *Clinical_document_header*

Este elemento sirve para indicar la cabecera del documento, dentro de éste se incluyen los datos de filiación del paciente.

- *Id*

Id representa la *Instancia Identificadora Única* (UID) de un documento clínico. Es un elemento que define de forma única y universal un documento, y los diferencia de todos los demás. El elemento id de este proyecto contiene los atributos EX (clave de identificación única de cada historial) y RT (indica el ámbito en el que es válido el identificador).

- *Document_type_cd*

Indica el tipo de documento que se va a crear. Contiene los atributos DN (tipo de documento), S (nombre del sistema de codificación) y V (código relacionado con el tipo

de documento). Para esta Tesis se trata de HC, que dentro del vocabulario LOINC le corresponde el código 11488-4.

- *Origination_dttm*

Representa la fecha en el que un documento original es creado. Aunque un documento sea revisado su valor sigue siendo el mismo. Contiene el atributo V (fecha de creación del documento). Dentro del estándar CDA todas las fechas se representan con el formato aaaa-mm-dd.

- *Originator*

Como se ha comentado anteriormente este elemento contiene datos del origen (humano) del documento. Relacionados con este elemento están: *originator.type_cd* que es la persona que crea el documento y *participation_tmr* que indica el la fecha de creación del documento.

- *Provider*

Contiene información sobre la persona u organización que proporciona el servicio médico.

- *Patient*

En este elemento se incluyen los datos personales del paciente.

- *Patient_type*: siempre toma el valor PATSBJ.
- *Id*: almacena el identificador del paciente, en este caso se ha tomado como identificador el Documento Nacional de Identidad (DNI).
- *GIV*: nombre del paciente.
- *FAM*: apellidos del paciente.
- *STR*: calle del domicilio del paciente.
- *HNR*: número del portal de la vivienda del paciente.
- *ADL*: este campo sirve para contener información adicional relativa a la dirección del paciente. En el presente proyecto almacena el número de planta de la vivienda del paciente.
- *CTY*: localidad de residencia del paciente.
- *STA*: provincia a la que pertenece la localidad de residencia del paciente.
- *ZIP*: código postal del domicilio del paciente.
- *CNT*: país de residencia del paciente.
- *Phon*: número de teléfono del paciente.

- *Local_header*: el estándar CDA permite definir etiquetas especiales si las fijadas por la DTD no se ajustan a la información que se quiere almacenar. En este caso concreto se ha utilizado una etiqueta de cabecera personalizada con el descriptor *email* para almacenar la dirección de correo del paciente.
- *Birth_dttm*: fecha de nacimiento del paciente.
- *Administrative_gender_cd*: sexo del paciente.
- *Local_header*: etiqueta con descriptor "numeroSS" para almacenar el número de la seguridad social del paciente.
- *Local_header compartidos:* este grupo de etiquetas se utiliza para definir los participantes que tienen acceso al historial y el tipo de permisos de cada uno.

Otras posibles etiquetas que no aparecen en el documento clínico y que pueden ser de importancia son:

- *ConfidentialityCode*

El código de confidencialidad es un componente contextual requerido de CDA, en el que el valor expresado en la cabecera se mantiene en verdadero para el documento completo, a menos que sea anulado por un valor anidado. En la codificación del nivel de confidencialidad se utiliza *Coding With Extensions* (CWE).

- *RecordTarget*

Representa la persona a la que pertenece ese documento clínico. Normalmente coincide con el sujeto sobre el que se está realizando las pruebas/observaciones, etc.; pero puede que no sea así, como por ejemplo el caso de un feto. Un documento clínico normalmente tiene un único participante *recordTarge*t. En el caso poco común de que un documento clínico (como una nota de encuentro en grupo) esté ubicado en más de una tabla de pacientes, se pueden establecer más de un *recordTarget*. El/los *recordTarget* de un documento se establecen en la cabecera y se transmiten al contenido jerarquizado, cuando no puede ser anulados.

- *Autor*

Representa las personas y/o máquinas que crearon el documento. Puede existir uno o más autores. En algunos casos, el rol o función del autor es inherente al código del documento.

La etiqueta *author* requiere los elementos: *time* y *assignedAutho*r, los elementos *assignedPerson y representedOrganization* son opcionales.

- *Custodian*

Representa la organización que está a cargo del mantenimiento del documento. La etiqueta custodian indica quién es el encargado del cuidado y la seguridad del documento. Cada documento CDA tiene asignado exactamente una organización encargada de su mantenimiento.

Vemos un ejemplo de cabecera de un documento CDA:

```
<clinical_document_header>
<id EX="1e00a7c07dbd190e0000011270338695" RT="TeleOftalWeb3.2"/>
<document_type_cd DN="Consultation note" S="LOINC" V="11488-4"/>
<origination_dttm V="2007-05-09"/>
<originator>
      <originator.type_cd V="AUT"/>
      <participation_tmr V="2007-05-09"/>
      <person>
            <id EX="123412344" RT="TeleOftalWeb3.0"/>
      </person>
</originator>
<provider>
      <provider.type_cd V="CON"/>
      <person>
            <id EX="123412344" RT="TeleOftalWeb3.0"/>
      </person>
</provider>
<patient>
      <patient.type_cd V="PATSBJ"/>
      <person>
            <id EX="12121212" RT="TeleOftalWeb3.0"/>
            <person_name>
                  <nm>
                        <GIV V="Raúl"/>
                        <FAM V="Gonzalez"/>
                  </nm>
            </person_name>
            <addr>
                  <STR V="Paseo San vicente"/>
                  <HNR V="14"/>
                  <ADL V="4d"/>
                  <CTY V="Valladolid"/>
                        <STA V="Valladolid "/>
                        <ZIP V="47007"/>
```

```
                                        <CNT V="España"/>
                  </addr>
                  <phon V="983123456"/>
                  <local_header
descriptor="email">rgonzalez@gmail.com</local_header>
            </person>
            <birth_dttm V="1982-02-11"/>
            <administrative_gender_cd S="2.15.840.1.113883.5.1" V="M"/>
            <local_header
descriptor="numeroSS">645363728273</local_header>
      </patient>
      <local_header descriptor="compartidos">
            <local_header descriptor="person">
                  <local_attr name="V" value="123412344"/>
                  <local_attr name="type" value="owner"/>
            </local_header>
      </local_header>
       </clinical_document_header>
```

4.5.8.4.2. CDA Level One Body

El cuerpo de los documentos CDA está formado por una serie de contenedores anidados, incluyendo datos no XML, *sections*, *paragraphs*, *list*, *tables*, etc. Los elementos *section* pueden contener a su vez elementos *paragraph*, *list*, y *table*. Los contenedores pueden tener elementos *caption* y *content*, que a su vez pueden estar codificados. En la figura 21 se puede observar cómo puede realizarse el anidamiento de elementos en el cuerpo de los documentos CDA.

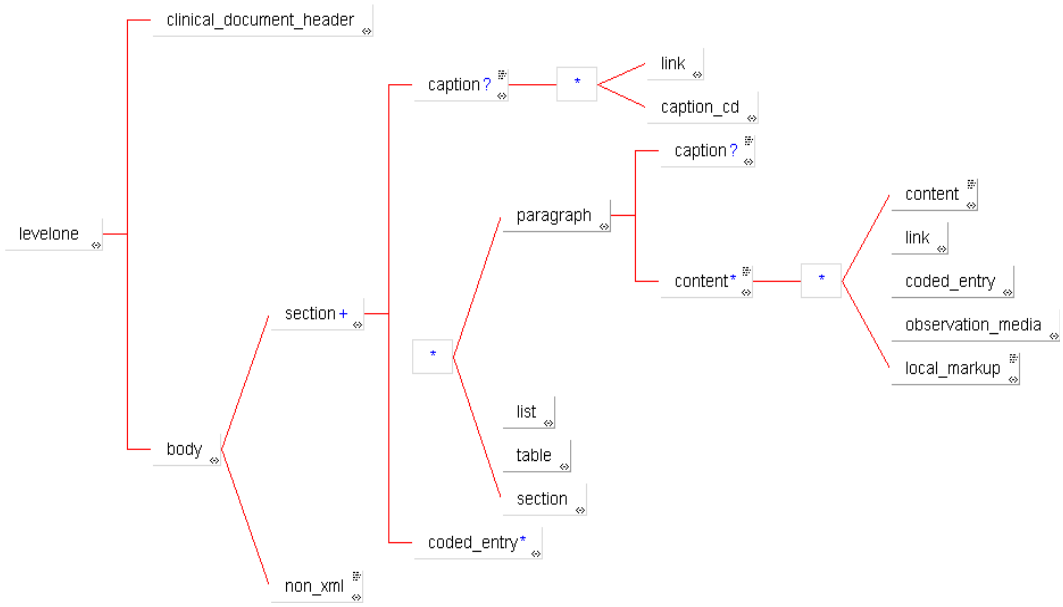


Figura 21. Esquema de los documentos CDA (Treins *et ál.* 2006).

Cada documento CDA tiene exactamente un cuerpo, asociado con la clase *ClinicalDocument* a través de la relación component. Un cuerpo de un CDA puede ser representado a través de un cuerpo estructurado (*structuredBody*) o uno no estructurado en XML (*nonXMLBody*). Un contenido XML estructurado siempre está insertado dentro de un elemento structuredBody, nunca como un archivo externo.

La clase de datos no XML (*NonXMLBody)* representa un cuerpo de documento que está en un formato distinto a XML. Contiene un elemento requerido, *text*, que se utiliza para referenciar datos que se almacenan externamente al documento CDA, o para codificar los datos directamente en línea. El elemento *text* tiene un atributo opcional, *mediaType*, que identifica el código del dato encapsulado e identifica un método para interpretar o presentar la información. La presentación de un cuerpo referenciado no-XML requiere una herramienta de *software* que reconozca el tipo particular MIME del bloque.

Los valores de *mediaType* más usuales son: "imagen/gif", "imagen/tiff", "text/rtf", "aplicación/pdf", "imagen/g3fax", "texto/html", "imagen/jpeg", "imagen/png", y "texto plano". Un elemento *text* puede contener una referencia que requiere de un atributo, value, que contiene la dirección URL del objeto externo al que apunta. Un ejemplo es el siguiente:

```
<component>
```

```
<nonXMLBody>
     <text mediaType="texto plano">
                     <reference value="paciente.txt"/>
                </text>
           </nonXMLBody>
</component>
```

Un cuerpo estructurado está compuesto por uno o más elementos *component*, que pueden estar compuestos a su vez por ninguna o varias secciones (*Section*), compuestas a su vez de entradas (*entry*), las cuales pueden referenciar observaciones, encuentros, etc. Cada uno de estos elementos está compuesto por elementos opcionales como *code, title, y text*. En este caso, el elemento *text* puede contener un elemento *content* que es usado para añadir más estructuración a la parte de texto de la entrada, o añadir información formal adicional. Un ejemplo es el siguiente:

```
<component>
        <structuredBody>
             <component>
                  <section>
                     <code code="10164-2"
codeSystem="2.16.840.1.113883.6.1"
                      codeSystemName="LOINC"/>
                     <title>Anamnesis</title>
                      <text>
                <content>
                                  El paciente presenta…
                           </content>.
                      </text>
                  </section>
             </component>
        </structuredBody>
</component>
```

Todos estos elementos pueden combinarse de la forma que mejor se adapte al documento que se quiere describir. Vamos a explicar más detalladamente el cuerpo de nuestro documento.

- *Section*

Esta etiqueta es el nodo básico dentro del cuerpo del documento. En los historiales definidos se pueden distinguir dos secciones almacenadas dentro de los correspondientes nodos *section*, por un lado la que contiene todas las revisiones del paciente y por otro la que almacena las imágenes asociadas al historial. Puede observarse en la figura 22.

Figura 22. Cuerpo del historial XML. Fuente: propia.

También puede observarse (figura 23) que cada revisión del paciente se almacena en un nodo *section* contenido dentro del nodo de revisiones.

Figura 23. Revisiones dentro de un historial XML. Fuente: propia.

Cada revisión se encuentra dividida en secciones que corresponden a cada una de las partes de información de los historiales del proyecto: filiación, motivo consulta, antecedentes personales oftalmológicos, antecedentes personales generales, antecedentes familiares, exploración, otras pruebas, juicio diagnóstico y por último tratamiento y recomendaciones. Como en los casos anteriores, cada uno de estos apartados se encuentra en un nodo *section*. Cada revisión lleva asociado un nodo *caption* donde se almacena la fecha y hora de creación de la misma.

```
<!-- ************************************************************************* --
>
<!--              NUEVA REVISION                                      -->
<!-- ************************************************************************** --
>
<!-- Cada nueva revision será una nueva sección dentro de Revisiones, y cuyo titulo
será la fecha -->
- <section>
   <caption>2006-03-26 17:01</caption>
   <!-- ************************************************** -->
   <!--        FILIACION                            -->
   <!-- ************************************************** -->
 + <section>
   <!-- ************************************************** -->
   <!--        MOTIVO CONSULTA                            -->
   <!-- ************************************************** -->
 + <section>
   <!-- ************************************************** -->
   <!--        AP OFTALMOLOGICOS                    -->
   <!-- ************************************************** -->
 + <section>
   <!-- ************************************************** -->
   <!--        AP GENERALES                         -->
   <!-- ************************************************** -->
 + <section>
   <!-- ************************************************** -->
   <!--        A FAMILIARES                         -->
   <!-- ************************************************** -->
 + <section>
   <!-- ************************************************** -->
   <!--        EXPLORACION                          -->
   <!-- ************************************************** -->
 + <section>
   <!-- ************************************************** -->
   <!--        OTRAS PRUEBAS                              -->
   <!-- ************************************************** -->
 + <section>
   <!-- ************************************************** -->
   <!--        JUICIO DIAGNOSTICO                   -->
   <!-- ************************************************** -->
 + <section>
   <!-- ********************************************************** -->
   <!--        TRATAMIENTO Y RECOMENDACIONES                    -->
   <!-- ********************************************************** -->
 + <section>
</section>
```

Figura 24. Revisión de un historial XML. Fuente: propia.

- *Paragraph*

La etiqueta *section* se puede seguir repitiendo agrupando los apartados generales, hasta llegar a los datos concretos insertados por los usuarios de la herramienta. La forma en la que se almacenan estos datos en el documento XML puede variar en función del formato de entrada en la herramienta, pudiendo tener listas,

etiquetas especiales, etc. El contenido de esta etiqueta es el dato en sí. Esta estructura puede apreciarse en la figura 25.

```
<section>
        <caption>Tratamiento</caption>
        <paragraph>
                <content>Reposo absoluto</content>
        </paragraph>
</section>
```

Figura 25. Parte de una revisión de un historial XML. Fuente: propia.

- *Observation_media*

En esta etiqueta se almacenan las imágenes modificadas y codificadas en base 64. Estas imágenes se almacenan como elementos de una lista dentro de su sección correspondiente. Cada imagen tiene un titulo donde se almacena el identificador de la imagen, generado a partir de una marca de tiempo y el identificador del usuario que la ha introducido. Además las imágenes tienen un nodo hijo que contiene la descripción y otros dos para la imagen original y la editada. Vemos dos imágenes en las que una de ellas la imagen original no ha sido modificada (figura 26) y otra en la que si (figura 27).

```
<paragraph>
        <caption>sagital</caption>
        <content>
        <observation_media>
                <observation_media.value ENC="B64" MT="image/jpeg">&amp;quot;</observation_media.value>
        </observation_media>
        </content>
</paragraph>
```

Figura 26. Código en el que la imagen sagital no ha sido modificada. Fuente: propia.

```
<paragraph>
        <caption>frontal</caption>
        <content>
        <observation_media>
                <observation_media.value ENC="B64" MT="image/jpeg">/9j/4AAQSkZJRgABAgAAA
        </observation_media>
        </content>
</paragraph>
```

Figura 27. Código en el que la imagen frontal ha sido modificada. Fuente: propia.

4.5.8.5. HL7 v3.0 Data Types

Como se ha indicado anteriormente los documentos CDA derivan su significado del modelo de información de HL7 RIM. Los elementos y atributos y las relaciones entre dichos elementos y atributos están obtenidos del modelo de referencia y plasmados en XML.

La versión 3 de los *Data Types* del HL7 es una especificación que intenta definir los tipos de datos para el intercambio de información clínica (Schadow 2000). Se definen tipos de datos para cadenas de caracteres, representación de datos multimedia, códigos, identificadores para conceptos y aparatos del mundo real, medidas, etc.

La especificación CDA permite componentes o elementos codificados, cuyos nombres en la especificación XML aparecen con una terminación *cd*. Estos componentes codificados tienen asociados dominios de vocabulario, tesauros, etc, que representan los grupos de valores permitidos para dichos componentes. Cada vocabulario tiene un identificador único asignado por HL7 y cada concepto dentro de un vocabulario tiene un código único. El identificador raíz de HL7 es el 2.16.840.1.113883. Como ejemplo, la definición para el identificador 2.16.840.1.113883.6.3 es el esquema de codificación *International Classification of Diseases revision 10* (ICDR10). De esta forma se puede observar en el siguiente fragmento de historia clínica el uso de estos códigos:

```
<caption>Most Responsible Diagnosis</caption>
<section>
      <caption>Unstable Angina
            <caption_cd V=I20.0 S=”2.16.840.1.113883.63”/>
      </caption>
      <paragraph>
<content>Y</content>
</paragraph>
</section>
```

Se puede observar que el titulo del párrafo es *Unstable Angina*. El código que representa dicho concepto es I20.0 y es tomado del vocabulario fuente ICD10. Esto permite que el elemento *caption* pueda ser interpretado y leído por un humano o bien procesado por una aplicación *software*.

Aunque la mayoría el estándar HL7 surgió hace más de una década, no ha sido hasta hace unos años cuando se ha empezado a utilizar realmente. Es necesario tener un mecanismo para verificar la integridad de los datos antes del intercambio cuando se emplea el estándar HL7 (Liou *et ál.* 2000).

4.6. Digital Imaging and Communications in Medicine (DICOM)

En relación al estándar DICOM, se ha llevado a cabo la revisión bibliográfica de más de cien artículos en revistas sobre telemedicina de alto índice de impacto y la mayoría de ellos trataban sobre el estándar DICOM en especialidades como son la radiología (Inamura *et ál.* 1996) y la cardiología (Christiaens *et ál.* 1999; Hibel *et ál.* 2000). También hay publicaciones que describen el diseño, desarrollo y posterior evaluación de aplicaciones de *software* libre basadas en dicho estándar con el fin de visualizar imágenes sobretodo en blanco y negro (Marcheschi *et ál.* 2003). También existen aplicaciones que convierten cualquier tipo de formato de imagen en DICOM y viceversa.

4.6.1. Introducción general

El estándar DICOM fue establecido en 1992 para intercambiar imágenes médicas en formato digital. Es un tanto complicado a causa de su propia y especifica "jerga". Se encuentra en constante cambio para acomodar nuevas tecnologías de tratamiento de imágenes y una mayor integración. Actualmente consta de dieciocho secciones diferentes, que tratan del protocolo y formatos DICOM, así como la especificación de conformidad. DICOM está basado en el estándar *The American Collage of Radiology* - *The Nacional Electrical Manufacturers Association* (ACR-NEMA) y fue lanzado por la ACR para satisfacer las necesidades de conectividad entre equipos de tratamiento de imágenes. DICOM está afiliado a diversas organizaciones de estándares americanas e internacionales que trabajan en campos relacionados. El estándar DICOM original fue denominado ACR-NEMA en referencia a esas organizaciones. Dicho estándar fue publicado por primera vez en 1985, y una segunda versión (2.0) fue publicada en 1988 (DICOM 2008). Con el objeto de hacer patente su relación hereditaria respecto del estándar ACR-NEMA, la versión original de DICOM ha sido denominada DICOM versión 3.0. Normalmente, cada año se publican versiones actualizadas del estándar DICOM. La historia de DICOM se remonta a principio de los años 80. En realidad, fue impulsado por los usuarios (es decir, la ACR). La mayoría de

los fabricantes se sentían relativamente cómodos dando soporte a estándares propietarios de comunicaciones e intercambio de datos, porque restringía a sus clientes a adquirir equipos de la misma compañía, o desarrollar *software* a medida para conectarse a los equipos de adquisición. El desarrollo de una estación de trabajo de propósito general, tal como las estaciones de trabajo 3-D, constituía un esfuerzo de primera magnitud porque requería soportar una librería de todos los diferentes tipos de cintas, discos flexibles y otros formatos de intercambio. La disponibilidad del estándar ACR-NEMA en los años 80 ha demostrado ser un arma de doble filo. Diversos fabricantes comenzaron a implementarlo encima de su propio protocolo de red no estandarizado, haciendo extensiones al estándar a medida que se necesitaba. Philips y Siemens se unieron e incluso desarrollaron su propia versión, llamada SPI. Las implementaciones tempranas de *Picture Archiving and Communication Systems* (PACS) y los usuarios que tratan con equipos de esa era todavía tienen que vérselas con esas implementaciones primerizas de ACR-NEMA 2.0 y trabajan con convertidores, cajas de interfaz, etc. Para poder ascender hasta el auténtico nivel de DICOM (DICOM 2008).

DICOM no es un estándar ANSI, tal como por ejemplo, el estándar HL7, que prevalece en todas partes de un hospital, exceptuando los departamentos de tratamiento de imágenes (DICOM 2008). La discusión sobre si debe convertirse o no en un estándar ANSI surge regularmente dentro del comité para los estándares DICOM, y a día de hoy se ha decidido que ello impondría más restricciones que los beneficios adicionales que pudiera reportar. A estas alturas, DICOM es muy respetado y reconocido como un autentico estándar internacional y no hay alternativas reales. Además, el proceso de estandarización es bastante eficiente. El comité de estándares DICOM se reúne varias veces al año durante un día en diversos lugares, principalmente en EE.UU, pero también ocasionalmente en Europa y Asia. Este comité está formado por miembros pertenecientes a fabricantes y organizaciones profesionales. Hay aproximadamente cincuenta miembros que aprueban nuevos temas de trabajo, es decir, determinan nuevas áreas susceptibles de ser estandarizadas y aprueban los resultados. En realidad, los estándares son escritos por varios grupos de trabajo, de los cuáles hay actualmente más de veinte, que cuentan con conocimientos expertos sobre el tema en particular. Hay un proceso de revisión muy exhaustivo, tanto interno como externo, porque cada cambio propuesto es objeto de "comentario públicos" por parte de cualquiera que desee comentar sobre el mismo, antes de ser sometido a votación por los miembros del comité

DICOM. Uno de los grupos de trabajo, el dedicado al estándar Base (*WG VI*) se asegura de que los nuevos añadidos encajen con la filosofía y la integridad generales del estándar y actúa a modo de "embudo" para los nuevos añadidos al estándar.

DICOM utiliza un lenguaje específico y conceptos que son habituales en el entorno Orientado a Objetos de la ingeniería de *Software*. Adicionalmente el propio estándar contiene diversas secciones que no se emplean casi nunca, lo cual se debe al proceso democrático mediante el cual fue definido el estándar.

DICOM está basado en conceptos que son habituales en el entorno de la *Object-Oriented Programming* (OOP), pero que aún no son de uso extendido. Por ejemplo, ACR-NEMA define imágenes, mientras que DICOM define objetos que incluyen imágenes, y siempre se refiere a esos objetos en el contexto de las operaciones aplicables a ellos, tales como almacenarlos, moverlos o buscarlos, crearlos, imprimirlos, etc. Los objetos en DICOM son denominados Objetos de Información, y las operaciones o servicios son llamados Clases de Servicio, que son definiciones generalizadas. Por consiguiente, en lugar de debatir sobre imágenes, los gurús de DICOM hablan de Clases de Servicio, Objetos y Clases de *Service-Object Pair* (SOP). Una ocurrencia específica de una clase es llamada Instancia, por ejemplo un CT_Store de un paciente en particular. El uso de nuevos conceptos y terminología requiere un periodo de aprendizaje y familiarización antes de llegar a entender y abarcar el estándar (NEMA Part 1 2008).

Una razón para explicar el lento desarrollo de DICOM hasta ahora es que en general el estándar DICOM todavía se utiliza únicamente para el tratamiento de imágenes médicas, lo cual constituye aún una aplicación muy particular comparada con otros estándares y aplicaciones. Esto significa que no hay muchos tutoriales disponibles para aclarar esta materia al público en general. Y, como sucede con cualquier publicación sobre un estándar, debe ser exacto y correcto, lo cual significa que, de la misma forma que sucede al emplear términos legales, la exactitud no siempre facilita la simplicidad y facilidad de lectura. Intentar entender e/o incluso implementar DICOM utilizando tan sólo el documento sobre el estándar es un verdadero reto, como mínimo. Se puede comparar con intentar aprender un nuevo idioma contando simplemente con un diccionario. Uno puede aprender todas las palabras que empiezan por A, B, etc. Pero eso no significa que pueda conversar en ese idioma. Al igual que sucede con un diccionario, el estándar DICOM también contiene muchas secciones y opciones que son raramente utilizadas. Por ejemplo, se puede desarrollar una conversación bastante buena

en cualquier idioma en particular usando tan sólo unos cuantos miles de palabras. Lo mismo sucede con DICOM. El estándar no identifica qué opciones se utilizan más frecuentemente, ni cuáles no se usan casi nunca o en absoluto (NEMA Part 1 2008).

El estándar DICOM está en continua expansión. Probablemente no pasa una sola semana sin que un grupo de trabajo DICOM se reúna en alguna parte del mundo para debatir sobre la definición de nuevos objetos y servicios. Es fácil ampliar el estándar debido a su naturaleza modular. No importa realmente si se está almacenando una imagen de TAC o un objeto recientemente definido tal como una imagen dental, hemodinámica o endoscopia pero se puede utilizar el mismo servicio. Se puede comparar esto con un taller que hace muebles, donde los objetos de información serían las piezas del mobiliario; y las herramientas, es decir, la sierra, el martillo, etc. son los servicios DICOM. No importa realmente con qué muebles o tipos de madera se está trabajando, en todos los casos se usan las mismas herramientas. Además, tal y como se ilustra en la figura 28, las propias imágenes están basadas sobre bloques de construcción –módulos- que permiten la especificación de otro objeto, reutilizando muchos de los módulos previamente definidos tales como los del paciente, estudio, equipo, etc. El resultado es que muchas nuevas aplicaciones fuera de la radiología (anatomía patológica, macroscopía, endoscopia, radioterapia, oftalmología) están ampliando el estándar DICOM y utilizándolo como la base para estandarizar sus comunicaciones de imágenes.

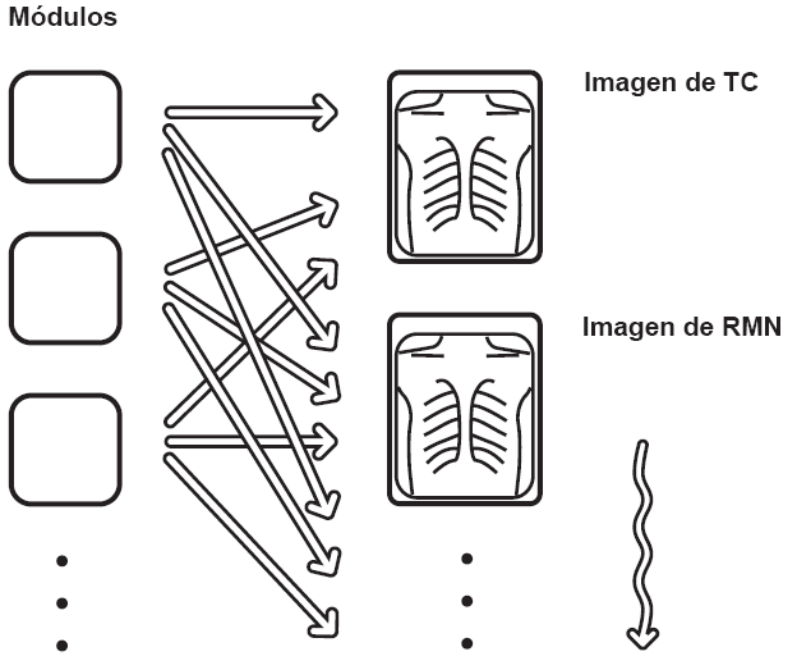


Figura 28. Relación Módulos imágenes (Otech 2008).

Algunas personas también están confundidas acerca de la versión del estándar. El estándar DICOM se denomina DICOM 3.0. El número "3.0" muestra la evolución que comenzó con los estándares ACR-NEMA 1.0 y 2.0. Se cambio el título de ACR-NEMA a DICOM para resaltar el hecho de que el estándar es realmente un estándar mundial, siendo las instituciones americanas NEMA y ACR simplemente miembros del comité DICOM, de la misma forma en que otros fabricantes y organizaciones de usuarios de Europa, Asia, y otros continentes son también miembros. Tanto el estándar DICOM, tal y como se publicó inicialmente en 1993, como la versión actual, aún se denominan DICOM 3.0. Sin embargo, está en constante crecimiento y expansión. De hecho, se intentó eliminar el "3.0" porque en realidad podría confundir y hacer pensar que hay una noción de versiones en el estándar DICOM. Se imprime y publica una revisión actualizada del estándar en un volumen integrado periódicamente, habitualmente cada año. Pero el estándar completó no consiste tan sólo en la última versión impresa, sino que incluye también todos los servicios adicionales, cambios y correcciones, tal y como fueron aprobados por lo miembros del comité DICOM y publicados en el sitio de NEMA en Internet, que se suele actualizar bimestralmente (después de la reunión del Grupo de Trabajo VI). Por consiguiente, la última y más completa versión del estándar DICOM no tiene típicamente más de dos meses de antigüedad.

Respecto a la implementación, DICOM es un requisito estándar para toda compra de nuevo equipo de diagnóstico por imagen, excepto para sistemas aislados que no se piensan conectar jamás a otros dispositivos. Los fabricantes han estado ofreciendo prestaciones DICOM desde comienzos de los años noventa. DICOM consta de varios servicios diferentes, esto es, puede transferir imágenes a otros dispositivos para almacenamiento, enviarlas a una impresora, recuperar una planificación que incluya la información demográfica de un paciente, actualizar el estado de un examen en un servicio de información, etc. Sin embargo, la implementación de los diversos servicios, aparte del almacenamiento, es aun un tanto limitada para algunos tipos de equipo. Algunos dispositivos limitan sus implementaciones a las capacidades básicas de Almacenamiento de Imágenes, Consulta/Recuperación *(Quero/Retrieve*), y, a veces, impresión *(Print).* Para identificar estos diversos servicios, a veces se habla de diferentes "niveles" de DICOM. A efectos de incluirlo en la documentación de los requisitos DICOM, un término mejor seria "perfiles de utilización" de DICOM, término

que se explicara más adelante, en la sección relativa a la definición de los requisitos DICOM.

Los fabricantes han utilizado DICOM para enviar, recuperar e imprimir imágenes. Recientemente, han comenzado a utilizar servicios DICOM para gestionar datos también. Por ejemplo, un equipo tal como una Tomografía Axial Computarizada (TAC) adquiere, por ejemplo, un estudio con 120 cortes axiales, y los envía a un sistema PACS utilizando el servicio de Almacenamiento DICOM. Para poder comunicar el número de imágenes que pertenecen a ese Componente especifico del Estudio, para solicitar del sistema PACS el Compromiso de almacenarlos, y para comunicar el sistema de información (SI) que elimine este estudio especifico de su lista de planificación, se necesitan esos servicios DICOM de imagen y gestión de la información.

El estándar incorpora unas mejoras principales respecto a las versiones anteriores del estándar:

1. Se aplica a un ambiente de red. Las versiones anteriores eran aplicables sólo a ambientes de punto a punto; para la operación en un ambiente de red una unidad de interfaz de Red (*Network Interface Unit,* NIU) era requerida. La Versión 3.0 de DICOM puede soportar los estándares de comunicación en red que se usan en industria, como los protocolos OSI Y TCP/IP.
2. Especifica como los dispositivos deben cumplir con el estándar para que los datos y las instrucciones sean intercambiadas. Las versiones anteriores han sido limitadas a la transferencia de datos, pero la Versión 3.0 DICOM específica, por el concepto de clases de servicio, la semántica de órdenes y datos asociados.
3. Especifica los niveles de conformidad. Las versiones anteriores especificaron un nivel mínimo de conformidad. DICOM la Versión 3.0, explícitamente, describe como una implementación debe estructurar una declaración de conformidad para seleccionar opciones específicas.
4. Está estructurado como un documento dividido en varias partes. Esto, facilita la rápida evolución del estándar simplificando la adición de rasgos nuevos. Las directrices ISO que definen como estructurar documentos de multi-partes han sido seguidas en la construcción del estándar DICOM.
5. Introduce objetos de información explícitos no sólo para imágenes y gráficos, también para estudios, informes, etc.

6. Especifica una técnica establecida para la identificación única de cualquier objeto de información lo que facilita las definiciones inequívocas de las relaciones entre objetos de información como ellos son interpretados a través de la red.

El estándar DICOM no especifica:

- Los detalles de cualquiera de los rasgos del estándar para que un dispositivo reclamando conformidad para una implementación.
- El juego total de rasgos y funciones de un sistema integrado por un grupo de dispositivos cada uno siguiendo la norma del estándar.
- Un procedimiento de pruebas y validación para evaluar de que forma se cumple con el estándar.

4.6.2. Objetivos de DICOM

El estándar DICOM facilita la interoperabilidad de dispositivos que cumplen con este. En particular:

- Dirige la semántica de comandos y datos asociados. En dispositivos que actúan recíprocamente, debe haber normas sobre como se espera que estos dispositivos reaccionen a esas ordenes y esos datos asociados, no solamente a como la información debe ser intercambiada entre dispositivos.
- Dirige la semántica de los servicios de fichero, formatos de fichero y directorios de información necesaria para la comunicación fuera de línea.
- Explicita en la definición de las exigencias de conformidad de las puestas en práctica del estándar. En particular, una declaración de conformidad debe especificar bastante información para determinar las funciones para las que pueden esperar la interoperabilidad con otro dispositivo reclamando la conformidad.
- Facilita la operación en un ambiente conectado a una red, sin la exigencia de unidades de interfaz de red.
- Está estructurado para acomodar la introducción de servicios nuevos, facilitando así el soporte para usos futuros de aplicaciones con imágenes médicas.
- Hace uso de normas existentes internacionales, y ellas mismas siguen las directrices de documentación establecidas para normas internacionales.

Si bien el estándar DICOM tiene el potencial para facilitar las puestas en práctica de soluciones PACS, el empleo del estándar sólo, no garantiza que todos los objetivos de un PACS sean cumplidos. El estándar facilita la interoperabilidad de sistemas siguiendo la conformidad en un ambiente de multivendedor, pero por si mismo no garantiza la interoperabilidad.

Ha sido desarrollado con énfasis en el diagnóstico médico como el practicado en radiología y en disciplinas relacionadas; sin embargo, se piensa que también es aplicable a una amplia gama de información de imágenes relacionadas en los ambientes clínicos.

4.6.3. DICOM y otros estándares

Como puede verse en la figura 29, hay fuertes vínculos entre DICOM y otras organizaciones, para garantizar que existe un auténtico soporte internacional, al objeto de evitar duplicidades y aumentar su credibilidad. En particular, dos entes de estándares internacionales tienen comités encargados de temas de Informática Sanitaria, a saber: el CEN con su TC CEN/TC 251 y la ISO, que cuenta con el Comité Técnico ISO/TC 215. DICOM cuenta con presencia en ellos porque estableció una alianza llamada de Tipo A con el Comité Técnico 215 de ISO cuando fue creado en 1999, por ello, ISO/TC 215 ha decidido no crear un grupo de trabajo para tratamiento de imágenes, sino delegar en DICOM para los estándares de tratamiento de imágenes bio-médicas. DICOM es un aliado oficial de ISO/TC 215.

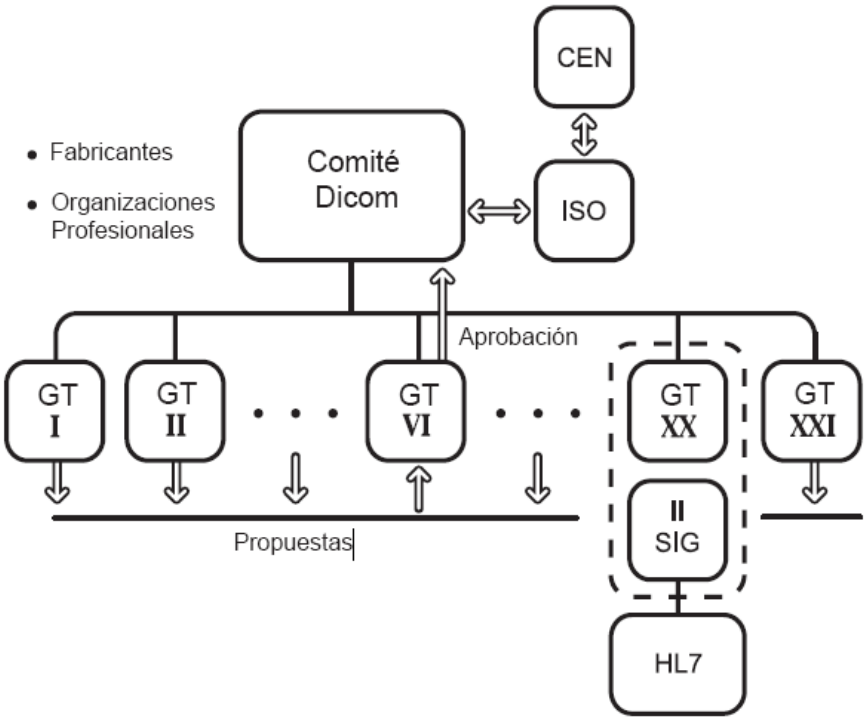


Figura 29. Propuestas de estándar (Otech 2008).

Los estándares no europeos no participan normalmente en el trabajo de los grupos de CEN, pero sin embargo si hubo miembros de DICOM que participaron en los trabajos iniciales. Los servicios Lista de Trabajo del Equipo (*Modalito Worklist*) y Compromiso de Almacenamiento (*Storage Commitment*) fueron producidos por Equipos de Proyecto del CEN. Los grupos de trabajo del estándar Europeo abandonaron la participación activa hace unos pocos años. Sin embargo, el comité DICOM y diversos grupos de trabajo, que también cuentan con miembros Europeos (así como de otros continentes no americanos) se reúnen regularmente en Europa. CEN ha adoptado DICOM por referencia, es decir, básicamente lo ha convertido en un estándar Europeo también. El Secretario Europeo es parte del Comité de Coordinación Europeo para la Industria Radiológica y Electrotécnica (COCIR), el equivalente Europeo del NEMA.

DICOM también se reúne conjuntamente con la organización de los estándares HL7, para la coordinación de ambos estándares y facilitar la integración del tratamiento de imágenes en las organizaciones sanitarias, Paralelamente, la convergencia con DICOM de un formato Japonés para medios de intercambio llamado *Image Sae and Carry* (IS&C) requirió mucho trabajo conjunto en el cual jugaron un importante papel Asociación de Industrias Japonesas de Sistemas Radiológicos (JIRA) y *Medical Information System Development Center* (MEDIS-DC).

DICOM también esta centrando su atención en la evolución de los estándares vinculados a Internet. La estrategia de DICOM e integrar las recomendaciones de Internet tan pronto como sean estables y ampliamente adoptadas por productos comerciales de consumo. En esta evolución, se tiene especial cuidado en asegurar que se mantenga la consistencia del estándar DICOM con su amplia base instalada. DICOM ya utiliza las intranets estándar de las organizaciones sanitarias, y es posible intercambiar por correo electrónico objetos con tipo *Multipurpose Internet Mail Extension* (MIME).

4.6.4. Relación entre las partes del estándar DICOM

Todas las partes del estándar DICOM están interrelacionadas. Como puede verse en la figura 30, la Parte 1 proporciona una vista general, explica algunos conceptos básicos, y es un buen comienzo cuando se es nuevo en el estándar DICOM. También explica el contenido de las diversas partes, y su interrelación. La Parte 2 especifica la conformidad DICOM. Se requiere que cada dispositivo conforme con DICOM especifique qué servicios DICOM proporciona, de forma que los usuarios potenciales, integradores de sistemas, y otros, pueden determinar su conectividad. La parte 2 especifica que conformidad DICOM debe cumplir un producto, y también proporciona una plantilla con diversos ejemplos de declaraciones de conformidad DICOM típicas. Esta parte es crítica para cualquiera que deba redactar una "Declaración de Conformidad DICOM", y también resulta útil para todo aquel que quiera ser capaz de interpretar una. Puesto que trata de todos los aspectos de DICOM, tiene relación con todas las demás Partes de la figura 30.

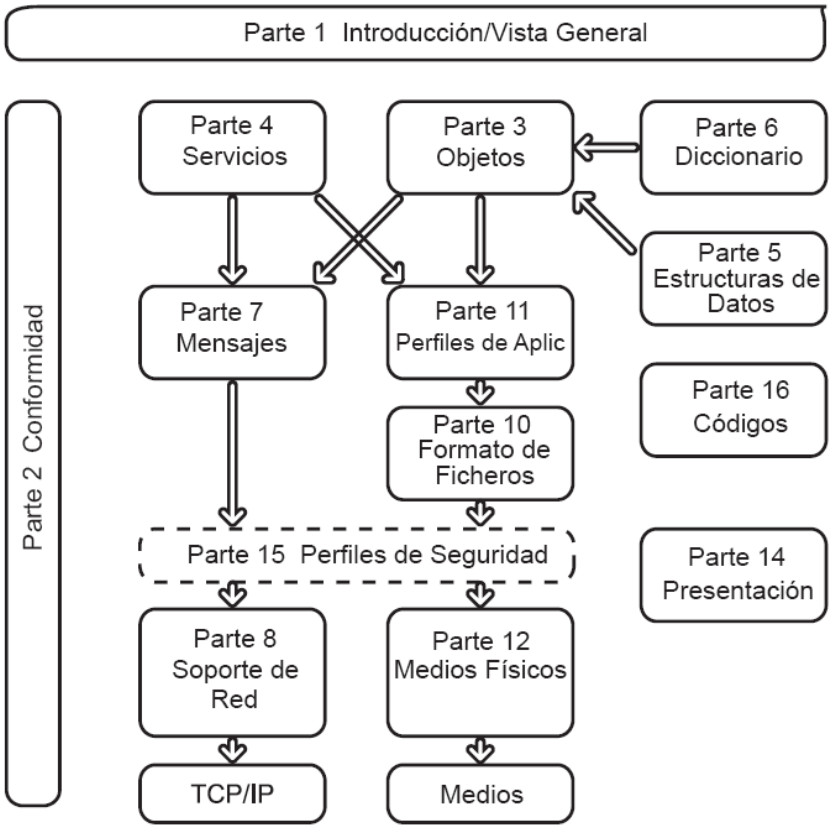


Figura 30. Relación entre partes del estándar DICOM (Otech 2008).

La parte 4 del estándar especifica las Clases de Servicio. Describe todos los servicios DICOM, desde impresión a almacenamiento y lista de trabajo del equipo. Se especifica con todo detalle la información intercambiada con estos servicios. Si se desea averiguar con precisión qué información se intercambia cuando, por ejemplo, se envía una imagen a una impresora, digamos para saber como cambiar el formato de impresión, se pueden encontrar los detalles en esta parte. Estos servicios utilizan comandos específicos para realizar sus funciones. DICOM cuenta con diversos comandos, dependiendo de la funcionalidad que necesita cumplir, y esos comandos están descritos en la Parte 7. La Parte 8 especifica exactamente la interfaz de DICOM con el protocolo estándar de comunicaciones, es decir, TCP/IP. Programadores a nivel de aplicación probablemente necesiten utilizar un juego de herramientas DICOM que se encargará de todos los detalles de la interfaz y seguramente no necesite mirar las Partes 7 y 8. Pero hay que tener en cuenta que estas partes también contienen todos los errores que se pueden dar durante la comunicación DICOM, incluyendo sus códigos y

descripciones, por consiguiente a efectos de resolución de problemas y soporte podría ser necesario utilizar estas partes como recurso.

Los servicios descritos en la Parte 4 utilizan los comandos descritos en la Parte 7, pero a menudo intercambian objetos de información, tales como imágenes, listas de planificación, etc. Estos objetos de información están todos descritos en la Parte 3, la parte más voluminosa y de mayor crecimiento. La razón para estar en perpetuo cambio es que continuamente se están definiendo nuevos objetos de imagen. Por ejemplo, si se desea saber exactamente cómo y cuándo se intercambia la información de máscara en una imagen de hemodinámica, se necesitara consultar la Parte 3. Esta Parte también especifica la definición del tipo (*Type*) para cada elemento de datos individual, es decir, si es obligatorio, opcional o condicional el que ese elemento específico forme parte del mensaje DICOM. La Parte 3 es el "corazón" del estándar. Hay dos parte que dan soporte a la Parte 3, a saber, la Parte 6, el Diccionario de Datos, que contiene una lista de todos los atributos individuales que se utilizan para construir esos objetos DICOM; y la Parte 5, las Estructuras de Datos que especifican la representación de su valor, que puede compararse a una definición de "tipo" en *software*. Todos los elementos DICOM tienen una representación especifica de su valor, por ejemplo, todos los nombres son del tipo PN (*Person Name*, Nombre de Persona), todas las fechas de tipo DA (*Date*, Fecha), etc. Para determinar cómo están definidas todas estas representaciones de valor, por ejemplo, que formato se utiliza para la fecha (*YYYYMMDD*), se debe consultar la Parte 5, que contiene las Estructuras de Datos y Semántica (*Data Structures and Semantics*). Así pues, la Parte 6, el Diccionario de Datos, y la Parte 5, las Estructuras de Datos, soportan ambas a la Parte 3, los Objetos DICOM. La Parte 9 especifica la comunicación punto-a-punto que utilizaba el conector de 50 contactos tal y como definía el predecesor de DICOM, es decir, ACR-NEMA. Ya no hay necesidad alguna de consultar esta parte, excepto por razones históricas, o si se da la circunstancia de "topar" con un dispositivo antiguo de los años 80 que aun tenga ese tipo de conexión. Las Partes 10, 11 y 12 tratan del formato de intercambio de los medios DICOM. La Parte 10 especifica el formato de archivos, incluyendo la estructura del Directorio. Hay que tener en cuenta que DICOM tiene una estructura de directorio muy particular, que permite leer un CD de intercambio DICOM, por ejemplo, y mostrar la información clave de todas las imágenes (que pueden ser miles en el caso de un CD o DVD) y recuperarlas. Puesto que este directorio es específico de DICOM, se necesita una aplicación DICOM para interpretar la información. EL sistema de archivos en si es el estándar de los PCs, lo que significa que

cuando se abre mediante, digamos, el Explorador de MS Windows, se pueden ver los archivos presentes. La Parte 11 contiene los perfiles de aplicación específicos, que están definidos para aplicaciones concretas, tales como hemodinámica, ultrasonidos, etc. Conteniendo la definición de un subconjunto especifico de imágenes y las posibilidades de codificación para almacenar estas imágenes. Los medios físicos se definen en la Parte 12. Si se desea saber, por ejemplo, que tipo de DVD soporta el estándar DICOM, ahí es donde se puede encontrar esa información. La Parte 13 fue definida tempranamente durante el proceso de estandarización de DICOM para estandarizar el protocolo de impresión punto-a-punto, y ha sido suprimida. La Parte 14 trata acerca de la calidad de imagen. Contiene la Función de Representación estándar para Escala de Grises DICOM, que mapea los valores de los píxeles a un rango estándar de valores de luminancia para la representación en pantalla, y a valores de densidad para representación impresa, tal como por ejemplo en película. Si se necesita alguna vez implementar esta característica o se tienen dudas sobre calibración, aquí es donde se puede encontrar esta información. La Parte 15 trata de la seguridad definida en DICOM. Se identifican perfiles específicos, dependiendo del nivel de seguridad que se desee soportar. La Parte 16, Recurso para el Mapeado de Contenido, contiene todos los códigos y plantillas. Los objetos DICOM más recientes utilizan códigos cada vez más a menudo para identificar de forma única ítems tales como partes del cuerpo, códigos de visualización, etc. Las plantillas son críticas para facilitar la interoperabilidad cuando se utilizan informes estructurados (*Structured Reports,* SR).

No hay que abrumarse por las numerosas partes de que constan los estándares DICOM, hay que utilizar la figura como una guía para encontrar lo que se necesita. En particular, la versión electrónica el estándar DICOM resulta conveniente porque permite moverse fácilmente entre las distintas partes, lo cual, desafortunadamente, es necesario en la mayoría de los casos.

4.6.5. Partes del Estándar DICOM

DICOM 3.0 consiste en las siguientes 18 partes, aunque alguna de ellas ha sido retirada:

PS 3.1. Introducción y visión general (NEMA Part 1 2008).

PS 3.2. Conformidad. Define los principios que las implementaciones deben seguir (NEMA Part 2 2008):

- Exigencias de conformidad. Esta parte especifica las exigencias generales que deben ser encontradas por cualquier realización. Hace referencia a secciones de conformidad de otras partes del estándar.
- Declaración de conformidad. Esta parte define la estructura de un estado de conformidad. Especifica la información que debe estar presente en una declaración de conformidad. Hace referencia a las secciones del estado de conformidad de otras partes del estándar.

PS 3.3. Definición de objetos. Especifica unas clases de objeto de información (IOD) que proporcionan una definición abstracta de entidades del mundo real, aplicables a la comunicación de imágenes digitales médicas. Cada clase de objeto de información consiste en la descripción de su objetivo y los atributos que lo definen. Un IOD no incluye los valores para los atributos que comprenden su definición. Para facilitar el crecimiento futuro del estándar y mantener la compatibilidad con las versiones anteriores del estándar, dos tipos de IOD´s están definidos: normalizado y compuesto (NEMA Part 3 2008).

PS 3.4. Especificaciones de las *Service Class.* Define un número determinado de Clases de Servicio. Una Clase de servicio asocia uno o más objetos de la información con uno o más comandos para ser representados por estos objetos. Las especificaciones de la clase de servicio indican los requisitos para los elementos de comando y cómo los comandos que resultan se aplican a los objetos de la información. Las especificaciones de la clase de servicio indican los requisitos para proveedores y usuarios de los servicios de comunicación. Además, define las características almacenadas por todas las clases de servicio, y como es estructurada una sentencia de conformidad a un a clase de servicio individual. Esto contiene un número de anexos normativos que describen las clases de servicio individuales en detalle (NEMA Part 4 2008).

PS 3.5. Estructura de datos y codificación. Especifica cómo las aplicaciones DICOM construyen y codifican la información del conjunto de datos resultantes de usar los objetos de la información y las clases de servicio definidas en la Parte 3 y 4 del estándar DICOM. El apoyo de un número de técnicas de compresión de imágenes estándar (por ejemplo JPEG) está especificado. Esta parte dirige las reglas de codificación necesarias para construir un flujo de datos para ser convertido en un mensaje como está especificado en Parte 7 del estándar DICOM. Este flujo de datos es producido por la colección de elementos de datos creada por el *set* de datos. También

define la semántica de un número genérico de funciones que son comunes a muchos de los objetos de la información. Además, define las reglas de codificación para conjuntos de caracteres internacionales mediante DICOM (NEMA Part 5 2008).

PS 3.6. Diccionario de datos. Es el registro centralizado que define la colección de todos los elementos de datos DICOM disponibles para representar la información.

Para cada elemento de datos, esta parte (NEMA Part 6 2008):

- Asigna una etiqueta única, que consiste en el grupo y el número de elemento
- Un nombre
- Especifica sus características de valor (la cadena de caracteres, el número entero, etc.)

  Especifica su valor de multiplicidad (cuantos valores por atributo)
- Si está retirado.

PS 3.7. Intercambio de mensajes. Especifica tanto el servicio como el protocolo usado por una entidad de aplicación en un ambiente de imágenes médicas para intercambiar mensajes sobre los servicios de apoyo de comunicaciones definidos en Parte 8 o 9. Un mensaje está compuesto por un flujo de órdenes definido en la parte 7 seguido por un flujo de datos opcional como lo definido en la parte 5 (NEMA Part 7 2008). Especifica:

- Las operaciones y notificaciones (servicios *DICOM Message Service Element,* DIMSE) hechos disponibles a las clases de servicio definidas en la Parte 4.
- Reglas para establecer y terminar asociaciones provistas por el apoyo de comunicaciones especificado en la Parte 7, y el impacto sobre las transacciones pendientes.
- Reglas que gobiernan el intercambio de comandos de petición y respuesta.
- Reglas de codificación necesarias para construir flujos de comandos y mensajes.

PS 3.8: Soporte de la comunicación de la red para el intercambio de mensajes. Especifica los servicios de comunicación y los protocolos de capas superiores necesarios para soportar, en una red, la comunicación entre entidades de aplicación DICOM según lo especificado en las Partes 7, 6, 5, 4 y 3. Estos servicios de comunicación y protocolos aseguran que la comunicación entre entidades de aplicación DICOM sea realizada de una manera eficiente y coordinada a través de la red (NEMA Part 8 2008).

PS 3.9. Retirado.

PS 3.10. Almacenamiento de los medios y formato de ficheros para el intercambio de datos. Especifica un modelo general para el almacenamiento de información de imágenes médicas sobre medios desmontables. El propósito de esta parte es proveer una estructura permitiendo el intercambio de varios tipos de imágenes médicas e información relacionada sobre un ancho rango de medios físicos de almacenaje (NEMA Part 10 2008). Además, especifica:

- Un modelo de capas para el almacenamiento de imágenes médicas e información relacionada sobre los medios de almacenamiento. Este modelo introduce el concepto de perfiles de aplicación de medios de almacenamiento, que especifican subconjuntos específicos de aplicación del estándar DICOM al cual una implementación de medios de almacenamiento puede demandar conformidad. Tal conformidad se aplica solo a la escritura, lectura y actualización de los contenidos del medio de almacenamiento.
- Un formato de fichero DICOM soportando el encapsulado de cualquier objeto de información.
- Un formato de fichero DICOM seguro soportando el encapsulado de un formato de fichero DICOM en un sobre criptográfico.
- Un servicio de fichero DICOM proveyendo independencia del formato de medio subyacente y medio físico.

PS 3.11. Perfiles de aplicación de almacenamiento de medios. Especifica un subconjunto del estándar DICOM para el cual una implementación puede demandar conformidad. Este subconjunto especifico de aplicación estará referenciado a un perfile de aplicación en esta sección. Tal sentencia de conformidad aplica al intercambio ínter operable de imágenes médicas e información relativa sobre medios de almacenamiento para usos clínicos específicos. Esto sigue la estructura, definida en la Parte 10, para el intercambio de varios tipos de información sobre medios de almacenamiento (NEMA Part 11 2008).

PS 3.12. Funciones de almacenamiento y formato de medios para el intercambio de datos. Esta parte del estándar DICOM facilita el intercambio de información entre aplicaciones en ambientes médicos gracias a especificar (NEMA Part 12 2008):

1. Una estructura para describir la relación entre el modelo de medios de almacenamiento y un medio físico específico y el formato del medio.
2. Especifica las características del medio físico y los formatos de medios asociados.

PS 3.13. Retirado.

PS 3.14. Función de visualización estándar de la escala de grises. Especifica una función de visualización estandarizada para la visualización consistente de imágenes en escala de grises. Esta función provee métodos para el calibrar un sistema de visualización particular con el propósito de presentar imágenes consistentemente sobre diferentes medios de visualización (ejemplo monitores e impresoras) (NEMA Part 14 2008).

La función de visualización elegida está basada en la percepción visual humana. La sensibilidad de contraste del ojo humano es claramente no lineal con el rango de luminiscencia de los dispositivos de visualización.

PS 3.15. Seguridad y perfiles de gestión del sistema. Especifica la seguridad y los perfiles de gestión del sistema para que las implementaciones puedan demandar conformidad. La seguridad y los perfiles de gestión del sistema son definidos mediante referenciación externa de protocolos estándar desarrollados, tales como *Dynamic Host Configuration Protocol* (DHCP), *Lightweight Directory Access Protocol* (LDAP), *Transport Layer* Security (TLS) e *Integrated Secure Communication Layer* (ISCL). Los protocolos de seguridad pueden usar técnicas de seguridad como llaves públicas y "tarjetas inteligentes". La codificación de datos puede usar varios esquemas de encriptación estandarizados (NEMA Part 15 2008).

Esta parte no dirige temas de políticas de seguridad. El estándar solo provee mecanismos que puedan ser usados para implementar políticas de seguridad con consideración del intercambio de los objetos DICOM. Esto es responsabilidad del administrador local para establecer la política de seguridad apropiada.

PS 3.16. Contenido de recursos mapeados. Especifica (NEMA Part 16 2008):

- Plantillas para estructurar documentos como objetos de información DICOM.
- Conjuntos de términos codificados para usar en objetos de información.
- Un léxico de terminaos definidos y mantenidos por DICOM.
- Traducciones especificas del país de términos codificados.

PS 3.17. Información aclaratoria. Especifica (NEMA Part 17 2008).

- **Anexos informativos y normativos que contienen información explicativa.**

  Esta parte del estándar DICOM contiene información explicatorio en forma de anexos informativos y normativos.

  De esta forma nos encontraremos con las siguientes partes:

  Anexo A Explicación de la orientación del paciente (Normativa).

  Anexo B Integración de modalidad de lista de trabajo y modalidad de desarrollo de pasos de procedimiento en el estándar original DICOM (informativo).

  Anexo C Ondas (Informativo).

  Anexo D Ejemplos de codificación SR (Informativo).

  Anexo E Mamografía CAD (Informativo).

  Anexo F CAD de pecho (Informativo).

  Anexo G Explicación de criterio de agrupación para IODs de grupo funcional multi-frame (Informativo).

  Anexo H Ejemplos de flujo de trabajo de identificación de ensayo clínico (Informativo).

  Anexo I Plantillas de ultrasonidos (Informativo).

  Anexo J Manejo de parámetros de identificación (Informativo).

  Anexo K Gestión de datos de protocolo representados de ultrasonidos.

  Anexo L Estructura de informe de hemodinámicas (Informativo).

  Anexo M Informes de ultrasonidos vascular (Informativo).

  Anexo N Informes de procedimiento de ecocardiogramas (Informativo).

  Anexo O Registro (Informativo).

  Anexo P Transformaciones y mapas (Informativo).

  Anexo Q Informes de imágenes de pechos (Informativo).

  Anexo R Configuración de casos de uso (Informativo).

  Anexo S Legado de transpón para gestión de configuración (Informativo).

  Anexo T Referencias de análisis cuantitativas (Informativo).

  Anexo U Casos de usos de oftalmología (Informativo).

  Anexo V Protocolos de manejo (Informativo).

  Anexo W Firmas digitales en casos de uso de informes estructurados (Informativo).

  Anexo X Informe basado en dictado con referencias de imágenes.

Anexo Y Funciones LUT VOI (Informativo).

Anexo Z Transformaciones de referencia de centros de rayos X (Informativo).

Anexo AA Casos de uso de información de dosis de radiación.

Anexo BB Impresión (Informativo).

Anexo CC Compromiso de almacenamiento (Informativo).

Anexo DD Lista de trabajo (Informativo).

Anexo EE Preguntas de información de paciente relevantes (Informativo).

Anexo FF Plantillas de informe de análisis cardiovascular CT/MR (Informativo).

Anexo GG Negociación de sintaxis de transferencia de datos de píxel referenciados JPIP.

Anexo HH Ejemplo de codificación de segmentación (Informativo).

PS 3.18. Acceso Web a DICOM *Persisten Objects* (WADO). Especifica las maneras por las cuales una pregunta para acceder a un objeto persistente DICOM pueden ser expresadas como una petición HTTP URL/URI, que incluyen un puntero a un objeto persistente DICOM especifico en la forma de una instancia UID (NEMA Part 18 2008).

4.6.6 . Suplemento 91: Clases SOP de imágenes fotográficas oftalmológicas

Este suplemento del estándar DICOM introduce tres nuevas IODs y clases SOP para técnicas de imagen fotográfica oftalmológica usando luces en el visible y cerca del visible. Las primeras dos IODs son para imágenes oftalmológicas, mientras que la tercera se usa para definir relaciones de imagen para pares estereoscópicos de imágenes. Las nuevas IODs se usarán con varios tipos de dispositivos de imágenes fotográficas oftalmológicas que usan luz y ópticas para producir imágenes de anatomía externa de la cornea, y también estructuras retinales internas. Las imágenes creadas pueden ser estructuras simples o multi estructuras, de colores o monocromos. Las imágenes fotográficas oftalmológicas pueden incorporan amplificación, tintes y filtros de color, y fotografía estereoscópica. El realizado de imágenes fotográficas oftalmológicas requiere mucha especificación e identificación detallada del equipamiento usado y de los atributos de adquisición relativos para la apropiada interpretación de las imágenes.

Este suplemento no direcciona tomografías coherentes ópticas (OCT) o endoscopias oftalmológicas. Para las endoscopias oftalmológicas, la existencia de VL

simple o multiframe o clases SOP de almacenamiento de imágenes endoscopias deberá ser usado.

Este suplemento propone cambios a las siguientes partes del estándar DICOM (DICOM Standards Committee 2004):

PS 3.3	*Information Object Definitions*
PS 3.4	*Service Class Specifications*
PS 3.6	*Data Dictionary*
PS3.16	*Content Mapping Resource*

Alcance y campo de aplicación

Los fabricantes de dispositivos de imagines fotográficas oftalmológicas han cambiado de películas a crear datos digitales directamente, y han identificado la necesidad de una definición de objeto de imagen DICOM compatible para el intercambio de información. El grupo de trabajo 9 evaluó el IOD de luz visible fotográfico y el nuevo IODs de captura secundario multiframe para la adecuada representación de imágenes fundidas. Alguna de las razones de que estas IODs existentes no son adecuadas son (DICOM Standards Committee 2004):

1. Equipamientos hechos para la realización de imágenes fotográficas oftalmológicas incluyen una amplia variedad de fuentes de luz, sistemas de lentes, y tipos de cámaras digitales. El visualizador de imágenes oftalmológicas algunas veces necesita especificar la descripción de equipo para una interpretación apropiada. El equipamiento general y los módulos de equipamiento SC especificados en el VL y en las definiciones de objeto de imagen SC se complementan con un módulo de equipamiento oftalmológico para una adecuada identificación.
2. La fotografía de fundido y los dispositivos angiográficos fluorescentes producen combinaciones de imágenes simples, múltiples, y conjuntos de datos de cine dependiente del tipo de equipamiento y procedimientos llevados a cabo. Las clases SOP propuestas proveerán la flexibilidad de cambio a todos estos formatos de imagen usando un módulo multiframe.
3. Un requerimiento básico importante para la realización de imágenes fotográficas oftalmológicas es identificar pares estereoscópicos, conjuntos de imágenes de ojo derecho e izquierdo, y conjuntos de imágenes angiográficas. Los pares

estereoscópicos de imágenes se guardan juntos a través del uso de una entidad de relación estereoscópica nueva.

4. Un requerimiento básico importante para los procedimientos basados en agentes de contraste oftalmológico es identificar el tiempo de sellado para conjuntos de imágenes angiográficas. La sincronización y atributos de tiempo se incluyen en esos módulos para proveer las relaciones temporales necesarias.
5. Porque los IODs propuestos rodean varias técnicas de imaginado fotográfico, una especificación de adquisición extendida y valores de código correspondientes serán requeridos para la descripción completa.

4.7. Internacional Organization for Standardization (ISO)/TC 215

ISO es una federación mundial de organismos nacionales de normalización que son miembros de ISO. El trabajo de preparación de las Normas Internacionales se realiza a través de Comités Técnicos ISO (ISO 2008). Los borradores de estándares internacionales (*Draft International Standards*) adoptados por los comités técnicos se circulan a los organismos miembros, como AENOR en España, para su votación. La publicación como un Estándar Internacional requiere la aprobación por al menos el 75% de los organismos miembros. ISO es conocida por su amplia gama de estándares utilizados en numerosos aspectos de sistemas de información, que tienen lugar dentro del *Joint Technical Committee* (JTC). Una de las contribuciones más populares y de amplio impacto es el conjunto de normas que soportan el modelo *Open Systems Interconnection* (OSI) para comunicaciones (Monteagudo 2003).

En 1999 ANSI tomó la iniciativa de promover la creación de un Comité ISO TC215 dedicado a Informática de la Salud para cuya constitución ha contado con el soporte de CEN y en el que participa AENOR. Las áreas de trabajo de este comité son:

- Mensajes y comunicación.
- Representación de conceptos médicos.
- Seguridad.
- Coordinación de modelado.
- Historias clínicas.
- Imágenes biomédicas.

En agosto de 1999 comenzaron los trabajos del ISO/TC215 para desarrollar un conjunto de estándares sobre los requerimientos de la arquitectura de referencia de la HCE. El proyecto ha conducido a la Especificación Técnica ISO 18308 *"Requirements for an Electronic Health Record Reference Architecture*". Conviene recalcar que el estándar no contiene requisitos funcionales para un sistema de HCE sino como dice el propio estándar: "…un conjunto de requisitos clínicos y técnicos para una arquitectura de historia clínica que soporta el uso, compartimiento e intercambio de registros electrónicos entre y a través de diferentes sectores de salud, diferentes países y diferentes modelos de asistencia sanitaria". Otro de los estándares para HCE de ISO es el ISO DTR 20514 *"EHR definition and scope"* que se encarga de definir la HCE y especificar sus objetivos.

Los usuarios principales de esta especificación técnica ISO de requisitos son los desarrolladores de estándares de arquitecturas, p. ej. CEN 13606, y otras arquitecturas de referencia tales como el Modelo de Referencia openEHR. El desarrollo de la ISO 18308 se ha producido en tres etapas. En la primera se realizó una búsqueda exhaustiva de material de referencia a través de la literatura y contactos directos seleccionándose 35 fuentes primarias incluyendo 20 fuentes originalmente recogidas por el proyecto EHCR-SupA en Europa. En la segunda fase se trabajó con los más de 700 requisitos identificados en la primera fase y se desarrolló una estructura jerárquica de "encabezamientos" bajo los cuales se podían organizar los requisitos. Tras eliminar las redundancias se quedaron reducidas al final de esta fase a 590. En la fase final se consolidó un conjunto de 123 requisitos listados bajo una estructura de 10 títulos y 60 subtítulos.

4.8. European Committee for Standardization / Technical Committee (CEN/TC) 251

Comité Europeo de Normalización (CEN) es una organización fundada en 1961 que contribuye a los objetivos de la Unión Europea y el Área Europea Económica con los estándares voluntarios técnicos que promueven el comercio libre, la seguridad de trabajadores y consumidores, interoperabilidad de redes, protección de medio ambiente, explotación de programas de investigación y desarrollo, y la consecución pública.

El Comité Técnico TC 251 *"Health informatics"* es uno de los múltiples sectores en los que se organiza el CEN. Su objetivo es alcanzar la compatibilidad y la interoperabilidad entre sistemas independientes. Esto implica unos requisitos de la estructura de información sanitaria para apoyar procedimientos clínicos y administrativos, exigencias de calidad y de seguridad, así como métodos técnicos de apoyar sistemas interoperables (CEN 2008).

CEN TC 251 se diferencia de otras organizaciones de estandarización ya que es una Organización de Desarrollo de Normas (SDO) europea entre internacional o SDOs de ámbito específico, cuyo objetivo es casi exclusivamente contenido tecnológico y no la tecnología de comunicación. El Plan de negocio del CEN/TC251 está integrado en un proceso de cambio. Tanto la nueva dirección del Comité Técnico (TC) como el mandato de la Comisión Europea (CE), que dirigen este campo, requerirán esfuerzos conjuntos del CEN, CENELEC Y ETSI para conseguir armonizar el trabajo con el de otras SDOs internacionales (CEN/TC 251 2008). Con respecto a su estructura, el CEN TC 251 se está compuesto por varios grupos de trabajo (WG). El grupo de trabajo 1 Modelos de información *(WG1 Information models),* en el que nos centraremos ya que se encarga de elaborar los estándares referentes a la arquitectura y modelos de información del HCE. Magdalena *et ál.* 1999 emplean diferentes módulos del estándar CEN/TC 251 en una red de telemedicina para cardiología: NEMESIS.

En 1999 CEN publicó el prENV 13606 "Comunicación con la historia clínica electrónica", en el que se proponía una serie de medidas de interoperabilidad para facilitar la comunicación entre sistemas heterogéneos. De forma que se consideraba que el formato de los datos clínicos se debía mantener y presentar en el sistema receptor aunque las arquitecturas subyacentes se modificaran. Esta prenorma explicada en estaba formada por cuatro partes (CEN/TC 251 2008):

- **prENV 13606-1. Arquitectura extendida *(Extended Architecture).*** En esta parte se definen los componentes de la arquitectura necesarios para permitir que el contenido de una historia clínica pueda ser construido, usado, compartido y mantenido. Por lo que respecta al modelo de dominio, el resultado será una descripción formal del contexto que envuelve a la HC.
- **prENV 13606-2. Dominio de lista de términos *(Domain Term List).*** Esta parte proporciona definiciones de un conjunto de términos usados por el personal sanitario. El objetivo es permitir que todos los profesionales sanitarios puedan

transmitir y reconocer el significado de la información clínica en función de la posición de estos términos dentro de la historia clínica.

- **prENV 13606-3. Reglas de distribución *(Distribution Rules).*** Define un conjunto de reglas para el intercambio de registros médicos cumpliendo las obligaciones legales de seguridad. Por intercambio se entiende cualquier método por el cual los registros médicos se ponen a disposición de los profesionales sanitarios. Incluye el acceso a un archivo físico de historias clínicas, la creación de una historia clínica virtual que se construye a partir de la información almacenada en diversas fuentes y la comunicación de historias clínicas, o partes de éstas, entre sistemas.
- **prENV 13606-4. Mensajes para el intercambio de registros de información *(Messages for the Exchange of Record Information).*** Esta parte del prestándar especifica diferentes tipos de mensajes que permiten el intercambio de información de HCE entre centros o personal sanitario. Estos mensajes permiten que la información contenida en una HCE pueda ser enviada de un profesional a otro. En esta parte se consideran dos propiedades de las comunicaciones, la legibilidad y la capacidad de procesamiento de información.

Con el fin de completar y adaptar el preestándar a nuevos requisitos, el prENV 13606 se sometió a un periodo de consulta pública. A partir de múltiples opiniones, la mayoría basadas en experiencias prácticas con sistemas comerciales, y la colaboración de representantes de HL7 y openEHR se desarrolló la versión revisada del ENV 13606. El objetivo de esta cooperación es armonizar el estándar con las especificaciones de openEHR y de HL7 para facilitar la interoperabilidad entre ellos, es decir, mejorar la capacidad de los sistemas heterogéneos de comunicarse e intercambiar procesos o datos El documento resultante de la revisión prEN13606 contiene cinco partes (CEN/TC 251 2008):

- **prEN 13606-1. Modelo de Referencia *(Reference Model).*** Esta parte es un refinamiento de la prENV 13606-1. En ella se define un modelo de información genérico para comunicar con la historia clínica electrónica de cualquier paciente. Hay que destacar que en la elaboración de esta parte se integraron contribuciones de otras organizaciones como HL7 y openEHR para contribuir a la interoperabilidad entre sus estándares.

- **prEN 13606-2. Especificación de Intercambio de Arquetipos *(Archetype Interchange Specification).*** Esta parte se incluye con motivo del modelo dual adoptado, estableciendo un modelo de información genérico y un lenguaje para representar y comunicar la definición de instancias individuales de arquetipos.
- **prEN 13606-3. Arquetipos de Referencia y Listas de Términos *(Reference Archetypes and Term Lists).*** Esta parte se aprovecha del prENV 13606-2, define un conjunto de arquetipos y más listas para soporte de otras partes de este estándar.
- **prEN 13606-4. Características de seguridad *(Security Features).*** Esta parte se basa en el prENV 13606-3. Establece conceptos del modelo de información que se precisa reflejar dentro de instancias de HCE individuales. Esto se realiza con el fin de conseguir una interacción apropiada con los componentes de seguridad que pudieran ser requeridos en cualquier implantación futura de HCE.
- **prEN 13606-5. Modelos de Intercambio *(Exchange Models).*** Esta parte desempeña el mismo papel que el prENV 13606-4, contiene un conjunto de modelos que se construyen sobre las partes anteriores de la norma y puede formar el soporte de comunicaciones basadas en mensajes o en servicios.

En concreto, en prEN 13606-1 se adopta el Modelo Dual para el diseño de la arquitectura de información para comunicación con la HCE. Su objetivo era concebir un modelo escalable para representar cualquier entrada posible de registros de datos sanitarios. Los motivos para la adopción de este modelo son: en primer lugar la gran cantidad de conceptos distintos existente en el dominio clínico. Segundo la continua variabilidad de dichos conceptos y por último la dificultad de representar estos datos por medios informáticos. Este modelo se compone de un Modelo de Referencia y un Modelo de Arquetipos.

Este modelo se corresponde conceptualmente con otros modelos como el de GEHR, el de prENV 13601-1 o el de openEHR. Se utiliza para representar las características genéricas de la información de la historia clínica, cómo se agregan, y la información de contexto requerida para cumplir con los requisitos éticos legales y de origen. De forma que define un conjunto de clases que forman los bloques constitutivos genéricos de la HCE y refleja las características que no cambian de una HCE. Este modelo de información necesita complementarse en el dominio del conocimiento por un método formal de comunicarse y compartir las estructuras jerárquicas dentro de las

HCE, los tipos de datos y los rangos de valores que pueden tomar los registros y otros condicionantes. Esto es así para asegurar la interoperabilidad, la consistencia y la calidad de los datos. Su representación se realiza mediante diagramas *Unified Modelling Language* (UML).

Este Modelo de Referencia asume que la información de una HC es inherentemente jerárquica. Por lo tanto, tiene que reflejar esta estructura y organización jerárquica, cumpliendo los requisitos para ser fiel al contexto clínico original y asegurar que se preserva el significado cuando los registros se comunican entre sistemas clínicos heterogéneos. Los requisitos contextuales de la HCE son clave para conseguir tal fidelidad y están relacionados con un conjunto de clases de bloques de construcción lógica con atributos apropiados propuestos para cada nivel en la jerarquía del Extracto de HCE.

Según un Modelo de Arquetipo, los arquetipos son metadatos usados para representar las características específicas de las diferentes clases de datos clínicos, que potencialmente se necesitaran representar para cumplir con los requisitos de las diferentes profesiones, especialidades o servicios. Cada caso de arquetipo tiene que ajustarse a un Modelo de Arquetipo, que formalmente está relacionado con el modelo de referencia. Así pues, un arquetipo proporciona:

- Una estructura para la información de la HCE basada en elementos del modelo de referencia.
- Una descripción semántica de alto nivel de los conceptos clínicos que puede ser procesada automáticamente por los sistemas de información sanitarios. Los arquetipos se usan por los sistemas para la creación y validación de datos, para la construcción de interfaces de usuario, para la consulta de datos y para compartir de conocimiento. El control de versión de revisiones de arquetipo asegurará que nuevas revisiones no invalidan datos creados con revisiones anteriores. La diversidad de información sanitaria almacenada en papel, que existe en cada organización sanitaria, puede dar lugar a un amplio rango de posibles arquetipos que podrían necesitarse dentro de una comunidad con HCE compartida. Además los arquetipos deberían ser creados por especialistas del dominio médico.

Tras superar las fases de análisis y periodos de aprobación correspondientes, cuatro de las cinco partes del prEN 13606 se han adoptado como estándar EN 13606-x (siendo x el número correspondiente a cada parte). Las partes 1, 2 y 4 se aprobaron

como estándares en el 2007, mientras que la parte 3 se ha aprobado en el 2008. Además la parte 5, prEN ISO 13606-5 *Interface specification* (ISO/DIS 13606-5:2008) se ha modificado y está en periodo de aprobación hasta marzo del 2010.

Es importante resaltar que el estándar EN 13606 está especialmente diseñado para la comunicación de la HCE. Permite una gran flexibilidad en la representación de las estructuras de información que se transmiten gracias al uso de arquetipos, que además sirven para describir semánticamente dicha información. Esto permite un procesamiento semántico de la misma y una reducción en los errores de comprensión de dicha información. EN 13606 es robusto frente a los cambios de las especificaciones, ya que los cambios en la definición de los arquetipos no requieren implementar de nuevo los sistemas subyacentes.

4.9. Common Object Request Broker Architecture (CORBA)

El *Object Management Group* (OMG) es el mayor consorcio de *software* que existe en el mundo y está formado por más de 800 compañías. Es una organización sin ánimo de lucro. Uno de los mayores logros de esta organización ha sido el desarrollo del estándar CORBA, uno de los estándares que ha desarrollado para la integración de sistemas por medio de la tecnología orientada a objetos (Monteagudo 2003).

CORBA define la infraestructura para la arquitectura *Object Management Architecture* (OMA) de OMG, especificando los estándares necesarios para la invocación de métodos sobre objetos en entornos distribuidos y heterogéneos. De manera intuitiva podemos ver a un objeto como una "pieza de *software*" que proporciona uno o varios servicios y a un método como uno de los servicios que ofrece el objeto. Grimson *et ál.* (1998) realizan una integración de un sistema de telemedicina sobre HCE basado en CORBA.

Pero lo que nos interesa es el CORBAmed, que es una adaptación del estándar CORBA al ámbito de la salud. Blobel *et ál.* (1998), describen las posibilidades de CORBA y las actividades de CORBAmed. CORBAmed es una división de CORBA encargada de definir una serie de servicios o *middleware* básico en el dominio de la sanidad. La misión de este grupo de trabajo se resume en estos dos puntos:

- Mejorar la calidad de la atención sanitaria y reducción de costes por medio del uso de tecnologías CORBA.
- Definición de interfaces estandarizadas orientadas a objetos entre servicios y funciones sanitarias.

Para el ámbito sanitario dónde habitualmente conviven un sinfín de sistemas distribuidos y heterogéneos CORBAmed intenta dar una especificación de los servicios necesarios para que estos sean interoperables entre sí. Siete de los servicios más importantes son:

- Servicio de Identificación de paciente (PIDS). Permite la asociación unívoca de diferentes agregados de información clínica a un único paciente. Se mantiene un índice maestro de paciente encargado de establecer las diferentes correlaciones de Identificación de pacientes involucrados en los diferentes sistemas de información. Esta es una tarea muy importante y crítica siendo la principal barrera en el desarrollo de entornos fiables para la consecución de la HCE.
- Servicio de terminología léxico-médica. Usado para:
    - Adquisición de la información: asistencia en el proceso de introducción de información codificada.
    - Mediación: transformación de mensajes o elementos de información desde una forma de representación a otra.
    - Indexación e inferencia: averiguar si algunas asociaciones pueden existir entre varios elementos de información.
    - Manipulación de conceptos compuestos: ayudar en la entrada, validación, traducción y simplificación de conceptos compuestos.
- Servicio de acceso a información multimedia (CIAS). Usado para extraer y gestionar recursos multimedia entre ellos por ejemplo imágenes médicas. Permite la recuperación y extracción entre sistemas heterogéneos que quieran interoperar entre sí.
- Servicio de decisión de acceso a los recursos. Usado para obtener autorizaciones, permisos de administración y acceso sobre la información médica.
- Servicio acceso a observaciones clínicas (COAS). Implementa la interfaz estandarizada de acceso público a la información clínica de una federación de sistemas heterogéneos. Requiere la implementación de pasarelas estandarizadas

para cada sistema de información clínico conectado, para importar, exportar, propagar, indexar los registros clínicos del paciente.

- Servicio de gestión de la información abreviada. Usado para compilar y gestionar resúmenes médicos para poder transmitirlos entre sistemas dispares. Este servicio es opcional.
- Servicio para facilitar la interpretación de los datos. Usado para la ayuda en la toma de decisiones. A partir de la información de salud extraída de las diferentes fuentes.

4.10. Good European Health Record (GEHR)

Dentro del programa de investigación europeo *Telematics* surgió GEHR para tener una arquitectura multimedia de los datos de los pacientes en formato electrónico. Este proyecto de informática avanzada en medicina ha desarrollado una arquitectura multimedia para poder usar y compartir: los expedientes electrónicos del *healthcare*, los requisitos clínicos, técnicos, educativos y ético-legales. El proyecto de GEHR implicó 21 organizaciones que participaban en siete países europeos junto con sus departamentos clínicos en el que se incluía diversas profesiones y disciplinas relacionadas con la medicina. Además este estándar ha desarrollado dos definiciones formales en apoyo su arquitectura: el modelo del objeto de GEHR y el formato del intercambio de GEHR. Para apoyar el desarrollo de los sistemas del expediente electrónico el proyecto ha producido un sistema de 2.000 nombres de HRI disponibles en 9 idiomas europeos y un sistema comprensivo de 47 dibujos anatómicos. Pruna *et ál.* 1998, emplean GEHR en un proyecto de investigación de HCE sobre la diabetes.

4.10.1. Los requisitos del paciente

La arquitectura de GEHR ha sido desarrollada gracias a una extensa investigación de las exigencias de los pacientes. Los médicos, las enfermeras y otras profesiones relacionadas con la medicina estuvieron implicados en derivar un sistema de requisitos en varias áreas dominantes. El GEHR resume que el expediente electrónico debe de ser legible y portable, sirviendo de apoyo para el cuidado clínico. Los expertos valoran esta oportunidad de creación de expediente electrónico, no obstante incluso con el uso de términos, de clasificaciones y de idiomas estándares, no puede ser asumido que otros individuos que vean el expediente electrónico tengan la las mismas valoraciones y que los significados no serán perdidos ni serán cambiados.

4.10.2. Descripción de la arquitectura de GEHR

Toda la información en un HCE está implícitamente relacionada con el paciente. La arquitectura de GEHR posee una estructura original de los datos y se ve como las entradas en el expediente se agrupan. Se ha hecho un gran esfuerzo en proponer una arquitectura genérica, flexible y lo más fiable posible. Las características de la arquitectura del GEHR son los siguientes:

- El HCE proporciona un envase particular para cada paciente particular .
- La transacción proporciona la mayor parte de las características necesitadas para los aspectos legales de la medicina. Gracias a esto los datos se pueden transferir con seguridad entre los diferentes sistemas de HCE.
- Es posible la agregación de otros formatos de HCE y de otras colecciones de HCE.

4.11. Integrating the Healthcare Enterprise (IHE)

Integrating the Healthcare Enterprise (IHE) es una iniciativa de profesionales de la sanidad (incluyendo colegios profesionales de médicos) y empresas proveedoras cuyo objetivo es mejorar la comunicación entre los sistemas de información que se utilizan en la atención al paciente. IHE define unos Perfiles de Integración que utilizan estándares ya existentes para la integración de sistemas de manera que proporcionen una interoperabilidad efectiva y un flujo de trabajo eficiente. IHE nos permite alcanzar el nivel de integración exigible en la era de la HCE.

Cada perfil de integración IHE describe una necesidad clínica de integración de sistemas y la solución para llevarla a cabo. Define también los componentes funcionales, a los que llamaremos Actores IHE, y especifica con el mayor grado de detalle posible las transacciones que cada Actor deberá llevar a cabo, basadas siempre en estándares como el de DICOM y HL7 (IHE 2008).

Los beneficios de IHE son:

- Hace que el uso de las tecnologías de la información avanzadas ayude en gran medida al personal sanitario a la hora de mejorar la calidad y eficiencia de la atención sanitaria.
- Aumenta la seguridad del paciente al garantizar la integridad de la información médica.

- Reduce el tiempo empleado en la solución de problemas tales como la pérdida de datos y la aparición de estudios no correspondientes, optimizando así el aprovechamiento de tiempo del personal.
- Proporciona al personal sanitario información bien estructurada sobre el paciente de modo que la toma de decisiones médicas se base en la mejor información posible.
- Define los problemas de integración más acuciantes
- Define soluciones técnicas
- Prueba soluciones de productos entre vendedores
- Promueve demostraciones de integración multivendedor
- Encauza la traducción de estas soluciones a productos
- Formación a proveedores del Sector Sanitario

En su primer año, la iniciativa de IHE se ha centrado en los *Hospital Information Systems* (HIS), *Radiology Information Systems* (RIS), Modalidades de Imagen y en PACS aplicando tanto el HL7 como el DICOM según lo resumido en la Tabla 4:

	HIS	RIS	PACS	Modalidades
HIS	HL7	HL7	-	-
RIS	HL7	HL7	HL7/DICOM	DICOM
PACS	-	HL7/DICOM	DICOM	DICOM
Modalidades	-	DICOM	DICOM	DICOM

Tabla 4. Modalidades de Imagen y en PACS. Fuente: propia.

En los siguientes años la iniciativa de IHE tomará en consideración otros estándares. Siemens siempre ha jugado un papel activo en desarrollar tecnologías y estándares, teniendo representación en más de 20 grupos de trabajo y comités en la mayoría de las organizaciones de estándares e iniciativas (DICOM, HL7, CORBAmed, IHE) y presidentes que lideraban los grupos de trabajo en áreas dominantes.

4.12. OpenEHR

OpenEHR es una fundación que realiza investigación abierta, desarrollo e implementación de registros electrónicos de salud. Sus resultados están basados en una combinación de 15 años de investigación en HCE, en los que se incluyen: plantillas, modelos de información y servicio, volumen demográfico y trabajo clínico.

La fundación OpenEHR fue creada para permitir el desarrollo de especificaciones, del *software* y de recursos abiertos para los sistemas de información de la salud, particularmente HCE. Publica todas sus especificaciones y las pone en práctica (como *software* abierto). También desarrolla "arquetipos" y una terminología para el uso con HCE (OpenEHR 2008).

Los objetivos de esta fundación son:

- Promover y publicar los requisitos para representar y comunicar la salud electrónica basada en la experiencia de la implementación y evolucionando con el tiempo como asistencia médica y conocimiento médico.
- Promover y publicar las arquitecturas de información de HCE, diccionarios de modelos y datos ya probados en aplicaciones.
- Evaluar la validación de las arquitecturas de HCE.
- Mantener las aplicaciones abiertas bajo la licencia.
- Colaborar con otros grupos y sistemas de información de salud, logrando interoperar en los campos relacionados con la informática de la salud.

Los sistemas de información de la salud pueden ser construidos definiendo modelos de la información, poniéndolos en ejecución en BD y usos y agregando terminologías clínicas. Sin embargo, construir un ambiente que contenga lo dicho anteriormente para el cuidado compartido de por vida requiere algo más. En primer lugar, necesitan ser los requisitos, sin los cuales el ningún modelar apropiado no puede ser hecho. Los requisitos en los cuales se basa el openEHR incluyen:

- Los requisitos del proyecto de GEHR.
- Requisitos técnicos de HCE escritos para el proyecto australiano de GEHR;
- Los requisitos se convirtieron sobre los años desde GEHR en Europa, documentada en Tesis doctoral del Dr Dipak Kalra.
- Los requisitos de la ISO 18308 para una salud electrónica registran arquitectura.

Las mejoras que ofrece esta fundación son las siguientes:

- Se reduce el mantenimiento del *software*, puesto que el *software* no tiene que ser cambiado cada vez el modelo de algunos cambios clínicos de los datos - solamente el cambio de los arquetipos.
- Aumenta la validez de los datos, porque los arquetipos se utilizan para validar toda la entrada de datos al HCE. Desde un punto de vista clínico, los datos son más dignos de confianza.
- Aumenta la interoperabilidad, gracias a esto se mejora el intercambio de la información entre clínicos dentro de un hospital, en una red del cuidado de la comunidad y en distancias más grandes.
- Estándar, el modelo de la referencia del openEHR se basa en estándares de la ISO y de CEN EHR, y es interoperable con los estándares del mensaje de HL7 y de EDIFACT. Esto permite a *software* openEHR ser integrada con otro *software* y sistemas.
- Integración con los sistemas, este estándar se puede integrar con los muchos sistemas que utilizan estándares bien conocidos de la interoperabilidad de los datos.

4.13. Estándares de Seguridad en HCE

La seguridad en un sistema de información debe contemplar todas las posibles amenazas que se identifiquen sobre todos los elementos del sistema de información: máquinas, programas, datos, redes y electrónica de red. Entre las amenazas se encuentran las personas, tanto con carácter voluntario como involuntario, y las catástrofes, como los incendios e inundaciones (Garbayo *et ál.* 2003).

A continuación, en la Tabla 5 se muestran los objetivos de seguridad y las medidas o mecanismos de seguridad que existen para garantizar su cumplimiento. Los dos mecanismos básicos de seguridad son las claves públicas y privadas, y los algoritmos de resumen de una dirección. Estos son los fundamentos para la construcción del resto de mecanismos de seguridad.

OBJETIVO	DESCRIPCIÓN	MEDIDAS
Identificación	Es el proceso de identificar al cliente de la aplicación o servicio.	Certificados digitales.
Confidencialidad	Consiste en asegurar que a la información sólo accede quien está autorizado para ello.	Cifrado, encriptación
Integridad	Conjunto de acciones que garantizan que la información no se ha transformado durante su procesado, transporte o almacenamiento	Firma digital.
No repudio.	Procedimiento para asegurar que ninguna de las partes implicadas ya identificadas pueda negar haber participado en alguna transacción.	Firma digital, auditoría.
Autorización	Determinar a qué información puede acceder y qué tareas puede acometer, un cliente autentificado. Este proceso determina los privilegios asociados a un perfil de usuario.	Cuestión organizativa que debe diseñar cada organización y llevar a cabo en sus sistemas particulares.
Auditoría	Es la posibilidad de poder rastrear los accesos realizados a la información y a las operaciones hechas sobre ella por cada usuario y las circunstancias en las que la hizo.	Registros de acceso y operaciones efectuadas sobre la información.
Disponibilidad	Forma parte de la seguridad poder disponer de la información cuando se necesite. Por lo que se deben proteger los sistemas de forma que se mantengan en funcionamiento y se pueda acceder a la información en cualquier momento.	Operación y nivel de servicios adecuados sobre los sistemas.

Tabla 5. Objetivos y medidas de seguridad (Garbayo *et ál.* 2003).

Mediante la combinación de todos ellos se consigue proteger los sistemas de información mediante el cifrado o encriptación, la firma y los certificados digitales. Los mecanismos básicos y técnicos se complementan con los de organización de autorización y auditoria, así como con los de operación y de nivel de servicio.

CEN ENV 13608-1:2000. ***Security for healthcare communication.***

Se tiene que establecer un marco general de seguridad, políticas de seguridad y procedimientos. Para ello hay que definir cuestiones como el acceso y las herramientas para llevarlo acabo. Las normas sobre seguridad y protección de datos incluyen despliegue de sistemas de certificados, claves públicas y firma electrónica. Por tanto se define unas normas de seguridad para la salud en las que se definen normas sobre integridad de datos y back-up y la seguridad en intercambio de mensajes.

Por ello se define la norma CEN TC251, la cual tiene por características:

- Identificación segura de usuario por contraseñas para la gestión de Asistencia Sanitaria y Seguridad de Autenticación (ENV 12251).
- También se define una autenticación fuerte utilizando tarjetas con microprocesador para la identificación segura de usuario (ENV 12251).
- Se define un algoritmo para los servicios de firma digital en sanidad ENV 12388.

Existen otra serie de organismos que han definido normas para la seguridad en la salud, en este caso ISO ha definido el uso de la infraestructura PKI (ISO/CD 17090). Además existen grupos de trabajo en seguridad como es ACR NEMA/DICOM. También el ASTM3 que se divide en varios subcomités y que define normas para:

- Privacidad, confidencialidad y Acceso.
- Seguridad de Datos y sistemas de Información Sanitaria.
- Trascripción Médica y Documentación.

5

Bases de datos

5.1. INTRODUCCIÓN 130
5.2. CONCEPTOS BÁSICOS 131
5.3. ¿POR QUÉ USAR BASES DE DATOS? 135
5.4. COMPONENTES PRINCIPALES DE UNA BASE DE DATOS 136
5.5. VENTAJAS EN EL USO DE BASES DE DATOS Y SGBD 138
5.6. TIPOS DE BASES DE DATOS 140
5.6.1. SEGÚN LA VARIABILIDAD DE LOS DATOS ALMACENADOS 140
5.6.2. SEGÚN EL CONTENIDO 140
5.7. MODELOS DE BASES DE DATOS 141
5.7.1. BASES DE DATOS JERÁRQUICAS 141
5.7.2. BASES DE DATOS EN RED 142
5.7.3. BASES DE DATOS RELACIONALES 142
5.7.4. BASES DE DATOS ORIENTADAS A OBJETOS 145
5.8. BASES DE DATOS EN TELEMEDICINA 146
5.9. BASES DE DATOS DE LA APLICACIÓN TELEOFTALWEB 147
5.9.1. MYSQL 148
5.9.2. BASES DE DATOS NATIVAS XML 152
5.9.3. ORACLE 162

5.1. Introducción

En el presente capítulo hablaremos de conceptos fundamentales para entender las BDs, tipos, modelos, BDs empleadas en telemedicina y las BDs seleccionadas para el desarrollo de nuestra aplicación, **TeleOftalWeb.** Antes de que existieran las BDs se utilizaban archivos secuenciales como almacenes de datos. Más tarde aparecieron los archivos indexados, dónde el acceso ya podía ser aleatorio. El sistema de archivos era el sistema más común de almacenamiento de datos. Para compartir los datos entre varias máquinas surgió el *Network File System* (NFS), sistema de archivos distribuido para un entorno de red de área local, y más tarde para evitar fallos en los sistemas de archivos aparecieron los sistemas *Redundant Array of Independent/Inexpensive Disks* (RAID), conjunto de discos redundantes independientes. Pero los programas y datos cada vez eran más complejos y grandes. Se requería de un almacenamiento que garantizara un cierto número de condiciones y que permitiera operaciones complejas sin que se violaran estas restricciones. Además cada usuario que accediera a los datos debía tener su trabajo protegido de las operaciones que hicieran el resto de usuarios (Silberschatz *et ál.* 2006).

Respondiendo a estas necesidades, surgieron las BDs jerárquicas donde los datos se situaban siguiendo una jerarquía. Sin embargo, las BDs jerárquicas tenían el problema de que los accesos a los datos eran unidireccionales, y era más complicado hacer el camino inverso. Por ejemplo, era fácil saber que cuentas tenía un cliente, pero no tanto saber de qué cliente era una cierta cuenta (Elmasri & Navathe 2007).

Para dar absoluta libertad a las relaciones entre tablas surgieron las BDs relacionales, *Relational DataBase Management System* (RDBMS), que trajeron dos cosas muy importantes: las propiedades atomicidad, consistencia, aislamiento y durabilidad (ACID) y un lenguaje común de acceso a los datos: *Structured Query Language* (SQL). Las propiedades de ACID son (Silberschatz *et ál.* 2006):

- **Atomicidad.** Cada transacción del usuario debe tratarse de forma atómica. De esta forma, o se ejecuta todo o nada. En todo sistema la información es muy importante y no es posible realizar una transacción a medias. Una transacción se ejecuta exactamente una vez y tiene carácter "atómico" (de subdivisión), es

decir, el trabajo se realiza en su totalidad o no se realiza en ningún caso.

- **Consistencia.** Las transacciones han de cumplir las restricciones definidas dentro de la BD. Si no las puede cumplir, se evita su ejecución. De esta forma se conserva la integridad y coherencia de los datos.

- **Aislamiento.** Una transacción es una unidad de aislamiento, permitiendo que transacciones concurrentes se comporten como si cada una fuera una única transacción que se ejecuta en el sistema. Las transacciones alcanzan el nivel más alto de aislamiento cuando se pueden serializar. En este nivel, los resultados obtenidos de un conjunto de transacciones concurrentes son idénticos a los obtenidos mediante la ejecución en serie de las transacciones.

- **Durabilidad.** Una vez se ha completado la transacción, los resultados de la misma han de ser permanentes y sobrevivir a posibles caídas del sistema o la BD.

Debido a que las RDBMS tienen que soportar todas estas propiedades, nunca serán tan rápidas como trabajar directamente sobre archivos, aunque internamente trabajen sobre ellos. La mayoría de desarrolladores prefieren hoy en día sacrificar la velocidad por las funcionalidades (Burbano 2006).

5.2. Conceptos básicos

Un dato es un conjunto de caracteres con algún significado, pueden ser numéricos, alfabéticos, o alfanuméricos.

La información en BD es un conjunto ordenado e integrado de datos los cuales son manejados en función de la necesidad del usuario. La integración evita inconsistencias y aumenta el control de la organización.

Así pues, una BD es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una BD compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y debido al desarrollo

tecnológico de campos como la informática y la electrónica, la mayoría de las BDs tienen formato electrónico, que ofrece un amplio rango de soluciones al problema de almacenar datos (Silberschatz *et ál.* 2006).

Otra definición propuesta por (Korth *et ál.* 2002) es la siguiente: una BD es una serie de datos organizados y relacionados entre sí, y un conjunto de programas que permitan a los usuarios acceder y modificar esos datos.

Un sistema gestor de BD (SGBD) es una colección de numerosas rutinas de *software* interrelacionadas que proporciona un sistema eficiente para extraer, almacenar y manipular información de la BD. Todas las peticiones de acceso a la base, se gestionan centralizadamente por medio del SGBD. Por lo tanto, el SGDB es un interfaz entre el usuario y la BD (Elmasri & Navathe 2007).

Uno de los propósitos principales de un sistema de BD es proporcionar a los usuarios una visión abstracta de los datos. Es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos. Un SGBD posee una arquitectura basada en tres niveles (Cabrera 2001):

- **Nivel físico.** Es el nivel más bajo de abstracción y el nivel real de los datos almacenados. Este nivel define cómo se almacenan los datos en el soporte físico, ya sea en registros o de cualquier otra forma, así como los métodos de acceso. Este nivel lleva asociada una representación de los datos, que es lo que denominamos Esquema Físico.
- **Nivel conceptual.** Es el correspondiente a una visión de la BD desde el punto de visto del mundo real. Es decir se trata con la entidad u objeto representado, sin importar como está representado o almacenado éste. Es la representación de los datos realizada por la organización, que recoge los datos parciales de los requerimientos de los diferentes usuarios y aplicaciones parciales. Incluye la definición de los datos y las relaciones entre ellos. Este nivel lleva asociado un Esquema Conceptual.
- **Nivel de visión.** Son partes del esquema conceptual. El nivel conceptual presenta toda la BD, mientras que los usuarios, por lo general, sólo tienen acceso a pequeñas parcelas de ésta. El nivel visión es el encargado de dividir estas parcelas. Un ejemplo sería el caso del empleado de una organización que tiene

acceso a la visión de su nómina, pero no a la de sus compañeros. El esquema asociado a éste nivel es el Esquema de Visión.

A estos niveles también se les llama nivel interno, nivel conceptual y nivel externo como se puede observar en la figura 31.

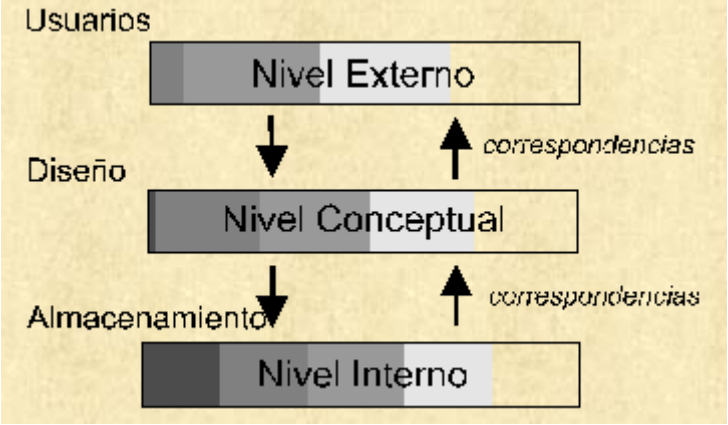


Figura 31. Niveles de la arquitectura de un SGBD (Cabrera 2001).

Los SGBD se componen de un lenguaje de definición de datos (*Data Definition Language:* DDL) y de un lenguaje de manipulación de datos (*Data Manipulation Language:* DML). Estos dos lenguajes son englobados por el lenguaje de gestión de BD (Elmasri & Navathe 2007).

El lenguaje SQL es el lenguaje de gestión de BD más conocido en la actualidad. Es un lenguaje estándar internacional, comúnmente aceptado por los fabricantes de generadores de BDs relacionales. Existen numerosos productos que soportan SQL, cada uno de ellos con pequeñas diferencias sin apenas importancia. El SQL estándar es el publicado por *American National Standards Institute* (ANSI) e ISO.

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las BDs. Trabaja con estructura cliente/servidor sobre una red de ordenadores. El ordenador cliente es el que inicia la consulta; el ordenador servidor es que atiende esa consulta. El cliente utiliza toda su capacidad de proceso para trabajar; se limita a solicitar datos al ordenador servidor, sin depender para nada más del exterior. Estas peticiones y las respuestas son transferencias de textos que cada ordenador cliente

se encarga de sacar por pantalla, presentar en informes tabulados, imprimir, guardar, etc., dejando el servidor libre.

- DDL es un lenguaje proporcionado por el SGBD que permite a los usuarios de la misma llevar a cabo las tareas de definición de las estructuras que almacenarán los datos así como de los procedimientos o funciones que permitan consultarlos. El DDL permite al administrador de la BD especificar los elementos de datos que la integran, su estructura y las relaciones que existen entre ellos, las reglas de integridad y los controles a efectuar antes de autorizar el acceso a la base.
- DML es utilizado para escribir programas que crean, actualizan y extraen información de las BDs. Siempre de acuerdo con las especificaciones y las normas de seguridad dictadas por al administrador.

Un lenguaje de manipulación de datos es un lenguaje que permite a los usuarios acceder o manipular los datos organizados mediante el modelo de datos apropiado. Hay dos tipos básicamente:

- DMLs procedimentales. Requieren que el usuario especifique qué datos se necesitan y cómo obtener esos datos.
- DMLs declarativos (o no procedimentales). Requiere que el usuario especifique qué datos se necesitan sin especificar cómo obtener esos datos.

La secuencia conceptual de operaciones que ocurren para acceder a cierta información que contiene una BD es la siguiente:

- El usuario solicita cierta información contenida en la BD.
- El SGBD intercepta este requerimiento y lo interpreta.
- El SGBD realiza las operaciones necesarias para acceder y/o actualizar la información solicitada.

5.3. ¿Por qué usar bases de datos?

Los sistemas tradicionales se denominaban sistemas orientados hacia procesos, debido a que, en ellos, se ponía el énfasis en los tratamientos que reciben los datos, los cuales se almacenan en archivos que son diseñados para una determinada aplicación.

Este planteamiento produce además de una ocupación inútil de memoria, un aumento de los tiempos de procesos, al repetirse los mismos controles y operaciones en los distintos archivos. Pero más graves todavía son las inconsistencias que a menudo se presentan en estos sistemas, debido a que la actualización de los mismos datos, cuando estos se encuentran es más de un archivo, no se puede realizar de forma simultánea en todos ellos.

Con el fin de resolver estos problemas y de lograr una gestión más racional del conjunto de datos, surge un nuevo enfoque que se apoya sobre una BD en la cual los datos son recogidos y almacenados, al menos lógicamente, una sola vez, con independencia de los tratamientos. El diagrama de flujo de información al emplear una BD se puede observar en la figura 32.

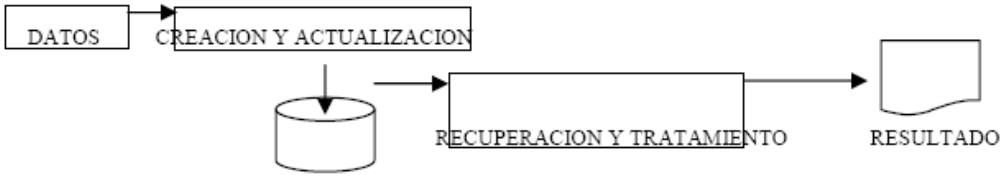


Figura 32. Diagrama de flujo de la información al utilizar una BD (Burbano 2006).

Según este enfoque se logran los siguientes cambios:

- Independencia de los datos respecto a los tratamientos y viceversa, lo que evita el importante esfuerzo que origina la reprogramación de las aplicaciones cuando se producen cambios en los datos.
- Coherencia de los resultados, con lo que se elimina el inconveniente de las divergencias en los resultados debidas a actualizaciones no simultáneas en todos los archivos.

- Mejor disponibilidad de los datos para el conjunto de los usuarios junto con una mayor transparencia respecto a la información existente.
- Mayor valor informativo, debido a que los distintos elementos están interrelacionados.
- Documentación de la información mejor y más normalizada, la cual está integrada con los datos.
- Mayor eficiencia en la recuperación, validación y entrada de los datos al sistema.

Además, en el momento de tomar una decisión hay que tener en cuenta posibles inconvenientes que es necesario valorar antes de tomar una decisión relativa a un cambio en la orientación del sistema de información:

- Instalación costosa.
- Personal especializado.
- Falla de rentabilidad a corto plazo.
- Desfase entre teoría y práctica.

5.4. Componentes principales de una base de datos

- *Hardware*. El *hardware* se refiere a los dispositivos de almacenamiento en donde reside la BD, así como a los dispositivos periféricos (unidad de control, canales de comunicación, etc.) necesarios para su uso.
- *Software*. Está constituido por un conjunto de programas que constituyen el SGBD. Este sistema maneja todas las solicitudes formuladas por los usuarios a la BD.
- Usuarios. Existen cuatro clases de usuarios relacionados con una BD.
- El programador de aplicaciones, quien crea programas de aplicación que utilizan la BD.
- El usuario final, quien accede a la BD por medio de un lenguaje de consulta o de programas de aplicación.

- El administrador de la BD (*DataBase Administrator,* DBA), quien realiza las siguientes acciones:
    - Se encarga del diseño físico de la BD y de su implementación.
    - Realiza el control de seguridad y de la concurrencia. En la Figura 33 se puede observar el diagrama de interacción del DBA con los usuarios.
    - Mantiene el sistema para que siempre se encuentre operativo y se encarga de que los usuarios y las aplicaciones obtengan buenas prestaciones.
    - Es responsable de copias y recuperación.
    - Configurar y monitorizar el sistema: parámetros, equilibrios de cargas, etc.
    - Debe conocer muy bien el SGBD que se esté utilizando, así como el equipo informático sobre el que esté funcionando.

Figura 33. Diagrama de interacción del DBA con los usuarios (Burbano 2006).

- El diseñador de la BD, quien se encarga de lo siguiente:
    - Identificar los datos, las relaciones entre los datos y las restricciones sobre los datos y sus relaciones.
    - Tener un profundo conocimiento de los datos de la empresa y también debe conocer sus "reglas de negocio".
    - Implicar a todos los usuarios de la BD para obtener un buen resultado.

5.5. Ventajas en el uso de bases de datos y SGBD

Según (Korth *et ál.* 2002) las principales ventajas de usar SGBD son las siguientes:

- Globalización de la información. Permite a los diferentes usuarios considerar la información como un recurso corporativo que carece de dueños específicos.
- Control sobre la redundancia de datos.
    - Desperdicio de espacio de almacenamiento.
    - Provoca la falta de consistencia de datos.
    - A veces la redundancia es necesaria para modelar las relaciones entre los datos, o bien es necesaria para mejorar las prestaciones.
- Eliminación de información inconsistente. Si el sistema está desarrollado a través de archivos convencionales, una cancelación de compra por ejemplo deberá operarse tanto en el archivo de facturas del sistema de control de cobranza como en el archivo de facturas del sistema de comisiones.
- Disminuir la dificultad para acceder a los datos. Más información sobre la misma cantidad de datos.
    - Ofrecer un entorno de datos que le facilite al usuario la gestión de los mismos.
    - Datos integrados. Más facilidad de extraer información
- Permite compartir información. Varios sistemas o usuarios pueden utilizar una misma entidad.
- Disminuir aislamiento de datos.
    - Los datos de diferentes aplicaciones integrados en la misma BD.
- Independencia de datos y tratamiento. La independencia de datos implica un divorcio entre programas y datos; es decir, se pueden hacer cambios a la información que contiene la BD o tener acceso a la BD de diferente manera, sin hacer cambios en las aplicaciones o en los programas. Lo que implica menor coste de mantenimiento.
- Mejora en la productividad y mantenimiento.
    - El SGBD proporciona muchas de las funciones estándar que el programador necesita escribir en un sistema de ficheros.

  - Se separan las descripciones de los datos de las aplicaciones.
- Disminuir anomalías en el acceso concurrente.
  - Múltiples usuarios.
- Disminuir problemas de seguridad.
  - Control de acceso a los datos, no todos los usuarios pueden visualizar alguna información.
  - Un sistema de BD confiable debe mantener un grado de seguridad que garantice la autentificación y protección de los datos.
- Disminuir problemas de integridad.
  - Los valores de datos almacenados en la BD deben satisfacer cierto tipo de restricciones de consistencia.
- Mejora en los servicios de copias de seguridad y de recuperación ante fallos.
  - Centralizado.

A pesar de las múltiples ventajas que ofrece un SGBD también posee ciertas desventajas:

- Complejidad. Hay que comprender la funcionalidad para poder sacar el mejor provecho del sistema.
- Tamaño. Los SGBD son programas complejos y muy extensos que requieren una gran cantidad de espacio en disco y de memoria para trabajar de forma eficiente.
- Coste económico del SGBD. El coste de un SGBD varía dependiendo del entorno y de la funcionalidad que ofrece.
- Coste del equipamiento adicional. Tanto el SGBD, como la propia BD, pueden hacer que sea necesario adquirir más espacio de almacenamiento.
- Vulnerable a los fallos debido a la centralización.

5.6. Tipos de bases de datos

Las BDs pueden clasificarse de varias maneras, de acuerdo al criterio elegido para su clasificación.

5.6.1. Según la variabilidad de los datos almacenados

- BDs estáticas. Éstas son BDs de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.
- BDs dinámicas. Éstas son BDs donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de este tipo puede ser la BD utilizada en un sistema de información de una tienda de informática, una farmacia, un videoclub, etc.

5.6.2. Según el contenido

- BDs bibliográficas. Sólo contienen un representante de la fuente primaria, que permite localizarla. Un registro típico de una BD bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque sino estaríamos en presencia de una BD a texto completo (o de fuentes primarias). Como su nombre lo indica, el contenido son cifras o números. Por ejemplo, una colección de resultados de análisis de laboratorio, entre otras.
- BDs de texto completo. Almacenan las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas.
- Directorios. Un ejemplo son las guías telefónicas en formato electrónico.
- Banco de imágenes, audio, video, multimedia, etc.
- BDs o "bibliotecas" de información biológica. Son BDs que almacenan diferentes tipos de información proveniente de las ciencias de la vida o médicas. Se pueden considerar en varios subtipos:
    - Aquellas que almacenan secuencias de nucleótidos o proteínas.
    - Las BDs de rutas metabólicas.

- BDs de estructura, comprende los registros de datos experimentales sobre estructuras 3D de biomoléculas.
- BDs clínicas (Ejemplo: **TeleOftalWeb**).
- BDs bibliográficas (biológicas).

5.7. Modelos de bases de datos

Este apartado sería otra clasificación de las BDs, pero esta clasificación es la más importante y conocida por lo que merece un apartado aparte. Las clasificaciones anteriores atendían en general al tipo de función que iban a desempeñar las BDs pero, esta se refiere al modelo de administración de datos. Existen tres modelos tradicionales: modelo jerárquico, modelo en red y modelo relacional. Hay otro modelo que se puede considerar como el paso anterior al modelo relacional: modelo Entidad-Relación. Asimismo, hace relativamente poco surgió otro modelo que, debido a las mejoras que introducía, se ha extendido mucho; este es el modelo orientado a objetos. En este apartado se hablará brevemente de todos ellos.

5.7.1. Bases de datos jerárquicas

Éstas son BDs que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas. En la figura 34, se puede observar un ejemplo de BD de tipo jerárquico.

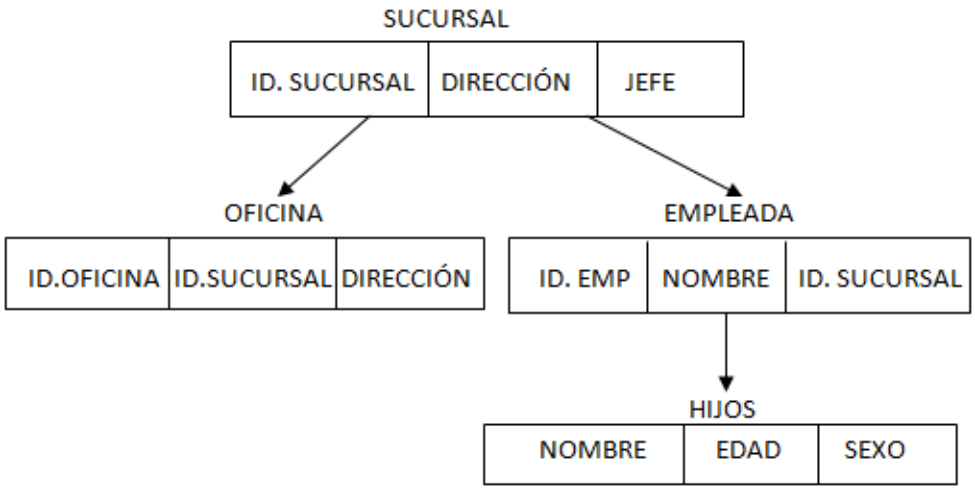


Figura 34. Ejemplo de estructura jerárquica de una BD. Fuente: propia.

Las BDs jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

5.7.2. Bases de datos en red

Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).

Los conceptos básicos en el modelo en red son:

- El tipo de registro, que representa un nodo.
- Elemento, que es un campo de datos.
- Agregado de datos, que define un conjunto de datos con nombre.

Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una BD en red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

5.7.3. Bases de datos relacionales

Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postulados sus fundamentos en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de BD. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Pese a que ésta es la teoría de las BDs relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es, pensando en cada relación como si fuese una tabla que está compuesta por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla). En la figura 35, se puede observar un ejemplo de modelo relacional de un Sistema de Información Hospitalario.

Relación	Entidades implicadas	Cardinalidad	Atributos
Cita	Paciente – Médico	N:M	Fecha, Hora
Pide	Médico – Análisis	1:N	
Anotación	Médico - Historial	N:M	Fecha
Consulta síntomas	Médico - Enfermedad	N:M	
Está indicado para	Medicamento - Enfermedad	N:M	Efecto
Consulta información	Médico - Medicamento	N:M	
Genera	Análisis - Resultado	1:1	
Análisis	Analista – Paciente	N:M	Fecha, procedimiento, muestra
Trabaja en	Analista - Laboratorio	1:N	
Está asociado a	Laboratorio - Hospital	N:M	Máximo análisis/día, remuneración
Recoge	Historial – Enfermedad	N:M	Gravedad, fecha
Anotación	Historial - Medicamento	N:M	Posología

Figura 35. Ejemplo de modelo relacional de un Sistema de Información Hospitalario. Fuente: propia.

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la BD. La información puede ser recuperada o almacenada mediante consultas que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a BD es SQL, que ya se habló de él cuando se vio los conceptos básicos y, como se dijo, es un estándar implementado por los principales motores o sistemas de gestión de BDs relacionales.

Durante su diseño, una BD relacional pasa por un proceso al que se le conoce como normalización de una BD, la cual consiste en convertir una relación en varias sub-relaciones, cada una de las cuales obedece a reglas que se describen en términos de dependencia. Las BDs relacionales se normalizan para:

- Evitar la redundancia de los datos.
- Evitar problemas de actualización de los datos en las tablas.
- Proteger la integridad de los datos.

5.7.3.1. El modelo Entidad-Relación

Este modelo está basado en una percepción del mundo real, que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre estos objetos.

Una entidad es todo aquello que exhibe autonomía, diferenciación y existencia en el mundo real que es distinguible de otros objetos. Por ejemplo cada persona es una

entidad y las cuentas bancarias pueden ser consideradas entidades. Las entidades se describen en la BD mediante atributos. Por ejemplo los atributos *nombre_cliente*, *dirección_cliente* pueden describir una entidad *cliente*. Adicionalmente *id_cliente*, se usa para identificar unívocamente a los clientes (dado que puede existir en el mundo real un cliente con el mismo nombre o dirección), de esta manera de asigna un identificador único para cada cliente.

Una relación es una asociación entre varias entidades, Por ejemplo una relación puede asociar un cliente con cada cuenta que tiene.

El conjunto de todas las entidades del mismo tipo, y el conjunto de todas las relaciones del mismo tipo, se denominan respectivamente conjunto de entidades y conjunto de relaciones.

La estructura lógica general de una BD se puede expresar gráficamente mediante un diagrama Entidad-Relación (E-R), que consta de los siguientes componentes:

- Rectángulos, que representan conjuntos de entidades.
- Elipses, que representan atributos.
- Rombos, que representan relaciones entre conjuntos de entidades.
- Líneas, que unen los atributos con los conjuntos de entidades y los conjuntos de entidades con las relaciones.

Cada componente se etiqueta con la entidad o relación que representa como se puede ver en la figura 36.

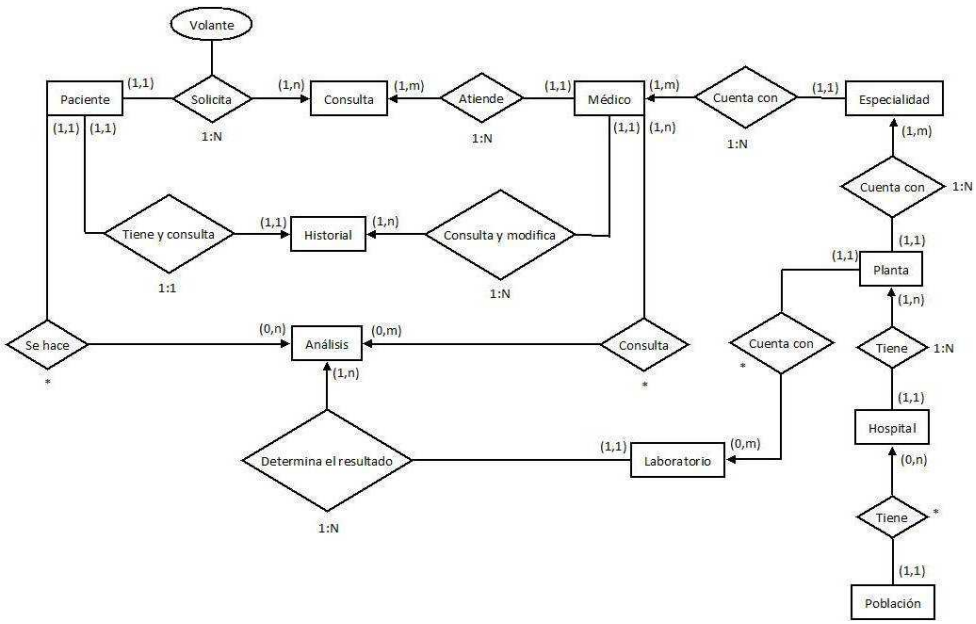


Figura 36. Ejemplo de diagrama E-R. Fuente: propia.

Un diagrama E-R es una representación conceptual de la información. Mediante una serie de procedimientos se puede pasar del modelo E-R a otros, como por ejemplo el modelo relacional.

5.7.4. Bases de datos orientadas a objetos

Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos (típicos en programación), trata de almacenar en la BD los objetos completos (estado y comportamiento).

Las limitaciones impuestas por el modelo relacional han surgido como obstáculos y, en consecuencia, los investigadores de BD inventaron nuevos modelos de datos que resuelven estas limitaciones del modelo de datos relacional.

Las aplicaciones de BDs tradicionales consisten en tareas de procesamiento de datos, tales como la banca, seguros, ventas, gestión de recursos humanos. Dichas aplicaciones presentan conceptualmente tipos de datos simples. Los elementos de datos básicos son registros bastante pequeños y cuyos campos son atómicos, es decir, no contienen estructuras adicionales. En los últimos años, la demanda ha incrementado la forma de abordar los tipos de datos más complejos. Por ejemplo, un conjunto de direcciones. Mientras una dirección completa puede ser vista como un elemento de datos atómico de tipo cadena de caracteres, esta forma de verlo escondería detalles como la calle, barrio, sector, provincia que pueden ser interesantes para una consulta. Por otra parte si una dirección se la presenta dividiéndole en varias componentes (calle, barrio, sector, provincia) las consultas escritas serían más complicadas, pues tendría que mencionar cada campo. Una alternativa mejor es permitir tipos de datos estructurados, que admiten un tipo dirección con sub-partes calle, barrio, sector, provincia.

Las BDs orientadas a objetos (BDOO) almacenan y manipulan información que puede ser digitalizada por objetos y proporcionan una estructura flexible con acceso ágil, rápido, con gran capacidad de modificación.

Además combinan las mejores cualidades de los archivos planos, las bases jerárquicas y las relacionales. Las BDOO representan el siguiente paso en la evolución de las BDs para soportar análisis, diseño y programación orientada a objetos. Permiten el desarrollo y mantenimiento de aplicaciones complejas ya que se puede utilizar un mismo modelo conceptual y así aplicado al análisis, diseño y programación, se reduce el

problema entre los diferentes modelos a través de todo el ciclo de vida, con un costo significativamente menor.

Las BDOO ofrecen un mejor rendimiento de la máquina que las BDs relacionales, para aplicaciones o clases con estructuras complejas de datos. Sin embargo, las BDOO coexistirán con las BDs relacionales como una forma de estructura de datos dentro de una BDOO.

En resumen, una BD orientada a objetos es una BD que incorpora todos los conceptos importantes de la programación orientada a objetos:

- Encapsulación. Ocultar datos del resto de los datos, impidiendo así accesos incorrectos a conflictos.
- Herencia. Reusabilidad del código.
- Polimorfismo. Sobrecarga de operadores o de métodos.

5.8. Bases de datos en telemedicina

En este trabajo, se ha llevado a cabo la **revisión bibliográfica de 154 artículos** que tratan de BDs en telemedicina en las revistas de alto índice de impacto *(IEEE-ITB, JAMIA, Telemedicine and e-Health Journal, Journal of Telemedicine and Telecare, Methods of Information in Medicine, Medical Informatics and the Internet in Medicine, Journal of Biomedical Informatics, Journal of Medical Systems e International Journal of Medical Informatics).* Algunos de ellos tratan sobre SGBD relacionales (Corwin *et ál.* 2007). Taylor *et ál.* emplean BDs de tipo relacional dinámicas en aplicaciones de odontología (Taylor *et ál.* 2002).

En lo relativo al tratamiento y almacenamiento de información, resulta obvio reconocer la necesidad de las BDs en cualquier sistema de información telemédico. En ellas, hay cuatro funciones básicas de procesado de información a cumplir: crear, leer, actualizar y eliminar la propia información. En inglés, estas funciones se resumen en su acrónimo CRUD (Kroenke 2000). Sin embargo, es muy importante explicar que cada BD realiza estas funciones (y otras mucho más avanzadas) de forma muy diversa, de modo que las diferentes aplicaciones a nivel de usuario tendrán disparidad de características, adaptadas a ciertas necesidades, así como tendrán igualmente debilidades o carencias debidas a su funcionamiento interno.

Hasta los últimos años la administración más aceptada era la basada en bases de datos relacionales, como MySQL o *Microsoft Access*. Sin embargo, como enunció el profesor Elliotte Rusty Harold, "C*uando la única herramienta de la que se dispone es un martillo, todo lo demás se asemeja a un clavo. De la misma forma, cuando la única herramienta que se dispone son bases de datos relacionales, todo se asemeja a una tabla"*. Efectivamente, dichas tareas de almacenamiento y procesado únicamente resultaban posibles en las plataformas desarrolladas hasta el momento, todas ellas basadas en asignaciones relacionales.

Es en este punto en el que la aparición del formato XML en el año 1998 supuso una gran revolución, al proponer una alternativa real y eficaz a la ya anticuada organización de las bases de datos, las cuales quedaban muy limitadas debido a las restricciones de posibilidad de tipos de datos de entrada y salida, así como la velocidad de manejo ante contenidos masivos de datos. Desde entonces, la continua evolución de la tecnología asociada a XML ha causado un gran avance hacia la interoperabilidad de los diferentes tipos de información, y ha forjado un progreso hacia un gran formato que sea capaz de englobar los demás formatos, y estandarizar los flujos de información que diferentes dispositivos hayan generado previamente en modelos no definidos hasta ese momento, supliendo así las carencias de las bases de datos relacionales y resultando en una organización más flexible y global (Staken 2001)

5.9. Bases de datos de la aplicación TeleOftalWeb

En este apartado nos centramos en las propiedades de las BDs empleadas en el desarrollo de las diferentes versiones de la aplicación **TeleOftalWeb** (MySQL 5.0, dbXML 2.0, Xindice 1.2, eXist 1.1.1 y Oracle 10g).

En el campo de la telemedicina las BDs nativas XML tienen gran importancia debido a la gran importancia que tiene en ella el almacenamiento y procesado eficiente de la información, estandarizando automáticamente una gran cantidad de información proveniente de distintos dispositivos. La elección de BDs de *Open Source* ha sido debida a la restricción económica, teniendo en cuenta que dichas BDs ofrecen las mismas prestaciones que las privativas, pero libres de cargo.

5.9.1. MySQL

MySQL es un **SGBD relacional**, multihilo, multiusuario y robusto con más de seis millones de instalaciones en todo el mundo. MySQL está diseñado para entornos de producción críticos, con alta carga de trabajo así como para integrarse en *software* para ser distribuido. MySQL es una marca registrada de MySQL AB.

El *software* MySQL tiene una doble licencia. Los usuarios pueden elegir entre usar el *software* MySQL como un producto *Open Source* (de código abierto) bajo los términos de la licencia GNU *General Public License* o pueden adquirir una licencia comercial estándar de MySQL AB (Gilfillan 2003). Su logo aparece en la figura 37.

Figura 37. Logo de MySQL (MySQL 2008).

Al contrario de proyectos como el servidor Apache, donde el *software* es desarrollado por una comunidad pública, y el *copyright* del código está en poder del autor individual, MySQL es propiedad y está patrocinado por una empresa privada, que posee el *copyright* de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado en Suecia por David Axmark, Allan Larsson, y Michael Widenius.

El *software* MySQL, como ya se ha dicho, lo desarrolla, distribuye y soporta MySQL AB. Es una compañía *Open Source* de segunda generación que une los valores y metodología *Open Source* con un exitoso modelo de negocio (Gilfillan 2003).

El sitio Web MySQL proporciona la última información sobre MySQL y MySQL AB, en donde se exponen las principales propiedades de MySQL, que se enuncian a continuación:

- **MySQL es un sistema de gestión de BD relacionales.** Como ya se ha comentado, una BD relacional almacena datos en tablas separadas en lugar de poner todos los datos en un gran almacén. Esto añade velocidad y flexibilidad.
- **El *software* MySQL es *Open Source*.** *Open Source* significa que es posible para cualquiera usar y modificar el *software*. Cualquiera puede bajar el *software* MySQL desde Internet y usarlo sin pagar nada. Si lo desea, puede estudiar el código fuente y cambiarlo para adaptarlo a sus necesidades. El *software* MySQL usa la licencia *General Public License* (GPL)*:* GNU, para definir lo que se puede y no se puede hacer con el *software* en diferentes situaciones. De todos modos, MySQL en una aplicación comercial, y se puede comprar una licencia comercial.
- **El servidor de BD MySQL es muy rápido, fiable y fácil de usar.** MySQL *Server* se desarrolló originalmente para tratar grandes BDs mucho más rápido que soluciones existentes y ha sido usado con éxito en entornos de producción de alto rendimiento durante varios años. *MySQL Server* ofrece hoy en día una gran cantidad de funciones. Su conectividad, velocidad, y seguridad hacen de *MySQL Server* altamente apropiado para acceder a BD en Internet. El servidor MySQL también tiene una serie de características prácticas desarrolladas en cooperación con los usuarios.
- ***MySQL Server* trabaja en entornos cliente/servidor o incrustados.** El *software* de BD MySQL es un sistema cliente/servidor que consiste en un servidor SQL multihilo que trabaja con diferentes programas y bibliotecas cliente, herramientas administrativas y un amplio abanico de APIs. También se proporciona el *MySQL Server* como biblioteca incrustada multihilo que se puede enlazar en una aplicación para obtener un producto más pequeño, rápido y fácil de administrar.
- **Una gran cantidad de *software* de contribuciones está disponible para MySQL.** Está escrito en C y C++ y funciona en casi todos los sistemas operativos existentes y bajo la mayoría de lenguajes de programación.

Las principales características técnicas de MySQL son las siguientes (MySQL 2008):

- **Interioridades y portabilidad.**
    - Escrito en C y en C++.
    - Funciona en diferentes plataformas.
    - Probado con un amplio rango de compiladores diferentes.
    - Usa GNU *Automake, Autoconf,* y *Libtool* para portabilidad.
    - APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.
    - Proporciona sistemas de almacenamiento transaccionales y no transaccionales.
    - Uso completo de multi-threaded mediante *threads* del kernel. Pueden usarse fácilmente múltiples CPUs si están disponibles.
    - Emplea tablas en disco B-tree muy rápidas con compresión de índice.
    - Relativamente sencillo de añadir otro sistema de almacenamiento. Esto es útil si desea añadir una interfaz SQL para una BD propia.
    - Un sistema de reserva de memoria muy rápido basado en threads.
    - *Joins* muy rápidos usando un multi-join de un paso optimizado.
    - Tablas *hash* en memoria, que son usadas como tablas temporales.
    - Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas.
    - El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor. También está disponible como biblioteca y puede ser incrustado (enlazado) en aplicaciones autónomas. Dichas aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible.

- **Tipos de columnas.**
    - Diversos tipos de columnas: enteros con/sin signo de 1, 2, 3, 4, y 8 bytes de longitud, *FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB,*

DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM, y tipos espaciales *OpenGIS*.

- Registros de longitud fija y longitud variable.

- **Seguridad.**

  - Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor.

- **Escalabilidad y límites.**

  - Soporte grandes BDs. Se puede usar *MySQL Server* con 60.000 tablas y cerca de 5.000.000.000.000 de registros.
  - Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2).Un índice puede usar prefijos de una columna para los tipos de columna CHAR, VARCHAR, BLOB, o TEXT.

- **Conectividad.**

  - Los clientes pueden conectar con el servidor MySQL usando *sockets* TCP/IP en cualquier plataforma. En sistemas Windows de la familia NT (NT, 2000, XP, o 2003), los clientes pueden usar *named pipes* para la conexión. En sistemas Unix, los clientes pueden conectar usando ficheros *socket* Unix.
  - En MySQL 5.0, los servidores Windows soportan conexiones con memoria compartida si se inicializan con la opción -shared-memory. Los clientes pueden conectarse a través de memoria.
  - La interfaz para el conector *Open Database Connectivity* (ODBC) *MyODBC* proporciona a MySQL soporte para programas clientes que usen conexiones ODBC. Por ejemplo, se puede usar MS Access para conectar al servidor MySQL. Los clientes pueden ejecutarse en Windows

o Unix. El código fuente de *MyODBC* está disponible. Todas las funciones para ODBC 2.5 están soportadas, así como muchas otras.

- La interfaz para el conector J MySQL proporciona soporte para clientes Java que usen conexiones JDBC. Estos clientes pueden ejecutarse en Windows o Unix.

- **Localización.**

  - El servidor puede proporcionar mensajes de error a los clientes en muchos idiomas.
  - Soporte completo para distintos conjuntos de caracteres, incluyendo latin1 (ISO-8859-1), german, big5, ujis, y más. Por ejemplo, los caracteres escandinavos 'â', 'ä' y 'ö' están permitidos en nombres de tablas y columnas. El soporte para Unicode está disponible
  - Todos los datos se guardan en el conjunto de caracteres elegido. Todas las comparaciones para columnas normales de cadenas de caracteres son *case-insensitive*.
  - La ordenación se realiza acorde al conjunto de caracteres elegido (usando colación Sueca por defecto). Es posible cambiarla cuando arranca el servidor MySQL. *MySQL Server* soporta diferentes conjuntos de caracteres que deben ser especificados en tiempo de compilación y de ejecución.

- **Clientes y herramientas**.

  - *MySQL Server* tiene soporte para comandos SQL para chequear, optimizar, y reparar tablas.
  - Todos los programas MySQL pueden invocarse con las opciones -help o -? para obtener asistencia en línea.

5.9.2. Bases de datos nativas XML

Este tipo de BDs son completamente distintas a las relacionales, las cuales en la actualidad tienen soporte para XML pero, no obstante, siguen almacenando toda la información de manera relacional, es decir en forma tabular (tablas, registros y

columnas) o, en caso contrario, almacenan todo el documento en formato *Binary Large Object* (BLOB). La principal característica que brindan estas BDs es la capacidad de obtener los resultados de las consultas en formato XML; es por ello que dichas BDs pertenecen a la categoría de "*XML-enabled database*".

Una BD nativa en XML ni tiene campos, ni almacena datos atómicos, lo que ella almacena son documentos XML. Por lo tanto a este tipo de BD se les denomina BD centradas en documentos, *data-centric databases*, o documentales. Además, preservan el orden del documento, las instrucciones de procesamiento, los comentarios, las secciones *Character Data* (CHDATA) y las entidades. Se lleva a cabo, también, validación de los documentos, a los cuales se les asocia un identificador único por el que serán reconocidos dentro del repositorio (Grupo Kybele 2004).

Este tipo de BD no utiliza SQL como lenguaje de consulta. En lugar de ello utilizan otros lenguajes, entre los que destacan **XPath y XQuery**, entre otros. Además, tienen una gran variedad de estrategias para actualizar y borrar documentos.

XPath es un lenguaje que nos permite "navegar" dentro de los documentos para alcanzar aquellos elementos que nos sean de interés. XPath modela un documento XML como un árbol de nodos. Hay diferentes tipos de nodos, incluyendo nodos elemento, nodos atributo y nodos texto. El problema que tienen las búsquedas en este tipo de BD es que no permiten realizar búsquedas muy complicadas, como por ejemplo ordenamiento y *cross join*, debido a que Xpath no fue creada realmente para búsquedas en BD, sino simplemente para búsquedas en un solo documento.

XQuery es un lenguaje de consultas diseñado por el W3C para consultar colecciones de datos XML. Es semánticamente similar a SQL, pero incluye algunas capacidades de programación. XQuery proporciona los medios para extraer y manipular información de documentos XML, o de cualquier fuente de datos que pueda ser representada mediante XML. Para ello utiliza expresiones Pat para acceder a determinadas partes del documento XML. Añade además unas expresiones similares a las usadas en SQL, conocidas como expresiones FLWOR. Las expresiones FWLOR toman su nombre de los 5 tipos de sentencias de las que pueden estar compuestas: FOR, LET, WHERE, ORDER BY y RETURN. También incluye la posibilidad de construir nuevos documentos XML a partir de los resultados de la consulta. Se puede usar una sintaxis similar a XML si la estructura (elementos y atributos) es conocida con antelación, o usar expresiones de construcción dinámica de nodos en caso contrario. El lenguaje se basa en el modelo en árbol de la información contenida en el documento

XML, que consiste en siete tipos distintos de nodo: elementos, atributos, nodos de texto, comentarios, instrucciones de procesamiento, espacios de nombres y nodos de documentos.

La Tabla 6 muestra las soluciones actuales de BDs XML, ya sean nativas o bien extensiones de BD.

	BD XML NATIVAS	EXTENSIONES DE BD		
		Almacenamiento de estructurado	Almacenamiento estructurado	Mapeo
Comercial	Tamino TEXTML GoXML DB EXcelon XIS	Oracle XML DB Microsoft SQLXML IBM DB2 XML Extender		Oracle XML DB Mapeo estructurado
Código abierto	dBXML eXist Xindice		Ozone/XML (OO)	
Investigación	Lore Natix PDOM TIMBER		Monet XML Shimura *et ál.* XML Cartridge	

Tabla 6. Soluciones de bases de datos XML. Fuente: propia.

Muchas de estas BDs nativas están actualmente en desarrollo o han sido proyectos frustrados. Al igual que en el mundo de las BDs relacionales hay unas pocas más importantes o más conocidas (Oracle, MySQL, *SQL Server*), en este ámbito de BDs nativas no hay ninguna más importante o más conocida que el resto.

5.9.2.1. dbXML

Hasta ahora hemos hablado de propiedades generales de las BDs nativas XML, que, por tanto, posee dbXML. A continuación nos centraremos en las características específicas de dbXML. Su logo aparece en la figura 38.

Figura 38. Logo de dbXML (dbXML 2008).

La idea principal detrás de dbXML es proporcionar un camino simple para almacenar y administrar un gran número de documentos XML. Esto se lleva a cabo guardando los documentos en colecciones, en donde cada documento individual es guardado en un formato comprimido y pre-procesado. Esto aumenta significativamente la velocidad de almacenamiento cuando se trabaja con datos XML. El motor de dbXML está optimizado para documentos XML de pequeño tamaño, de no más de 50K. El servidor puede, no obstante, almacenar documentos mayores, pero ya no sería un escenario ideal.

Almacenar los documentos en colecciones proporciona un mecanismo simple para hacer consultas y manipular los documentos. Para hacer un paralelismo con una BD relacional, se puede considerar de una manera un tanto aproximada una colección equivalente a una tabla y cada documento de la colección equivalente a una fila de esa tabla. Una de las diferencias principales aparte, obviamente, del uso de XML, es que en dbXML el esquema de lo que se almacene en la BD no es fijo. Esto significa que se tiene una flexibilidad grande en los tipos de documentos que pueden ser almacenados. Si se quiere, se pueden mezclar documentos con esquemas totalmente diferentes en la misma colección. Probablemente no sea muy beneficioso hacerlo, pero sí lo es para poder almacenar y hacer consultas en documentos que son similares pero que no tienen la misma estructura. Por ejemplo, pensemos en un catálogo de productos en donde cada tipo de producto tiene cierta información especializada. En este caso todos los productos tendrán algunos datos comunes y otros especializados. En dbXML se podrían almacenar todos estos productos juntos y después consultar los datos generales de todos o restringir la búsqueda a un tipo particular de producto y buscar sus datos específicos (Staken 2001).

DbXML es, como MySQL, un *software* de código abierto que se desarrolló usando *Java 2 Standard Edition version 1.4* y utiliza la API de Java para la construcción de aplicaciones. Posee un motor multihilo optimizado para datos XML. El servidor dbXML tiene un útil conjunto de herramientas y de comandos que permiten realizar

todas las funciones básicas de administración.

Para actualizar los documentos almacenados utiliza el lenguaje **Xupdate**, que no se debe confundir con **Xpath**, ya que este último se encarga simplemente de señalar los nodos que se quieren actualizar o consultar. *xupdate* es, por tanto, un lenguaje definido en XML y diseñado para actualizar documentos XML para lo cual hace un uso extenso de Xpath para seleccionar los nodos a actualizar y realizar un procesamiento selectivo.

A pesar de todas sus virtudes, el proyecto dbXML que creó y desarrolló *dbXML Group* se abandonó hace casi cuatro años, siendo su última versión, *dbXML 2.0 Released*, sacada el 24 de Agosto de 2004. Es por esta razón que no se dispone de mucha información sobre esta BD.

5.9.2.2. Exist

La BD eXist es un esfuerzo de código abierto para desarrollar una BD nativa XML, que pueda ser integrada fácilmente en la variedad de escenarios posibles que traten con XML. Esta BD está completamente escrita en Java y puede ser desplegada de varias maneras, ya sea corriendo como proceso de servidor, dentro de un motor *servlet* o directamente incrustado en una aplicación.

Mediante el uso de la amplia sintaxis de XPath, los usuarios pueden consultar una parte de la jerarquía de colecciones o incluso todos los documentos contenidos en la BD. Es la BD nativa XML más adecuada para solicitudes que tratan con colecciones de documentos XML, ya que realiza las búsquedas y actualizaciones de forma eficiente. Se ofrece acceso a los desarrolladores a través de HTTP, XML-RPC, SOAP y WebDAV. Las aplicaciones Java pueden utilizar el API XML:DB (The XML:DB Project 2008), que es una interfaz común para acceder a BDs nativas XML.

Como ya se ha comentado XML utiliza lenguajes de programación como **XPath** o **XQuery** para navegar por la estructura jerárquica de un documento XML. Por ejemplo, la expresión:

```
book//section/title
```

Localiza todos los elementos “titulo” que sean hijos de un elemento “sección” que tengan un antepasado llamado “libro”. La doble barra especifica que debe haber un

camino que conduce de un elemento “libro” a un elemento “sección”. La barra simple indica relación padre-hijo. De acuerdo con la versión 2.0 de XPath que es la que utiliza la BD eXist, el resultado de una expresión es una secuencia de distintos nodos en el documento XML que se esté consultando. Para establecer filtros adicionales se utilizan expresiones de predicado. El predicado se evalúa para cada nodo de la secuencia, y devuelve una variable booleana. Los nodos en la secuencia para los que el predicado devuelve *false* se descartan. Por ejemplo, para encontrar todas las secciones cuyo título contiene la cadena "XQuery" en su nodo hijo, se puede utilizar la expresión:

```
book//section[contains(title,'XQuery')]
```

De esta manera sólo si al evaluar el predicado en el nodo se cumple la condición, se devuelve la información que se ha consultado. Pese a la simplicidad, esta forma de realizar las consultas se vuelve ineficiente para grandes colecciones de documentos.

Para empezar a utilizar la BD eXist se ha de descargar la última distribución desde la página oficial (Exist 2008). Se incluye un instalador que se inicia mediante un ejecutable .jar que se incluye y que configura automáticamente todos los *Paths* y variables de entorno para el correcto funcionamiento del sistema gesto de BD. Una vez finalizada la instalación se nos ofrece la posibilidad de inicializar la BD o de abrir el administrador de la misma para poder configurarla. Para ello durante la instalación se pedía una contraseña, que es necesaria para iniciar el administrador. La ventana que se abre para introducir la contraseña se puede ver en la figura 39.

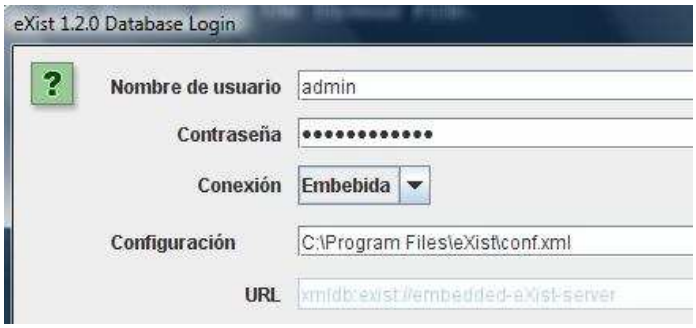


Figura 39. Ventana identificación usuario, eXist. Fuente: propia.

Una vez introducida se accede al administrador, donde se pueden crear y borrar colecciones, crear usuarios y dar permisos, acceder a los documentos almacenados en la

BD, etc. Para acceder a la BD desde una aplicación en Java será necesario introducir un usuario, *login* y contraseña, con permisos sobre esa colección, por lo que habrá que configurar los permisos para el correcto funcionamiento de la aplicación. En la figura 40 se puede ver el aspecto del administrador de la BD eXist. En la figura 41 se ve la ventana mediante la cual se establecen permisos sobre una colección.

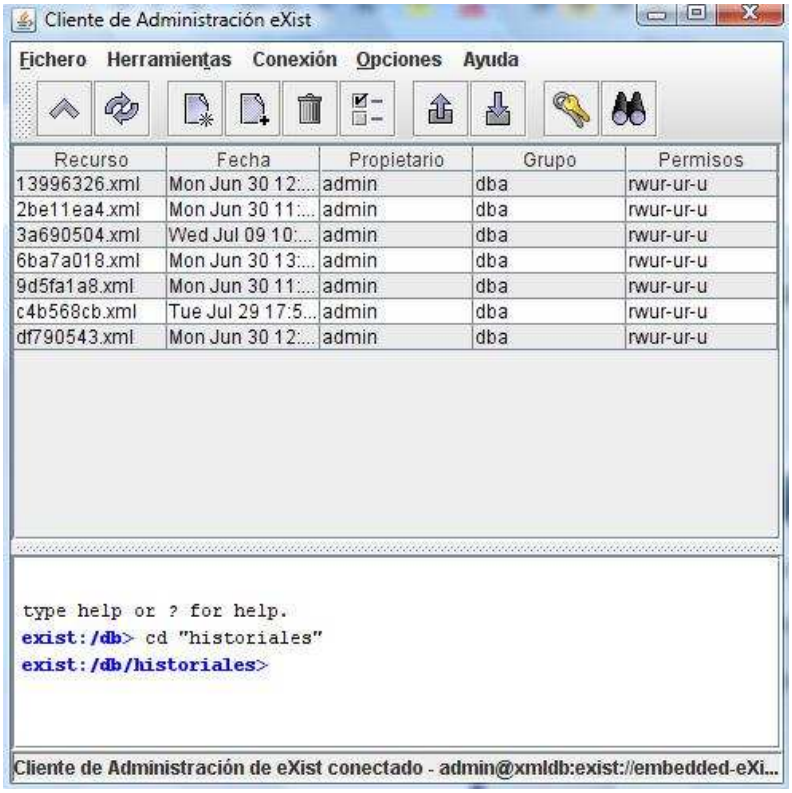


Figura 40. Administrador de la base de datos eXist. Fuente: propia.

Figura 41. Establecimiento de permisos, eXist. Fuente: propia.

Una vez configuradas las colecciones y los permisos, está todo listo para poder acceder a la BD. Para que la BD se inicialice habrá que ejecutar el correspondiente *Startup* de la BD. Se abrirá una ventana, que deberá permanecer abierta para que siga activa. Esta ventana se puede ver en la figura 42.

Figura 42. Inicialización de la base de datos eXist. Fuente: propia.

Una vez inicializada para acceder a una colección se tendrá que utiliza el siguiente *path*. A partir del directoria raíz “db” cual cuelgan todas las colecciones.

```
xmldb:exist://localhost:8888/exist/xmlrpc/db/mi-coleccion
```

El puerto en el que escucha la BD se configura durante la instalación de la misma. El logo de la BD eXist aparece en la figura 43.

Figura 43. Logo de eXist (eXist 2008).

5.9.2.3. Xindice

Apache Xindice es una BD diseñada desde cero para almacenar datos XML o una BD de nativa XML. En la actualidad Xindice utiliza XPath 1.0 para su consulta y

XUpdate 1.0 para su actualización. Apache proporciona una implementación del API XML:DB para desarrollo en Java y el acceso a Xindice desde otros lenguajes utilizando el API XML-RPC. Xindice es la continuación del proyecto que se llamaba dbXML Core. El código fuente de dbXML fue donado a la *Apache Software Foundation* en diciembre de 2001.

Xindice está diseñada para almacenar colecciones de documentos XML. En Xindice el almacén de datos de basa en una instancia de BD que puede ser utilizada como una colección de documentos Esta instancia de BD puede contener cualquier número de colecciones hijo. Por defecto la instancia de BD se llama "db" y todas las rutas se iniciarán con /db. Es posible cambiar el nombre de la instancia de BD si se desea, aunque no es necesario hacerlo. Las colecciones están referenciadas en forma similar a como se trabaja en un sistema jerárquico de ficheros. Un ejemplo de referencia a una colección sería el siguiente:

```
/db/mi-colección/mi-colección-hijo
```

Para acceder a un documento concreto dentro de esa colección bastaría con añadir el identificador del documento al final del *path* que se utiliza para acceder a la colección:

```
/db/mi-colección/mi-colección-hijo/id-documento
```

También se puede acceder a una colección o documento en una máquina remota, especificando el servidor y el puerto al que se ha de acceder, por ejemplo:

```
host.dominio.com:8888/db/mi-colección/mi-colección-hijo/id-
documento
```

El puerto que se ha de utilizar para un correcto funcionamiento de Xindice es el 8888. Otra forma de acceder es utilizando una *Uniform Resource Identifier* (URI) de Xindice. Se puede utilizar tanto para acceso remoto como local:

```
xmldb:xindice://host.dominio.com:8888/db/mi-coleccion

xmldb:xindice://localhost:8888/db/mi-coleccion
```

Para acceder de forma local, se puede utilizar una forma abreviada de la URI que se ha especificado. Esta versión es la que se ha utilizado para la realización del código Java para acceder a la BD que se utiliza en este proyecto:

```
xmldb:xindice:///db/mi-coleccion
```

Para poder acceder a la BD lo primero que hay que hacer es instalar el sistema gestor de BD Xindice en la máquina en la que lo vayamos a utilizar. Las versiones 1.1 y 1.2 de Xindice no son un servidor independiente. Xindice funciona tanto bajo Apache Tomcat como bajo Jetty. La instalación es sencilla. Una vez bajado el código de la página de Apache Xindice (Xindice 2008), basta con desplegar el archivo WAR en el servidor que se vaya a utilizar y estará listo para usar. Cambiando el nombre a este archivo a "xindice.war" podremos acceder a la BD desde cualquier navegador utilizando la siguiente dirección:

```
http://localhost:puerto-servidor/xindice
```

De esta forma para iniciar la BD basta con iniciar el servidor que estemos utilizando. Se acceder así a una interfaz desde la cual se puede acceder a todas las colecciones y a todos los documentos que estén almacenados en la BD. Además desde esta interfaz se puede administrar la BD, creando o borrando colecciones y documentos.

En las figuras 44 y 45 se puede ver cómo es la interfaz que se ofrece para administrar la BD Xindice, para acceder a las coleccione y a los documentos.

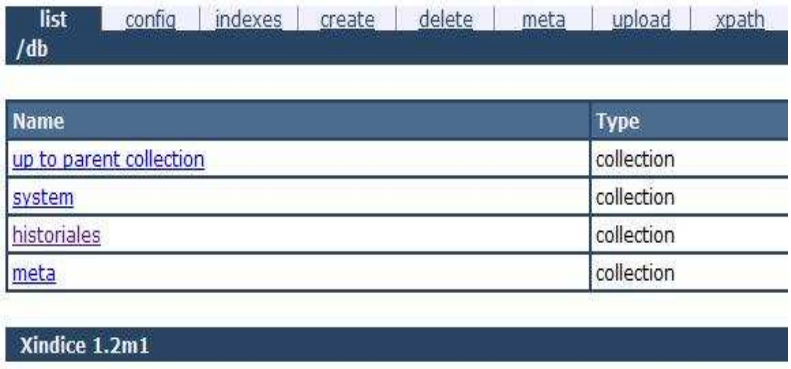


Figura 44. Interfaz de la base de datos Xindice, colecciones. Fuente: propia

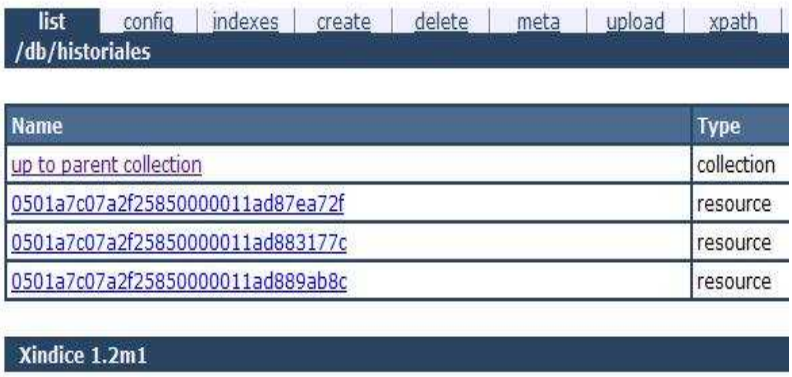


Figura 45. Interfaz de la base de datos Xindice, documentos. Fuente: propia.

Una vez iniciada la BD y creadas las colecciones que se desean ya se puede acceder a ellas desde una aplicación Java, ya sea a documentos XML completos o a campos concretos dentro de ellos. El logo de la BD Xindice aparece en la figura 46.

Figura 46. Logo de Xindice (Xindice 2008).

5.9.3. Oracle

Oracle es un sistema de gestión de BD relacional fabricado por *Oracle Corporation*. Oracle es básicamente un herramienta cliente/servidor para la gestión de BD, es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hace que solo se vea en empresas muy grandes y multinacionales, por norma general. Su logo aparece en la figura 47.

Figura 47. Logo de Oracle (Oracle 2008).

Se considera a Oracle como uno de los sistemas de BD más completos, destacando su:

- Escalabilidad.
- Soporte de transacciones.
- Estabilidad.

- Multiplataforma.

La tecnología Oracle se encuentra prácticamente en todas las industrias del mundo y en las oficinas de 98 de las 100 empresas *Fortune 100*. Oracle es la primera compañía de *software* que desarrolla e implementa aplicaciones para empresas 100% activado por Internet a través de toda su línea de productos: BDs, aplicaciones comerciales y herramientas de desarrollo de aplicaciones y soporte de decisiones. Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del *Microsoft SQL Server* de Microsoft y de la oferta de otros SGBD con licencia libre como PostgreSQL, MySQL o *Firebird* (Silberschatz *et ál.* 2006).

5.9.3.1. Historia de Oracle Corporation

Oracle Corporation es una de las mayores compañías de *software* del mundo. Sus productos van desde BDs hasta sistemas de gestión. Cuenta además, con herramientas propias de desarrollo para realizar potentes aplicaciones, como *Oracle Designer*, o *JDeveloper*. Su jefe ejecutivo actual es Larry Ellison.

Oracle surge a finales de los 70 bajo el nombre de *Relational Software* a partir de un estudio de George Koch sobre SGBD que *Computer World* definió como uno de los más completos jamás escritos sobre la materia. Este artículo incluía una comparativa de productos que erigía a *Relational Software* como el más completo desde el punto de vista técnico. Esto se debía a que usaba la filosofía de las BD relacionales, algo que por aquella época era todavía desconocido.

Corría el año 1977 cuando Larry Ellison fundó *Software Development Laboratories*. En el año 1979 cambia el nombre a *Relational Software Inc*. y presenta su nuevo producto Oracle V2 como una versión comercial de un sistema de administración de BD. Esta versión no soportaba transacciones, pero sí toda la funcionalidad SQL de consultas y uniones.

En 1983 RSI cambia su nombre definitivo a *Oracle Corporation*, y lanza Oracle V3, agregando el manejo de transacciones a través de las instrucciones *COMMIT* y *ROLLBACK*. De hecho, el producto es recodificado en C lo que permite expandir las plataformas de ejecución para incluir los entornos Unix, cuando hasta aquí era solo sobre Digital VAX/VMS.

En 1984, Oracle V4 soporta consistencia de lectura y en 1985 Oracle V5 empieza a soportar el modelo Cliente/Servidor para unirse al auge de la aparición de las redes. Además se soporta la ejecución de consultas distribuidas.

1989 trajo la aparición del *Enterprise Resource Planning* (ERP) de Oracle, conocido como *Oracle Financials*, junto a la versión 6 del motor, que agrega un lenguaje procedural *Procedural Language/Structured Query Language* (PL/SQL) a nivel de fila y las posibilidades de hacer *back up* sin la necesidad de bajar los procesos. Para convertirse en una BD completa, en 1992 aparece Oracle V7h, donde la h viene de *datawareHouse*, aunque lo más significativo es el soporte de la integridad referencial, el almacenamiento y ejecución de programas escritos en PL/SQL dentro del motor y la definición de *triggers* (disparadores) de BD.

Internet era ya una realidad y los nuevos paradigmas de programación empezaban a aparecer para intentar desplazar a los paradigmas imperativos cuando en 1997 Oracle V8 comienza a soportar desarrollos orientados a objetos y el almacenamiento y ejecución de contenido multimedia. En 1999 sale a la luz Oracle 8i para estar a tono con los requerimientos de Internet, de donde se deriva la i del nombre. Además, el motor incorpora una *Java Virtual Machine* (JVM) interna para soportar el almacenamiento y ejecución de código Java dentro del motor.

El nuevo siglo comienza y en 2001 Oracle 9i trae más de 400 características nuevas incluyendo la habilidad de manipular documentos XML, opciones de alta disponibilidad, BD en *cluster*. Un avance importante se hace sobre la definición de BDs virtuales, autenticación vía LDAP y en la autoadministración de la BD.

Por último, en 2003 *Oracle Corporation* lanza Oracle 10g incorporando el manejo y administración de BD "en parrilla" (o *grid*, en inglés), un conjunto de BD cuya administración de espacio, recursos y servicios pueden realizarse como si fueran una sola. La última versión lanzada al mercado fue Oracle 11g, comercializada en 2007.

Oracle tiene su sede en la localidad californiana de Redwood City (Estados Unidos). Según la clasificación *The 2006 Software 500* correspondiente al año 2006, ocupa el primer lugar en la categoría de las BDs y el séptimo lugar a nivel mundial de las compañías de tecnologías de la información. Es el proveedor mundial líder de *software* para administración de información. Por casi dos décadas, *Oracle Corporation*

ha estado resolviendo complejos retos en el manejo de información para empresas de todos los tipos y tamaños.

5.9.3.2. Características generales del SGBD Oracle

Estructura física y lógica.

Las estructuras físicas tales como los archivos del sistema operativo, son almacenadas en dispositivos tangibles como son cintas magnéticas, discos y otros. A cada archivo le corresponde un espacio en el sistema operativo. Oracle requiere de varios archivos para su funcionamiento, los cuales conforman su estructura física.

A la estructura lógica le corresponde un espacio por unidad, pero sus limitaciones son independientes de las localizaciones de espacio físico.

Uso de memoria.

El uso de memoria en el SGBD Oracle tiene como propósito lo siguiente:

- Almacenar los códigos de los programas para empezar a ejecutarse.
- Almacenar los datos necesarios durante la ejecución de un programa.
- Almacenar información sobre como es la transferencia entre procesos y periféricos.

Programas y archivos que componen Oracle.

El SGBD de Oracle está compuesto por tres componentes principales:

1. El *kernel* de Oracle.

   El *kernel* es el corazón del RDBMS Oracle, el cual maneja las siguientes tareas:

   - Manejar el almacenamiento y definición de los datos.
   - Suministrar y limitar el acceso a los datos y la concurrencia de los usuarios.
   - Permitir los backup y la recuperación de los datos.
   - Interpretar el SQL y PL/SQL.

Así como el *kernel* es un sistema operativo, el *kernel* de Oracle es cargado a la memoria al inicio de las operaciones y es usado por cada BD existente en el equipo. Se compone de los siguientes programas:

- El *Program Global Area* (PGA).

Es también llamado *Process Global Area*, consta de datos e información de control de los procesos, asegurando el uso correcto de éstos. El PGA contiene información acerca de las conexiones y los procesos que se realizan en Oracle, su tamaño es variable en longitud, pero no es dinámico. El PGA se activa al conectarse un usuario.

- El *System Global Area* (SGA).

Se puede llamar *Shared Global Area*. Se podría definir como una serie de *buffers* en memoria residente, a través de la cual todas las transacciones y el almacenamiento de datos fluyen. El SGA es localizado en memoria al iniciarse una instancia de la BD y desaparece al pararse. Su tamaño no puede ser cambiado, pero si puede ser visto. Su longitud está definida por los parámetros del archivo de iniciación INIT.ORA. Está compuesto por:

  - Diccionario Caché. Es un conjunto de tablas que son usadas para proveer información asociada con la BD, donde se definen los nombres de usuarios, privilegios, objetos, retenciones y espacios que conforman un SGBD Oracle.
  - Los *Redo Log Buffers*. Es un espacio reservado en memoria que contiene los cambios realizados a los bloques de datos, permitiendo la reconstrucción de la BD en caso de ocurrir un accidente.
  - Los *Database Buffers*. Es un espacio reservado en memoria para las operaciones normales de la BD, el cual depende del tamaño especificado en el archivo de inicialización (INIT.ORA).

2. Las instancias del sistema de BD.
3. Los archivos relacionados al sistema de BD.

5.9.3.3. Oracle Database 10g

La BD de Oracle que usaremos en la aplicación es la versión 10g *Release 2 Enterprise Edition*, por tanto nos centraremos en hablar sobre las propiedades y características de esta BD, que, aunque es relativamente actual (salió al mercado en 2005), no es la más reciente. La última versión es la 11g.

Oracle Database 10g es un producto revolucionario (Gordon 2005), ya que supone la mayor introducción de nuevas funcionalidades de Oracle. Empezando con la revisión 1 (*Release* 1) en 2004 y siguiendo con la revisión 2 (*Release* 2) en 2005. Esta última es la revisión más innovadora de Oracle, liderando la industria de las BDs e introduciéndola en un nuevo terreno en *clustering*, automatización, alta disponibilidad, y más cosas. Ha tenido el mayor impacto en la industria de las BDs que ha producido una revisión de *software*. *Oracle Database 10g* cambia fundamentalmente la forma en que los centros de datos buscan y operan (Loney 2004).

Oracle 10g transforma los centros de datos de colecciones de sistemas monolíticos separados y configurados para manejar aplicaciones individuales, en un pequeño número de fuentes consolidadas y compartidas del servidor y recursos de almacenamiento llamados *grids* de infraestructura. La arquitectura escalar única de Oracle permite a aplicaciones de todo tipo aprovisionar dinámicamente a servidores y recursos de almacenamiento adicionales según se necesite al conocer las demandas de cambios en el procesamiento. Ésto proporciona importantes ahorros en el coste e importantes avances en la calidad de servicio (Loney 2004).

Los ahorros en el coste vienen de muchos sitios. Oracle reduce el coste del *hardware* significativamente permitiendo que incluso los sistemas más grandes puedan ser construidos mediante pequeños servidores de bajo coste y módulos de almacenamiento baratos. Las nuevas propiedades de automatización de Oracle 10g reducen significativamente el coste de la administración de la BD. También por el hecho de que la BD Oracle elimina la necesidad de tener otras costosas herramientas y utilidades de administración (Loney 2004).

Uno de los avances en calidad de servicio más significativos es la automatización de las funciones de administración de la BD. No sólo ganan en

productividad los administradores de la BD, sino que adquieren el poder para dar nuevos niveles de optimización del rendimiento, alta disponibilidad y más. Oracle 10g mejora virtualmente en todas las áreas de las BDs. Esto incluye una mejor disponibilidad, rendimiento, inteligencia de negocio, movimientos de los datos, seguridad y desarrollo de aplicaciones, entre otras cosas (Loney 2004).

La BD Oracle 10g *Enterprise Edition* es ideal para empresas que necesita soporte de un alto volumen de transacciones *online* y aplicaciones con almacenamiento intensivo de datos. Proporciona escalabilidad probada en todas las configuraciones de *hardware* y puede ser usada para manejar grandes cantidades de información, con un gran nivel de seguridad. Además, proporciona ventajas que protegen los datos de errores humanos, reduce el tiempo asociado al mantenimiento rutinario e incluye capacidades de auto-administración para ayudar a bajar los costes operacionales. Al ser la primera BD diseñada para computación en parrilla (*Grid Computing*), reduce significativamente los costes de infraestructura mediante el uso eficiente de los servidores compartidos de bajo coste y los componentes de *hardware* estandarizado (Loney 2004). En la figura 48 se puede observar el logo de la BD Oracle 10g.

Figura 48. Logo de Oracle Database 10g (Oracle 2008).

- Soporta todos los entornos. *Oracle Database 10g Enterprise Edition* está disponible en todos los sistemas operativos soportados por Oracle, incluidos Windows, Linux y Unix, y soporta todas las configuraciones de *hardware*, desde pequeñas máquinas de un solo procesador hasta entornos *Symmetric Multi-Processing* (SMP). Los entornos *cluster* y *grid* son también soportados con la opción de Oracle *Real Application Cluster* (Loney 2004).
- Manejo de todos los datos. *Oracle Database 10g Enterprise Edition* soporta todos los tipos de datos relacionales estándar además del almacenamiento de XML, texto, documentos, imágenes, audio, video y datos de localización.

Además, a esta lista se pueden añadir también datos espaciales complejos con la opción de Oracle *Spatial option*. El acceso a los datos se realiza mediante interfaces estándar tales como SQL, JDBC, SQLJ, ODBC, OLE DB y ODP .NET, SQL/XML, XQuery y WebDAV. La lógica de negocio desarrollada en la BD puede ser escrita en Java o en PL/SQL (Pérez 2004). Oracle 10g *Enterprise Edition* puede almacenar hasta 8 Exabytes de datos en una sola BD. Los datos almacenados fuera de la BD pueden ser cargados en paralelo, mientras que los que están almacenados en una BD existente Oracle pueden ser transportados a través de las plataformas del sistema y añadidos sin la necesidad de descargar los datos (Loney 2004). Una vez almacenados, todos los datos pueden ser transformados, indexados y resumidos usando operaciones paralelas. Las aplicaciones de Inteligencia de Negocio se beneficiarán particularmente de la indexación por mapeo de bits, de las capacidades de unión, de la re-escritura transparente de consultas y de las operaciones de consulta paralelas de la edición *Enterprise*. Todo ello da como resultado una mejora significativa en el rendimiento de las consultas. La opción de particionamiento de Oracle puede ser usada para simplificar las operaciones comunes de administración en grandes entornos de BD, con soporte para realizar estrategias de partición de datos.

- Corre en todas la aplicaciones. Para los entornos de procesamiento de transacciones *online* demandadas, *Oracle Database 10g Enterprise Edition* soporta el despliegue de un gran número de usuarios, permitiendo a la aplicación un rápido y fácil escalado desde decenas a miles de decenas de usuarios *online*. Para la inteligencia de negocio, esta versión de Oracle proporciona capacidades analíticas, estadísticas y de modelado que pueden ser accedidas directamente desde cualquier entorno basado en SQL. Estas capacidades pueden ser expandidas con el uso de las opciones de Oracle *On-Line Analyticas Processing* (OLAP) y *Data Mining* (minería de datos), proporcionando un motor de cálculo OLAP de gran rendimiento y la habilidad de contar datos “in-situ” (Loney 2004).
- Integrar toda la información. *Oracle Database 10g Enterprise Edition* soporta consultas distribuidas y transacciones entre dos o más BD, e incluye soporte para conectar vía ODBC las BDs más comunes (no de Oracle). Además, están disponibles entradas transparentes para BDs más específicas, proporcionando una solución de integración de información muy optimizada. Esta edición

también proporciona un marco de trabajo para capturar, organizar y procesar eventos en la BD, tales como aquellos que son causados por cambios de datos o creados por las aplicaciones de negocio. Estos eventos, junto con los cambios en los datos asociados o los mensajes de aplicación, pueden ser automáticamente propagados y aplicados a una o más BDs o aplicaciones, proporcionando una solución integrada para la consulta de mensajes y la repetición de datos (Loney 2004; Pérez 2004).

- Disponible todo el tiempo. *Oracle Database 10g Enterprise Edition* proporciona capacidades únicas para asegurar la disponibilidad de las aplicaciones más críticas. La opción de Oracle *Real Application Cluster* (RAC) soporta el despliegue transparente de una sola BD a través de un *cluster* de servidores *hardware*, proporcionando tolerancia cero al fallo de *hardware* o a las interrupciones planeadas. También tiene la tecnología *Fast-Start Fault Recovery*, que automáticamente realiza la recuperación de un fallo de la BD en unos segundos, haciendo el tiempo de recuperación rápido y predecible (Loney 2004). *Oracle Database 10g Enterprise Edition* proporciona capacidades únicas de protección de datos. La administración del almacenamiento automático interno de Oracle copia los datos en dispositivos de almacenamiento que estén disponibles para su protección contra fallos de almacenamiento. Se han implementado nuevos algoritmos de validación de datos en conjunción con dispositivos de almacenamiento comunes, eliminando una importante clase de fallos causados por la corrupción. Se proporcionan *backups* automáticas a un disco y recuperaciones desde éste basados en área de recuperación, asegurando que las *backups* estén siempre listas para llevarse a cabo, eliminando la probabilidad de error y mejorando el tiempo de recuperación. Se suministra una combinación automática de las *backups* rápidas e incrementales con las imágenes de *backup* existentes, con lo que se reduce significativamente el tiempo requerido para realizar *backups online* y minimizar el almacenamiento necesario para las operaciones de *backup* diarias (Loney 2004). *Oracle Database 10g Enterprise Edition* también proporciona un conjunto de capacidades únicas de *flashback* que ayudan al administrador a diagnosticar y deshacer de forma sencilla el efecto de los errores humanos incluyendo cambios en una sola fila, cambios hechos por una transacción malintencionada, todos los cambios hechos a una o más tablas (incluyendo la eliminación de una tabla), y

todos los cambios hechos a una BD entera. Con esta versión de Oracle, el tiempo necesario para corregir un error humano es igual o menor que el tiempo que lleva realizar ese error, revolucionando la recuperación de la causa más común de pérdidas de datos (Pérez 2004). *Oracle Database 10g Enterprise Edition* está también diseñada para proteger las operaciones de negocio críticas del impacto del mantenimiento de rutina. Pueden ser añadidos y usados por Oracle nuevo *hardware*, memoria y almacenamiento sin la necesidad de reiniciar los sistemas. En la BD, las tablas pueden ser reubicadas o su tipo de almacenamiento puede ser cambiado, pueden ser añadidos o rehechos nuevos índices, y pueden añadirse columnas, ser eliminadas y cambiadas de nombre sin interrupciones para los usuarios finales que están accediendo a los datos. Esta edición también soporta actualizaciones de la BD y parches y revisiones del sistema operativo, permitiendo verdaderas operaciones 24x7 (Loney 2004; Pérez 2004).

- Seguridad probada. La BD Oracle proporciona la mayor seguridad disponible en la industria actual. En la década pasada Oracle ha completado exitosamente 17 evaluaciones de seguridad independientes. La consolidación de los datos, los requerimientos de privacidad y las regulaciones gubernamentales tales como *Health Insurance Portability and Accountability Act* (HIPAA) requiere características de seguridad sofisticadas. *Oracle Database 10g Release* 2 proporciona a la industria características de seguridad líder como el nivel de seguridad de fila, el de columna, la encriptación de datos, la administración de llaves, la autentificación de Proxy, el contexto de la aplicación y los roles seguros de la aplicación. A estas características se suman las características de seguridad comunes, como las revisiones, la comprobación de la complejidad de las contraseñas, los roles de la BD, los procedimientos almacenados y las funciones. La seguridad avanzada de Oracle (*Oracle Advanced Security*) protege la privacidad y la confidencialidad de los datos existentes en la red. Todas las comunicaciones con la BD Oracle pueden ser encriptadas. También proporciona soluciones robustas de autentificación para Oracle 10g. La seguridad de etiquetas de Oracle (*Oracle Label Security*) proporciona una solución ideal para los clientes que necesitan proteger información privada. Basada en tecnologías de seguridad multinivel, la seguridad de etiquetas restringe el acceso a los datos usando etiquetas sensibles y controles de seguridad (Loney 2004).

- Rápido de instalar, fácil de manejar, fácil de desarrollar. *Oracle Database 10g Enterprise Edition* proporciona una instalación muy rápida en todos los entornos. La BD es pre-configurada para un uso productivo completo con una automatizada administración del espacio, almacenamiento y memoria, *backups* y recuperaciones automáticas, y un administrador de estadísticas automático también. La consola de control de BD incluida en el producto proporciona un interfaz basado en Web que muestra de un vistazo, el estatus de la BD y del entorno de *cluster*, y permite acciones de administración de la BD desde cualquier navegador conectado al sistema. *Oracle Database 10g Enterprise Edition* con la opción RAC toma también ventaja de la solución propuesta para *clusterware*, eliminando la complejidad de tener que instalar y configurar el *cluster*. Las capacidades de administración de almacenamiento automático distribuyen eficientemente los datos almacenados entre los discos disponibles, asegurando un rendimiento óptimo, y eliminando la necesidad de otros administradores de almacenamiento (Loney 2004; Pérez 2004). La interfaz de administración de la BD Oracle 10g se muestra en la figura 49.

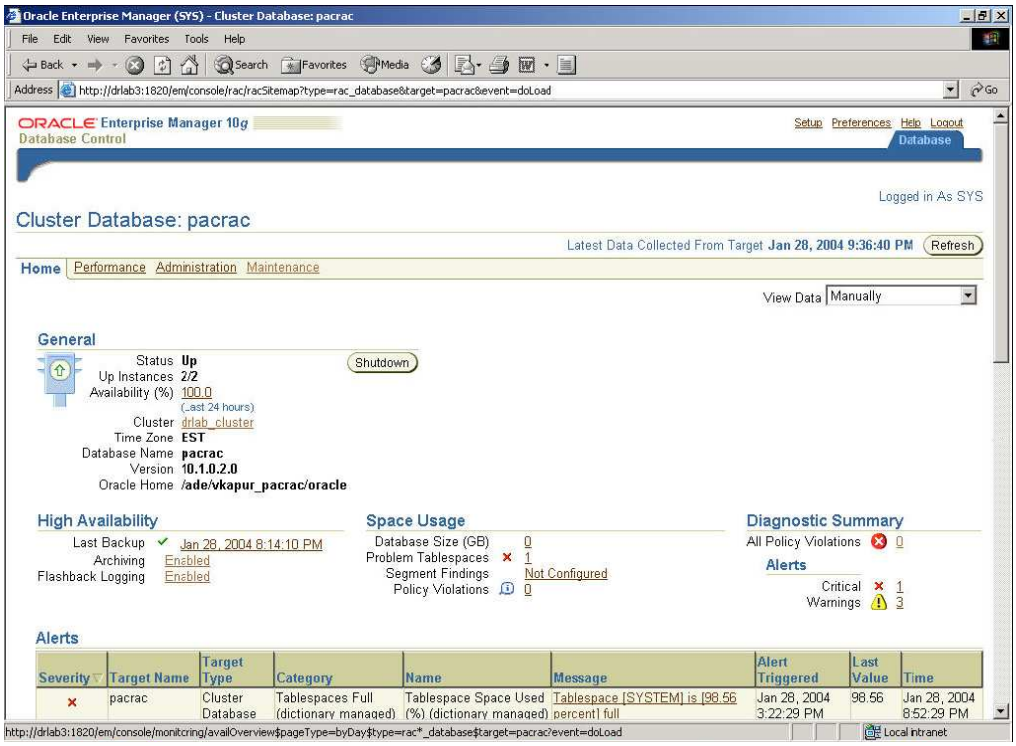


Figura 49. Interfaz de administración basado en Web de la BD. Fuente: propia

Además de estas capacidades, hay otras de administración automatizada que dinamizan operaciones y reducen costes operacionales. El paquete de diagnósticos de Oracle proporciona un conjunto de diagnósticos automáticos y capacidades de monitoreo dentro de la BD, mientras que el paquete de puesta a punto (o "tuneo") de Oracle ofrece una solución fácil de usar que automatiza la compleja tarea que consume tanto tiempo del *tuning* de aplicaciones. El paquete de administración de cambios de Oracle analiza las complejas dependencias asociadas con el cambio de la aplicación y automáticamente realiza los cambios requeridos en la BD, reduciendo errores, mientras que el paquete de administración de la configuración de Oracle reduce la labor asociada al manejo de múltiples BD, automatizando la instalación, parcheando y clonando la BD y manteniendo bajo control las configuraciones del sistema a través de una mejor política práctica y de un intenso rastreo de cambios (Pérez 2004).

Para facilitar el desarrollo, *Oracle Database 10g Release* 2 provee de soporte para la plataforma Windows con la integración de *Visual Studio* y procedimientos almacenados en los lenguajes de tiempo de ejecución más comunes.

- Diseñado para *grid.* La computación en *grid* es el uso combinado de un gran número de servidores de bajo coste y de dispositivos de almacenamiento actuando como un único recurso compartido. Siendo la primera BD diseñada para el *grid*, *Oracle Database 10g Enterprise Edition* permite adoptar la computación en *grid* en tres sencillos pasos, con la mínima inversión, interrupción cero y rápida recuperación de la inversión:

    - A través de la estandarización de los servidores de bajo coste y del almacenamiento.
    - A través de la automatización *end-to-end* de las tareas de administración diarias, permitiendo un solo administrador para manejar simultáneamente cientos de servidores.
    - Mediante el aprovisionamiento dinámico de todas las BDs y servidores de aplicación.

Oracle Database 10g Enterprise Edition y la opción RAC hace fácil la instalación y configuración del entorno de BD que usan estos servidores de bajo coste, incluyendo el aprovisionamiento dinámico de recursos y el balanceo de la carga de

trabajo a través del entorno. La administración de almacenamiento automático permite utilizar almacenamiento de bajo coste, mientras sigue proporcionando los mayores niveles de protección y rendimiento. Y la capacidad de auto-administración de Oracle 10g automatiza el manejo de las operaciones en el *grid* (Pérez 2004).

Con *Oracle Database 10g Enterprise Edition*, las ventajas de la computación en *grid* son tangibles: flexibilidad incremental, sistemas auto-administrables, mejora de la disponibilidad, rendimiento y escalabilidad a un menor coste.

6

Métodos

6.1. INTRODUCCIÓN 176
6.2. ARQUITECTURA DE LA APLICACIÓN 176
6.3. TECNOLOGÍAS EMPLEADAS 177
6.3.1. MÓDULOS DE TELEOFTALWEB EMPLEANDO LAS BDS DBXML 2.0 Y MYSQL 5.0 178
6.3.2. MÓDULOS DE TELEOFTALWEB EMPLEANDO LAS BDS XINDICE 1.2 Y MYSQL 5.0 191
6.3.3. MÓDULOS DE TELEOFTALWEB EMPLEANDO LAS BDS EXIST 1.1.1 Y MYSQL 5.0 198
6.3.4. MÓDULOS DE TELEOFTALWEB EMPLEANDO ORACLE 10G 205
6.4. MODELADO DE DATOS 238
6.4.1. BASE DE DATOS MYSQL 240
6.4.2. BASES DE DATOS NATIVAS XML 244
6.4.3. BASE DE DATOS ORACLE 10G 252
6.5. MODELOS DE COLAS 253
6.5.1. MODELADO DE LA APLICACIÓN Y ESTIMACIONES 257
6.6. PROGRAMAS DE MEDICIÓN DEL TIEMPO DE CARGA 262
6.6.1. ALTERNATIVAS PARA REALIZAR LAS PRUEBAS 262
6.6.2. LA APLICACIÓN FIREBUG 263

6.1. Introducción

En este capítulo se llevará a cabo una descripción de la arquitectura de **TeleOftalWeb**, las tecnologías empleadas en el desarrollo de la misma, así como el modelado de datos que se ha seguido en su diseño, el modelo de colas y los programas empleados para calcular los diferentes tiempos de respuesta de la aplicación en sus distintas versiones.

6.2. Arquitectura de la aplicación

La aplicación **TeleOftalWeb** ha sido desarrollada empleando las tecnologías Java *Servlet* y JSP. Los HCE se almacenan en una BD nativa XML, en las tres primeras versiones de la aplicación. En las Figuras 50 y 51 se muestra la arquitectura de las diferentes versiones de la aplicación. Se trata de una arquitectura de tres capas con dos BD (una relacional y otra XML) en tres de las versiones y una única BD Oracle, en la cuarta versión. El servidor de aplicación se comunica con las BDs. En la comunicación con la BD nativa XML, se emplean los lenguajes XPath y XUpdate. XPath es un lenguaje que permite construir expresiones que recorren y procesan un documento XML. XUpdate es un lenguaje enfocado a la actualización de BDs XML.

La aplicación verifica el estándar CDA Release 1.0 y 2.0. Gracias al empleo de tecnologías XML y a las especificaciones HL7, **TeleOftalWeb** cumple uno de los estándares fundamentales de HCE.

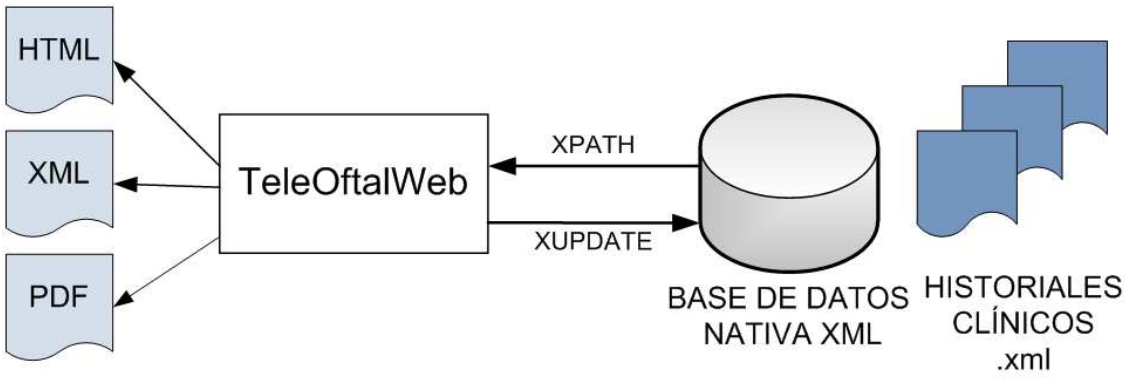


Figura 50. Arquitectura de la aplicación **TeleOftalWeb** con la BDs nativas XML. Fuente: propia.

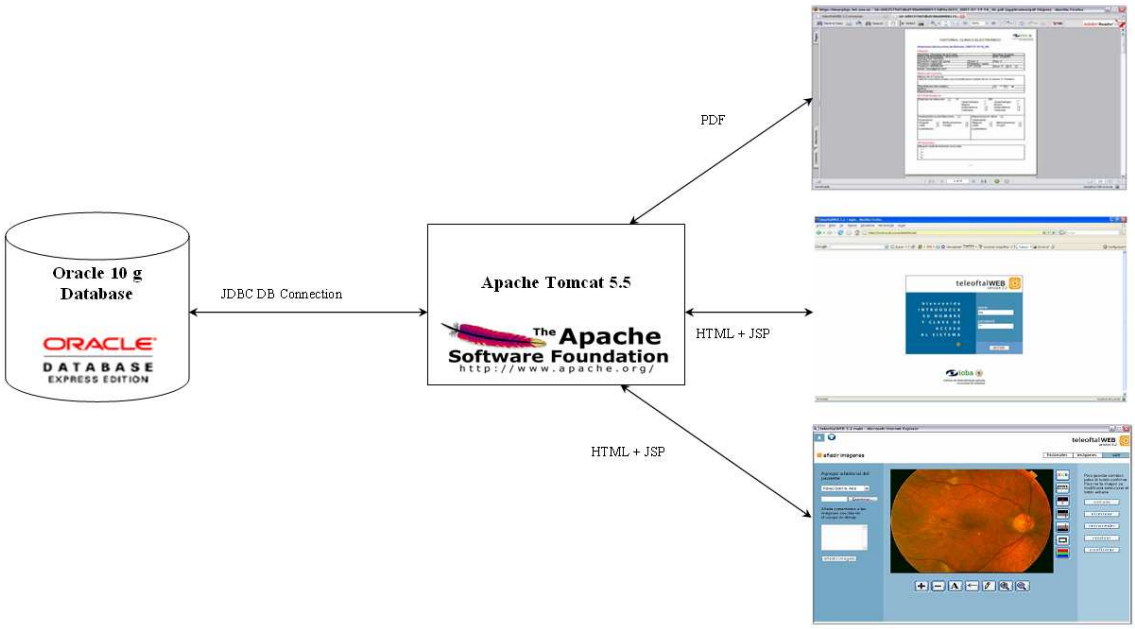


Figura 51. Arquitectura de la aplicación **TeleOftalWeb** con la BD Oracle 10g. Fuente: propia.

6.3. Tecnologías empleadas

Las tecnologías como los servicios Web y XML son las más adecuadas para facilitar el intercambio de información relativa a la HC y hacer que dicha información permanezca accesible en cualquier momento y lugar. Sin embargo, el uso de todas estas tecnologías no es suficiente para permitir la interoperabilidad entre sistemas e instituciones, por lo que fue necesario adoptar un estándar médico que permita dicha comunicación e intercambio de HCEs.

Para el almacenamiento de esta información se usa una BD, que servirá tanto para almacenar datos relacionales de los usuarios de la aplicación, como los historiales clínicos de los pacientes en forma de documentos XML.

Con todas estas ideas en mente, el desarrollo de la nueva versión de la aplicación **TeleOftalWeb** se ha realizado haciendo uso de las siguientes tecnologías:

- **Java, *servlets* y JSPs.** Java es un lenguaje de programación orientado a objetos. Los servlets y JSPs son tecnologías Java que se han utilizado en la lógica embebida en la aplicación Web. El hecho de utilizar el lenguaje de programación Java nos asegura la independencia respecto a la plataforma.
- **XML.** XML es un lenguaje de marcado extensible, lo que significa que con XML se puede crear nuevas etiquetas personales según las necesidades a la hora

de definir un documento. Esta característica de XML proporciona gran flexibilidad y permitirá la interoperabilidad entre aplicaciones con independencia de los datos. Para la formación de los historiales electrónicos se usa el estándar CDA R2 del HL7, que es un modelo genérico basado en XML para especificar la estructura y la semántica de documentos clínicos. Los historiales clínicos oftalmológicos utilizados por el IOBA están adaptados a este estándar.

- ***Secure Socket Layer* (SSL).** SSL es un protocolo criptográfico que proporciona comunicaciones seguras en Internet. Proporciona sus servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente con un algoritmo de cifrado simétrico, típicamente el RC4 o IDEA. En la aplicación **TeleOftalWeb** se hace uso de este protocolo para cifrar las comunicaciones entre cliente, servidor y BD.
- ***Hypertext Transfer Protocol over Secure Socket Layer* (HTTPS).** Es la versión segura del protocolo HTTP. HTTPS utiliza un cifrado basado SSL para crear un canal cifrado más apropiado para el tráfico de información sensible que el HTTP. **TeleOftalWeb** emplea este protocolo para cifrar las comunicaciones entre cliente, servidor y BD.

A continuación, se muestra la programación realizada en las diferentes versiones de **TeleOftalWeb**.

6.3.1. Módulos de TeleOftalWeb empleando las BDs dbXML 2.0 y MySQL 5.0

El código Java generado para **TeleOftalWeb** se divide en cuatro módulos según el tipo de funcionalidades que implementan. Cada uno de esto módulos está representado por una clase. Las principales funcionalidades que debe implementar la lógica de la aplicación se corresponde con las diversas consultas y actualizaciones que se realizan a las diferentes bases de datos. Por otro lado, en cuanto a las peticiones realizadas a la BD nativa XML, algunas de ellas dan como resultado elementos o texto que no requieren un procesamiento adicional; sin embargo, cuando se recuperan grandes bloques, como por ejemplo una sección de una revisión, el fragmento XML resultante debe ser procesado para poder mostrar la información de forma correcta en el interfaz de usuario. De esta forma, los módulos en los que se divide la aplicación son los siguientes:

- **Consultas MySQL.** Este módulo se implementa en la clase *BDmysql* y se encarga de todos los aspectos relacionados con la base de datos relacional MySQL, desde la gestión de las conexiones hasta la realización de las consultas e inserción de usuarios en la misma.
- **Consultas dbXML.** Este módulo se implementa en la clase *BDxml* y se ocupa de todas las transacciones necesarias con la base de datos XML.
- **Parseo XML.** Se implementa en la clase *ParseXML* y se encarga de procesar los resultados de algunas consultas a los documentos XML.
- **Otros.** Este módulo engloba múltiples clases que se encargan de realizar acciones auxiliares necesarias a lo largo de la lógica de la aplicación. .

En la figura 52 puede observarse un esquema de comunicación entre los diferentes módulos:

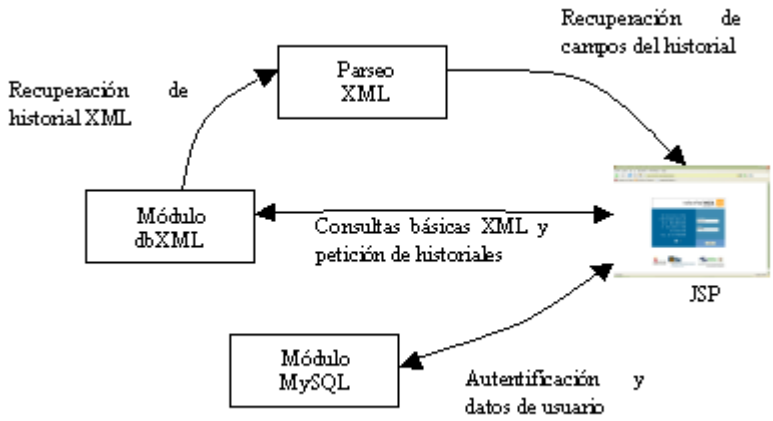


Figura 52.Esquema de comunicación entre módulos. Fuente: propia.

A continuación se analizarán con detalle los módulos de cada una de las clases Java pertenecientes a los módulos descritos.

Consultas MySQL

Este módulo se encarga de todas las consultas relacionadas con la base de datos MySQL. Estas consultas se producen en el proceso de autentificación de usuarios, modificación de datos de usuario, eliminación y adicción de usuarios. Este módulo

implementa la clase *BDmysql*, que se puede observar en la figura 53, y pertenece al paquete *utilidades*.

BDmysql	
Atributos	
Connection connection	**private ArrayList** usuarios
private boolean connectionFree	
Métodos	
public Bdmysql	**public** Collection **getUsuarios**
protected synchronized Connection **getConnection**	**public** Collection **searchUsuarios**
protected synchronized void releaseConnection	**public** boolean **checkLogin**
public Usuario **getUsuario**	**public** String **validarUsuario**
public int **insertarUsuario**	**public** String **getNumColegiado**
public int **borrarUsuario**	**public** String **modificarFechaAcceso**
public int **modificarUsuario**	**public** void **close**

Figura 53. Clase MySQL. Fuente: propia.

Atributos:

- **Connection connection.** Almacena la conexión abierta con la base de datos recuperada del *datasource*.
- **private bolean connectionFree.** Variable lógica que almacena si la conexión con la base de datos está libre o no.
- **private ArrayList usuarios.** Array que almacena el listado de usuarios del sistema recuperado de la base de datos.

Métodos:

- **public Bdmysql().** Constructor de la clase que recupera el *datasource* y solicita una conexión con la base de datos.
- **protected synchronized Connection getConnection ().** Recupera y devuelve la conexión con la base de datos si está libre, y espera por ella en caso contrario.
- **protected synchronized Connection releaseConnection ().** Libera la conexión con la base de datos una vez que se han terminado de realizar las peticiones necesarias.
- **public Usuario getUsuario (String id).** Recupera un usuario de las BDs y lo devuelve en un objeto del tipo Usuario.

- **public int insertarUsuario (Usuario usuario).** Inserta un usuario en la base de datos.
- **public int borrarUsuario (String id).** Elimina un usuario de la base de datos.
- **public int modificarUsuario (Usuario usuario).** Actualiza el usuario proporcionado en la BD.
- **public Collection getUsuarios ().** Obtiene el listado de usuarios del sistema y lo devuelve en una colección de objetos Usuario que almacena en uno de los atributos de la clase.
- **public Collection searchUsuarios (String apellidos, String numcolegiado, String id, String tipousuario).** Devuelve un listado de los usuarios del sistema que cumplen los requisitos especificados en los atributos del método.
- **public boolean checkLogin (String login).** Comprueba si un determinado nombre de usuario es válido y no está siendo ya utilizado en el sistema. Devuelve true si se puede utilizar el nombre de usuario proporcionado y false en caso contrario.
- **public String validarUsuario (String login, String password, String tipo).** Comprueba si un usuario con un determinado nombre tiene la clave y tipo de acceso proporcionado, devolviendo el identificador el mismo en caso afirmativo o null en caso contrario.
- **public String getNumColegiado (String id).** Recupera el número de colegiado de un usuario determinado.
- **public String modificarFechaAcceso (String uid).** Actualiza las fechas del primer (en caso de ser la primera vez que se actualiza) y último acceso del usuario con el identificador proporcionado a la herramienta.
- **public void close ().** Cierra la conexión abierta con la base de datos cuando se destruye el contexto de la aplicación.

Consultas dbXML

Este módulo se implementa mediante la clase *BDxml* que se puede observar en la figura 54. En esta clase se definen los métodos que se encargan del proceso de gestión de historiales dentro de la base de datos nativa XML. Al igual que el resto de clases principales de la aplicación se encuentra dentro del paquete *utilidades*.

BDxml	
Atributos	
private static final String BDHOST	**private static final** String BDPASS
private static final String BDPORT	**private static final** String BDSECURE
private static final String BDUSER	
Métodos	
public static dbXMLClientImpl **inicializar**	**public static** Collection **getParticipantes**
public static String **insertNewDocument**	**public static** Collection **getRevisiones**
public static int SubstituteDocument	**public static boolean checkPermisos**
public static int insertNewBloc	**public static** HashMap **getEstadisticas**
public static Collection **getHistoriales**	**public** String **getNumColegiado**
public static Collection **searchHistoriales**	**public** String **modificarFechaAcceso**
public static int borrarHistorial	**public static** Filiacion **getFiliacion**
public static int eliminarHistorial	**public static** String **getDiagnosticos**
public static int compartirHistorial	**public static** MotivoConsulta **getMotivoConsulta**
public static int insertarImagen	**public static** APOftalmologicos **getAPOftalmologicos**
public static int borrarImagen	**public static** APGenerales **getAPGenerales**
public static Collection **searchImagenes**	**public static** AFamiliares **getAFamiliares**
public static String **getImagenenNombre**	**public static** Exploracion **getExploracion**
public static Collection **getImagenesOriginales**	**public static** Complementa **getComplementa**
public static int actualizaImagenEditada	

Figura 54. Clase dbXML. Fuente: propia.

Atributos:

- **private static final String BDHOST.** En este atributo se especifica la dirección Ip de la máquina que alberga la base de datos.
- **private static final String BDPORT.** Especifica el puerto por el que debe realizarse la conexión a la base de datos. El puerto por defecto es el 7280 y en el presente proyecto no se ha modificado.
- **private static final String BDUSER.** Define el usuario con el que se realiza la conexión a la base de datos. Estos parámetros se definen dentro del código debido a que el API de la base de datos nativa utilizada no permite definir *datasources* de forma sencilla.
- **private static final String BDPASS.** Define la contraseña del usuario con el que la aplicación se conecta a la BD.

- **private static final String BDSECURE.** Determina el mecanismo de seguridad utilizado para las transacciones con la base de datos. En la aplicación se ha fijado el valor "secure", lo que supone que la comunicación con la base de datos se realiza a través de un canal seguro SSL.

Métodos:

- **public static dbXMLClientImpl inicializar ().** Inicializa la conexión con la base de datos.
- **public static String insertNewDocument (String collection, String document).** Inserta un nuevo documento en una determinada colección de la base de datos nativa.
- **public static int SubstituteDocument (String collection, String docKey, String document).** Substituye un documento completo de una colección determinada por otro.
- **public static int insertNewBloc (String collection, String idHistorial, String Xpath, String document, int posicion).** Inserta una sección de una revision en un historial. Mediante el atributo Xpath y posición se especifica el lugar del documento dónde debe insertarse.
- **public static Collection getHistoriales (String collection, String numcolegiado).** Obtiene el listado de historiales visibles para un usuario y los permisos que tiene para ellos.
- **public static Collection searchHistoriales (String collection,String numcolegiado, String apellidos, String dni, String fechaCreacion, String origen).** Realiza una búsqueda atendiendo a los parámetros proporcionados de los historiales visibles para un usuario.
- **public static int borrarHistorial (String collection,String numcolegiado, String IdHistorial).** Elimina de forma lógica un historial para un usuario eliminando sus permisos del mismo.
- **public static int eliminarHistorial (String collection, String IdHistorial).** Elimina físicamente un historial de la BD.
- **public static int compartirHistorial (String collection, String numcolegiadoOrigen, String numcolegiadoCompartir, String IdHistorial).** Añade permisos a un determinado usuario para un historial.

- **public static int insertarImagen (String collection, String numcolegiadoOrigen, String IdHistorial, String nombreImagen, String fechaCreacion, String descripcionImagen, String imagenOriginalB64,String imagenModificadaB64).** Inserta una imagen y todos sus campos en un historial.
- **public static int borrarImagen (String collection, String numcolegiado, String IdHistorial,String nombreImagen).** Elimina una imagen de un historial atendiendo a su identificador.
- **public static Collection searchImagenes (String collection, String numcolegiado, String idHistorial, String nombreImagen, String fechaCreacion, String diagnosticoImagen).** Realiza una búsqueda de imágenes dentro de los historiales visibles para un usuario atendiendo a los parámetros proporcionados.
- **public static String getImagenenNombre (String collection, String IdHistorial, String nombreImagen):** Recupera una determinada imagen de un historial atendiendo a su identificador.
- **public static Collection getImagenesOriginales (String collection, String IdHistorial).** Obtiene todas las imágenes originales asociadas a un historial.
- **public static int actualizaImagenEditada (String collection, String numcolegiado, String IdHistorial, String nombreImagen, String imagenB64).** Almacena la imagen editada de una determinada imagen en un historial.
- **public static Collection getParticipantes (String collection, String IdHistorial).** Obtiene un listado de los participantes de un historial.
- **public static Collection getRevisiones (String collection, String numcolegiado, String IdHistorial).** Obtiene un listado de las revisiones y fechas de las mismas que contiene un historial.
- **public static boolean checkPermisos (String collection, String numcolegiado, String IdHistorial, String permisos).** Comprueba si un usuario tiene un determinado permiso de acceso a un historial, devolviendo true en ese caso y false en el contrario. A este método se llama antes de realizar cualquier operación que necesite cierta seguridad en la base de datos.
- **public static HashMap getEstadisticas (String collection, String numcolegiado).** Obtiene las estadísticas de historiales de un determinado usuario.
- **public static Filiacion getFiliacion (String collection, String IdHistorial, String numcolegiado, String fechaRevision).** Obtiene la sección correspondiente a los

datos de filiación del paciente de una determinada revisión de un historial. Una vez recuperada la sección de la base de datos se llama al módulo de parseo XML para devolver un objeto Filiacion, mucho más fácil de manejar desde el archivo jsp.

- **public static String getDiagnosticos (String collection, String IdHistorial, String numcolegiado, String fechaRevision).** Obtiene los diagnósticos anteriores a una revision de un historial determinado.
- **public static MotivoConsulta getMotivoConsulta (String collection, String IdHistorial, String numcolegiado, String fechaRevision).** Obtiene la sección de los datos referentes al motivo de la consulta de una revisión. Una vez recuperada la sección de la base de datos se llama al módulo de parseo XML para devolver un objeto MotivoConsulta, mucho más fácil de manejar desde el archivo jsp.
- **public static APOftalmologicos getAPOftalmologicos (String collection, String IdHistorial, String numcolegiado, String fechaRevision).** Obtiene la sección de los datos referentes a los antecedentes personales oftalmológicos de una revisión. Una vez recuperada la sección de la base de datos se llama al módulo de parseo XML para devolver un objeto APOftalmologicos, mucho más fácil de manejar desde el archivo jsp.
- **public static APGenerales getAPGenerales (String collection, String IdHistorial, String numcolegiado, String fechaRevision).** Obtiene la sección de los datos referentes a los antecedentes personales generales de una revisión. Una vez recuperada la sección de la base de datos se llama al módulo de parseo XML para devolver un objeto APGenerales, mucho más fácil de manejar desde el archivo jsp.
- **public static AFamiliares getAFamiliares (String collection, String IdHistorial, String numcolegiado, String fechaRevision).** Obtiene la sección de los datos referentes a los antecedentes familiares de una revisión. Una vez recuperada la sección de la base de datos se llama al módulo de parseo XML para devolver un objeto AFamiliares, mucho más fácil de manejar desde el archivo jsp.
- **public static Exploracion getExploracion (String collection, String IdHistorial, String numcolegiado, String fechaRevision).** Obtiene la sección de los datos referentes a la exploración de una revisión. Una vez recuperada la sección de la base de datos se llama al módulo de parseo XML para devolver un objeto Exploracion, mucho más fácil de manejar desde el archivo jsp.

- **public static Complementa getComplementa (String collection, String IdHistorial, String numcolegiado, String fechaRevision).** Obtiene la sección de los datos complementarios de una revisión. Una vez recuperada la sección de la base de datos se llama al módulo de parseo XML para devolver un objeto Complementa, mucho más fácil de manejar desde el archivo jsp.

Parseo XML

Se implementa en la clase *ParseXML* y se encarga de procesar los resultados de algunas consultas a los documentos XML. Para ello hace uso de clases auxiliares definidas en el paquete *datos* almacenando en ellas la información ya procesada. Estas clases auxiliares consisten básicamente en un conjunto de atributos que pueden establecerse y recuperarse con su correspondiente método *set* y *get*, respectivamente. El procesamiento del documento XML se realiza mediante el modelo DOM. Se optó por esta decisión frente SAX debido a su mayor velocidad en el procesamiento no secuencial de documentos, debido en parte a que almacena en memoria la representación completa del documento y en parte a la posibilidad de acceder a cualquier nodo del mismo de forma jerárquica. El hecho de almacenar todo el documento en memoria no supone ningún problema ya que las revisiones se recuperan de forma fragmentada asegurando así que el consumo de memoria no sea elevado.

ParseXML	
Atributos	
Métodos	
public static Filiacion **ParseFiliacion**	**public static** String **ParseParagraph**
public static MotivoConsulta **ParseMotivoConsulta**	**public static** String **ParseParagraphNumero**
public static APOftalmologicos **ParseAPOftalmologicos**	**public static** HashMap **ParseMultipleParagraph**
public static APGenerales **ParseAPGenerales**	**public static** HashMap **ParseMultipleImage**
public static AFamiliares **ParseAFamiliares**	**public static** String **ParseLista**
public static Exploracion **ParseExploracion**	**public static** String **tipoNodo**
public static Complementa **ParseComplementa**	**public static** String **ObtenerContenido**
public static String **getConcatText**	

Figura 55. Clase parseXML. Fuente: propia.

Métodos:

- **public static Filiacion ParseFiliacion (Filiacion filiacion, Element nodo).** Procesa la sección correspondiente a los datos de filiación del paciente almacenando la información en el objeto que recibe.
- **public static MotivoConsulta ParseMotivoConsulta (MotivoConsulta motivoconsulta, Element nodo).** Procesa la sección correspondiente a los datos del motivo de la consulta del paciente almacenando, la información en el objeto que recibe.
- **public static APOftalmologicos ParseAPOftalmologicos (APOftalmologicos apoftalmologicos, Element nodo, String apartado).** Procesa la sección correspondiente a los antecedentes personales oftalmológicos del paciente, almacenando la información en el objeto que recibe.
- **public static APGenerales ParseAPGenerales (APGenerales apgenerales, Element nodo, String apartado).** Procesa la sección correspondiente a los antecedentes personales generales del paciente, almacenando la información en el objeto que recibe.
- **public static AFamiliares ParseAFamiliares (AFamiliares afamiliares, Element nodo, String apartado).** Procesa la sección correspondiente a los antecedentes familiares del paciente, almacenando la información en el objeto que recibe.

- **public static Exploracion ParseExploracion (Exploracion exploracion, Element nodo, String apartado).** Procesa la sección correspondiente a los datos de exploración del paciente, almacenando la información en el objeto que recibe.
- **public static Complementa ParseComplementa (complementarios del paciente).** Almacena la información en el objeto que recibe.
- **public static String getConcatText (Element nodo).** Obtiene todo el texto concatenado de un determinado nodo.
- **public static String ParseParagraph (Element nodo).** Obtiene el texto de una estructura como la siguiente, tomando como argumento el nodo seccion: Complementa complementa, Element nodo, String apartado): Procesa la sección correspondiente a los datos

```
- <section>
    <caption>Titulo de la sección</caption>
  - <paragraph>
      <content>Texto</content>
    </paragraph>
  </section>
```

Figura 56. Estructura XML párrafo. Fuente: propia.

- **public static String ParseParagraphNumero (Element nodo, int i).** Obtiene el texto del parrafo número i de una estructura como la siguiente, tomando como argumento el nodo seccion:

```
- <section>
    <caption>Titulo de la sección</caption>
  - <paragraph>
      <content>Texto1</content>
    </paragraph>
  - <paragraph>
      <content>Texto2</content>
    </paragraph>
  - <paragraph>
      <content>Texto3</content>
    </paragraph>
  </section>
```

Figura 57. Estructura XML múltiples párrafos. Fuente: propia.

- **public static HashMap ParseMultipleParagraph (Element nodo).** Obtiene los títulos y contenidos de una estructura como la siguiente, tomando como argumento

el nodo sección principal, y devolviendo en un objeto tipo HashMap dos vectores, uno con los títulos de las subsecciones y otro con los contenidos de las mismas:

```
- <section>
    <caption>Titulo sección principal</caption>
  - <section>
      <caption>Titulo subsección 1</caption>
    - <paragraph>
        <content>Texto 1</content>
      </paragraph>
    </section>
  - <section>
      <caption>Titulo subsección 2</caption>
    - <paragraph>
        <content>Texto 2</content>
      </paragraph>
    </section>
  </section>
```

Figura 58. Estructura XML múltiples párrafos (subsecciones). Fuente: propia.

- **public static HashMap ParseMultipleImage (Element nodo).** Obtiene el media type y contenido de las imágenes de la estructura utilizada para los esquemas de ojo, tomando como argumento el nodo de la sección principal:

```
- <section>
    <caption>imagenes</caption>
  - <section>
      <caption>nombre1</caption>
    - <paragraph>
      - <content>
        - <observation_media>
            <observation_media.value
              MT="image/jpeg">Base64Imagen1</observation_media.value>
          </observation_media>
        </content>
      </paragraph>
    </section>
  - <section>
      <caption>nombre2</caption>
    - <paragraph>
      - <content>
        - <observation_media>
            <observation_media.value
              MT="image/jpeg">Base64Imagen2</observation_media.value>
          </observation_media>
        </content>
      </paragraph>
    </section>
  </section>
```

Figura 59. Estructura XML esquemas de ojo. Fuente: propia.

- **public static String ParseLista (Element nodo, int numcolumnas).** Obtiene el contenido de los elementos de una lista.
- **public static String tipoNodo (int tipo).** Obtiene el tipo de nodo en formato texto del código que recibe como atributo.
- **public static String ObtenerContenido (HashMap hash, String tipoBuscar, String caption).** Obtiene el contenido asociado a un objeto caption dentro de un objeto HashMap.

Otros

Este módulo engloba múltiples clases que se encargan de realizar acciones auxiliares necesarias a lo largo de la lógica de la aplicación. Algunas de ellas no han sido desarrolladas para este proyecto sino que están disponibles en diversos trabajos de código abierto. Los métodos implementados por estas clases son muchos y muy diversos, por lo que no se entrará a analizarlos con detalle sino que solamente se dará una explicación de las funcionalidades de dichas clases. Las clases que constituyen éste módulo son:

- **Base64.** Esta clase se encarga de la codificación y decodificación de los archivos binarios de las imágenes a base 64, permitiendo almacenar las imágenes directamente dentro de los documentos XML como texto plano y recuperarlas posteriormente.
- **ContextListener.** Se ocupa de cargar en memoria una instancia de la clase BDmysql en la inicialización del contexto de la aplicación y de cerrar las conexiones abiertas del pool de conexiones cuando se destruye el contexto de la aplicación.
- **FileInfo.** Permite recuperar diversos parámetros de los archivos de imagen subidos mediante http-upload en los formularios multi-parte.
- **HttpMultiPartParser.** Permite recuperar los campos de un formulario multi-parte de forma rápida y sencilla.
- **MuestraImagen:** Extiende la clase HttpServlet del API Java Servlets y permite recuperar el texto base 64 de una imagen concreta de un historial y devolver el archivo binario escribiéndolo en la salida estándar para mostrar la imagen en el interfaz de usuario.

- **Utilidades.** Implementa diversos métodos de formateo de fechas, construcción de listas, etc.

6.3.2. Módulos de TeleOftalWeb empleando las BDs Xindice 1.2 y MySQL 5.0

Como ya se ha comentado Apache Xindice es una BD diseñada desde cero para almacenar datos XML. En la actualidad, Xindice utiliza XPath 1.0 para su consulta y XUpdate 1.0 para su actualización. Apache proporciona una implementación del API XML:DB para desarrollo en Java y el acceso a Xindice desde otros lenguajes utilizando el API XML-RPC. Xindice es la continuación del proyecto que se llamaba dbXML Core.

Una vez tenemos la base de datos Xindice instalada, necesitamos algunos ajustes más para poder acceder a ella correctamente desde una aplicación Java. Para que un programa que vaya a acceder a Xindice funcione correctamente se han de incluir en la propia aplicación y en el servidor de aplicaciones sobre el que corre la base de datos una serie de librerías. En este apartado se explicará qué librerías se necesitan para que la aplicación funcione de la forma correcta.

La aplicación **TeleOftalWeb** hace uso de bastantes librerías, que se han de mantener, exceptuando la librería de dbXML. Una vez eliminada esta librería podemos incluir las nuevas librerías que son necesarias para Xindice. A continuación se enumeran estas librerías:

- **xindice.jar**. Contiene las principales clases para Xindice que son usadas.
- **xml-apis.jar**. Contiene los APIs de Java para XML.
- **xerces.jar**: Contiene el parseo Xerces XML.
- **xalan.jar**. Esta librería incluye el motor Xalan XSLT.
- **commons-logging.jar**. Contiene el paquete *Jakarta Commons Logging.*

Además de éstas, hay otras cuatro librerías que son necesarias y que contienen implementaciones del API XML:DB y del API XUpdate. Estas cuatro librerías son las que se nombran a continuación:

- **xmldb-common.jar.**
- **xmldb-api.jar.**
- **xmldb-api-sdk.jar.**
- **xmldb-xupdate.jar.**

Una vez incluidas estas librerías ya se puede empezar a desarrollar la adaptación de la aplicación para que funcione con la BD nativa XML Xindice en lugar de con dbXML.

Tras configurar adecuadamente la BD, se puede acceder a ella desde una aplicación Java. En este apartado se presenta un ejemplo para acceder a un documento XML que esté almacenado en la base de datos

Lo primero que hay que hacer es importar las librerías necesarias para el correcto funcionamiento de la aplicación. Ésto se puede ver en el siguiente código:

```
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.*;
import org.apache.xindice.client.xmldb.DatabaseImpl;
```

Estas librerías contenidas en el archivo xmldb-api.jar contienen las clases necesarias para el acceso a bases de datos nativas XML, así como para realizar consultas o actualizaciones. Estas librerías nos permiten definir un objeto de la clase *Collection*, que es diferente a la clase *Collection* que está definida en *java.util*, y que permite cargar una colección almacenada en la base de datos. La clase *DatabaseImpl* que está incluida en el archivo xindice.jar, es la que nos permite definir una instancia para acceder a la base de datos Xindice. Por lo tanto, el siguiente paso es definir un objeto *Collection*, y acceder a la base de datos. Para ello se necesita el siguiente código:

```
Collection col = null;
String driver = "org.apache.xindice.client.xmldb.DatabaseImpl";
Class c = Class.forName(driver);
Database database = (Database) c.newInstance();
DatabaseManager.registerDatabase(database);
```

La primera línea crea un objeto *Collection* llamado “col”, y el resto carga el “driver” de la base de datos, que en este caso es Xindice. Para esto se define una variable String con el nombre de la clase que se quiere cargar, y mediante el método *forName()* de la clase *Class* se carga la clase que necesitamos. Las últimas dos líneas hacen uso de clases definidas en el API XML:DB para crear una instancia de la base de datos a partir de la clase que hemos cargado.

Una vez se ha conectado con la BD podemos hacer la operación que necesitemos. En este ejemplo se pide a la base de datos que nos muestre un documento concreto. Para ello necesitamos saber el nombre de la colección y el identificador del documento XML. Se supone que dicha información está en las variables de nombre "coleccion" y "idhistorial". Por lo tanto, se pide que se cargue en el objeto "col", de la clase *Collection*, la colección que le pedimos, y una vez cargada se le pide el documento al que queremos tener acceso. Esto se puede ver a continuación:

```
col=DatabaseManager.getCollection("xmldb:xindice:///db"+coleccion);
XMLResource document = (XMLResource) col.getResource(idhistorial);
System.out.println((String) document.getContent());
```

En la primera línea se recupera la colección en la que se encuentra en el historial, y que se encuentra en la variable "coleccion". Para que este código funcione, el servidor de aplicaciones debe estar escuchando en el puerto 8888. Una vez cargado se crea un objeto de la clase *XMLResource*, que también forma parte del API XML:DB, para almacenar el documento que se necesita. En la última línea se muestra por pantalla el contenido de dicho documento XML.

Los módulos en los que se divide la aplicación son los mismos que para el caso de la BD dbXML 2.0, en este caso sólo cambian las consultas a la nueva BD XML. Este nuevo módulo se implementa mediante la clase *BDxml*, que se puede ver en la figura 60. En esta clase se definen los métodos que se encargan del proceso de gestión de historiales dentro de la base de datos nativa XML, en este caso Xindice. Al igual que el resto de clases principales de la aplicación se encuentra dentro del paquete *utilidades*.

Figura 60. Clase BDXML para Xindice. Fuente: propia.

Al realizar la adaptación se han conservado todos los métodos utilizados para comunicar la aplicación con dbXML. Además, dichos métodos devuelven los mismos tipos de variables y resultados, por lo que no es necesario modificar otros módulos diferentes al que realiza las consultas a la base de datos nativa XML. A continuación se describen brevemente cada uno de los métodos.

- **public static String insertNewDocument (String collection, String document).** Inserta un nuevo documento en una determinada colección de la base de datos nativa.

- **public static int SubstituteDocument (String collection, String docKey, String document).** Substituye un documento completo de una colección determinada por otro.
- **public static int insertNewBloc (String collection,String idHistorial, String Xpath, String document, int posicion).** Inserta una sección de una revisión en un historial. Mediante el atributo Xpath y posición se especifica el lugar del documento dónde debe insertarse.
- **public static Collection getHistoriales (String collection, String numcolegiado).** Obtiene el listado de historiales visibles para un usuario y los permisos que tiene para ellos.
- **public static Collection searchHistoriales (String collection,String numcolegiado, String apellidos, String dni, String fechaCreacion, String origen).** Realiza una búsqueda atendiendo a los parámetros proporcionados de los historiales visibles para un usuario.
- **public static int borrarHistorial (String collection,String numcolegiado, String IdHistorial).** Elimina de forma lógica un historial para un usuario eliminando sus permisos del mismo.
- **public static int eliminarHistorial (String collection, String IdHistorial).** Elimina físicamente un historial de la base de datos.
- **public static int compartirHistorial (String collection, String numcolegiadoOrigen, String numcolegiadoCompartir, String IdHistorial).** Añade permisos a un determinado usuario para un historial.
- **public static int insertarImagen (String collection, String numcolegiadoOrigen, String IdHistorial, String nombreImagen, String fechaCreacion, String descripcionImagen, String imagenOriginalB64,String imagenModificadaB64).** Inserta una imagen y todos sus campos en un historial.
- **public static int borrarImagen (String collection, String numcolegiado, String IdHistorial,String nombreImagen).** Elimina una imagen de un historial atendiendo a su identificador.
- **public static Collection searchImagenes (String collection, String numcolegiado, String idHistorial, String nombreImagen, String fechaCreacion, String diagnosticoImagen).** Realiza una búsqueda de imágenes dentro de los historiales visibles para un usuario atendiendo a los parámetros proporcionados.

- **public static String getImagenenNombre (String collection, String IdHistorial, String nombreImagen).** Recupera una determinada imagen de un historial atendiendo a su identificador.
- **public static String getImagenModificada (String collection, String IdHistorial, String nombreImagen).** Obtiene las imagen modificada correspondiente a una imagen almacenada en un historial.
- **public static Collection getImagenesOriginales (String collection, String IdHistorial).** Obtiene todas las imágenes originales asociadas a un historial.
- **public static int actualizaImagenEditada (String collection, String numcolegiado, String IdHistorial, String nombreImagen, String imagenB64).** Almacena la imagen editada de una determinada imagen en un historial.
- **public static Collection getParticipantes (String collection, String IdHistorial).** Obtiene un listado de los participantes de un historial.
- **public static Collection getRevisiones (String collection, String numcolegiado, String IdHistorial).** Obtiene un listado de las revisiones y fechas de las mismas que contiene un historial.
- **public static boolean checkPermisos (String collection, String numcolegiado, String IdHistorial, String permisos).** Comprueba si un usuario tiene un determinado permiso de acceso a un historial, devolviendo true en ese caso y false en el contrario. A este método se llama antes de realizar cualquier operación que necesite cierta seguridad en la base de datos.
- **public static HashMap getEstadisticas (String collection, String numcolegiado).** Obtiene las estadísticas de historiales de un determinado usuario.
- **public static Filiacion getFiliacion (String collection, String IdHistorial, String numcolegiado, String fechaRevision).** Obtiene la sección correspondiente a los datos de filiación del paciente de una determinada revisión de un historial. Una vez recuperada la sección de la base de datos se llama al módulo de parseo XML para devolver un objeto Filiacion, mucho más fácil de manejar desde el archivo jsp.
- **public static String getDiagnosticos (String collection, String IdHistorial, String numcolegiado, String fechaRevision).** Obtiene los diagnósticos anteriores a una revisión de un historial determinado.
- **public static MotivoConsulta getMotivoConsulta (String collection, String IdHistorial, String numcolegiado, String fechaRevision).** Obtiene la sección de

los datos referentes al motivo de la consulta de una revisión. Una vez recuperada la sección de la base de datos se llama al módulo de parseo XML para devolver un objeto MotivoConsulta, mucho más fácil de manejar desde el archivo jsp.

- **public static APOftalmologicos getAPOftalmologicos (String collection, String IdHistorial, String numcolegiado, String fechaRevision).** Obtiene la sección de los datos referentes a los antecedentes personales oftalmológicos de una revisión. Una vez recuperada la sección de la base de datos se llama al módulo de parseo XML para devolver un objeto APOftalmologicos, mucho más fácil de manejar desde el archivo jsp.
- **public static APGenerales getAPGenerales (String collection, String IdHistorial, String numcolegiado, String fechaRevision).** Obtiene la sección de los datos referentes a los antecedentes personales generales de una revisión. Una vez recuperada la sección de la BD se llama al módulo de parseo XML para devolver un objeto APGenerales, mucho más fácil de manejar desde el archivo jsp.
- **public static AFamiliares getAFamiliares (String collection, String IdHistorial, String numcolegiado, String fechaRevision).** Obtiene la sección de los datos referentes a los antecedentes familiares de una revisión. Una vez recuperada la sección de la base de datos se llama al módulo de parseo XML para devolver un objeto AFamiliares, mucho más fácil de manejar desde el archivo jsp.
- **public static Exploracion getExploracion (String collection, String IdHistorial, String numcolegiado, String fechaRevision).** Obtiene la sección de los datos referentes a la exploración de una revisión. Una vez recuperada la sección de la base de datos se llama al módulo de parseo XML para devolver un objeto Exploracion, mucho más fácil de manejar desde el archivo jsp.
- **public static Complementa getComplementa (String collection, String IdHistorial, String numcolegiado, String fechaRevision).** Obtiene la sección de los datos complementarios de una revisión. Una vez recuperada la sección de la base de datos se llama al módulo de parseo XML para devolver un objeto Complementa, mucho más fácil de manejar desde el archivo jsp.
- **public static String getXML (String collection, String documento).** Devuelve un documento XML completo. La finalidad de este método es usarlos para la creación del PDF de una revisión.

Todos estos métodos que se han explicado han sido modificados para que realicen las mismas funciones, que se realizaban utilizando la base de datos dbXML, con Xindice. En esta versión cada método accede a la base de datos Xindice y realiza las mismas consultas y actualizaciones que realizaba con dbXML. Además, como Xindice es la continuación del proyecto dbXML Core, la manera de procesar los datos que devuelve es similar y no ha sido necesario introducir demasiados cambios en el procesamiento de la información.

6.3.3. Módulos de TeleOftalWeb empleando las BDs eXist 1.1.1 y MySQL 5.0

La BD eXist es un esfuerzo de código abierto para desarrollar una base de datos nativa XML, que pueda ser integrada fácilmente en la variedad de escenarios posibles que traten con XML. Esta base de datos está completamente escrita en Java y puede ser desplegada de varias maneras, ya sea corriendo como proceso de servidor, dentro de un motor *Servlet* o directamente incrustado en una aplicación.

La BD eXist usa la versión 2.0 de XPath para realizar las consultas, y XUpdate 1.0 para las actualizaciones. Las aplicaciones Java pueden utilizar el API XML:DB, que es una interfaz común para acceder a bases de datos nativas XML.

Con la base de datos eXist ya instalada, es hora de realizar los últimos ajustes para poder acceder a ella correctamente desde una aplicación Java. Al igual que ocurría con Xindice para que el sistema gestor de la base de datos funcione correctamente se han de incluir en la aplicación y en el servidor de aplicaciones una serie de librerías. En esta sección se puede ver qué librerías son necesarias para el correcto funcionamiento de eXist.

Las librerías que se han de incluir incluyen también las necesarias para la aplicación **TeleOftalWeb**, excepto la referente a la base de datos dbXML, como ya se comentó en el caso de Xindice. Además de éstas, hay otras siete librerías que son necesarias y que contienen implementaciones del API XML:DB y del API XUpdate entre otros. Estas siete librerías son las que se nombran a continuación:

- **xmldb-common.jar.**
- **xmldb-api.jar.**
- **xmldb-api-sdk.jar.**
- **xmldb-xupdate.jar.**
- **antlr.jar.**

- **log4j.jar.**
- **xmlrpc-1.2-patched.jar.**

Como se puede ver es necesario un mayor número de librerías que las que se necesitaban para Xindice. Una vez incluidas estas librerías ya se puede empezar a desarrollar la adaptación de la aplicación para que funcione con la base de datos nativa XML eXist en lugar de con dbXML.

Al igual que se hizo para la base de datos Xindice, en este apartado se va a mostrar con un ejemplo cómo se puede acceder a un documento XML que esté almacenado en la base de datos eXist. Como se va a ver el hecho de que Xindice y eXist usen el API XML:DB para acceder a la base de datos hace que el código que se utiliza en ambas sea muy parecido. Lo que cambia es el API de la base de datos que utilizan. En este caso se utiliza el API eXist, para conectar con la base de datos del mismo nombre.

El primer paso, como ya ocurría con Xindice es importar las librerías necesarias para el correcto funcionamiento de la aplicación. Esto se puede ver en el siguiente código.

```
import org.xmldb.api.base.*;
import org.xmldb.api.modules.*;
import org.xmldb.api.*;
import org.exist.xmldb.DatabaseImpl;
```

Estas librerías contenidas en el archivo xmldb-api.jar contienen las clases necesarias para el acceso a bases de datos nativas XML, así como para realizar consultas o actualizaciones. Por este motivo son las mismas para las dos bases de datos.

Las librerías que se han importado permiten definir un objeto de la clase *Collection*, que es diferente a la clase *Collection* que está definida en *java.util*, y que permite cargar una colección almacenada en la base de datos. La clase *DatabaseImpl* que está incluida en el archivo exist.jar, es la que nos permite definir una instancia para acceder a la base de datos eXist. Por lo tanto, el siguiente paso es definir un objeto *Collection*, y acceder a la base de datos. Para ello se utiliza el siguiente código:

```
Collection col = null;
String driver = "org.exist.xmldb.DatabaseImpl ";
Class c = Class.forName(driver);
```

```
Database database = (Database) c.newInstance();
DatabaseManager.registerDatabase(database);
```

Se puede ver que se mantiene la misma estructura que para Xindice, sólo que se accede a la base de datos eXist. La primera línea crea un objeto *Collection* llamado "col", y el resto carga el "driver" de la base de datos, que en este caso es eXist. Para esto se define una variable String con el nombre de la clase que se quiere cargar, y mediante el método *forName()* de la clase *Class* se carga la clase que necesitamos. Las últimas dos líneas hacen uso de clases definidas en el API XML:DB para crear una instancia de la base de datos a partir de la clase que hemos cargado.

Una vez se ha conectado con la base de datos podemos hacer la operación que necesitemos. En este ejemplo se pide a la base de datos que nos muestre un documento concreto. Para ello necesitamos saber el nombre de la colección y el identificador del documento XML. Se supone que dicha información está en las variables de nombre "coleccion" y "idhistorial". Por lo tanto, se pide que se cargue en el objeto "col", de la clase *Collection*, la colección que le pedimos, y una vez cargada se le pide el documento al que queremos tener acceso. Esto se puede ver a continuación:

```
col=DatabaseManager.getCollection
("xmldb:exist://localhost:8080/exist/xmlrpc/db" + coleccion, "login",
"contraseña");
XMLResource document = (XMLResource) col.getResource(idhistorial);
System.out.println((String) document.getContent());
```

Aquí empiezan las diferencias con Xindice. El *path* mediante el cual se accede a eXist es distinto. Además se necesita especificar un *login* y una contraseña. Como se puede ver la base de datos eXist escucha en el puerto 8080, que es el puerto que hemos especificado en la instalación. En la primera línea se recupera la colección en la que se encuentra en el historial, y que se encuentra en la variable "coleccion". Una vez cargado se crea un objeto de la clase *XMLResource*, que también forma parte del API XML:DB, para almacenar el documento que se necesita. En la última línea se muestra por pantalla el contenido de dicho documento XML. A penas hay diferencias con el código para Xindice que realiza la misma operación.

Como ya ocurría para Xindice, a partir de este ejemplo se podrían realizar el resto de funciones que son necesarias para la aplicación, como insertar un documento, hacer una consulta o actualizar información. Para ello habría que cambiar las acciones que se hacen sobre la base de datos, pero el código que permite la comunicación con ella seguirá siendo el mismo. En el siguiente apartado se comentará como se han realizado todas estas acciones para adaptar la aplicación **TeleOftalWeb** para que funcione con eXist.

Como ya se comentó en apartados anteriores el módulo de consultas a la base de datos XML se implementa mediante la clase BDXML. En esta clase se definen los métodos que se encargan del proceso de gestión de historiales dentro de la BD nativa XML, en este caso eXist. Al igual que el resto de clases principales de la aplicación se encuentra dentro del paquete *utilidades*.

Figura 61. Clase BDXML para eXist. Fuente: propia.

En esta versión cada método accede a la base de datos eXist y realiza las mismas consultas y actualizaciones que realizaba con dbXML.

La adaptación de los métodos de la clase *BDxml* presenta mayor complicación en esta versión de la aplicación, por no se eXist, como lo es Xindice, una continuación del proyecto dbXML Core. En los siguientes apartados se verá como se han solventado las dificultades que ha presentado el desarrollo de esta versión de la aplicación.

Entre las diferencias con la versión para Xindice, cabe destacar la modificación de la clase *ParseXML*. Esta modificación se debe a incompatibilidades con el API DOM, que se utilizaba para devolver los datos cómo un árbol de nodos. En esta versión, esto no ha sido posible. Varios métodos de la clase *BDxml* se comunicaban con la clase

ParseXML, pasándole un objeto de la clase *Node*. La clase *ParseXML* procesaba el objeto y devolvía la información de la forma adecuada para que se mostrara de manera correcta al usuario. Los métodos implicados en esta modificación son los siguientes:

- **getFiliacion.**
- **getMotivoConsulta.**
- **getAPOftalmologicos.**
- **getAPGenerales.**
- **getAFamiliares.**
- **getExploracion.**
- **getComplementa.**

De esta manera se ve que, pese a que las clases *BDxml* tienen los mismos métodos en las dos versiones de la aplicación, la forma en la que se llevan a cabo las acciones puede ser distinta.

Las consultas a las bases de datos son uno de los procedimientos más utilizados. Como ya se ha comentado las consultas en eXist se llevan a cabo mediante el lenguaje XPath 2.0. Por lo tanto en la aplicación será necesario definir una variable *String*, que contenga la cadena XPath 2.0 que se va a utilizar. Una vez se conecte con la base de datos las consultas se pueden realizar de dos maneras: a toda la base de datos o a un documento en concreto. Para estos ejemplos se supone que la cadena XPath está contenida en la variable "query". Para realizar una consulta sobre toda la base de datos, una vez se ha conectado, se necesita el siguiente código:

```
XPathQueryService service = (XPathQueryService) col.getService
("XPathQueryService", "2.0");
ResourceSet resultSet = service.query(query);
```

Como se puede ver se crea un objeto de la clase *XPathQueryService*, al que se le asigna realizar una consulta XPath sobre la colección que se necesita. Para realizar una consulta sobre un único documento XML se necesita conocer el identificador del documento. La manera de llevar a cabo la consulta es muy parecida:

```
XPathQueryService service = (XPathQueryService) col.getService
("XPathQueryService", "2.0");
```

```
ResourceSet resultSet = service.queryResource
(idHistorial,query);
```

En este caso se puede ver como una vez definido el objeto de la clase *XPathQueryService*, en lugar de usar el método *query*, se utiliza el método *queryResource*, que necesita un segundo parámetro, además de la cadena XPath. Este parámetro es el identificador del documento XML. Una vez se realiza la consulta, la manera de proceder para recibir la información que se ha pedido es la siguiente:

```
ResourceIterator results = resultSet.getIterator();
while(results.hasMoreResources()){
        Resource res = results.nextResource();
        System.out.println( (String) res.getContent());
}
```

De esta manera se crea un objeto de la clase *ResourceIterator* con el que se puede recorrer toda la información que nos ha devuelto la consulta, de una manera sencilla. En este código se le presenta al usuario el resultado por pantalla. Para la aplicación se realizan algunos procesados sobre la información para que se le presente al usuario de la manera correcta al acceder a la aplicación **TeleOftalWeb**.

Como se puede ver no existen diferencias significativas con el código que hace las mismas funciones para Xindice. Una de las diferencias que se pueden apreciar en un primer vistazo es que eXist utiliza la versión 2.0 de XPath. Este cambio lleva asociada una serie de cambios que se pueden ver en la sección 6.4. Estos cambios no sólo afectan a la clase BDxml, sino también al archivo accionesHistorial.jsp, ya que en él aparece expresiones XPath 1.0 que no funcionan bien en esta versión.

Las actualizaciones en la base de datos eXist se llevan a cabo mediante el lenguaje XUpdate 1.0, el mismo que utiliza Xindice. La forma de preceder es análoga a la forma en que se realizan las consultas:

```
XUpdateQueryService service = (XUpdateQueryService)
col.getService ("XUpdateQueryService", "1.0");
service.queryResource (idHistorial, query);
```

Una vez ejecutado este código se actualiza el contenido que se necesita. El método *queryResource* devuelve el número de nodos que han sido modificados en el documento XML.

6.3.4. Módulos de TeleOftalWeb empleando Oracle 10g

Una vez instalada la aplicación y la base de datos Oracle el siguiente paso es crear las tablas usadas en MySQL en Oracle. Para ello se hizo uso de una herramienta proporcionada en la página Web de Oracle que sirve para migrar tablas y datos de otras bases de datos como MySQL, Access y Microsoft SQL *Server* a Oracle. Esta herramienta se llama *Oracle SQL Developer Migration Workbench* y no requiere de instalación.

Antes de activar esta herramienta hay que asegurarse de que están funcionando las bases de datos de MySQL y de Oracle, es decir, que hay una instancia de cada base de datos activa. Una vez comprobado esto se arranca la aplicación y se llevan a cabo los siguientes pasos:

I. Preparar el entorno para la migración.

1. **Crear una conexión a la base de datos Oracle.** Para ello se clica en el icono + verde del navegador de conexiones para abrir el diálogo de nueva conexión. Una vez abierto se elige la pestaña de Oracle y se rellena los campos con los datos de la base de datos Oracle. Es importante conectarse con un usuario que tenga derechos de creación de usuarios y otros privilegios, por lo que se conectó con el usuario SYSMAN, que cumple todas las condiciones necesarias. Además se asociará a esta conexión un repositorio de migración, clicando con el botón derecho sobre la conexión y eligiendo la opción *Associate Migration Repository*.

2. **Configurar el *driver* JDBC.** Hay que descargar el *driver* JDBC indicado para MySQL. Se descargó la versión 5.04, ya que la aplicación se había probado con ese controlador. Una vez descargado se descomprime y se extrae el archivo jar. Después, se tiene que configurar la herramienta para que sepa donde se encuentra este controlador. Para ello elegimos *Tools -> Preferences…*, expandimos la opción *Database* en el árbol a mano izquierda, clicamos en *Third Party JDBC*

Drivers, después en *Add Entry*... y ponemos la dirección donde está alojado el *driver*.

3. **Añadir la conexión *third party* de la base de datos que queremos migrar, MySQL.** Para ello, repetimos el paso 1 pero se elige ahora la pestaña de MySQL y se rellenan los campos con los datos de dicha base de datos.

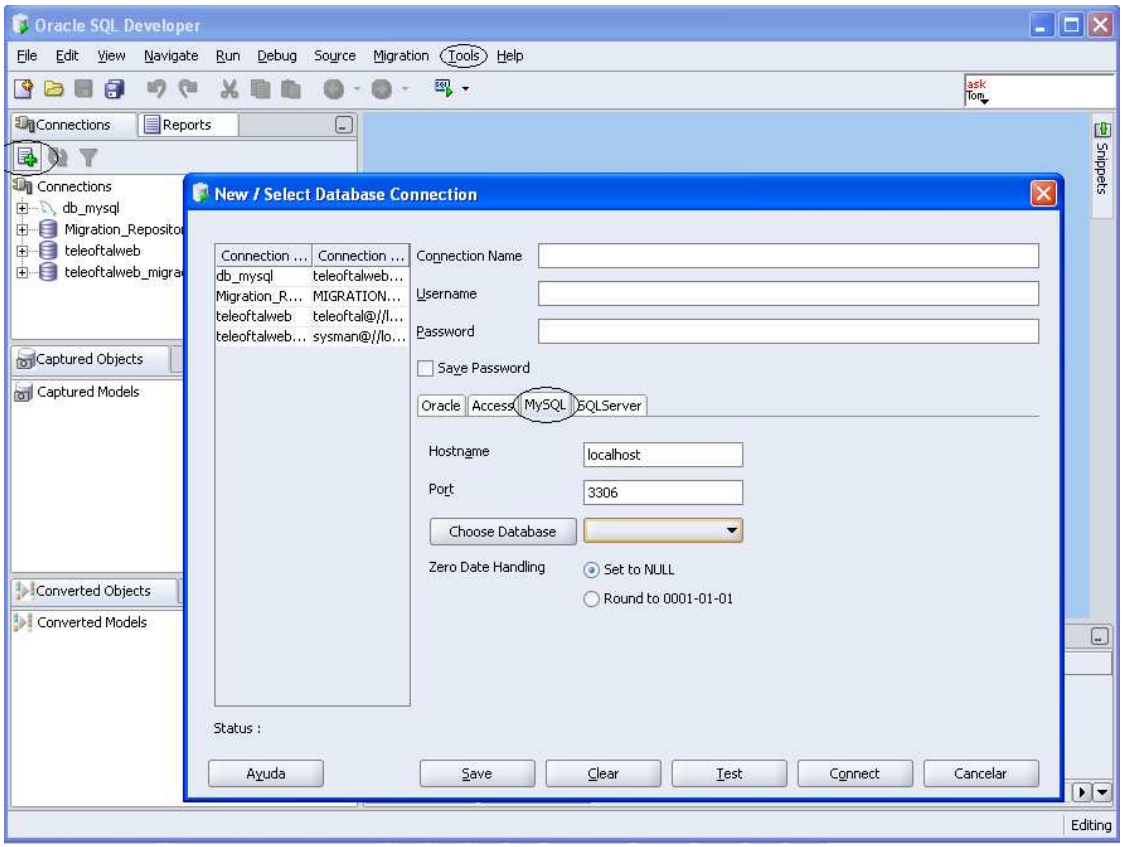


Figura 62. Pantalla de la herramienta (Paso I). Fuente: propia.

En la figura 62 están marcadas con un óvalo las pestañas indicadas en los pasos anteriores. La pantalla principal es la pantalla que aparece al añadir una conexión nueva (después de clicar en el botón + verde) y donde introduciremos los datos de MySQL (paso 3). Para introducir la conexión a Oracle se elegiría la pestaña Oracle.

II. Migración.

1. **Capturar la BD.** Es el primer paso en el proceso de migración. Se usa para capturar el estado actual de la base de datos *third party* y es necesario para

proporcionar a la herramienta una "vista en el tiempo" de la base de datos. Para capturar la base de datos MySQL se hace doble click en la conexión a MySQL creada en el apartado anterior para conectar con la base de datos y después se clica en la conexión con el botón derecho del ratón y se elige el item *Capture MySQL* del menú desplegado. La herramienta comenzará a capturar información de la BD y saldrá una pantalla mostrando las actividades actuales durante el proceso. Una vez completado, la base de datos capturada se muestra en la ventana *Captured Objects*.

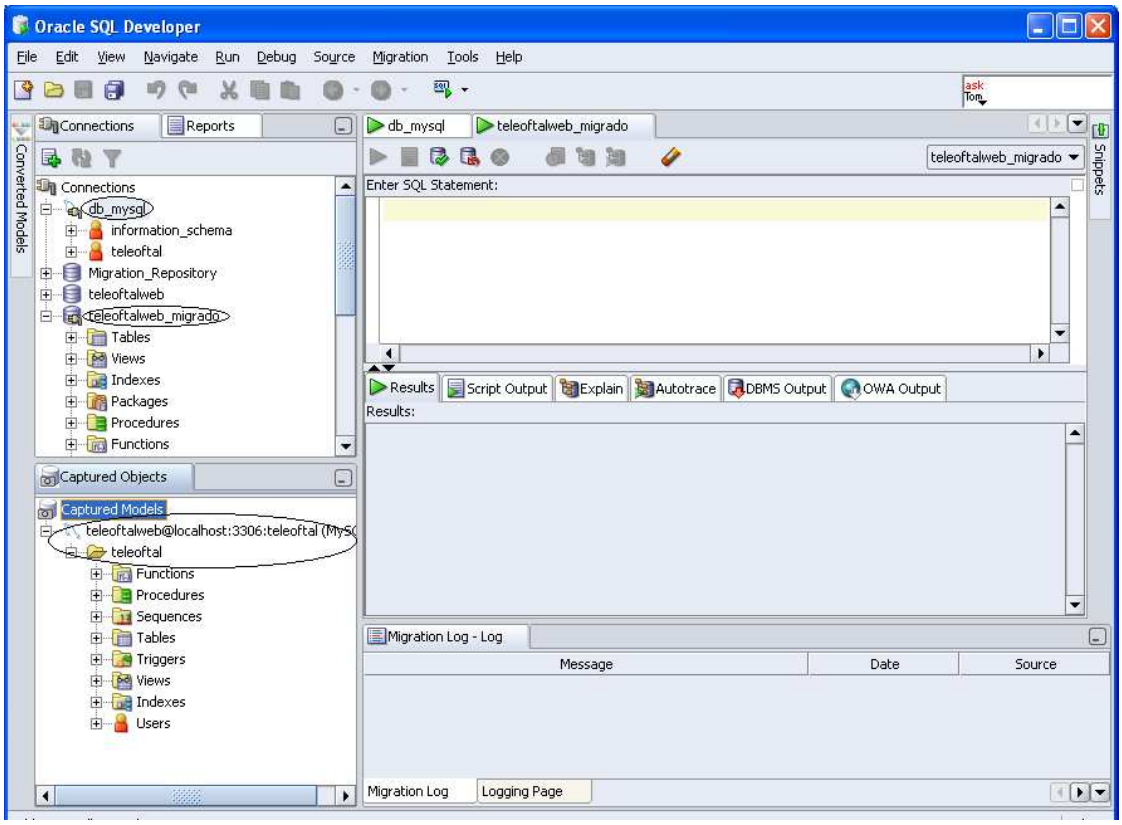


Figura 63. Pantalla de la herramienta (Paso II-1). Fuente: propia.

En la figura 63 están señaladas las conexiones a MySQL (la primera) y a Oracle (la segunda) que hemos llamado **TeleOftalWeb**_*migrado*. Asimismo, se ve en la pestaña *Captured Objects* el modelo capturado.

2. **Convertir el modelo capturado.** El siguiente paso en el proceso de migración es convertir el modelo de la base de datos MySQL en un modelo específico de Oracle. El modelo capturado contiene tipos de datos, nombres de esquemas, etc. definidos por la base de datos MySQL que tienen que ser convertidos a Oracle.

Para convertir el modelo capturado hay que navegar a la pestaña *Captured Objects*, seleccionar y clicar con el botón derecho del ratón en el nodo creado debajo de *Captured Models* y elegir *Convert to Oracle*. Una vez hecho esto aparece un diálogo de conversión de los tipos de datos que permite especificar las transformaciones de los tipos que son requeridas para convertir los datos específicos de MySQL a Oracle. Se aceptan los valores por defecto (en la mayoría de los casos es lo adecuado), aunque se pueden cambiar, y se presiona *Apply*. Este proceso lleva a cabo las transformaciones en el modelo de datos.

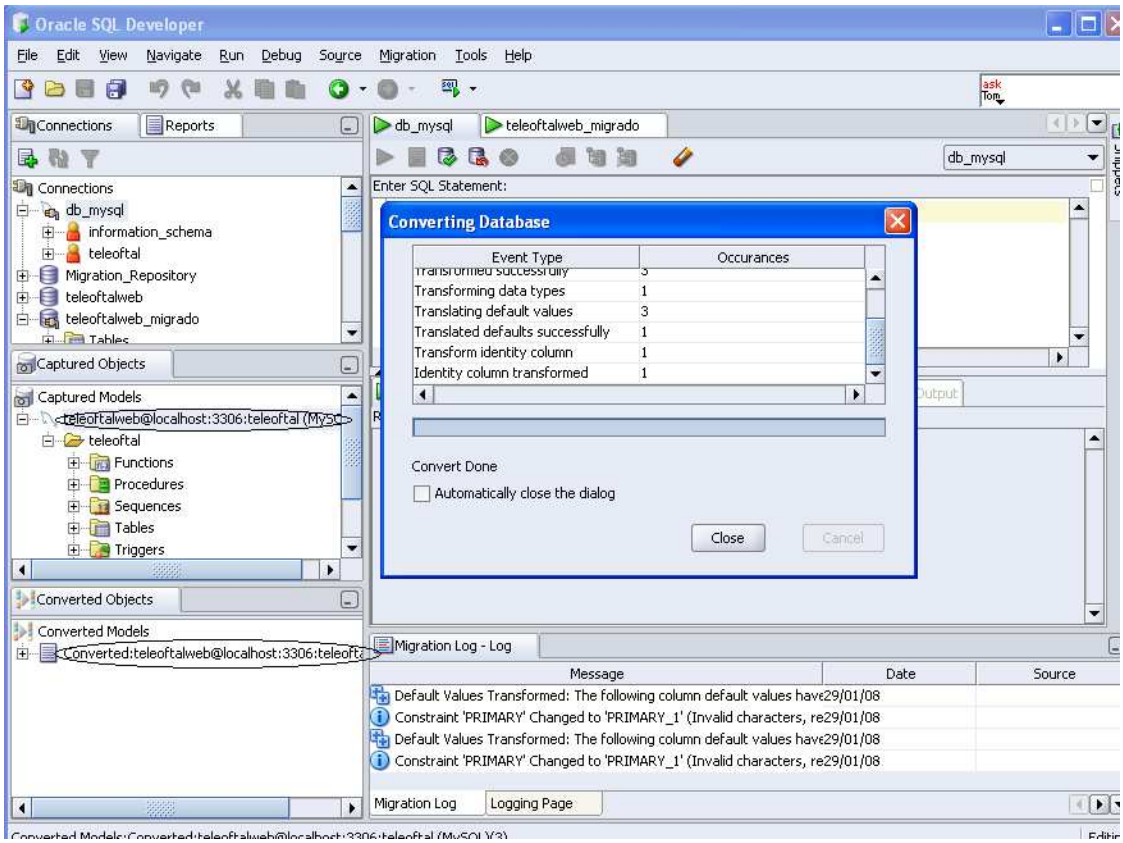


Figura 64. Pantalla de la herramienta (Paso II-2). Fuente: propia.

En la figura 64 está marcado en primer lugar el modelo capturado que hay que convertir y, en segundo lugar, el modelo convertido. Se ve en la pantalla central el proceso llevado a cabo.

3. **Generar scripts SQL para el modelo convertido.** Una vez el proceso de conversión ha sido completado, la herramienta tiene un modelo de cómo es la

base de datos convertida, el cual se usará para generar un *script* SQL para la creación del esquema de la base de datos. Para generar este *script* hay que navegar hasta la pestaña *Converted Objects* (debajo de la pestaña *Captured Objects*), seleccionar y clicar con el botón derecho del ratón en el modelo convertido debajo de *Converted Models* y dar a *Generate*. La herramienta genera las sentencias SQL necesarias para crear el esquema de la base de datos y, después de cerrar la pantalla que aparece será abierta una ventana con las sentencias SQL generadas.

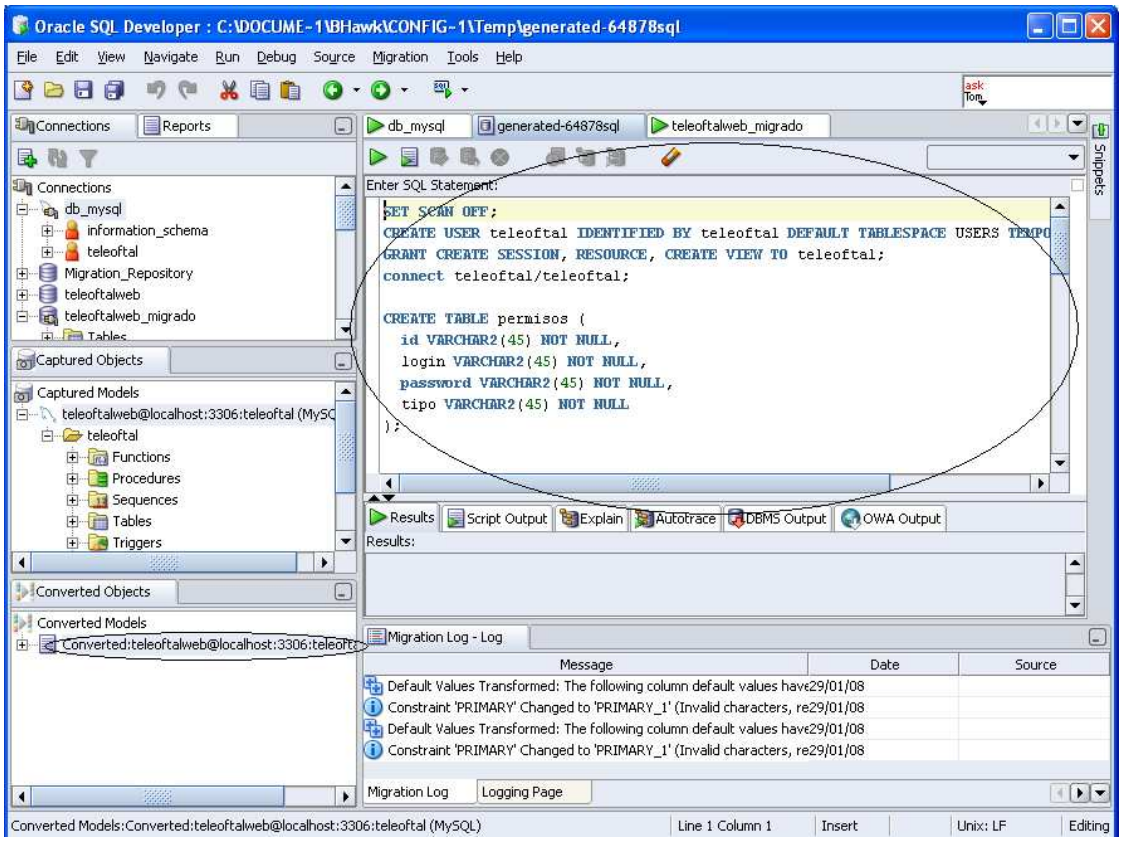


Figura 65. Pantalla de la herramienta (Paso II-3). Fuente: propia.

En la figura 65 se muestra el modelo convertido del cual hay que generar el *script* y el *script* generado. El *script* completo se muestra a continuación.

```
SET SCAN OFF;
CREATE USER teleoftal IDENTIFIED BY teleoftal DEFAULT
TABLESPACE USERS TEMPORARY TABLESPACE TEMP;
GRANT CREATE SESSION, RESOURCE, CREATE VIEW TO teleoftal;
```

```
      CREATE USER test IDENTIFIED BY test DEFAULT TABLESPACE
USERS TEMPORARY TABLESPACE TEMP;
      GRANT CREATE SESSION, RESOURCE, CREATE VIEW TO test;
      connect teleoftal/teleoftal;

      CREATE TABLE permisos (
        id VARCHAR2(45) NOT NULL,
        login VARCHAR2(45) NOT NULL,
        password VARCHAR2(45) NOT NULL,
        tipo VARCHAR2(45) NOT NULL
      );

      ALTER TABLE permisos
      ADD CONSTRAINT PRIMARY PRIMARY KEY
      (
        id
      )
      ENABLE
      ;
      GRANT SELECT, INSERT, DELETE, UPDATE, REFERENCES ON
permisos TO test;

      CREATE TABLE usuarios (
        id VARCHAR2(45) NOT NULL,
        nombre VARCHAR2(45) NOT NULL,
        apellidos VARCHAR2(45) NOT NULL,
        calle VARCHAR2(45),
        fechaultacceso DATE DEFAULT '9999-01-01 00:01:01',
        telefono VARCHAR2(45) NOT NULL,
        email VARCHAR2(45) NOT NULL,
        fechanacimiento VARCHAR2(45),
        especialidad VARCHAR2(45),
        numcolegiado VARCHAR2(45) NOT NULL,
        fechaprimacceso DATE DEFAULT '9999-01-01 00:01:01',
        numportal VARCHAR2(45),
        piso VARCHAR2(45),
        poblacion VARCHAR2(45),
        provincia VARCHAR2(45),
        codpostal VARCHAR2(45)
      );
```

```
      ALTER TABLE usuarios
      ADD CONSTRAINT PRIMARY_1 PRIMARY KEY
      (
        id
      )
      ENABLE
      ;
      GRANT SELECT, INSERT, DELETE, UPDATE, REFERENCES ON
usuarios TO test;

      connect test/test;
      connect teleoftal/teleoftal;
      connect test/test;
```

Se puede comprobar cómo se crea el esquema de base de datos con el usuario *teleoftal* y la contraseña *teleoftal* al cual se le da ciertos privilegios y se le asigna el *tablespace USERS*. También se crean las tablas *PERMISOS* y *USUARIOS* tal y como se encontraban en la base de datos MySQL.

4. **Ejecutar el *script* generado.** Para crear la base de datos es necesario ejecutar el *script* generado. Este *script* es abierto en una ventana que tiene la capacidad de ejecutar *scripts* SQL. Como se ha visto, el *script* empieza creando un usuario para el esquema de bases de datos generado, por lo que es necesario que el usuario de Oracle desde donde se va a ejecutar el *script* tenga derechos de creación de usuario, que, como se dijo en el paso 1, así es. Se comprueba el *script* para asegurase de que es correcto y se ejecuta.

   Se pueden hacer cambios manuales y en nuestro caso se hicieron algunos, ya que el formato del tipo de dato fecha de MySQL es distinto al de Oracle. Como se observa en el *script* hay dos campos que usan tipo de dato *DATE* (fecha) y tienen un valor por defecto que son *fechaultacceso* y *fechaprimacceso*. El formato *DATE* de Oracle es "DD/MON/YY" (día/mes/año) mientras que estamos introduciendo una fecha del tipo "YYYY-MM-DD HH:MI:SS" (año-mes-día horas:minutos:segundos). Como los formatos son distintos, se tuvo que hacer la siguiente modificación en el *script*:

```
...
fechaultacceso TIMESTAMP DEFAULT '9999/01/01 00:01:01',
...
fechaprimacceso TIMESTAMP DEFAULT '9999/01/01 00:01:01',
...
```

Se cambia *DATE* por *TIMESTAMP* porque este último guarda un momento en el tiempo con horas, minutos y segundos (como se vió en el capítulo 5). Además hay que escribir la siguiente sentencia antes de la creación de la tabla *USUARIOS* para modificar el formato de *TIMESTAMP* de modo que sirva este formato para el valor por defecto:

```
ALTER SESSION SET NLS_TIMESTAMP_FORMAT = 'DD/MM/YYYY
HH24:MI:SS';
```

El formato natural de *TIMESTAMP* es 'DD/MON/YY HH24:MI:SSXFF TZR' donde HH24 son las horas (de 0 a 24), SS son los segundos, FF son las fracciones de segundo y TZR es la zona horaria. X sirve para imprimir las fracciones de segundo .

Con estos cambios ya se puede ejecutar el *script*, para lo cual se elige en la lista de la esquina de arriba a la derecha de la ventana SQL la conexión en la que se ejecutará el *script* (la de Oracle) y se presiona F5 para ejecutarlo. Una vez ejecutado se comprueba que no ha habido ningún fallo. Si no ha habido ninguno, ya se habrán construido los objetos de la base de datos MySQL en Oracle, incluidas las tablas.

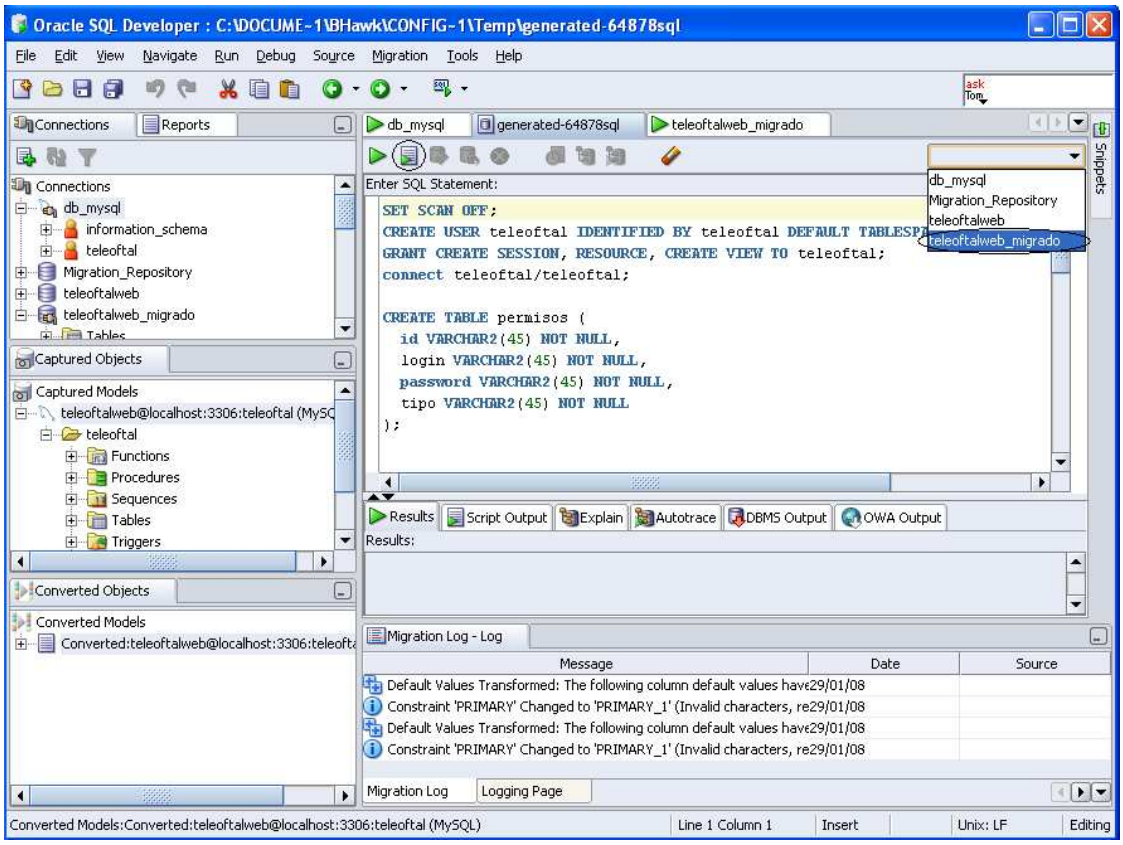


Figura 66. Pantalla de la herramienta (Paso II-4). Fuente: propia.

En la figura 66 están marcadas la conexión a Oracle que se usa para ejecutar el *script* (**TeleOftalWeb**_*migrado*) arriba a la derecha y el botón de ejecución del *script* (en vez de pulsar F5).

5. **Migrar los datos a Oracle.** Esta herramienta no sólo crea las tablas correspondientes a las de MySQL, sino que además migra los datos que teníamos en MySQL a estas nuevas tablas en Oracle. Para ello es necesario crear una conexión a Oracle con el nuevo esquema creado en el *script* (*teleoftal*). Creada esta conexión ya estamos preparados para migrar los datos: desde el menú *Migration* elegimos *Migrate Data*, lo que invocará una ventana que nos pedirá la información necesaria para migrar los datos. En el campo *Source Connection* introducimos la conexión a MySQL, en el campo *Target Connection* introducimos la conexión creada teleoftal y en el campo *Converted Model* se selecciona el modelo convertido creado en el paso 2. Se da a *OK* y comienza el proceso de migración. Una vez acabado ya tenemos los datos almacenados en MySQL en Oracle.

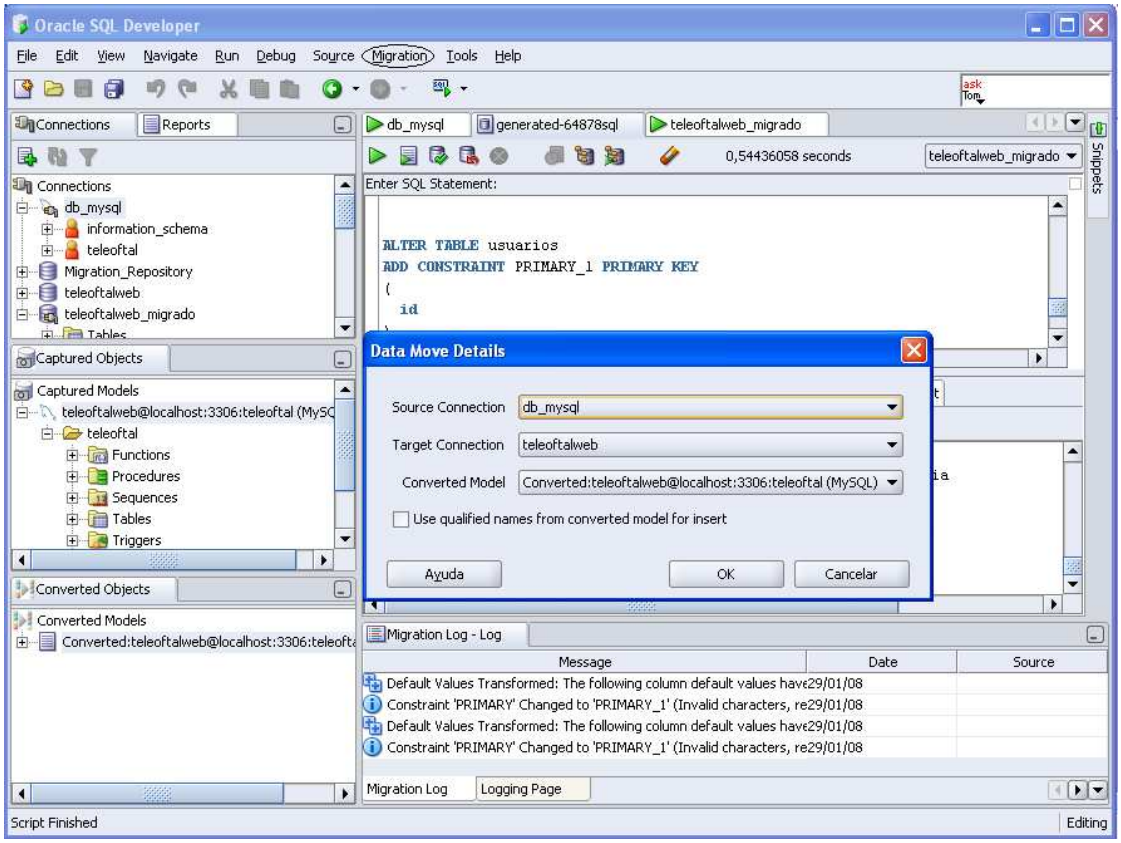


Figura 67. Pantalla de la herramienta (Paso II-5). Fuente: propia.

En la figura 67 se muestra el menú *Migration* desde donde se accede a la opción *Migrate Data* que abre la ventana central mostrada en la figura, en la cual se introduce la información indicada en el paso 5.

Adaptación de la aplicación a Oracle (referente a MySQL)

Una vez migradas las tablas con sus datos a Oracle, el siguiente paso es reescribir el código necesario de la aplicación para que pueda llevar a cabo las mismas funciones que se realizaban con MySQL. Para ello se deben llevar a cabo dos acciones. La primera es reescribir la clase Java *BDmysql*, que se encargaba de realizar las consultas en la base de datos MySQL, así como la inserción, modificación y borrado de datos. La siguiente acción consiste en modificar en todas las JSPs y clases Java que llamen a la clase MySQL estas llamadas, ya que se cambió el nombre de la clase *BDmysql* a *BDoracle*.

La única modificación esencial que hay que hacer en la clase *BDmysql* (a partir de aquí la llamaremos ya *BDoracle*) es cambiar la conexión de la base de datos a

Oracle. Para ello definimos un *Data Source*, *OracleDataSource*, en el se introduce la URL de la base de datos, y el nombre y la contraseña *teleoftal*, ya que queremos acceder a las tablas guardadas en este esquema. El resto de métodos o se dejaron tal cual o las modificaciones fueron mínimas. En el apéndice C se muestra todo el código generado correspondiente a esta clase. A continuación explicamos brevemente los atributos y los métodos de esta clase.

Atributos

- **Connection connection.** Almacena la conexión con la BD Oracle recuperada del datasource OracleDataSource.
- **private bolean connectionFree.** Variable lógica que almacena si la conexión con la base de datos está libre o no.
- **private ArrayList usuarios.** Array que almacena el listado de usuarios del sistema recuperado de la base de datos, con todos los datos de usuario (campos de la tabla USUARIOS).

Métodos

- **public BDoracle ().** Constructor de la clase que abre el datasource y solicita una conexión con la base de datos, que almacenará en el atributo connection.
- **protected synchronized Connection getConnection ().** Recupera y devuelve la conexión connection con la base de datos si está libre, y espera por ella en caso contrario.
- **protected synchronized Connection releaseConnection ().** Libera la conexión con la BD connection una vez que se han terminado de realizar las peticiones necesarias de un momento dado.
- **public Usuario getUsuario (String id).** Extrae los datos del usuario con identificador id de la base de datos y los devuelve en un objeto del tipo Usuario. Este método sólo extrae los datos de la tabla USUARIOS.
- **public int insertarUsuario (Usuario usuario).** Inserta el usuario usuario en la base de datos. Este método inserta todos los datos del usuario, los de la tabla USUARIOS y los de la tabla PERMISOS. Devuelve el número de filas afectadas si se ha insertado el usuario ó 0 si se ha producido un error.

- **public int borrarUsuario (String id).** Elimina el usuario con identificador id de la base de datos (las dos tablas). Devuelve el número de filas afectadas al borrarse el usuario o 0 si se ha producido un error.
- **public int modificarUsuario (Usuario usuario).** Actualiza los datos de las dos tablas del usuario con identificador id almacenado en la base de datos. Si se ha modificado correctamente el método devuelve el número de filas afectadas y si no ha sido así devuelve 0.
- **public Collection getUsuarios ().** Obtiene el listado de usuarios del sistema y lo devuelve en una colección de objetos Usuario que almacena en el atributo de la clase usuarios.
- **public Collection searchUsuarios (String apellidos, String numcolegiado, String id, String tipousuario).** Devuelve un listado de los usuarios del sistema que cumplen los requisitos especificados en los parámetros del método, que son los apellidos del usuario (apellidos), el número de colegiado del usuario (numcolegiado), la identidad el usuario (id) y el tipo de usuario (tipousuario). Estos parámetros de búsqueda se pueden usar por separado o con cualquier combinación entre ellos.
- **public boolean checkLogin (String login).** Comprueba si un determinado nombre de usuario (login) es válido y no está siendo ya utilizado en el sistema. Devuelve true si se puede utilizar el nombre de usuario proporcionado y false en caso contrario.
- **public String validarUsuario (String login, String password, String tipo).** Comprueba si un usuario con un determinado nombre, login, tiene la clave y tipo de acceso proporcionado, password y tipo respectivamente. Devuelve el identificador del mismo en caso afirmativo o null en caso contrario.
- **public String getNumColegiado (String id).** Recupera el número de colegiado del usuario con identificador id.
- **public String modificarFechaAcceso (String uid).** Actualiza las fechas del primer (en caso de ser la primera vez que el usuario accede a la aplicación) y último acceso del usuario con el identificador id proporcionado.
- **public void close().** Cierra la conexión abierta con la base de datos cuando se destruye el contexto de la aplicación.

Para la comprobación de cada método por separado se construyó un programa en Java que fuera realizando operaciones de tal modo que usara todos los métodos de la clase BDoracle.

Al realizar estas comprobaciones se encontró un problema: al llamar al método modificarFechaAcceso o al método insertarUsuario ocurría un error debido a que el tipo de dato TIMESTAMP de las columnas fechaprimacceso y fechaultacceso no aceptaba la fecha que se le introducía (daba error de mes no válido, entre otros), por lo que después de muchas pruebas se decidió borrar estas dos columnas y volver a crearlas de tipo VARCHAR2 con el valor por defecto "9999-01-01 00:01:01". Una vez hecho esto, se solucionó el problema.

Después de comprobar el correcto funcionamiento de la clase *BDoracle* en el programa desarrollado por el alumno, se procedió a su implantación en el servidor TomCat, para lo cual, como ya se ha dicho, se cambió todas las llamadas a *BDmysql* por llamadas a *BDoracle* en las JSPs y en la clase Java *ContextListener*, que es la clase que realiza la función de *listener* ("oyente") de la base de datos en la aplicación; y se introdujo la nueva clase creada en el directorio que contiene las clases y las nuevas librerías usadas en el directorio de las librerías. Ya estaba todo dispuesto para probar la clase *BDoracle* en la aplicación.

Después de probarla se encontró un nuevo problema: si al introducir un usuario se dejaba un campo en blanco como, por ejemplo, la calle o la ciudad, después, cuando se abría este usuario en la aplicación, aparecía en este campo la palabra *null*. Esto no era algo deseado, con lo que se usó la función SQL soportada por Oracle NVL para solucionarlo. Esta función tiene el siguiente formato:

NVL(*valor_introducido*, *valor_si_nulo*)

y devuelve *valor_introducido* si este valor no es nulo, o *valor_si_nulo* en el caso de que *valor_introducido* sea nulo. De este modo se usó esta función en los métodos *insertarUsuario* y *modificarUsuario* de la siguiente manera:

```
      PreparedStatement preparedStatement =
this.connection.prepareStatement
      ("INSERT INTO usuarios (id, nombre, apellidos, telefono, email,
calle, numportal, piso, poblacion, provincia, codpostal,
```

```
fechanacimiento, especialidad, numcolegiado) VALUES (?, ?, ?, ?, ?,
NVL(?,' '), NVL(?,' '), NVL(?,' '), NVL(?,' '), NVL(?,' '), NVL(?,'
'), ?, ?, ?)");
```

Esto quiere decir que si el dato a introducir es nulo, entonces guardamos un espacio en el campo correspondiente de la tabla *USUARIOS*. Esta función sólo se usa con los datos que pueden ser nulos, como se observa en el ejemplo anterior.

Como se ha podido comprobar, el proceso completo de migrar una base de datos MySQL a Oracle es bastante sencillo. Esto no es debido solamente a la herramienta de migración de Oracle, sino a que ambas bases de datos son muy parecidas: ambas son relacionales y usan el mismo lenguaje SQL (al menos en nuestro caso). Aunque la herramienta de migración ayuda, ésta se podría haber llevado a cabo manualmente, creando las tablas y el esquema necesarios en primera instancia y migrando los datos de una base de datos a otra, construyendo un programa Java sencillo, en segundo lugar.

Adaptación de la aplicación a Oracle (referente a la BD nativa XML)

Una vez comprobado el correcto funcionamiento de la aplicación tras la migración de MySQL a Oracle, el siguiente paso es migrar la base de datos dbXML a Oracle.

En esta caso, las dos bases de datos son de naturaleza totalmente distinta: dbXML es una base de datos nativa XML de código libre mientras que Oracle es una base de datos relacional propietaria con ciertas características orientadas a objeto. Está claro que en este caso la migración no va a ser, ni mucho menos, tan sencilla como en el caso anterior.

Para esta nueva migración no existe ninguna herramienta de Oracle que la haga de un modo “automático”, no tiene sentido porque ahora se trata de migrar documentos XML, no tablas relacionales. Además, la base de datos de Oracle no fue diseñada para guardar documentos XML como tales y poder realizar búsquedas en estos documentos, sino que se le ha añadido esta funcionalidad posteriormente. Por ello la nueva tabla podría ser simplemente una columna de tipo XMLType, cuya creación no tiene misterio. Por todo esto, no hay más remedio que hacerlo de modo “manual”. Para ello hay que adaptar primero el código de la aplicación para poder introducir historiales y consultarlos en Oracle tal como se hacía con dbXML y, después, intentar migrar los

historiales existentes en la BD dbXML a la de Oracle. Es decir, se invierte el orden de los pasos de la migración de MySQL.

De todos modos, antes de cambiar el código, hay que crear una tabla que guarde estos historiales. Para ello se creó la tabla *HISTORIALES* con dos columnas, *key_column* y *XML_column* que guardan, respectivamente, el identificador del historial y el documento XML que compone el historial. Esta tabla se vio con más detalle en el capítulo 5.

Este es el paso más crítico del presente proyecto y también el más complicado y laborioso. Requiere cambiar todas las consultas y la mayor parte del código de la clase que se encarga de todo lo referente al almacenamiento, modificación, eliminación y consultas de los historiales clínicos, *BDxml*. Esta clase es la clase que trata directamente con la base de datos dbXML. Como se ha comentado, es una base de datos nativa cuyo lenguaje de consultas es el *XUpdate*. Sin embargo, en el capítulo 5 se vio que Oracle tiene sus propias funciones de consulta. *XUpdate* y estas funciones específicas de Oracle de consulta de datos XMLType no tienen nada en común excepto la utilización de XPath para la señalización de los nodos. Aún así hubo que cambiar la forma de escribirlos, ya que Oracle no lee comillas simples cuando lee XPath. Es decir, si en el XPath hay una expresión como la que sigue:

```
...text()='Revisiones'...
```

se cambió de la siguiente manera para que Oracle lo entendiera:

```
...text()=\"Revisiones\"...
```

Por tanto, todas las comillas simples de los XPath se cambiaron por comillas dobles. A excepción de esto, en la mayoría de los casos estas expresiones XPath servían tal cual, en otros se tuvieron que cambiar.

Para ayudar en la modificación de esta clase (a la cual no se cambió el nombre) se creó un programa en Java, como en el caso anterior, para comprobar el correcto funcionamiento de los métodos que se iban cambiando.

A continuación analizaremos los métodos por separado de forma breve, con sus cometidos y funciones. Estos métodos se encuentran en la clase *BDoracle* del apéndice

C; están en esta clase, porque, se juntó el código de las dos clases iniciales *BDoracle* y *BDxml* en una sola, *BDoracle*.

- Los atributos de la clase, el constructor y sus métodos de conexión y desconexión son los mismos que las de la clase BDoracle, excepto el atributo de tipo ArrayList usuarios. Los métodos de conexión son **getConnection**() (aparte del constructor de la clase **BDxml()**) y los de desconexión son **releaseConnection**() y **close**(), como en el caso anterior.
- **public void insertNewDocument (String idHistorial, String document).** Inserta un documento nuevo en la BD. Recibe dos parámetros correspondientes a las dos columnas de la tabla HISTORIALES, el identificador del historial (idHistorial) y el historial en sí (document). Para la creación del XMLType se usa la clase XMLType incluida en la librería oracle.xdb y su método createXML que toma como parámetros la conexión de la base de datos y el documento que será convertido a tipo XMLType, document. La inserción se hace mediante una sentencia SQL INSERT INTO, con lo que no es necesario usar ninguna función de XMLType.
- **public int substituteDocument (String idHistorial, String document).** Sustituye un documento anteriormente almacenado correspondiente a idHistorial por document. Como en el método anterior, hay que usar el método de XMLType createXML para convertir document en XMLType. Se usa una sentencia SQL UPDATE sin necesitar ninguna función de XMLType.
- **public int eliminarHistorial (String idHistorial).** Este método elimina el historial almacenado correspondiente a idHistorial. Borra toda la fila de la tabla usando una sentencia SQL DELETE.
- **public int insertNewBloc (String idHistorial, String Xpath, String document, int posicion).** Inserta un nuevo bloque dentro de un historial almacenado en la base de datos. idHistorial indica el historial en donde se insertará el bloque, Xpath la posición donde se inserta, document es el bloque y posición es el número que indica que parte de la revisión a introducir es. En un principio hay tres partes (tres bloques):

1. Bloque 1. Corresponde a la primera parte de la revisión. Por ello se usa la función *appendChildXML* en la sentencia SQL, para introducir la nueva revisión al final de la sección del historial *Revisiones*.
2. Bloque 2. Corresponde a la parte intermedia de la revisión. Se usa la función *updateXML* para cambiar el bloque por el comentario que indica donde hay que introducir la segunda parte (final de la primera parte de la revisión).
3. Bloque 3. Corresponde a la última parte de la revisión. Como en el bloque 2 se usa *updateXML* para cambiar el bloque por el comentario que indica donde introducir la última parte de la revisión…

- **public Collection getHistoriales (String numColegiado).** Obtiene los historiales que son visibles para el usuario (de los que sea propietario o compartido) cuyo número de colegiado sea numColegiado. Para ello se hace una consulta de varios datos del historial, con lo que, aparte de utilizar la función extractValue, se hace uso de la función XMLSecuence, ya que se obtendrán datos de distintos historiales y queremos almacenar cada uno de forma independiente. Este método devuelve una colección de los historiales requeridos con los datos consultados.
- **public Collection searchHistoriales (String numColegiado, String apellidos, String dni, String fechaCreacion, String origen).** Este método realiza una búsqueda selectiva de los historiales visibles para el usuario con número de colegiado numColegiado. La búsqueda se puede realizar buscando por los apellidos del paciente (apellidos), por el dni del mismo (*dni*), por la fecha de creación del historial (*fechaCreacion*), o por el origen del historial (*origen*), es decir, si el historial es propiedad del médico o es compartido por otro. La búsqueda puede ser de un solo parámetro o de cualquier combinación de ellos. Este método es parecido al anterior pero usando más condiciones en la función *existsNode*. También es necesario usar *XMLSecuence* con *extractValue*. Devuelve una colección con los historiales resultado de la búsqueda.
- **public boolean checkPermisos (String numColegiado, String idHist, String permisos).** Comprueba que el usuario con número de colegiado numColegiado tiene determinados permisos para el historial con identificador idHist. El parámetro permisos puede ser owner o shared, dependiendo de si se quiere

comprobar que el usuario es propietario del historial o, al contrario, este es compartido por otro usuario. En este método se hace uso de la función extractValue.

- **public int borrarHistorial (String numColegiado, String idHistorial).** Borra el historial idHistorial de forma lógica a la vista del usuario con número de colegiado numColegiado, siempre y cuando el usuario no sea propietario del historial. Es decir, si antes de realizar este método el usuario podía ver un historial por ser compartido por otro, después de realizarlo el usuario ya no puede verlo, pero el historial no se ha borrado de la base de datos, simplemente ha dejado de ser compartido por el usuario propietario del historial y, por tanto, ha dejado de ser visible al usuario en cuestión. Se usa la función deleteXML para borrar el usuario del nodo compartidos. Este método devuelve las filas afectadas por la operación si se ha borrado el historial o -1 si se ha producido un error.
- **public int compartirHistorial (String numColegiadoOrigen, String numColegiadoCompartir, String idHistorial).** Comparte el historial idHistorial del usuario con número de colegiado numColegiadoOrigen con el usuario de número numColegiadoCompartir. Para ellos se hace uso de la función appendChildXML, que se usa para poner el usuario con el que se comparte el historial al final del nodo compartidos. Devuelve las filas afectadas por la operación o -2 si se ha producido algún error.
- **public HashMap getEstadisticas (String numColegiado).** Obtiene las estadísticas sobre los historiales del usuario con número de colegiado numColegiado. Estas estadísticas son el número de historiales pertenecientes al usuario, el número de historiales que comparte con otros médicos y el número de historiales que son compartidos por otros. Se usa la función extractValue para extraer el valor del atributo tipo del usuario del nodo compartidos (owner o shared). Devuelve un HashMap con los números requeridos.
- **public Filiacion getFiliacion (String idHistorial, String numColegiado, String fechaRevision).** Obtiene y procesa los datos de filiación de una revisión creada en la fecha fechaRevision del historial idHistorial introducida por el usuario con número de colegiado numColegiado. Se usa la función extract para extraer la parte de la revisión correspondiente a la filiación. Como son varios

nodos no se puede usar extractValue. Este documento devuelto se convierte a tipo XMLType para poder obtener el DOM de él, usando el método getDOM() de la clase XMLType. Una vez obtenido el DOM del documento ya se puede llamar al método ParseFiliacion de la clase ParseXML, para analizar y extraer los valores de los nodos de este documento. Devuelve estos valores en un objeto Filiacion, que se encarga de almacenar de una manera ordenada todos los valores extraídos.

- **public String getDiagnosticos (String idHistorial, String numColegiado, String fechaRevision).** Obtiene los diagnósticos anteriores a la revisión creada en la fecha fechaRevision del historial idHistorial cuyo propietario es el usuario con número de colegiado numColegiado. Es necesario usar extractValue junto con XMLSecuence porque puede haber varias revisiones anteriores a la dada y, por tanto, varios diagnósticos que hay que almacenar de forma independiente. Devuelve un String con los diagnósticos.
- **public MotivoConsulta getMotivoConsulta (String idHistorial, String numColegiado, String fechaRevision).** Obtiene y procesa los datos del motivo de la consulta de la revisión de fecha fechaRevision del historial idHistorial perteneciente al usuario con número de colegiado numColegiado. Como en el caso de getFiliacion, se extrae con extract una parte del documento XML de la que se obtiene el DOM para poder analizar sus nodos mediante el método ParseMotivoConsulta de ParseXML. Se devuelve el objeto MotivoConsulta con los valores requeridos.
- **public APOftalmologicos getAPOftalmologicos (String idHistorial, String numColegiado, String fechaRevision).** Obtiene y procesa los datos de antecedentes personales oftalmológicos de la revisión de fecha fechaCreación dentro del historial idHistorial del usuario con número de colegiado numColegiado. Se extrae con extract la parte de la revisión referente a estos datos, se obtiene el DOM y se procesa el documento con el método ParseAPOftalmologicos. Este método devuelve el objeto APOftalmologicos con los valores extraídos.
- **public APGenerales getAPGenerales (String idHistorial, String numColegiado, String fechaRevision).** Obtiene y procesa los datos de antecedentes personales generales de la revisión de fecha fechaRevision del

historial idHistorial cuyo propietario es el usuario con número de colegiado numColegiado. Se extrae la parte de la revisión requerida con extract, se obtiene el DOM y se analiza con el método de ParseXML ParseAPGenerales para devolver el objeto APGenerales con los datos necesarios extraídos del documento.

- **public AFamiliares getAFamiliares (String idHistorial, String numColegiado, String fechaRevision).** Obtiene y procesa los datos de antecedentes familiares de la revisión de fecha fechaCreacion del historial idHistorial del usuario con número de colegiado numColegiado. Como en los casos anteriores, se extrae la parte de la revisión deseada con extract, se obtiene el DOM de esta parte y se analiza con ParseAFamiliares. Este método devuelve el objeto AFamiliares con los valores extraídos.
- **public Exploracion getExploracion (String idHistorial, String numColegiado, String fechaRevision).** Obtiene y procesa los datos de exploración de la revisión creada en la fecha fechaRevision del historial idHistorial cuyo propietario es el usuario con número de colegiado numColegiado. Se extrae la parte de la revisión referente a la exploración con extract, se obtiene el DOM y se analiza con el método ParseExploracion para devolver el objeto Exploracion con los datos extraídos del documento.
- **public Complementa getComplementa (String idHistorial, String numColegiado, String fechaRevision).** Obtiene y procesa los datos complementarios de la revisión de fecha fechaCreacion del historial idHistorial del usuario cuyo número de colegiado es numColegiado. Para ello, se extrae la parte de la revisión requerida con la función extract, se obtiene el DOM y se procesa con el método ParseComplementa. Este método devuelve el objeto Complementa con los valores obtenidos.
- **public int insertarImagen (String numColegiadoOrigen, String idHistorial, String nombreImagen,String fechaCreacion, String descripcionImagen, String imagenOriginalB64,String imagenModificadaB64).** Inserta una imagen en el historial idHistorial perteneciente al usuario cuyo número de colegiado es numColegiadoOrigen. La imagen se introduce codificada en base 64 y lo que realmente se está insertando es un documento XML en el que se introducen los datos que son los parámetros del método: el nombre de la imagen

nombreImagen, la fecha de creación de la imagen fechaCreacion, la descripción de la imagen descripcionImagen, la imagen original codificada en base 64 imagenOriginalB64 y la imagen modificada codificada en base 64 imagenModificadaB64. Aunque esta era la primera idea, hubo un problema que se verá más adelante que nos obligó a introducir las imágenes codificadas en una tabla aparte. Este método hace uso de la función appendChildXML para introducir el documento XML referente a las imágenes (creado con el método de XMLType createXML) en último lugar en la sección Imágenes. El método devuelve las filas afectadas si todo ha ido bien o -2 si ha habido algún error.

- **public int borrarImagen (String numColegiado, String idHistorial, String nombreImagen.:** Elimina la imagen nombreImagen del historial idHistorial cuyo propietario es el usuario con número de colegiado numColegiado. Hace uso de la función deleteXML para borrar todo la parte del historial referente a esa imagen. Devuelve las filas afectadas si no se ha producido ningún error o -1 en caso contrario.
- **public Collection searchImagenes (String numColegiado, String idHistorial, String nombreImagen, String fechaCreacion, String diagnosticoImagen).** Busca imágenes dentro del historial idHistorial del usuario con número de colegiado numColegiado. Los parámetros de búsqueda son el resto de parámetros del método: el nombre de la imagen (nombreImagen), la fecha de creación de la imagen (fechaCreacion), o la descripción de la imagen (diagnosticoImagen). Estos parámetros pueden ser usados separados o con cualquier combinación entre ellos. Este método usa la función extractValue en combinación con XMLSequence para almacenar cada dato extraído por separado y poder almacenarlos correctamente en la colección que se devuelve.
- **public String getImagenenNombre (String idHistorial, String nombreImagen).** Obtiene una imagen original del historial idHistorial por su nombre, nombreImagen. En un principio se usaba la función extractValue, pero después de un problema con el almacenamiento de imágenes no fue necesario su uso. Devuelve la imagen original codificada en base 64.
- **public Collection getImagenesOriginales (String idHistorial).** Obtiene todos los originales de las imágenes del historial idHistorial. Lleva a cabo el mismo cometido que searchImagenes pero sin condiciones de búsqueda. Por esto

mismo se usan las funciones extractValue junto con XMLSequence. Devuelve una colección con todos los datos de las imágenes requeridos.

- **public int actualizaImagenEditada (String numColegiado, String idHistorial, String nombreImagen, String imagenB64).** Actualiza la imagen editada, imagenB64, de la imagen nombreImagen del historial idHistorial del usuario con número de colegiado numColegiado. Antes del problema comentado en getImagenenNombre y en insertarImagen, este método usaba la función updateXML para cambiar la imagen modificada por la nueva edición de la imagen. Sin embargo, después se dejó de usar. Devuelve las filas afectadas si todo ha ido bien o -1 en caso de que haya habido algún error.
- **public String getXML (String document).** Extrae el historial XML completo document y lo almacena en una variable de tipo String. Hace uso de la función extract para la extracción del documento junto con el método createXML de la clase XMLType para poder obtener el documento XML con su formato.
- **public String getImagenModificada (String idHistorial, String nombreImagen).** Método que extrae de la base de datos la imagen modificada cuyo nombre es nombreImagen del historial idHistorial. Devuelve la imagen modificada en una variable de tipo String.
- **public Collection getParticipantes(String idHistorial).** Obtiene todos los participantes del historial idHistorial. Para ello hace uso de la función extractValue junto con XMLSecuence, para extraer los participantes del nodo compartidos y almacenarlos virtualmente de forma independiente. Este método devuelve una colección con los participantes y sus permisos (si es propietario o compartido).
- **public Collection getRevisiones (String numColegiado, String idHistorial).** Obtiene las fechas de las revisiones del historial idHistorial del usuario con número de colegiado numColegiado. Usa la función extractValue junto con XMLSecuence para obtener cada fecha por separado y poder introducirlas en una colección, que es lo que devuelve este método.

Si nos fijamos, hay varios métodos, como este último getRevisiones, o searchImagenes, o compartirHistorial, entre otros, que toman como parámetro el número de colegiado del usuario propietario del historial, cuando sólo sería necesario el

identificador del historial idHistorial. Esto es así porque dentro del método se llama a checkPermisos para comprobar que este usuario es realmente el propietario del historial en cuestión, ya que, de no serlo, no podría realizar modificaciones en él.

En otros métodos que también toman como parámetro el número de colegiado pero que no hacen ninguna modificación en el historial, como todos los aquellos que obtienen y procesan los datos de las revisiones (excepto getFiliacion), este parámetro sirve para comprobar que el usuario que accede al historial tiene permisos para acceder a él (es propio o compartido). Para hacer esta comprobación se usa la función existsNode con esta condición, no es necesario usar checkPermisos.

Problemas encontrados y soluciones

1. Identificador de los historiales.

En la antigua versión de **TeleOftalWeb**, los historiales que se guardaban en la base de datos dbXML tenían un identificador de historial que consistía en un número aleatorio que generaba la propia base de datos al introducir un documento en la misma. Con Oracle esto no sucede, por lo que se tuvo que generar el identificador del historial de manera "manual". Para ello se usan el número de colegiado del usuario que va a crear el historial y la fecha actual de creación. Estos dos datos componen el identificador del historial.

Al hacer esto, deja de ser necesario el método substituteDocument. Esto se debe a que antes, para introducir un historial se introducía primero una frase en la base de datos con insertNewDocument sólo para obtener el identificador del historial y, una vez obtenido, se sustituía la frase introducida por el historial en sí con substituteDocument. Ahora no es necesario introducir nada en la base de datos para obtener el identificador de historial, sino que éste es creado por la aplicación, con lo que se puede introducir directamente el historial y no es necesario por tanto el método *susbstituteDocument*.

2. Introducción de bloques de revisión.

Cuando se probó el funcionamiento del método *insertNewBloc* para introducir los bloques de una revisión, se comprobó que, a pesar de que no salía ningún error, estos bloques no eran insertados en el historial. Esto era porque en la versión anterior, estos bloques empezaban con un comentario y Oracle no procesa los documentos XML que empiezan por comentario. La solución fue simple: introducir los comentarios dentro

del primer nodo *section*. Una vez cambiado esto se consiguió introducir los bloques correctamente, excepto el 3, como se ve a continuación.

3. Introducción del bloque 3 de revisión.

Al intentar introducir el bloque 3 de una revisión ocurría un error incluso después de solucionar el problema del comentario inicial. Este error ocurría porque las función de XMLType *appendChildXML* y *updateXML* sólo sirven para introducir un nodo completo (que puede contener hijos), en nuestro caso un nodo *section*. Sin embargo, no se pueden introducir varios nodos. El bloque 3 contiene 3 nodos distintos, los referentes a *OTRAS PRUEBAS*, *JUICIO DIAGNÓSTICO* y *TRATAMIENTO Y RECOMENDACIONES*. Por esto mismo ocurría un error y no se podía introducir el último bloque. Para solucionar esto se podían llevar a cabo dos acciones: se podían introducir los nodos de *JUICIO DIAGNÓSTICO* y *TRATAMIENTO* dentro del nodo *OTRAS PRUEBAS*, con lo que sólo se introducía un solo nodo principal; o bien, se podía dividir el bloque 3 en tres bloques: bloque 3, bloque 4 y bloque 5, cada uno con cada nodo mencionado. En un principio se decidió realizar la solución menos engorrosa en lo que a cambio de código se refiere: la primera. En un principio, porque como veremos más adelante, finalmente se optaría por la segunda opción.

De todas formas, este cambio inicial se tuvo que tener en cuenta a la hora de extraer datos de un historial guardado referentes al *JUICIO DIAGNÓSTICO* y al *TRATAMIENTO*, ya que la ruta para acceder a ellos había cambiado.

4. Utilización de *XMLSequence*.

Cuando se explicaron los métodos de la clase *BDxml*, se comentó que algunos tenían que usar la función *XMLSequence* para obtener los datos por separado. Aquí explicamos de una manera más detallada por qué es necesario esto. Para comprenderlo mejor se verá esto con un método, por ejemplo *getParticipantes*.

Si se usa este método con la función *extract*, sin *XMLSecuence*, el código generado sería el siguiente:

```
      String queryNumColegiado=
      "//clinical_document_header/local_header[@descriptor=\"compartid
os\"]/local_header[@descriptor=\"person\"]/local_attr[@name=\"V\"]/@va
lue";
```

```
      String queryTipo=
      "//clinical_document_header/local_header[@descriptor=\"compartid
os\"]/local_header[@descriptor=\"person\"]/local_attr[@name=\"type\"]/
@value";

      String query = "SELECT
      x.XML_Column.extract('"+queryNumColegiado+"').getStringVal(), "+
      "x.XML_Column.extract('"+queryTipo+"').getStringVal() "+
            "FROM example1 x WHERE x.key_column=?";

      ArrayList participantes = new ArrayList();

      try {
          this.getConnection();
          OraclePreparedStatement preparedStatement=

(OraclePreparedStatement)this.connection.prepareStatement(query);
          preparedStatement.setString(1,idHistorial);
          ResultSet rs = preparedStatement.executeQuery();
          while ( rs.next() ) {
                   participantes.add(rs.getString(1));
                   participantes.add(rs.getString(2));
                   }
      […]
      return participantes;
```

Esta consulta parece que es correcta y que almacenaremos en *participantes* el número de colegiado del médico participante en primer lugar y el tipo de participación (propio o compartido) en segundo. Pues bien, si ejecutamos el siguiente código en un historial con dos participantes, Fernando Cortés (propietario) y Juan Molina (compartido):

```
      Collection participantes=bdxml.getParticipantes(idHist);
              if (participantes != null) {
                      if (participantes.size () > 0) {
                          for            (Iterator            iterator
=participantes.iterator(); iterator.hasNext(); ) {

                        System.out.println(iterator.next());
                          }
```

```
                    }

            }
```

Se obtiene la siguiente salida:

```
Fernando CortésJuan Molina
ownershared
```

Como se puede comprobar, ha almacenado las dos consultas, participantes y tipo, dentro de una fila en cada consulta. Obviamente este resultado no es el deseado. Es necesario cambiar esto.

No se puede usar la función *extractValue* porque sólo extrae la información de un solo nodo, no de varios, como en este caso. La solución es usar *XMLSequence* para formar una tabla virtual que almacene estos datos por separado, de modo que podamos acceder a ellos y guardarlos en el *ArrayList participantes* de forma independiente. El nuevo código usado para construir la sentencia que se pasa a Oracle sería el siguiente:

```
      String Xpath=
      "//clinical_document_header/local_header[@descriptor=\"compartid
os\"]/local_header[@descriptor=\"person\"]";//local_attr[@name=\"V\"]"
;
      String query="SELECT
      extractValue(value(col),'//local_attr[@name=\"V\"]/@value'), "+
"extractValue(value(col),'//local_attr[@name=\"type\"]/@value') "+
      "FROM example1 x, TABLE(XMLSEQUENCE(extract(x.XML_Column,
'"+Xpath+"'))) col WHERE x.key_column=?";
```

Ahora se ha usado la función *extractValue* para extraer los datos requeridos. Es posible usarla porque con *XMLSequence* y la función *TABLE* se ha creado una tabla que guarda en sus filas cada nodo que apunta el XPath. Así, *extractValue* extrae de cada fila un solo valor, no varios, y además este valor es tratado de forma independiente, de modo que al introducir la información extraída en el *ArrayList participantes* de igual manera que en el caso anterior y ejecutar las mismas líneas de código para mostrar los valores de *participantes*, obtenemos la siguiente salida:

```
Fernando Cortés
owner
```

```
Juan Molina
shared
```

Este resultado es el deseado para mostrarlo en la aplicación. Se ha comprobado la importancia de la función *XMLSequence* en el método usado para este ejemplo, *getParticipantes*, la cuál se extiende al resto de métodos que hacen uso de esta función.

5. Problema con *XMLSequence*.

Una vez visto el uso de la función *XMLSequence* vamos a hablar de un problema de compilación que surgió la primera vez que se intentó usar esta función. Al ejecutar una consulta que contuviera dicha función ocurría un error que decía que Oracle no reconocía *XMLSequence*. Después de investigar y preguntar en la página de soporte a los usuarios de Oracle se llegó a la conclusión de que Oracle no tenía compilada dicha función en su núcleo, por lo que se tuvo que navegar en el programa de administración a la página de las funciones y buscar la función *XMLSequence*. Esta función como tal no se encontró, pero sí se encontraron otras funciones llamadas *XMLSEQUENCEFROMXMLTYPE, XMLSEQUENCEFROMREFCURSOR 1* y *XMLSEQUENCEFROMREFCURSOR 2*. Se comprobó que estas funciones eran no válidas, por lo que se tuvieron que compilar, consiguiendo que fueran válidas. Al volver a probar una sentencia con *XMLSecuence* todo salió correctamente, con lo que el problema había sido resuelto.

En la figura 68 se muestra la página de administración de la base de datos Oracle que muestra algunas de las funciones que se incorporan bajo el esquema *SYS*. Se pueden ver las funciones anteriormente mencionadas encuadradas en un rectángulo rojo. Del mismo modo se ha encuadrado la validez de estas funciones. Antes de compilarlas las funciones eran "*INVALID*". Asimismo, si nos fijamos en la fecha de modificación de la función (la cuarta columna), vemos que data del 27 de diciembre del 2007 (día de la compilación), mientras que el resto datan del 30 de agosto de 2005 (día de la creación).

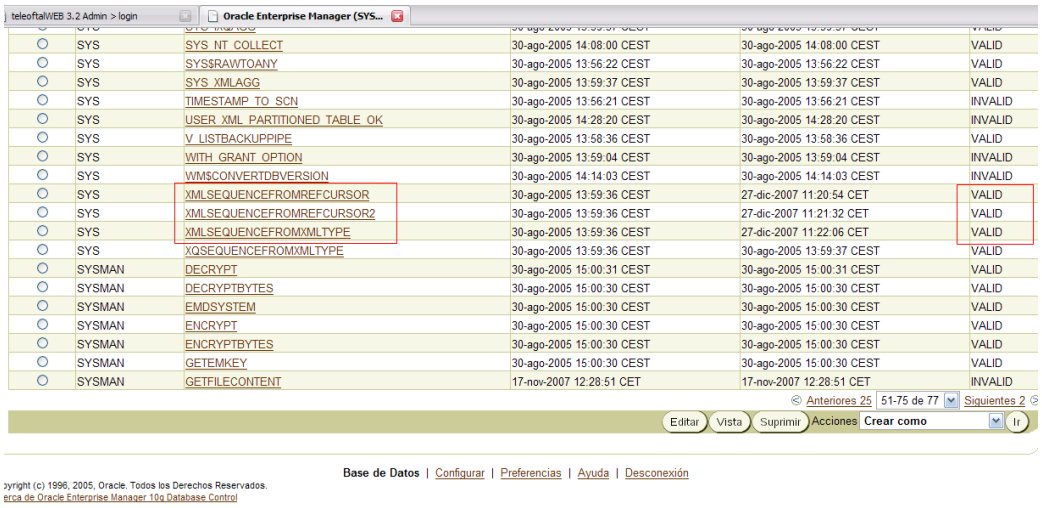


Figura 68. Página que muestra algunas funciones de Oracle bajo el esquema SYS. Fuente: propia.

Este problema se tuvo probando la aplicación en el ordenador del alumno, mientras que en el servidor de la Escuela estas funciones se encontraban validadas desde la instalación de la base de datos.

6. Extracción de los datos del historial.

Al probar los métodos que extraen una parte del historial para extraer ciertos datos de dicha parte, no se consiguió obtener estos datos correctamente. Nos estamos refiriendo a los métodos que hacen llamadas a los métodos de la clase *ParseXML*. Estos métodos son *getFiliacion, getDiagnosticos, getMotivoConsulta, getAPOftalmologicos, getAPGenerales, getAFamiliares, getExploracion y getComplementa*.

En estos métodos se hace siempre lo mismo: se obtiene el documento XML que se quiere extraer y después se obtiene el DOM de este documento. Esto es necesario para poder tener el documento en auténtico formato XML y poder hacer búsquedas en él. Una vez obtenido el DOM el último paso es llamar al método correspondiente de la función *ParseXML*. Con el método correspondiente nos referimos a que si estamos en *getFiliacion* por ejemplo, entonces usamos el método *parseFiliacion*, mientras que si estamos en *getExploracion* usamos *parseExploracion*.

Estos métodos de la clase *ParseXML* se encargan del procesamiento de los documentos XML extraídos en los métodos anteriormente mencionados. Para ello van pasando por todos los nodos y extrayendo los datos deseados. Estos datos son guardados en una instancia de una clase creada para ese propósito. Por ejemplo, si estamos en *parseFiliacion* (dentro de *getFiliacion*) guardaremos los datos en un objeto

de tipo *Filiacion*; si estamos en *parseExploracion*, estos datos se guardan en un objeto de tipo *Exploracion*. Todas estas clases se encuentran en el paquete *datos* de la aplicación.

Como hemos dicho, al probar estos métodos no se conseguía obtener los datos del documento extraído, por lo que el fallo debía de encontrarse en los métodos de *ParseXML*. Y así era, el problema estaba en que en estos métodos siempre se empieza haciendo la siguiente comprobación:

```
if(nodo.getNodeType()==Node.ELEMENT_NODE){
```

Esta condición nunca se cumple para Oracle. Con dbXML, al extraer las partes del historial usando estos métodos, el primer nodo siempre es un nodo elemento, mientras que ahora con Oracle el primer nodo no es un nodo elemento, sino un nodo documento. Por tanto, la solución a este problema fue la eliminación de esta condición en todos los métodos de la clase *ParseXML*. Se podría haberla cambiado por la siguiente condición:

```
if(nodo.getNodeType()==Node.DOCUMENT_NODE){
```

pero no fue necesario, ya que los métodos funcionan perfectamente sin esta comprobación, por lo que no tiene sentido añadir código innecesario.

7. Inserción de imágenes en la BD

Uno de los problemas que conllevó más cambios en la clase *BDxml* fue el problema que surgió cuando se intentó introducir una imagen real codificada en base 64 dentro de un nodo de la sección *Imágenes*. Al intentar introducir una imagen ocurría un error en la base de datos y es que ningún nodo que se introduzca en una columna de tipo XMLType puede sobrepasar en tamaño los 64 KB. Este problema es el que se menciona cuando se ha hablado de los métodos *insertarImagen*, *getImagenenNombre* y *actualizaImagenEditada*.

Cuando se comprobó el método *insertarImagen* por primera vez se usaron parámetros simples, cadenas de caracteres pequeñas, con lo que no ocurrió ningún error. Se pensó entonces que se podía introducir imágenes tal y como se había hecho hasta ahora (dentro del HC). Sin embargo, cuando se probó la nueva clase en la aplicación, se

observó que, al intentar introducir una imagen real, la aplicación se quedaba colgada y la imagen no había sido insertada en la base de datos. Tras hacer unas pruebas con una imagen real codificada en base 64 se obtuvo el error de límite de tamaño en los nodos de XMLType.

Al conocer cuál era el error se buscó información sobre posibles formas de solucionarlo, y se encontró con que este error ocurre en todas las versiones de Oracle que soportan XMLType hasta la 10, pero que la versión 11g está diseñada para no tener restricciones de tamaño en los nodos. Después de conocer este hecho se comprendió que sólo había dos posibles soluciones: cambiar la base de datos a Oracle 11g o almacenar estas imágenes de otra forma. Se eligió la opción más rápida y sencilla: la segunda.

Para llevarla a cabo se decidió crear una nueva tabla llamada *IMAGENES* que almacenara la imagen original y la imagen editada ambas codificadas en base 64. Esta tabla viene a sustituir a los nodos de la sección *Imágenes* que guardan dichas imágenes. La tabla tiene, además, otras columnas: una columna para guardar el nombre de la imagen y otra para guardar el identificador del historial donde pertenece la imagen. A continuación enunciamos cada columna de la tabla y su objetivo:

- *ID_IMAGEN*: Guarda el nombre de la imagen. Esta columna es necesaria para poder buscar la imagen en la tabla por su nombre, para identificarla. Es de tipo *VARCHAR2* con una longitud máxima de 40 caracteres, tamaño suficiente para guardar los nombres de las imágenes.
- *IMAGENORIGINAL*: Guarda la imagen original codificada en base 64. Es de tipo *CLOB* ya que estas imágenes codificadas son grandes y dependiendo del tamaño de la imagen pueden serlo aún más. Como vimos, el tipo *CLOB* sirve para guardar datos de tipo carácter de hasta 8 terabytes, y las imágenes *DICOM* ocupan mucho, del orden de 1.5 MB por lo que se decidió usar este tipo de columna.
- *IMAGENMODIFICADA*: Guarda la imagen editada codificada en base 64. Al igual que la columna anterior, es de tipo *CLOB*.
- *ID_HISTORIAL*: Almacena el identificador del historial al que pertenece la imagen. Esta columna es necesaria porque se puede dar el hecho de que imágenes de historiales distintos tengan el mismo nombre, por lo que es necesario distinguirlas usando esta columna.

En la figura 69 se muestra el esquema de la tabla creada.

Columnas

	Nombre	Tipo de Dato	Tamaño	Escala	No NULO	Valor por Defecto
	ID_IMAGEN	VARCHAR2	40		☐	
	IMAGENORIGINAL	CLOB			☐	
	IMAGENMODIFICADA	CLOB			☐	
	ID_HISTORIAL	VARCHAR2	40		☐	

Figura 69. Tabla *IMAGENES*. Fuente: propia.

Al cambiar la forma de guardar las imágenes no sólo fue necesario cambiar la forma de introducirlas en la base de datos, sino también el modo de acceder a ellas. Fue necesario cambiar todos los métodos que realizan operaciones con imágenes. Es por esto que cuando hemos hablado de los métodos *getImagenenNombre* y *actualizaImagenEditada* decíamos que antes de darnos cuenta de este problema (cuando se almacenaba todo como documento XML) usábamos funciones típicas de XMLType, pero que después de percatarnos del problema y cambiar el modo de guardar imágenes estas funciones desaparecían, porque ahora las imágenes se guardan en una tabla aparte, no en una columna de tipo XMLType.

Para introducir las imágenes, se solía usar el siguiente código XML que se añadía al historial (usando Java):

```
String xml=
"<item>"+
      "<caption>"+nombreImagen+"</caption>"+
      "<paragraph>"+
                  "<caption>Fecha</caption>"+
                  "<content>"+fechaCreacion+"</content>"+
      "</paragraph>"+
      "<paragraph>"+
                  "<caption>Descripcion</caption>"+
                  "<content>"+descripcionImagen+"</content>"+
      "</paragraph>"+
      "<paragraph>"+
                  "<caption>Imagen Original</caption>"+
                  "<content>"+
                        "<observation_media>"+
```

```
                                    "<observation_media.value
ENC=\"B64\"
MT=\"image/jpeg\">"+nombreImagen+"</observation_media.value>"+
                              "</observation_media>"+
                        "</content>"+
            "</paragraph>"+
            "<paragraph>"+
                        "<caption>Imagen Modificada</caption>"+
                        "<content>"+
                              "<observation_media>"+
                                    "<observation_media.value
ENC=\"B64\"
MT=\"image/jpeg\">"+nombreImagen+"</observation_media.value>"+
                              "</observation_media>"+
                        "</content>"+
            "</paragraph>"+
      "</item>";
```

Como se puede ver, con la introducción en la aplicación de la tabla *IMAGENES*, hay nodos que sobran, por lo que se eliminaron. Estos nodos sobrantes son los dos últimos nodos `<paragraph>` y sus descendientes. El resto de nodos se dejaron como estaban.

8. Obtención de los historiales en formato PDF.

TeleOftalWeb tiene una opción para convertir las revisiones de los historiales almacenados (en documentos XML) en archivos PDF que muestra por pantalla, con todos los datos guardados, lo que facilita su rápida visualización y permite su impresión.

Para llevar a cabo esta transformación, la aplicación extrae el historial completo y lo guarda en un archivo XML que se almacena en una carpeta con esa finalidad. Después se realiza un procesamiento de dicho historial línea por línea, buscando líneas de referencia para poder almacenar imágenes y datos que se mostrarán en el PDF y para buscar y extraer la revisión que se quiere transformar. Esta revisión es guardada como archivo XML en la carpeta anterior. Las imágenes que se mostrarán en el archivo PDF también son guardadas como JPEG en una carpeta destinada a dicho almacenamiento.

Una vez que se tiene la revisión en un archivo, se llama a la clase X*ml_to_Pdf*, que realiza un procesamiento de dicho archivo en conjunción con una hoja de estilos XSL-FO (*eXtensible Stylesheet Language Formatting Objects*). Esta hoja de estilos es

un documento XML en el que se especifica cómo se van a formatear los datos de la revisión para presentarlos en un archivo PDF. En ella figuran tanto los datos como el formato que se les va a aplicar. Así se lleva a cabo la composición del archivo PDF extrayendo los datos de la revisión guardada y transportándolos al archivo PDF. Una vez completada la operación muestra este archivo por pantalla y borra todos los archivos intermedios usados: los documentos xml y las imágenes almacenadas que se muestran en el archivo PDF.

Obviamente, para la obtención correcta del PDF de una revisión, fue necesario cambiar ciertas cosas. Hubo que adaptar este proceso para la base de datos Oracle.

Para empezar, fue necesario cambiar el identificador de los historiales. El archivo XML que almacena el historial extraído de la base de datos se guarda con su identificador, es decir el archivo guardado se llamará así: *identificador.xml*. Como dijimos, el identificador se compone del número de colegiado del usuario de la aplicación y de la fecha de creación de historiales separados por un espacio. Sin embargo, para poder almacenarlo usando Java es necesario que no haya espacios en el nombre del archivo ni símbolos que no sea el guión bajo "_". Por ello se tuvo que modificar el identificador del historial de modo que se sigue componiendo por los dos datos mencionados pero se han sustituido los espacios y todos los símbolos por guión bajo "_". De este modo si antes teníamos el identificador *2008-02-01 11:32 845822395*, ahora tendremos el identificador *2008_02_01_11_32_845822395*.

Otro problema que se encontró es que no se guardaba bien la revisión a partir del historial completo. Esto era debido a que las líneas de referencia que se comparaban con las líneas del historial completo no coincidían con las líneas del historial, aunque el contenido de ambas líneas fuera el mismo. Esto ocurría porque las tabulaciones del historial extraído con dbXML son distintas a las del historial extraído con Oracle. Para solucionar este problema se tuvo que adaptar las líneas de referencia antiguas a las nuevas tabulaciones que producía la extracción de un historial desde la base de datos Oracle. Una vez hecho esto, la revisión se pudo guardar en un archivo aparte, con las imágenes que se mostrarán en el PDF.

Finalmente, para que el archivo PDF se pudiera crear se extraían los datos de la revisión y se les daba formato usando la hoja de estilos XSL-FO. Sin embargo, esta hoja está desarrollada para obtener el PDF de los historiales de la antigua versión, con lo que el archivo no se creaba bien debido al cambio del formato de los historiales llevado a cabo cuando se introdujeron las secciones *JUICIO DIAGNÓSTICO* y *TRATAMIENTO*

Y RECOMENDACIONES dentro de la sección *OTRAS PRUEBAS*. Por esto, finalmente se decidió usar cinco bloques de revisión en vez de tres como se pensó inicialmente. De este modo, el bloque 3 se dividió en tres bloques:

- Bloque 3: Almacena el documento XML referente a la sección *OTRAS PRUEBAS*.
- Bloque 4: Almacena la parte del historial referente a la sección *JUICIO DIAGNOSTICO*.
- Bloque 5: Guarda la parte referente a la sección *TRATAMIENTO Y RECOMENDACIONES*.

De este modo no hizo falta cambiar la hoja de estilos para crear el archivo PDF correctamente y además conseguimos que los nuevos historiales creados en la base de datos de Oracle tengan el mismo formato que los historiales existentes en dbXML. De este modo se conseguiría que, una vez migrados dichos historiales, se podrían obtener sus PDFs. Es más, también se podría conseguir abrir sus revisiones en la aplicación. Sin embargo, como se verá en el siguiente apartado los PDFs no podrán ser creados a partir de los historiales migrados.

6.4. Modelado de datos

En la versión 3.2 de la aplicación **TeleOftalWeb** se utiliza un doble modelo de BD. Por un lado, se utiliza una BD relacional implantada sobre el servidor MySQL Server 5.0 en la que se almacena toda la información referente a los usuarios del sistema, tanto datos personales como contraseñas de autentificación en la herramienta. Por otro, se hace uso de la BD nativa XML dbXML 2.0 para almacenar los historiales clínicos de los pacientes y las imágenes asociadas a los mismos. La razón de utilizar dos BD tan diferentes es para separar la información más estructurada, como son los historiales clínicos, en un formato que permita su fácil intercambio y procesamiento (XML). Por otro lado, la información de usuarios es utilizada solamente en los procesos de gestión interna y autenticación, para lo que las BDs relacionales siguen siendo mucho más eficientes. Otro aspecto por el que la información de usuario no se ha introducido en la BD nativa es que el lugar natural de almacenamiento para este tipo de información sería un servidor LDAP.

Las dos BDs están relacionadas mediante un único campo, el número de colegiado, dato que es almacenado en una variable de sesión dentro del servidor Web una vez que el usuario se ha autenticado de forma correcta.

Para el correcto funcionamiento de la aplicación es necesario que los puertos utilizados para la comunicación con las BDs y securización estén abiertos en el sistema. Los puertos utilizados para dichos fines son:

- MySQL: 3306.
- dbXML: 7280.
- HTTPS: 443.
- HTTP: 80.

En la figura 70 se ilustra a grandes rasgos el esquema de comunicación de la aplicación. Inicialmente un usuario se conecta de forma remota y se negocian los parámetros de la encriptación SSL. Una vez establecida la sesión encriptada el usuario debe autentificarse en el sistema para poder acceder a las funcionalidades de la herramienta. Los datos introducidos por el usuario se contrastarían con la BD relacional, y en caso de ser correctos se almacenaría el identificador de usuario en la sesión del servidor. A partir de este momento el usuario puede realizar diversas acciones de consulta y creación de revisiones, pudiendo acceder solamente a aquellas que son visibles para el usuario con el que se identificó.

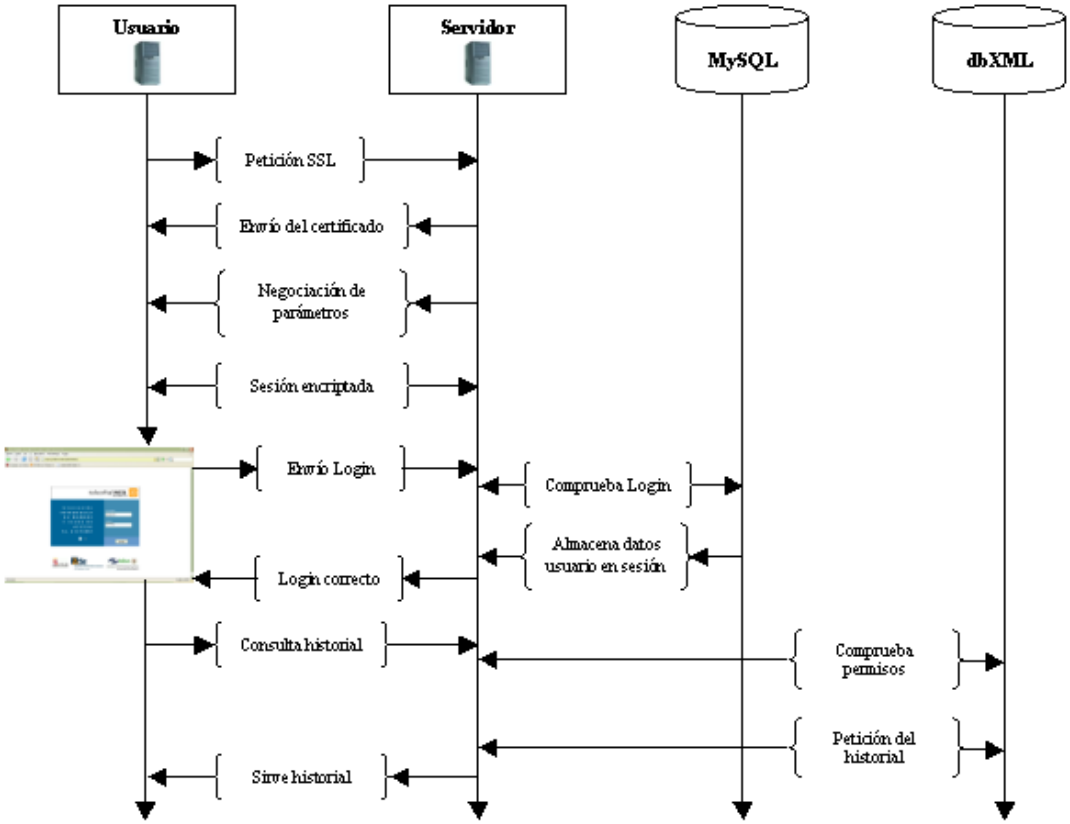


Figura 70. Esquema de comunicación de la aplicación empleando la BD dbXML 2.0. Fuente: propia.

A continuación se describirá con detalle el modelo de datos utilizado en cada una de estas BDs. Para el resto de BDs nativas XML el esquema de comunicación de la aplicación es idéntico al mostrado en la figura 70.

6.4.1. Base de datos MySQL

MySQL es un sistema de gestión de BD relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como *software* libre en un esquema de licenciamiento dual. Por un lado lo ofrece bajo la GNU GPL, pero, empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia que les permita ese uso, desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como el *Apache*, donde el *software* es desarrollado por una comunidad pública, y el *copyright* del código está en poder del autor individual, MySQL es propiedad y está patrocinado por una empresa privada, que posee el

copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado.

La razón de porque se ha querido usar este tipo de BD reside en sus ventajas, que se citan a continuación:

- Acceso a las BD de forma simultánea por varios usuarios y/o aplicaciones.
- Seguridad, en forma de permisos y privilegios, determinados usuarios tendrán permiso para consulta o modificación de determinadas tablas. Esto permite compartir datos sin que peligre la integridad de la BD o protegiendo determinados contenidos.
- Potencia: SQL es un lenguaje muy potente para consulta de BD, usar un motor nos ahorra una enorme cantidad de trabajo.
- Portabilidad: SQL es también un lenguaje estandarizado, de modo que las consultas hechas usando SQL son fácilmente portables a otros sistemas y plataformas.
- Escalabilidad: es posible manipular BD enormes, del orden de seis mil tablas y alrededor de cincuenta millones de registros, y hasta 32 índices por tabla.
- MySQL está escrito en C y C++ y probado con multitud de compiladores y dispone de APIs para muchas plataformas diferentes.
- Conectividad: es decir, permite conexiones entre diferentes máquinas con distintos sistemas operativos. Es corriente que servidores Linux o Unix, usando MySQL, sirvan datos para ordenadores con Windows, Linux, Solaris, etc. Para ello se usa TCP/IP, tuberías, o *sockets Unix*.
- Es multihilo, con lo que puede beneficiarse de sistemas multiprocesador.
- Permite manejar multitud de tipos para columnas.
- Permite manejar registros de longitud fija o variable.

En **TeleOftalWeb**, esta BD se encarga de almacenar todos los datos relativos a los usuarios del sistema: datos personales y datos de acceso a la herramienta. Está formada por dos tablas, los usuarios que contiene los datos personales de los usuarios del sistema y los permisos que contiene el identificador del usuario, nombre de usuario, contraseña y tipo de usuario. La clave primaria de ambas tablas es el DNI del usuario que es único para cada persona. Esta clave primaria relaciona de forma unívoca cada entrada de una tabla con una entrada de la otra.

A continuación se muestra la descripción de cada uno de los campos de estas tablas.

Tabla de usuarios:

- Id: es un campo alfanumérico que contiene el DNI del usuario. Es de carácter obligatorio.
- Nombre: es un campo alfanumérico que contiene el nombre del usuario. Es de carácter obligatorio.
- Apellidos: es un campo alfanumérico que contiene los apellidos del usuario. Es de carácter obligatorio.
- Calle: es un campo alfanumérico que contiene la dirección de la residencia del usuario. No es un campo obligatorio.
- Fechaultacceso: es un campo de tipo fecha que contiene la fecha en la que el usuario realizó su último acceso a la herramienta. No es un campo obligatorio.
- Teléfono: es un campo alfanumérico que contiene el teléfono de contacto del usuario. Es de carácter obligatorio.
- Email: es un campo alfanumérico que contiene la dirección de correo de contacto del usuario. Es de carácter obligatorio.
- Fechanacimiento: es un campo alfanumérico que contiene la fecha de nacimiento del usuario. No es un campo obligatorio.
- Especialidad: es un campo alfanumérico que contiene la especialidad médica del usuario. Es de carácter obligatorio.
- Numcolegiado: es un campo alfanumérico que contiene el identificador de colegiado del usuario. Es de carácter obligatorio.
- Fechaprimacceso: es un campo de tipo fecha que contiene la fecha en la que el usuario realizó su primer acceso a la herramienta. No es un campo obligatorio.
- Numportal: es un campo alfanumérico que contiene número de bloque de la vivienda del usuario. No es un campo obligatorio.
- Piso: es un campo alfanumérico que contiene el número de planta de la vivienda del usuario. No es un campo obligatorio.
- Población: es un campo alfanumérico que contiene el nombre de la población dónde reside el usuario. No es un campo obligatorio.
- Provincia: es un campo alfanumérico que contiene la provincia dónde reside el usuario. No es un campo obligatorio.

Column Name	Datatype	NOT NULL	AUTO INC	Flags
id	VARCHAR(45)	✓		BINARY
nombre	VARCHAR(45)	✓		BINARY
apellidos	VARCHAR(45)	✓		BINARY
calle	VARCHAR(45)			BINARY
fechaultacceso	DATETIME			
telefono	VARCHAR(45)	✓		BINARY
email	VARCHAR(45)	✓		BINARY
fechanacimiento	VARCHAR(45)			BINARY
especialidad	VARCHAR(45)			BINARY
numcolegiado	VARCHAR(45)	✓		BINARY
fechaprimacceso	DATETIME			
numportal	VARCHAR(45)			BINARY
piso	VARCHAR(45)			BINARY
poblacion	VARCHAR(45)			BINARY
provincia	VARCHAR(45)			BINARY
codpostal	VARCHAR(45)			BINARY

Figura 71. Modelo de datos: tabla de usuarios. Fuente: propia.

Tabla de permisos:

- Id: es un campo alfanumérico que contiene el DNI del usuario. Es un campo obligatorio.
- Login: es un campo alfanumérico que contiene nombre de usuario para acceder a la herramienta. Es un campo obligatorio.
- Password: es un campo alfanumérico que contiene la clave de acceso a la herramienta del usuario. Es un campo obligatorio.
- Tipo: es un campo alfanumérico que contiene el tipo de acceso que tiene el usuario a la herramienta. Es un campo obligatorio.

Column Name	Datatype	NOT NULL	AUTO INC	Flags
id	VARCHAR(45)	✓		BINARY
login	VARCHAR(45)	✓		BINARY
password	VARCHAR(45)	✓		BINARY
tipo	VARCHAR(45)	✓		BINARY

Figura 72. Modelo de datos: tabla de permisos. Fuente: propia.

La relación entre las dos tablas anteriores puede verse en la figura 73:

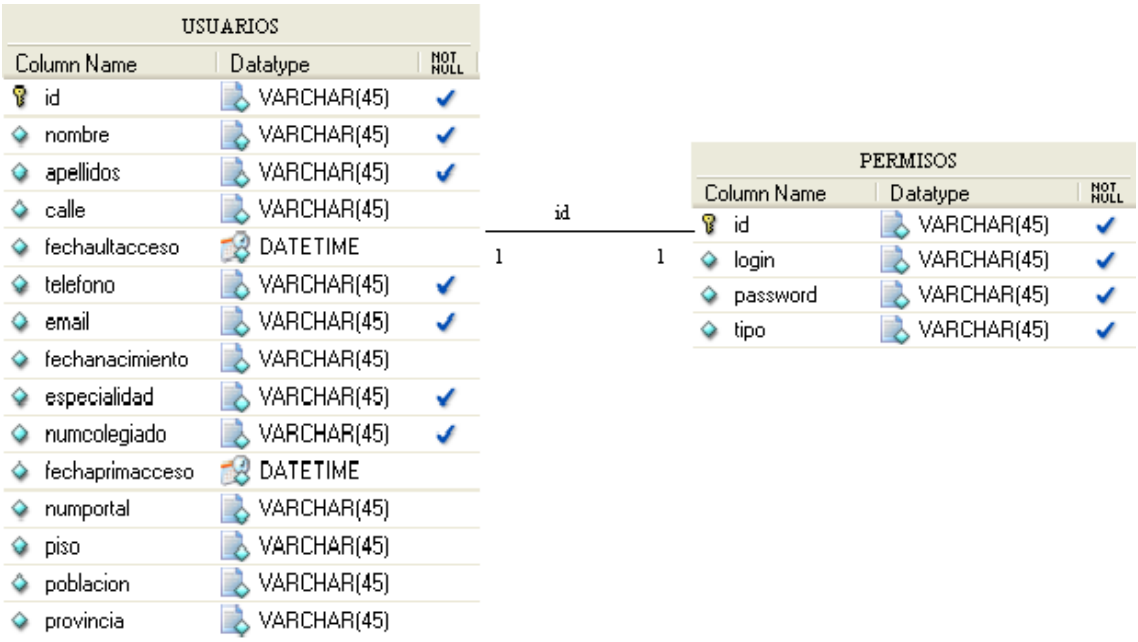


Figura 73. Modelo de datos: relación entre ambas tablas. Fuente: propia.

6.4.2. Bases de datos nativas XML

La BD nativa XML utilizada en la aplicación **TeleOftalWeb** es una herramienta que permite crear repositorios de documentos *xml* agrupados en colecciones. La gran ventaja de estas herramientas es la posibilidad de consultar y recuperar partes concretas de un documento mediante el lenguaje Xpath. Así mismo, permite realizar inserciones y actualizaciones dentro de los documentos de la colección. El resto de funcionalidades son muy similares a las de cualquier BD relacional, incluyendo la posibilidad de crear usuarios con diferentes permisos, indexación de contenidos, etc.

En la figura 74 se puede observar la ventana de la consola de administración de la BD dbXML 2.0. La consola de administración tiene 3 secciones bien diferenciadas, por un lado la columna de la izquierda, donde se puede acceder a todas las colecciones desplegadas en la BD, así como a las funcionalidades de creación de usuarios, indexación, etc., por otro lado se puede observar que la parte central de la consola es un editor dónde se pueden visualizar y modificar los documentos xml almacenados, y por último la parte superior proporciona una herramienta de consulta a las colecciones de documentos, pudiendo simular sentencias de búsqueda a los desarrolladores. En la figura 74, se puede observar la colección de documentos creada para la aplicación TeleOftal, “historiales”, desplegada a la izquierda y uno de los historiales clínicos abierto en la parte central. En el capítulo de BD se pueden consultar las interfaces de las otras dos BDs empleadas en esta Tesis, eXist y Xindice.

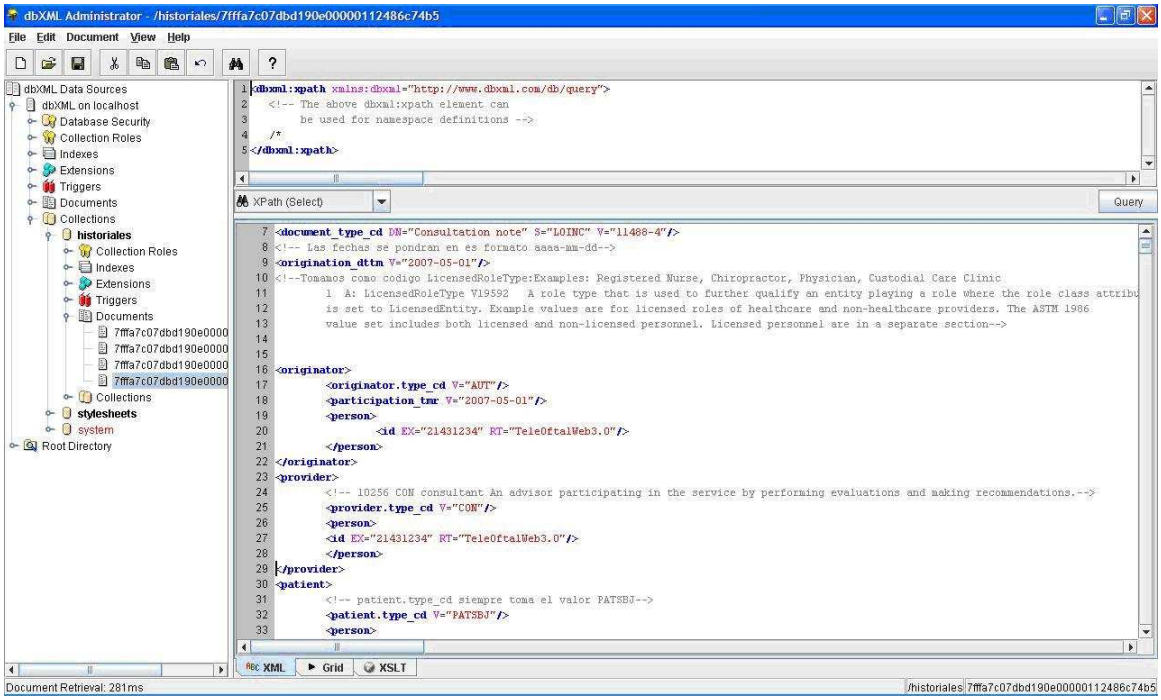


Figura 74. Interfaz de la base de datos dbXML. Fuente: propia.

Los HC almacenados en documentos *xml* siguen la estructura fijada por la DTD del estándar CDA Release 1.0 del HL7. Esta DTD fija la existencia de una cabecera, dónde se almacena la información relativa a la creación del historial y los datos personales del paciente, y un cuerpo del documento en el que se puede almacenar cualquier tipo de información estructurada en nodos *section*, *paragraph*, *content*, etc. Para más información esta DTD está puede consultarse en el Apéndice B de este proyecto.

A continuación se describirá brevemente la estructura seguida en este proyecto para adaptar el formato de historial utilizado por el IOBA al estándar CDA v 1.0:

- **Cabecera.** En las siguientes figuras pueden verse los campos definidos en utilizados en la cabecera y que se analizarán a continuación.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
- <levelone>
  - <clinical_document_header>
      <!-- IMPORTANTE PARA ENTENDER EL CAMPO ID LEER "D6.2 CDA Harmonised Message Spec v3.0
      FINAL FINAL 100502.doc" -->
      <id EX="6701a7c07dbd190e0000010a37134255" RT="TeleOftalWeb3.2" />
      <document_type_cd DN="Consultation note" S="LOINC" V="11488-4" />
      <!-- Las fechas se pondran en es formato aaaa-mm-dd -->
      <origination_dttm V="2006-03-26" />
    - <!--
        Tomamos como codigo LicensedRoleType:Examples: Registered Nurse, Chiropractor,
        Physician, Custodial Care Clinic
              1  A: LicensedRoleType V19592   A role type that is used to further qualify an e
              is set to LicensedEntity. Example values are for licensed roles of healthcare ar
              value set includes both licensed and non-licensed personnel. Licensed personnel
      -->
    - <originator>
        <originator.type_cd V="AUT" />
        <participation_tmr V="2006-03-26" />
      - <person>
          <id EX="22342424" RT="TeleOftalWeb3.2" />
        </person>
      </originator>
    - <provider>
        <!-- 10256 CON consultant An advisor participating in the service by performing
        evaluations and making recommendations. -->
        <provider.type_cd V="CON" />
      + <person>
      </provider>
```

Figura 75. Campos de la cabecera CDA. Fuente: propia.

- Id. Representa el identificador del historial y es único en todo el repositorio de la BD. El atributo RT indica el ámbito en el que es válido el identificador.
- Document_type_cd. Este campo indica el tipo de documento médico que contiene el documento. En el caso de este proyecto se trata de historiales clínicos, que dentro del vocabulario LOINC le corresponde el código 11488-4.
- Origination_dttm. Indica la fecha de creación del historial. Dentro del estándar CDA todas las fechas se representan con el formato aaaa-mm-dd
- Originator. Este nodo contiene información sobre la persona que ha originado la creación del historial. Se indican datos como la fecha de participación identificador del médico, etc.
- Provider. Contiene información sobre la persona u organización que proporciona el servicio médico.

```
- <patient>
    <!-- patient.type_cd siempre toma el valor PATSBJ -->
    <patient.type_cd V="PATSBJ" />
  - <person>
      <id EX="13543224-s" RT="TeleOftalWeb3.2" />
    - <person_name>
      - <nm>
          <GIV V="Juan" />
          <FAM V="López García" />
        </nm>
      </person_name>
    - <addr>
        <STR V="c/ Algún Lugar" />
        <HNR V="3" />
        <ADL V="3ºB" />
        <CTY V="Valladolid" />
        <STA V="Valladolid" />
        <ZIP V="4700" />
        <CNT V="España" />
      </addr>
      <phon V="983-00-00-00" />
      <local_header descriptor="email">lopgarju@teleoftal.com</local_header>
    </person>
    <birth_dttm V="1945-12-12" />
    <!-- (M) male (F) female (UN) The gender of a person could not be uniquely defined
    as male or female, such as hermaphrodite -->
    <administrative_gender_cd S="2.15.840.1.113883.5.1" V="M" />
    <local_header descriptor="numeroSS">111111111</local_header>
  </patient>
```

Figura 76. Campo de la cabecera CDA. Fuente: propia.

- Patient. Este nodo contiene toda la información referente al paciente que está recibiendo el servicio. Los campos que contiene son:
  - *Patient_type*. Siempre toma el valor PATSBJ.
  - *Id*. Almacena el identificador del paciente, en este caso se ha tomado como identificador el DNI.
  - *GIV*. Nombre del paciente.
  - *FAM*. Apellidos del paciente.
  - *STR*. Calle del domicilio del paciente.
  - *HNR*. Número del portal de la vivienda del paciente.
  - *ADL*. Este campo sirve para contener información adicional relativa a la dirección del paciente. En el presente proyecto almacena el número de planta de la vivienda del paciente.
  - *CTY*. Localidad de residencia del paciente.
  - *STA*. Provincia a la que pertenece la localidad de residencia del paciente.
  - *ZIP*. Código postal del domicilio del paciente.
  - *CNT*. País de residencia del paciente.
  - *Phon*. Número de teléfono del paciente.

- *Local_header*. El estándar CDA permite definir etiquetas especiales si las fijadas por la DTD no se ajustan a la información que se quiere almacenar. En este caso concreto se ha utilizado una etiqueta de cabecera personalizada con el descriptor "email" para almacenar la dirección de correo del paciente.
- *Birth_dttm*. Fecha de nacimiento del paciente.
- *Administrative_gender_cd*. Sexo del paciente.
- *Local_header*. Etiqueta con descriptor "numeroSS" para almacenar el número de la seguridad social del paciente.

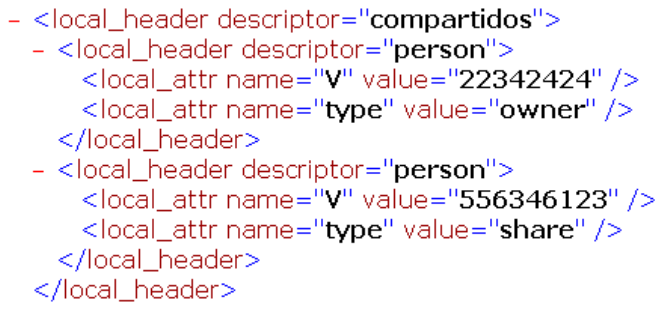


Figura 77. Campos de la cabecera CDA (*local_header*). Fuente: propia.

  - Local_header "compartidos". Este grupo de etiquetas se utiliza para definir los participantes que tienen acceso al historial y el tipo de permisos de cada uno.

- **Cuerpo.** El nodo básico dentro del cuerpo del documento es el de tipo *section*. En los historiales definidos se pueden distinguir dos secciones almacenadas dentro de los correspondientes nodos *section*, por un lado la que contiene todas las revisiones del paciente y por otro la que almacena las imágenes asociadas al historial. Esto puede observarse en la figura 78.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
- <levelone>
  + <clinical_document_header>
  - <body>
      <!--
      ****************************************************************************************
      <!--                                    REVISIONES
                                                                    -->
      <!--
      ****************************************************************************************
    + <section>
      <!--
      ****************************************************************************************
      <!--                                          IMAGENES
                                                                           -->
      <!--
      ****************************************************************************************
    + <section>
    </body>
  </levelone>
```

Figura 78. Estructura del cuerpo del historial XML. Fuente: propia.

Como puede observarse en la figura 79, cada revisión del paciente se almacena en un nodo *section* contenido dentro del nodo de revisiones.

```
<!--
************************************************************************************
<!--                                  REVISIONES
                                                           -->
<!--
************************************************************************************
- <section>
    <caption>Revisiones</caption>
    <!-- ************************************************************************** --
    >
    <!--             NUEVA REVISION                                  -->
    <!-- *************************************************************************** --
    >
    <!-- Cada nueva revision será una nueva sección dentro de Revisiones, y cuyo titulo
    será la fecha  -->
  + <section>
    <!-- ************************************************************************** --
    >
    <!--             NUEVA REVISION                                  -->
    <!-- *************************************************************************** --
    >
    <!-- Cada nueva revision será una nueva sección dentro de Revisiones, y cuyo titulo
    será la fecha  -->
  + <section>
    <!--  INSERTAR AQUI NUEVA REVISION   -->
  </section>
```

Figura 79. Esquema de revisiones dentro de un historial XML. Fuente: propia.

A su vez cada revisión se encuentra dividida en secciones que corresponden a cada una de los grupos de información diferenciada que puede distinguirse en los historiales, esto es: filiación, motivo consulta, antecedentes personales oftalmológicos, antecedentes personales generales, antecedentes familiares, exploración, otras pruebas, juicio diagnóstico y por último tratamiento y recomendaciones. De la misma forma que

en los casos anteriores, cada uno de estos apartados se encuentra en un nodo *section*. Cada revisión lleva asociado un nodo *caption* donde se almacena la fecha y hora de creación de la misma.

```
<!-- ************************************************************************** --
>
<!--               NUEVA REVISION                                        -->
<!-- *************************************************************************** --
>
<!-- Cada nueva revision será una nueva sección dentro de Revisiones, y cuyo titulo
será la fecha -->
- <section>
    <caption>2006-03-26 17:01</caption>
    <!-- ************************************************** -->
    <!--       FILIACION                                -->
    <!-- ************************************************** -->
  + <section>
    <!-- ************************************************** -->
    <!--       MOTIVO CONSULTA                                -->
    <!-- ************************************************** -->
  + <section>
    <!-- ************************************************** -->
    <!--       AP OFTALMOLOGICOS                        -->
    <!-- ************************************************** -->
  + <section>
    <!-- ************************************************** -->
    <!--       AP GENERALES                             -->
    <!-- ************************************************** -->
  + <section>
    <!-- ************************************************** -->
    <!--       A FAMILIARES                             -->
    <!-- ************************************************** -->
  + <section>
    <!-- ************************************************** -->
    <!--       EXPLORACION                              -->
    <!-- ************************************************** -->
  + <section>
    <!-- ************************************************** -->
    <!--       OTRAS PRUEBAS                                  -->
    <!-- ************************************************** -->
  + <section>
    <!-- ************************************************** -->
    <!--       JUICIO DIAGNOSTICO                       -->
    <!-- ************************************************** -->
  + <section>
    <!-- ********************************************************** -->
    <!--       TRATAMIENTO Y RECOMENDACIONES                    -->
    <!-- ********************************************************** -->
  + <section>
</section>
```

Figura 80. Estructura de una revisión de un historial XML. Fuente: propia.

Esta estructura de secciones anidadas se puede seguir repitiendo agrupando los apartados generales, hasta llegar a los datos concretos insertados por los usuarios de la herramienta. La forma en la que se almacenan estos datos en el documento XML puede variar en función del formato de entrada en la herramienta, pudiendo tener listas, etiquetas especiales, etc. Sin embargo, la estructura que más se repite consiste en una sección con un título que contiene el nombre del dato que contiene. Anidado dentro de

esta sección se puede encontrar un nodo de tipo *paragraph* cuyo contenido es el dato en sí. Esta estructura puede apreciarse en la figura 81.

```
<!--       *****    REMITIDO PO OTRO MEDICO (SI / NO)    ******  -->
- <section>
    <caption>Remitido por otro médico</caption>
  - <paragraph>
      <content>SI</content>
    </paragraph>
  </section>
```

Figura 81. Estructura XML de un dato del historial. Fuente: propia.

Las imágenes se almacenan como elementos de una lista dentro de la sección correspondiente. Cada uno de estos elementos tiene un titulo donde se almacena el identificador de la imagen, generado a partir de una marca de tiempo y el identificador del usuario que la ha introducido. Además las imágenes tienen un nodo hijo que contiene la descripción y otros dos para la imagen original y la editada. Las imágenes se almacenan dentro de una etiqueta especial denominada *observation_media* y codificadas en base 64. Se pueden observar estos elementos en la figura 82:

```
- <section>
    <caption>Imagenes</caption>
  - <list>
    - <item>
        <caption>1147984243015-222222222</caption>
      - <paragraph>
          <caption>Fecha</caption>
          <content>2006-05-18</content>
        </paragraph>
      - <paragraph>
          <caption>Descripcion</caption>
          <content>manchas dispersas</content>
        </paragraph>
      - <paragraph>
          <caption>Imagen Original</caption>
        - <content>
          + <observation_media>
          </content>
        </paragraph>
      - <paragraph>
          <caption>Imagen Modificada</caption>
        - <content>
          + <observation_media>
          </content>
        </paragraph>
      </item>
    </list>
```

Figura 82. Estructura XML de una imagen de un historial. Fuente: propia.

Como se ha podido comprobar a lo largo de este apartado, todos los elementos del historial contienen un elemento *caption* que los identifica. Dicho identificador y la

estructura jerárquica del documento permiten realizar un procesamiento sencillo para recuperar los datos de interés y mostrarlos en el interfaz gráfico cuando un usuario consulta una revisión.

6.4.3. Base de datos Oracle 10g

En la BD Oracle 10g almacenamos los datos de usuario y la información de acceso a la plataforma, así como los HCE e imágenes. La aplicación **TeleOftalWeb** va a ser usada por médicos que tendrán que ser dados de alta en ella y para ello se guardarán ciertos datos personales suyos, además del *login* y la contraseña de acceso. En la figura 83, se puede observar el modelado de datos empleando la BD Oracle 10g. Tiene 3 tablas: "usuarios", "permisos" e "historiales". En la tabla "usuarios" se almacena de información del personal médico. En la tabla "permisos" se almacena la información de acceso Web. Inicialmente, los HCEs son almacenados en dicha BD deacuerdo al estándar HL7/CDA Release 1.0 y posteriormente siguiendo el Release 2.0.

Se puede ver que todos los datos son de tipo VARCHAR2 (datos de texto) con una longitud de 45 caracteres, excepto FECHAULTACCESO Y FECHAPRIMACCESO, que tienen 25. Estos dos últimos datos contienen fechas que se guardarán con el formato mostrado en la columna *Valor por Defecto*. La columna *No NULO* indica qué datos pueden ser introducidos como valores nulos y cuáles no.

En la tabla "permisos" LOGIN y PASSWORD servirán para identificarse en la aplicación. TIPO almacena el tipo de usuario (dos valores posibles: *admin* o usuario). ID guarda la identidad del médico y sirve para poder relacionar los datos personales del usuario almacenado con estos datos de acceso a la aplicación, ya que este campo es el mismo campo ID de la tabla *USUARIOS*. La tabla "historiales" consta de dos columnas: La columna *KEY_COLUMN* almacena el número de identificación del historial, que será formado con el número de colegiado del médico propietario del historial y la fecha y la hora de creación del historial. La columna *XML_COLUMN* es una columna de tipo XMLType que almacenará los HCEs.

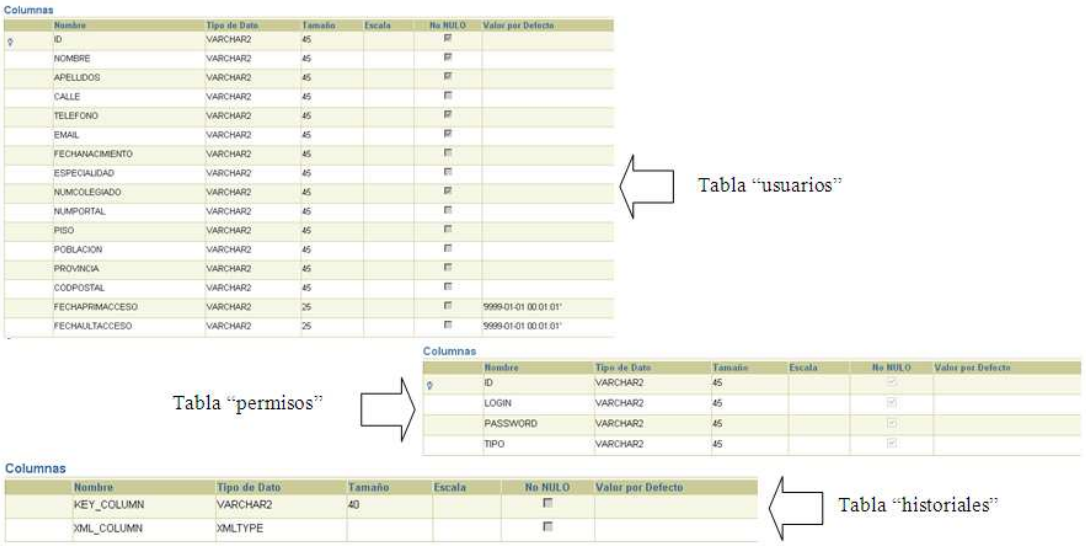


Figura 83. Modelado de datos en Oracle 10g. Fuente: propia.

6.5. Modelos de colas

El estudio de las colas de un sistema es importante porque proporciona tanto una base teórica del tipo de servicio que podemos esperar de un determinado recurso, como la forma en la cual dicho recurso puede ser diseñado para proporcionar un determinado grado de servicio a sus clientes.

La teoría de colas es el estudio matemático del comportamiento de líneas de espera. Esta se presenta, cuando los "clientes" llegan a un "lugar" demandando un servicio a un "servidor", el cual tiene una cierta capacidad de atención. Si el servidor no está disponible inmediatamente y el cliente decide esperar, entonces se forma la línea de espera.

Los sistemas de colas son modelos de sistemas que proporcionan servicio. Como modelo, pueden representar cualquier sistema en donde los trabajos o clientes llegan buscando un servicio de algún tipo y salen después de que dicho servicio haya sido atendido. Podemos modelar los sistemas de este tipo tanto como colas sencillas o como un sistema de colas interconectadas formando una red de colas.

Origen

El origen de la Teoría de Colas está en el esfuerzo de Agner Kraup Erlang (Dinamarca, 1878 - 1929) en 1909 para analizar la congestión de tráfico telefónico con el objetivo de cumplir la demanda incierta de servicios en el sistema telefónico de

Copenhague. Sus investigaciones acabaron en una nueva teoría denominada teoría de colas o de líneas de espera. Esta teoría es ahora una herramienta de valor en negocios debido a que un gran número de problemas pueden caracterizarse, como problemas de congestión llegada-salida.

Modelo de formación de colas

Cuando se habla de líneas de espera, se refieren a las creadas por clientes o por las estaciones de servicio. Los clientes pueden esperar en cola simplemente porque los medios existentes son inadecuados para satisfacer la demanda de servicio; en este caso, la cola tiende a ser explosiva, es decir, a ser cada vez más larga a medida que transcurre el tiempo. Las estaciones de servicio pueden estar esperando por que los medios existentes son excesivos en relación con la demanda de los clientes; en este caso, las estaciones de servicio podrían permanecer ociosas la mayor parte del tiempo. Los clientes puede que esperen temporalmente, aunque las instalaciones de servicio sean adecuadas, por que los clientes llegados anteriormente están siendo atendidos. Las estaciones de servicio pueden encontrar temporal cuando, aunque las instalaciones sean adecuadas a largo plazo, haya una escasez ocasional de demanda debido a un hecho temporal. Estos dos últimos casos tipifican una situación equilibrada que tiende constantemente hacia el equilibrio, o una situación estable.

La teoría de colas incluye el estudio matemático de las colas o líneas de espera y provee un gran número de modelos matemáticos para describirlas.

Se debe lograr un balance económico entre el costo del servicio y el costo asociado a la espera por ese servicio. La teoría de colas en sí no resuelve este problema, sólo proporciona información para la toma de decisiones.

Objetivos de la Teoría de Colas

Los objetivos de la teoría de colas consisten en:

- Identificar el nivel óptimo de capacidad del sistema que minimiza el coste global del mismo.
- Evaluar el impacto que las posibles alternativas de modificación de la capacidad del sistema tendrían en el coste total del mismo.
- Establecer un balance equilibrado (“óptimo”) entre las consideraciones cuantitativas de costes y las cualitativas de servicio.

- Hay que prestar atención al tiempo de permanencia en el sistema o en la cola: la "paciencia" de los clientes depende del tipo de servicio específico considerado y eso puede hacer que un cliente "abandone" el sistema.

Elementos existentes en un modelo de colas

- Fuente de entrada o población potencial: Es un conjunto de individuos que pueden llegar a solicitar el servicio en cuestión. Podemos considerarla finita o infinita.
- Cliente: Es todo individuo de la población potencial que solicita servicio.
- Capacidad de la cola: Es el máximo número de clientes que pueden estar haciendo cola.
- Disciplina de la cola: Es el modo en el que los clientes son seleccionados para ser servidos. Las disciplinas más habituales son:
  - La disciplina First in First Out (FIFO): según la cual se atiende primero al cliente que antes haya llegado.
  - La disciplina Last in First Out (LIFO) o pila: que consiste en atender primero al cliente que ha llegado el último.
  - La Random Selection of Service (RSS), que selecciona a los clientes de forma aleatoria.
- Mecanismo de servicio: Es el procedimiento por el cual se da servicio a los clientes que lo solicitan. Para determinar totalmente el mecanismo de servicio debemos conocer el número de servidores de dicho mecanismo (si dicho número fuese aleatorio, la distribución de probabilidad del mismo) y la distribución de probabilidad del tiempo que le lleva a cada servidor dar un servicio. En caso de que los servidores tengan distinta destreza para dar el servicio, se debe especificar la distribución del tiempo de servicio para cada uno.

La cola, propiamente dicha, es el conjunto de clientes que hacen espera, es decir los clientes que ya han solicitado el servicio pero que aún no han pasado al mecanismo de servicio.

El sistema de la cola: es el conjunto formado por la cola y el mecanismo de servicio, junto con la disciplina de la cola, que es lo que nos indica el criterio de qué cliente de la cola elegir para pasar al mecanismo de servicio.

La distribución más usada para los tiempos de servicio es la exponencial, aunque es común encontrar la distribución degenerada o determinística (tiempos de servicio constantes) o la distribución Erlang (Gamma).

Notación de Kendall

Por convención los modelos que se trabajan en teoría de colas se etiquetan. Las distribuciones que se utilizan son:

- M: Distribución exponencial (markoviana)
- D: Distribución degenerada (tiempos constantes)
- E k : Distribución Erlang
- G : Distribución general
- M / M / s: Modelo donde tanto los tiempos entre llegada como los tiempo de servicio son exponenciales y se tienen s servidores.
- M / G / 1: Tiempos entre llegada exponenciales, tiempos de servicio general y 1 sólo servidor.

Medidas de rendimiento para evaluar un sistema de colas

El objetivo último de la teoría de colas consiste en responder cuestiones administrativas pertenecientes al diseño y a la operación de un sistema de colas. En una estructura de producción, el administrador puede desear evaluar el impacto de la compra de una nueva máquina que pueda procesar los productos con más rapidez. Existen muchas medidas de rendimiento diferentes que se utilizan para evaluar un sistema de colas en estado estable. Para diseñar y poner en operación un sistema de colas, por lo general, los administradores se preocupan por el nivel de servicio que recibe un cliente, así como el uso apropiado de las instalaciones de servicio de la empresa.

Modelo M/M/1

Población de clientes infinita, llegadas de clientes probabilística según Poisson; una línea de espera y un solo servidor o canal de atención con tiempo de servicio exponencial.

Supuesto: Condición Estable; cuando $m > l$, es decir la tasa de servicio promedio es mayor que la tasa de llegadas promedio.

Fórmulas más importantes

El tiempo de respuesta del sistema (en el modelo M/M/1) se calcula empleando la ecuación (1).

$$R = \frac{\rho/\lambda}{1-\rho} = \frac{1/\mu}{1-\rho} = \frac{1}{\mu-\lambda} \qquad (1)$$

Donde λ es la tasa de llegada y μ es el tiempo de servicio.

El factor de utilización del sistema es el que se observa en (2).

$$\rho = \frac{\lambda}{\mu} \qquad (2)$$

Para que el sistema sea estable el factor de utilización tiene que ser menor que 1 (Trivedi, 2001).

6.5.1. Modelado de la aplicación y estimaciones

El modelado de **TeleOftalWeb**, siguiendo teoría de colas, puede verse en la Figura 84. El cliente A desde donde se accede a la aplicación es un navegador Web. A continuación se encuentra la red de área local (LAN) 1, Ethernet, que se conecta al Gateway (GW) 1. A través de GW1 se lleva a cabo la salida a Internet. La red de área amplia (WAN) envía y recibe la información de ambos extremos. En el otro lado de la conexión, está el servidor (donde se encuentra la bases de datos) que se conecta a la red WAN a través del GW 2.

A partir del modelo de colas de red de **TeleOftalWeb** (véase figura 84) estimamos los tiempos de respuesta medios, empleando las bases de datos comentadas en el apartado A. Asumimos que la tasa de llegada de una petición en nuestro sistema sigue una distribución de Poisson (Huang & Liou, 2007).

Las bases de datos empleadas en la evaluación son Oracle 10 g y 3 bases de datos nativas XML, dbXML 2.0, Xindice 1.2 and eXist 1.1.1, como ya se ha citado en lo largo de la presente Tesis Doctoral.

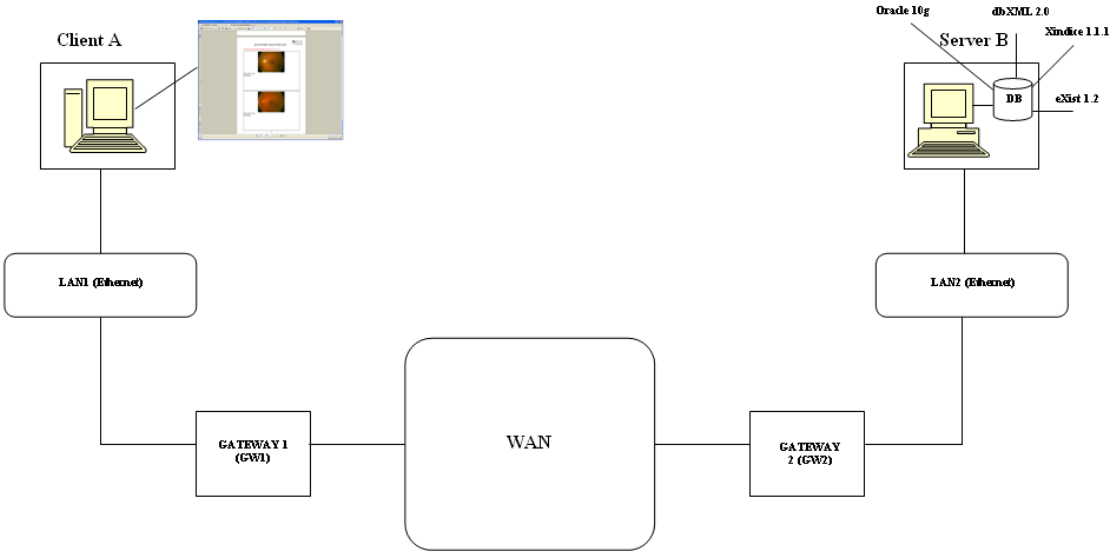


Figura 84. Modelado de la arquitectura de **TeleOftalWeb**. Fuente: propia.

Asumimos que la tasa de llegada de una petición en nuestro sistema sigue una distribución de Poisson (Huang & Liou 2007). Para modelar el tiempo medio de respuesta del sistema, es necesario estimar la tasa de llegada λ y el tiempo de servicio $1/\mu$ de cada cola del sistema. Se puede observar que hay 7 colas en el sistema. En la Tabla 7 se pueden observar las colas del sistema, su descripción, tipo y los tiempos de servicio asociados a las mismas.

Las colas Q_{GW1}, Q_{GW2}, Q_{CA} son de tipo CPU; la cola Q_{DB} es de tipo BD y las colas Q_{LAN1}, Q_{WAN}, Q_{LAN2} son tipo red. Los tiempos de servicio asociados a las colas de tipo CPU y red van a ser los mismos para los cuatro diferentes modelos de bases de datos. Lo que va a variar será la cola de la BD.

Es difícil estimar el tiempo de servicio de la red a la hora de transmitir HCEs. En este caso, aplicamos el modelo de Heidemann *et ál.* (Huang & Liou 2007). Este modelo estable un tiempo mínimo teórico de 129 ms para enviar una página Web de 5 kB a través de una conexión de 1 Mbps.

Para la cola de la base de datos tenemos que hacer una distinción entre: $Q_{DBOracle}$, Q_{DBXML}, $Q_{XINDICE}$ y Q_{EXIST}.

Abreviatura de la Cola	Descripción	Tipo de cola	Tiempos de servicio
Q_{CA}	Cola del cliente A	Cliente A (CPU)	T1 + T13
Q_{LAN1}	Cola de la red LAN1	Red	T2 + T12
Q_{GW1}	Cola del GW 1	GW1 (CPU)	T3 + T11
Q_{WAN}	Cola de la red WAN	Red	T4 + T10
Q_{GW2}	Cola del GW 2	GW2 (CPU)	T5 + T19
Q_{LAN2}	Cola de la red LAN2	Red	T6 + T8
Q_{DB}	Cola de la BD en el servidor B	BD	T7

Tabla 7. Colas del sistema vs. Tiempos de servicio. Fuente: propia.

El servidor es un Intel Pentium 4 a 2.8 GHz con Tecnología Hyper-Threading, 1 GB de memoria de sistema, sistema operativo Windows Server 2003 con Service Pack 2. El cliente es un Intel Pentium 4 a 1.2 GHz con tecnología Hyper-Threading, 1 GB de memoría y sistema operativo Windows XP Home Edition.

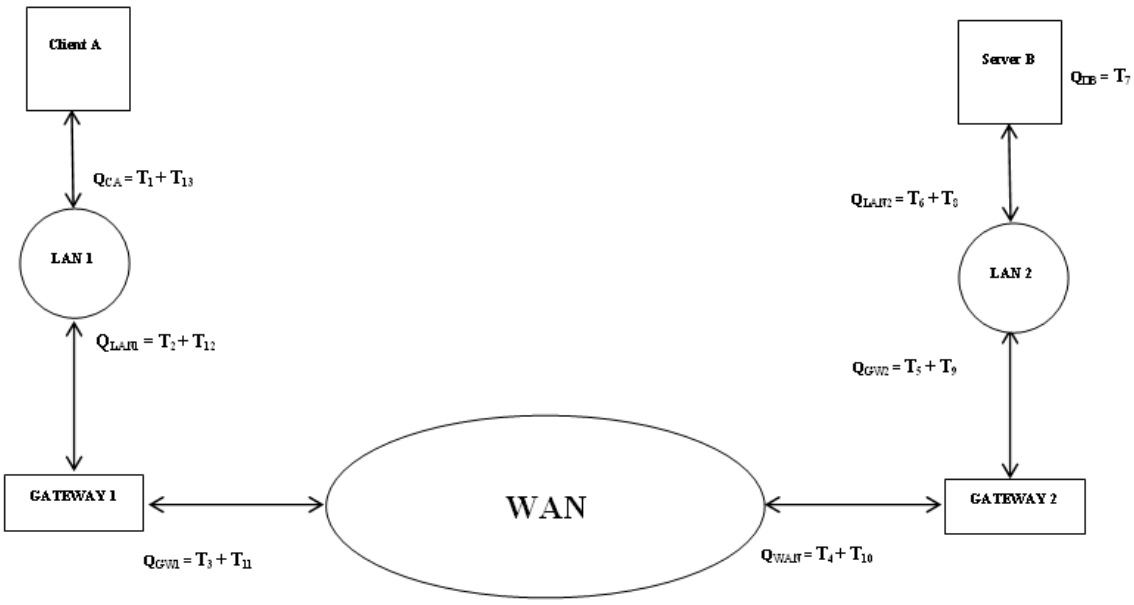


Figura 85. Modelo de colas de **TeleOftalWeb**. Fuente: propia.

El tiempo de servicio de cada una de las colas de base de datos va a depender del número de HCEs almacenados en el sistema y del número de consultas simultáneas a la base de datos. Para la base de datos Oracle 10g estimamos los tiempos de respuesta medios por consulta con diferente número de HCEs y consultas simultáneas. A través de *Oracle Automatic Database Diagnostic Monitor* podemos consultar dichos tiempos. En la Tabla 8 se muestran los tiempos de respuesta asociados a las consultas de las diferentes bases de datos propuestas. Para calcular los tiempos de consulta asociados a las bases de datos nativas XML se han realizado pruebas propias análogas a las realizadas por el benchmark XMach-1 (Böhme & Rahm, 2003) para ocho consultas.

El tiempo de respuesta para Oracle presenta una tendencia polinómica de grado 2 creciente y positiva en el almacenamiento (Ulloa, 2006). Para estimar el tiempo de servicio de la CPU de Host A y los tiempos de servicio asociados a los GWs realizamos las estimaciones oportunas a partir de las especificaciones técnicas de los fabricantes de los mismos.

NÚMERO DE HCES	TIEMPO MEDIO DE CONSULTA CON ORACLE 10G (S)	TIEMPO MEDIO DE CONSULTA CON DBXML (S)	TIEMPO MEDIO DE CONSULTA CON XINDICE (S)	TIEMPO MEDIO DE CONSULTA CON EXIST (S)
10	0,059	0,034	0,145	0,052
100	0,065	0,226	0,476	0,061
800	0,069	1,696	1,879	0,066
1000	0,074	2,121	2,541	0,072
5000	0,377	8,617	9,789	0,299

Tabla 8. Tiempo medio de consulta con diferentes BDs. Fuente: propia.

Se asume que en la interacción entre A y B emplean el protocolo de nivel de aplicación HTTPS. En la capa de transporte se emplea el protocolo TCP sobre IP.

Una vez estimados todos los tiempos de servicio asociados a las diferentes colas podemos calcular el tiempo medio de respuesta de una consulta desde el cliente A al servidor B. Así pues, el tiempo de respuesta medio de una consulta se puede calcular de la siguiente manera:

$R_{QUERY} = R_{CA} + R_{GW1} + R_{LAN1} + R_{WAN} + R_{LAN2} + R_{GW2} + R_{DB}$

donde, $R_{CA} = 1 / (\mu_{CA} - \lambda_{CA})$, $R_{GW1} = 1 / (\mu_{GW1} - \lambda_{GW1})$, $R_{LAN1} = 1 / (\mu_{LAN1} - \lambda_{LAN1})$, $R_{WAN} = 1 / (\mu_{WAN} - \lambda_{WAN})$, $R_{LAN2} = 1 / (\mu_{LAN2} - \lambda_{LAN2})$, $R_{GW2} = 1 / (\mu_{GW2} - \lambda_{GW2})$ y $R_{DB} = 1 / (\mu_{DB} - \lambda_{DB})$

$\mu_{CA} = 1/ (T_1 + T_{13})\ s^{-1}$

$\mu_{GW1} = 1/ (T_2 + T_{12})\ s^{-1}$

$\mu_{LAN1} = 1/ (T_3 + T_{11})\ s^{-1}$

$\mu_{WAN} = 1/ (T_4 + T_{10})\ s^{-1}$

$\mu_{LAN2} = 1/ (T_5 + T_9)\ s^{-1}$

$\mu_{GW2} = 1/ (T_6 + T_8)\ s^{-1}$

$\mu_{DB} = 1/T_7\ s^{-1}$

Donde T_i es el tiempo de servicio asociada a las diferentes colas.

Consideramos que:

$T_1 = T_8 = 0{,}5$ s en todos los casos de estudio.

$T_{10} = T_{12} = T_3 = T_5 = 0{,}005$ s en todos los casos de estudio

Calculamos los tiempos de respuesta del sistema empleando las diferentes bases de datos para distintas tasas de llegada (consultas/s), dependiendo del nº de HCEs intercambiados por hora. Por ejemplo, para 20 HCEs/hora tenemos que:

$\mu_{CA} = 1/ (T_1 + T_{13}) = 1/(0{,}5 + 0{,}129) = 1{,}589\ s^{-1}$, $\lambda_{CA} = 0{,}0076\ s^{-1}$

$\mu_{GW1} = 1/ (T_2 + T_{12}) = 1/(0,129 + 0,005) = 7,462\ s^{-1}$, $\lambda_{GW1} = 0,0076\ s^{-1}$

$\mu_{LAN1} = 1/ (T_3 + T_{11}) = 1/(0,005 + 0,129) = 7,462\ s^{-1}$, $\lambda_{LAN1} = 0,0076\ s^{-1}$

$\mu_{WAN} = 1/ (T_4 + T_{10}) = 7,462\ s^{-1}$, $\lambda_{WAN} = 0,0076\ s^{-1}$

$\mu_{LAN2} = 1/ (T_5 + T_9) = 7,462\ s^{-1}$, $\lambda_{LAN2} = 0,0076\ s^{-1}$

$\mu_{GW2} = 1/ (T_6 + T_8) = 1/(0,129 + 0,5) = 1,589\ s^{-1}$, $\lambda_{GW2} = 0,0076\ s^{-1}$

$\mu_{DB} = 1/T_7\ s^{-1}$, $\lambda_{DB} = 0,0076\ s^{-1}$

T_7 va a depender del modelado de base de datos empleado.

6.6. Programas de medición del tiempo de carga

En esta sección se presentan las alternativas que se presentan para realizar las mediciones de los tiempos de carga de la aplicación. Además se explicará el funcionamiento de una de las aplicaciones que se ha utilizado para realizar las mediciones.

6.6.1. Alternativas para realizar las pruebas

Hay varias formas de medir el tiempo de carga de una Web. Una posibilidad consiste en incluir en el código de la aplicación una función que haga de temporizador y que mida el tiempo que se tarda en completar la operación que pide el usuario. Esta forma no es muy adecuada porque al introducir código en la aplicación se están introduciendo más operaciones que harán variar el tiempo y además no es muy eficaz, por tener que incluir el código en todas las páginas de la aplicación que se quieren medir (Firebug 2008).

Con el fin de mejorar el tiempo de carga de una Web, existen una serie de programas que miden el tiempo que tarda en cargarse cada parte de una página Web, y que permiten obtener mediciones de forma rápida y sin influir en el tiempo de carga. Estas aplicaciones varían según el navegador utilizado. La comparativa se va a realizar para dos navegadores, por lo que serían necesarios dos programas distintos.

El navegador MF tiene una serie de complementos con los que se ofrece al usuario una serie de nuevas funcionalidades. Un programa muy completo es Firebug (Firebug 2008). Este programa no sólo ofrece la capacidad de medir el tiempo de carga de una Web, sino que ofrece una serie de herramientas de desarrollo. Existe otro

programa, más sencillo, que sólo realiza la medición del tiempo de carga de páginas Web. Este programa se llama *Life Of Request Info* (LORI). Este complemento permite ver de manera rápida el tiempo que tarda en cargarse la Web y el tamaño de los datos que se han cargado.

El navegador Internet Explorer también tiene una serie de complementos que permiten hacer las mediciones. Estos programas no son tan completos como los del navegador Firefox, pero permiten hacer las mediciones que se necesitan. Entre ellos están Web Development Helper (Web Development 2008) y DebugBar (DebugBar 2008). Ambos pretenden ofrecer las funcionalidades que ofrece Firebug para Firefox, aunque no son tan completos.

Otra forma de realizar las medidas, es hacerlo directamente desde páginas Web que permiten hacer mediciones de tiempos de carga de páginas Web. Estas páginas parecen una buena solución, pero no permiten hacer muchas de las pruebas que se realizan y que miden el tiempo de acceso a la BD.

Para realizar las pruebas se decidió usar las aplicaciones Firebug para MF y Web Development Helper para IE. Estas son las aplicaciones que nos ofrecen una mejor información sobre los tiempos de carga de una página Web.

6.6.2. La aplicación Firebug

Como se ha comentado la aplicación Firebug, es una aplicación que ofrece una serie de funcionalidades para el desarrollo de páginas Web. Una de estas funciones es la medición del tiempo de carga de una página. En este apartado se va a explicar la forma en la que se realizan las mediciones con esta aplicación. La aplicación que se utiliza para Internet Explorer es similar, y se utiliza de la misma manera.

Firebug es un complemento de MF que permite saber mientras navegamos, el tiempo de carga de las páginas (Firebug 2008). Para ello, al ejecutar la aplicación se abre una ventana donde se puede elegir la función que quieres usar. Para este proyecto sólo se va a utilizar la medición de tiempos de carga. En la figura 86 y la figura 87 se puede ver el aspecto de la aplicación.

Figura 86. Interfaz Firebug, carga de páginas. Fuente: propia.

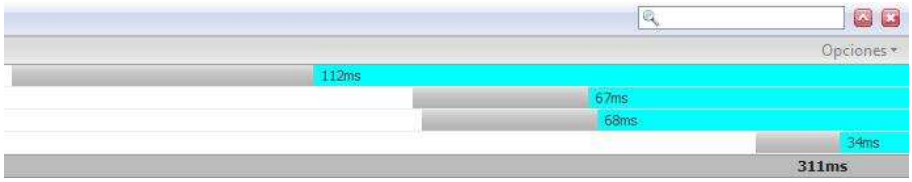


Figura 87. Interfaz Firebug, tiempos de carga. Fuente: propia.

Al cargar las páginas, se cargan distintos elementos como archivos JSP, imágenes, etc. Con Firebug se puede ver cuántos elementos se cargan, cuánto tarda en cargarse cada uno y el tiempo total de carga de la página Web. El tiempo de carga se da en milisegundos, con lo que se obtiene la precisión que se necesita para las pruebas del presente trabajo.

Todo funciona automáticamente, por lo que para hacer las pruebas, sólo hay que navegar por las páginas que se quieren medir y anotar los resultados que nos da la aplicación.

7

Resultados

7.1. INTRODUCCIÓN 266
7.2. MÓDULO GESTOR 266
7.3. MÓDULO USUARIO 268
7.4. EXPERIENCIA Y EVALUACIÓN DE LA APLICACIÓN 273
7.5. COMPARATIVA DE DIFERENTES VERSIONES DE LA APLICACIÓN 278
7.5.1. TIEMPOS DE RESPUESTA DE LA APLICACIÓN EMPLEANDO MODELOS DE COLAS 278
7.5.2. TIEMPOS DE RESPUESTA DE LA APLICACIÓN. PRUEBAS PROPIAS 280

7.1. Introducción

Una vez explicados los métodos que se van a usar en esta Tesis Doctoral, en el presente capítulo se resumen los resultados obtenidos. En los dos primeros apartados se muestran los módulos gestor y usuario de la aplicación **TeleOftalWeb** (De la Torre *et ál.* 2007; De la Torre *et ál.* 2008; De la Torre *et ál.* 2009). En el tercer apartado se incluyen los resultados obtenidos en la experiencia práctica realizada con dicha aplicación. En el cuarto apartado aparecen las comparativas a nivel de tiempos medios de carga de las cuatro versiones de **TeleOftalWeb** (De la Torre *et ál.* 2010).

7.2. Módulo gestor

El gestor puede acceder a la plataforma Web (figura 88) a través de cualquier navegador Web introducciendo *login* y *password.* En la figura 89, se pueden ver los usuarios (médicos) de la aplicación. Las funciones del módulo gestor con las que se citan a continuación (véase **Apéndice B. Manual de usuario**):

- Crear nuevos usuarios.
- Mostrar toda la información de usuario (Nombre, Apellidos, DNI, Nº Colegiado, Dirección, Teléfono, etc.).
- Borrar usuarios.
- Modificar la información de los médicos.
- Mostrar las estadísticas de los usuarios.
- Mostrar los HCE de cada médico registrado en la plataforma.
- Buscar usuarios por diferentes criterios tales como apellidos, número de identificación, tipo de usuario y/o colegiado.

Figura 88. Pantalla inicial del módulo gestor de la aplicación. Fuente: propia.

En la figura 89 se observa un listado de los usuarios del sistema: gestores, médicos especialistas y pacientes. En la parte derecha de la pantalla aparecen las acciones que se pueden realizar, que son: abrir la ficha de un usuario o gestor, eliminarlo, modificar sus datos, crear nuevos usuarios, ver los HCE asociados al usuario y ver sus estadísticas (donde aparecen el número de HCE compartidos con otros especialistas y los suyos propios). En la parte izquierda de la pantalla aparece un menú de búsqueda atendiendo a diferentes criterios como son: Apellidos, Número de Colegiado, DNI y tipo de usuario.

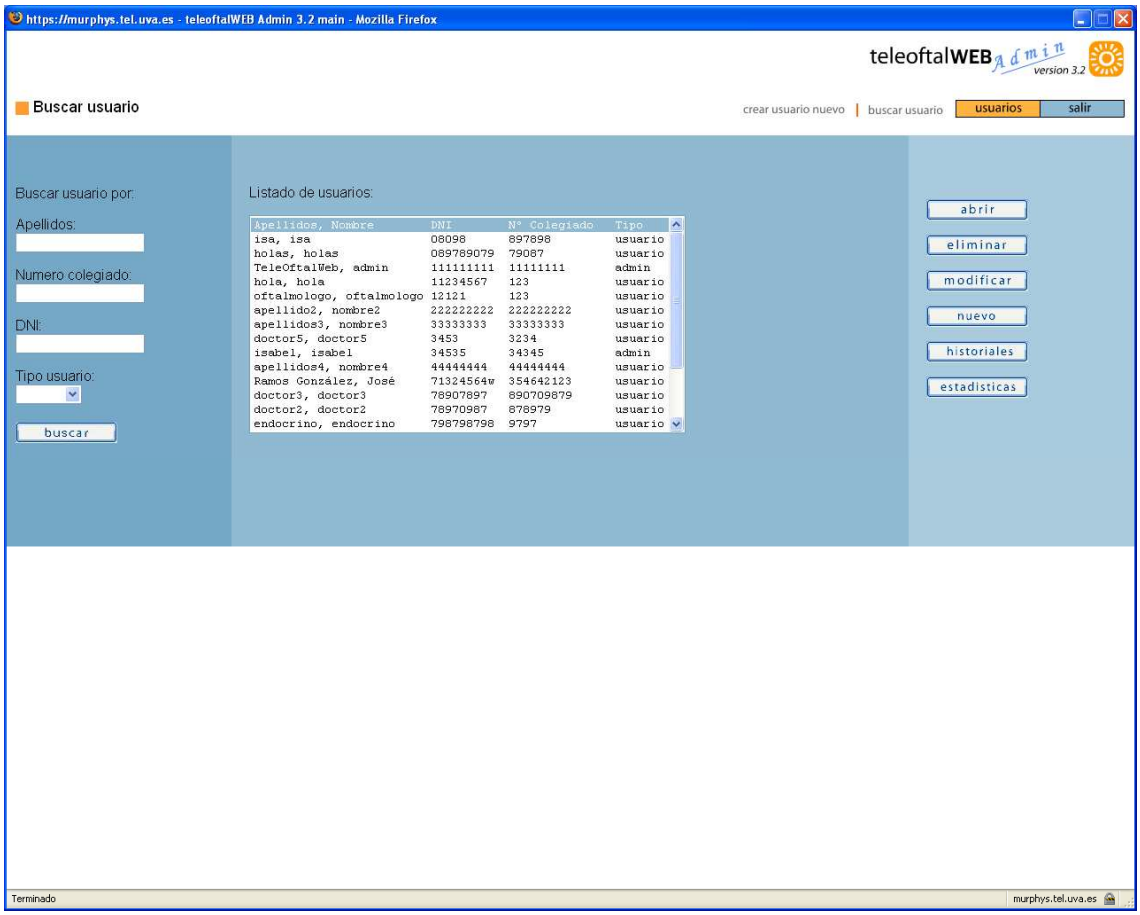


Figura 89. Usuarios de la aplicación **TeleOftalWeb**. Fuente: propia.

7.3. Módulo usuario

Los usuarios autorizados pueden acceder a la plataforma introducciendo su *login* y *password*. Previamente han debido ser registrados en el módulo gestor. Una vez que se han introducido en la aplicación pueden realizar las siguientes acciones.

- Crear nuevos HCE (Indicar las partes de los HC).
- Crear nuevas revisiones en un historial.
- Buscar diferentes HCE y revisiones.
- Añadir nuevas imágenes en los historiales. Dichas imágenes pueden tener cualquier tipo de formato (DICOM, JPEG, GIF, *BitMap* (BMP), TIFF, u otros).
- Editar y borrar imágenes. El editor de imágenes permite cambiar la forma, color, aumentar y disminuir la imagen. También permite modificar el brillo, contraste, fortmato RGB, cambio de escala, añadir texto, flechas, etc.

- Buscar imágenes de acuerdo a diferentes criterios como son: número de identificación de imagen, apellidos del paciente, fecha de creación de la imagen, u otros comentarios.

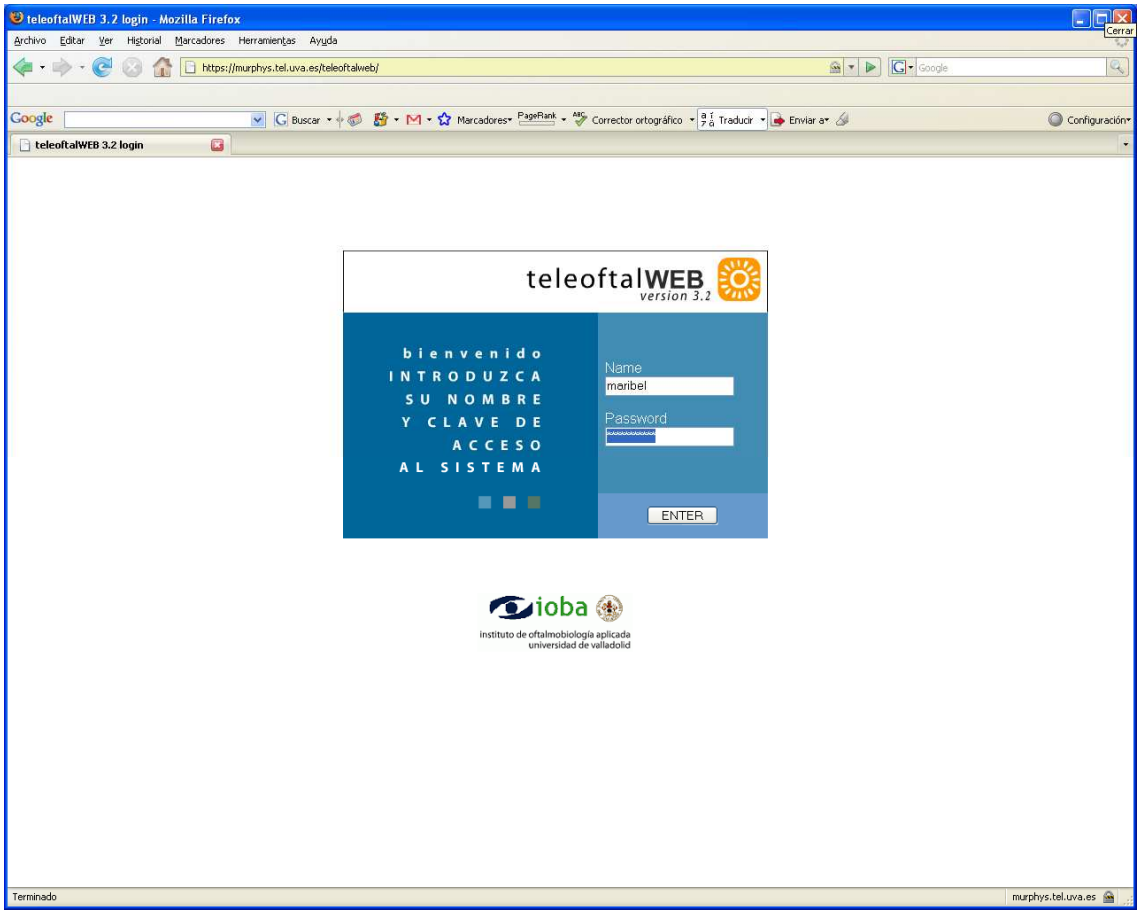


Figura 90. Pantalla inicial del módulo usuario de la aplicación. Fuente: propia.

Una explicación más detallada de todas estas acciones del módulo de usuario se pueden ver en el **Apéndice B. Manual de usuario.**

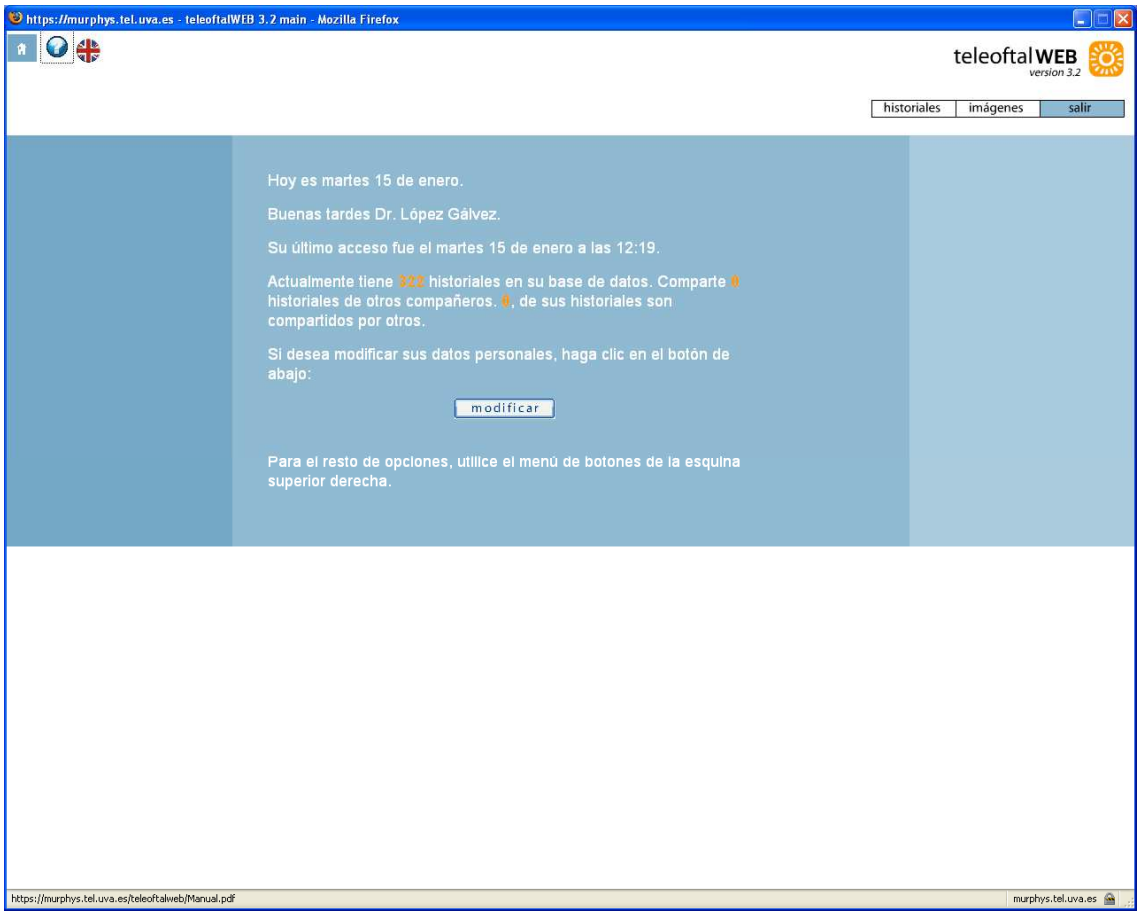


Figura 91. Pantalla de bienvenida de la aplicación. Fuente: propia.

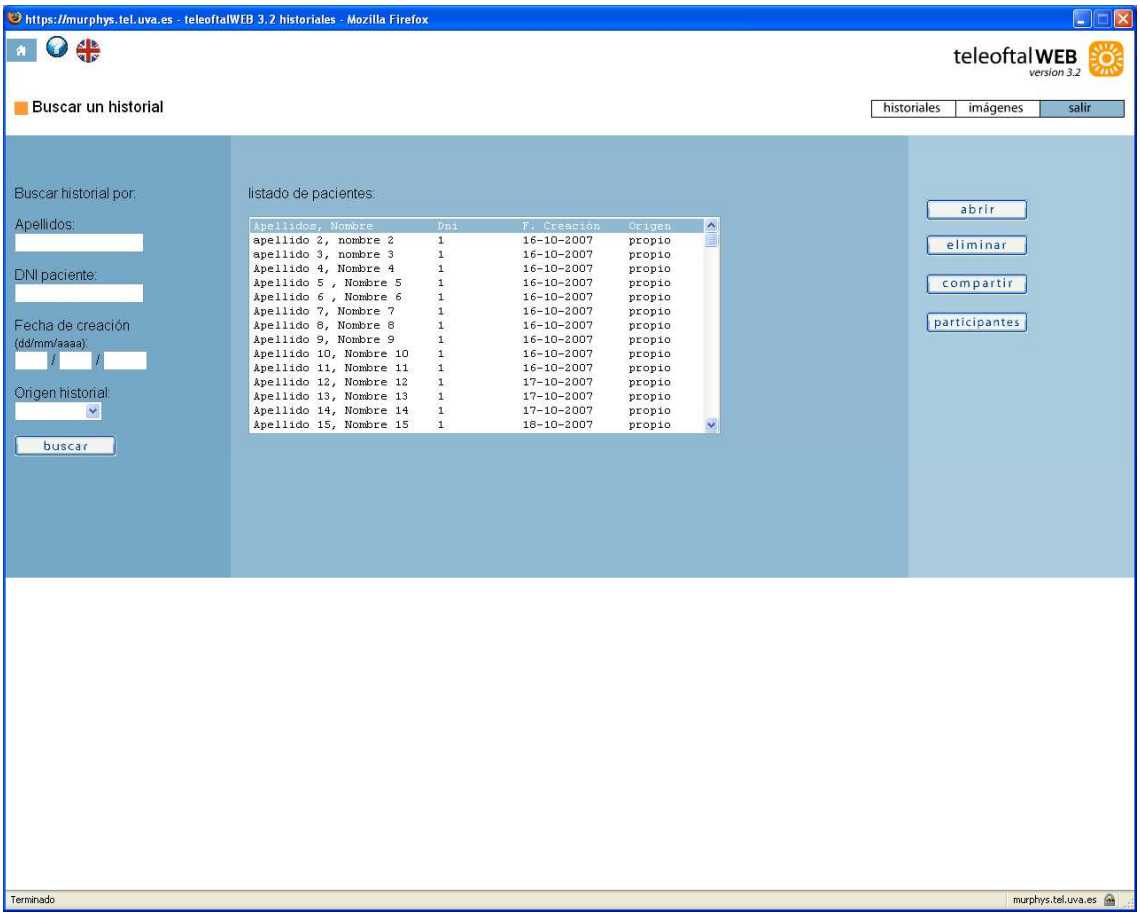


Figura 92. Historiales de un médico que emplea la aplicación. Fuente: propia.

En el módulo de imágenes, podemos añadir nuevas imágenes a un HCE. Una vez añadida, mediante el botón “abrir” o haciendo doble clic sobre la imagen deseada, podremos acceder a la pantalla que se observa en la figura 93, en la que se muestra la imagen ampliada y se pueden realizar varias operaciones sobre la misma. En la parte de la derecha de la pantalla nos saldrán una serie de botones que nos permitirán: mediante el botón “editada” abrir la imagen editada, para ver los cambios que se han realizado en ella. También, podemos eliminar la imagen del historial, retroceder a la pantalla anterior, o bien podemos deshacer todos los cambios que hayamos hecho a la imagen mediante el botón “resetear”, o podemos guardar los cambios realizados pulsando el botón “confirmar” (véase **Apéndice B. Manual de usuario**).

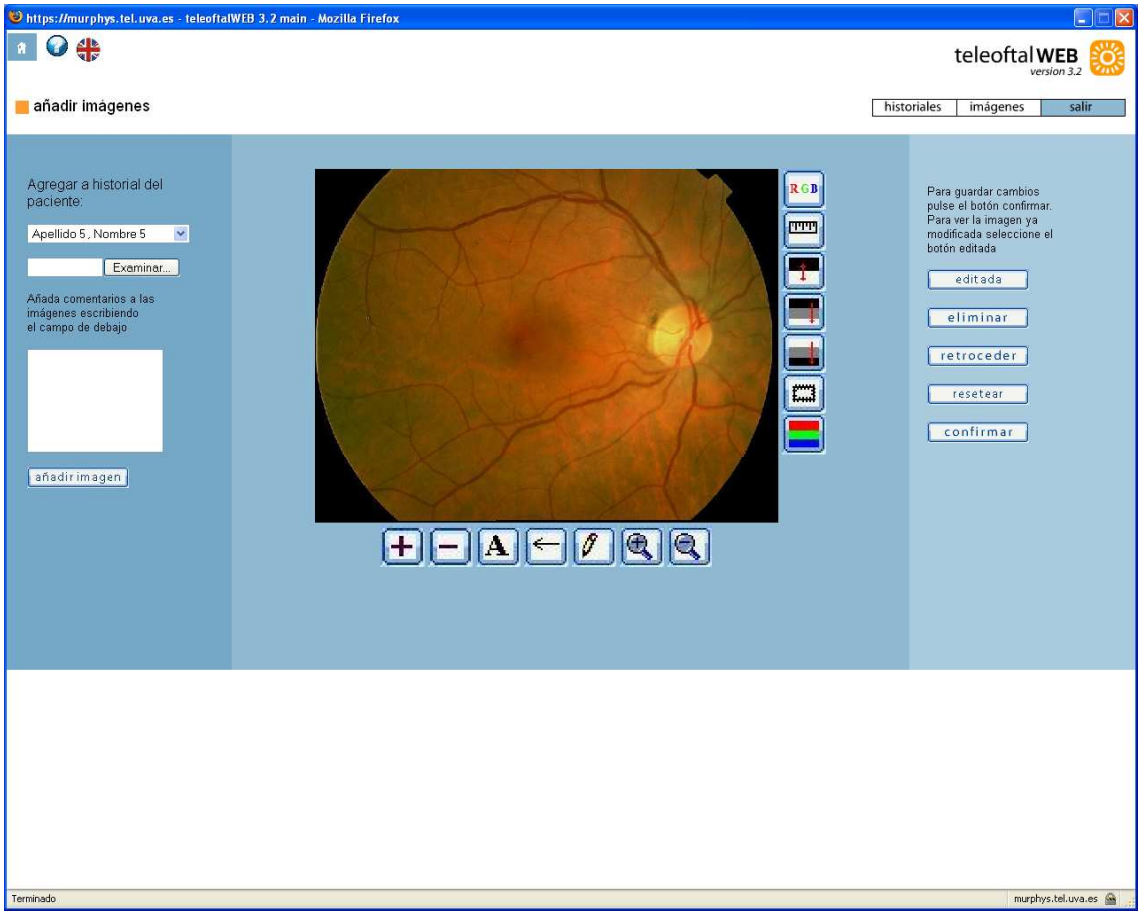


Figura 93. Editor de imagen. Fuente: propia.

También se pueden realizar búsquedas de las imágenes asociadas a un HCE. Para ello tenemos 4 posibles criterios de búsqueda: por historial, por identificador, por fechas de creación y por diagnóstico o comentarios. Una vez incluidos los términos de búsqueda en el recuadro correspondiente se pulsaría el botón "buscar" y aparecería una pantalla como la que aparece en la figura 94 (véase **Apéndice B. Manual de usuario**).

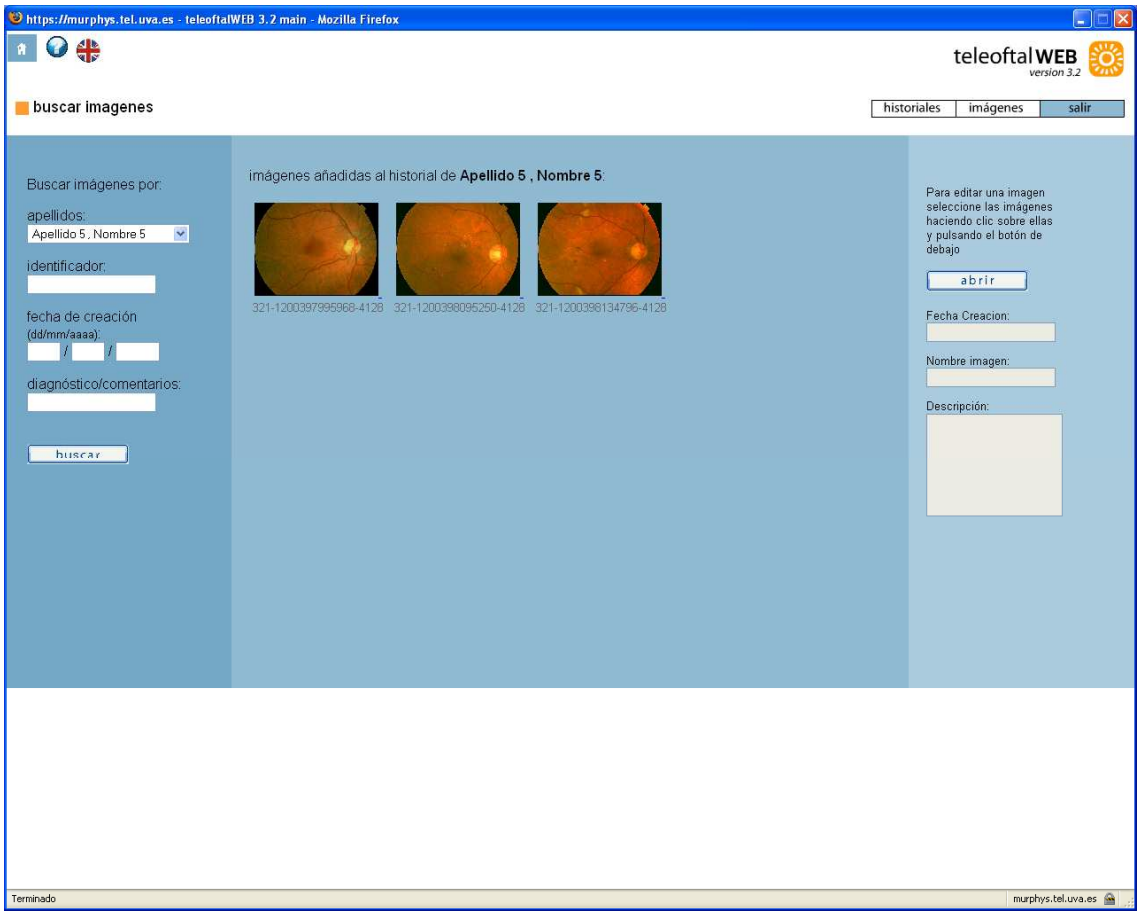


Figura 94. Buscador de imágenes de la aplicación. Fuente: propia.

7.4. Experiencia y evaluación de la aplicación

TeleOftalWeb ha sido probada por oftalmólogos del IOBA. Actualmente, hay más de 1000 HCE y 2000 retinografías que han sido introducidos durante seis meses. Todos los pacientes son diabéticos y participaron en el programa de telemedicina para el *screening* de la RD que se realizó en los pueblos de la provincia de Valladolid durante los años 1994-1999.

Los médicos emplearon la aplicación con diferentes navegadores Web. El HCE también puede ser asociado con retinografías en cualquier formato digital e imprimirse en PDF (véase figuras 95 y 96). El proceso de introducir un HC con dos revisiones en el sistema fue de aproximadamente diez minutos por paciente.

Una encuesta con diez cuestiones sobre la usabilidad de la aplicación fue realizada a los seis médicos que emplearon el sistema. Se empleo *System Usability Scale* (SUS) para realizar los cuestionarios. En la figura 97, se puede observar la

encuesta Web sobre usabilidadad de TeleOftalWeb realizada a los médicos que emplearon la aplicación. Los resultados de la misma se aprecian en la Tabla 9.

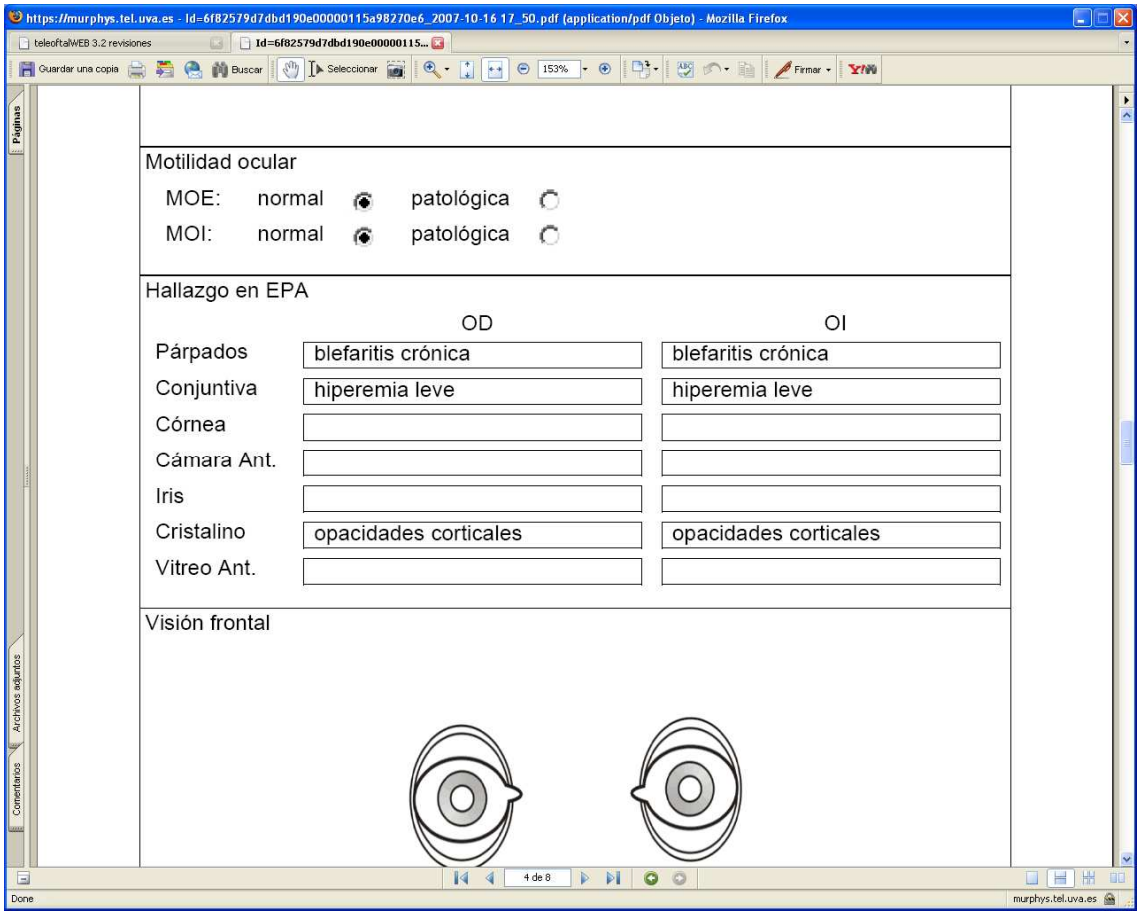


Figura 95. Historial de un paciente en formato PDF. Fuente: propia.

Además, empleamos el lenguaje *Extensible Stylesheet Language Formatting Objects* (XSL-FO) para formatear los datos XML. Se trata de un documento XML en el que se especifica cómo se van a formatear unos datos para presentarlos en pantalla, papel u otros medios. Uno de los formatos de salida de los HCEs y las retinografías es PDF. El proceso para crear los documentos PDF suele ser mediante la transformación del documento XML que contenga los datos a presentar con una hoja de estilos *Extensible* XSLT. De esta forma los datos (XML) se independizan del formato que proporcionará la hoja de transformación XSLT.

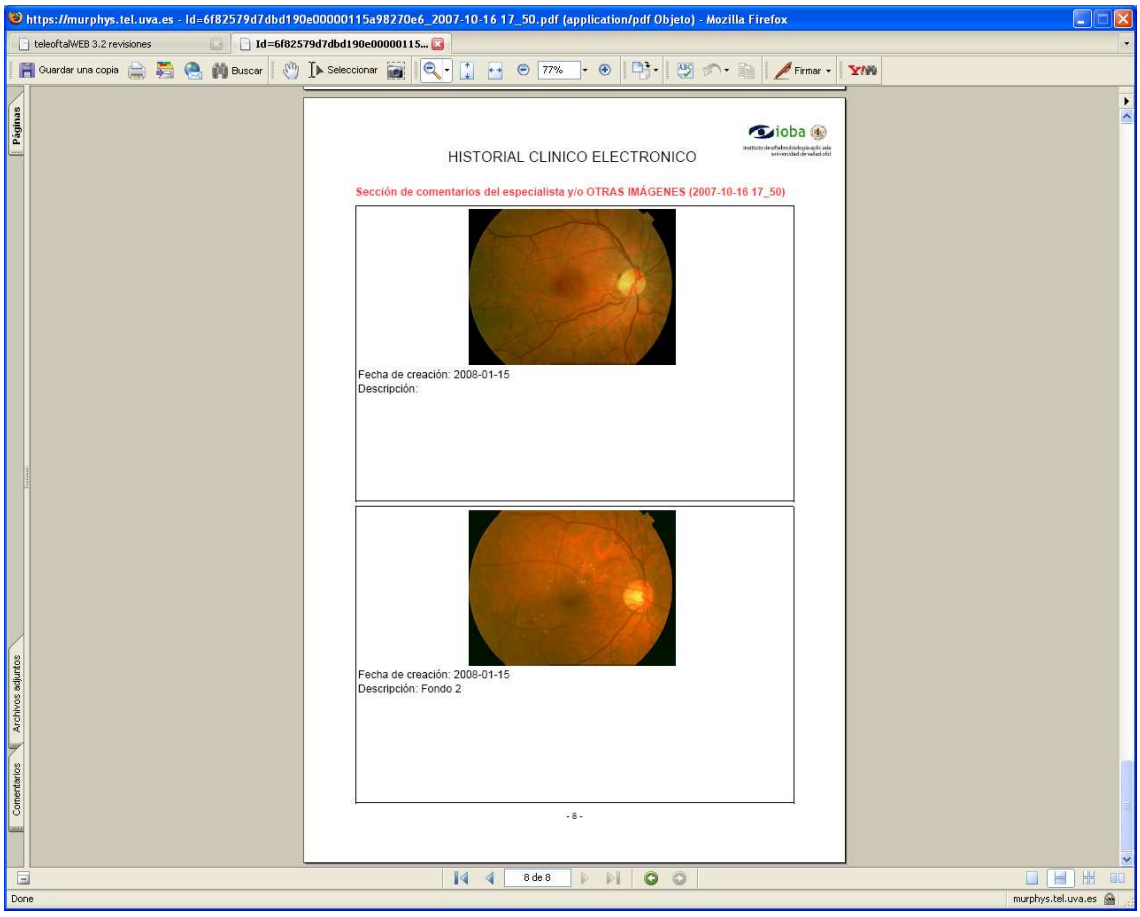


Figura 96. Historial en PDF con imágenes DICOM. Fuente: propia.

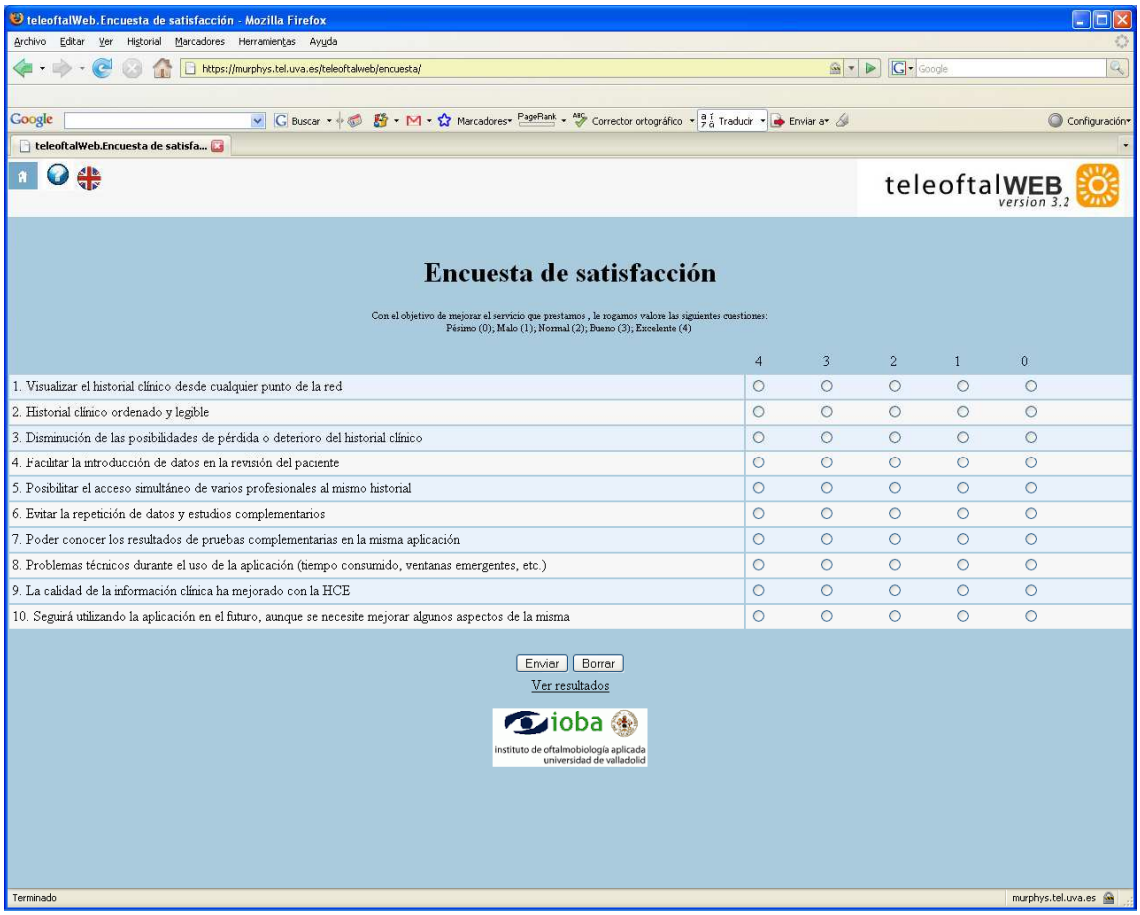


Figura 97. Encuesta de satisfacción de la aplicación. Fuente: propia.

Preguntas\Médicos	1	2	3	4	5	6
1. Visualizar el historial desde cualquier punto de la red	4	4	4	4	4	4
2. HC ordenado y legible	3	4	4	3	4	4
3. Disminución de las posibilidades de pérdida o deterior del HC	3	4	4	4	3	3
4. Facilitar la introducción de datos en la revisión del paciente	3	3	3	4	3	3
5. Posibilita el acceso simultáneo de varios profesionales al mismo HC	4	3	3	3	4	4
6. Evita la repetición de datos y estudios complementarios	4	4	4	4	4	4
7. Poder conocer los resultados de pruebas complementarias en la misma aplicación	4	4	4	4	4	2
8. Problemas técnicos durante el uso de la aplicación (tiempo consumido, ventanas emergentes, etc.)	2	2	1	2	2	4
9. La calidad de la información clínica ha mejorado con la HCE	2	2	2	2	2	2
10. Seguirá utilizando la aplicación en el futuro, aunque se necesiten mejorar algunos aspectos de la misma	4	4	3	4	4	4
SUS score	52,5	50	52,5	50	50	50

Tabla 9. Resultados de la encuesta de usabilidad de la aplicación. Fuente: propia.

Después de analizar los resultados obtenidos en la misma (véase Tabla 9), se pudo observar que la aplicación Web es útil y fácil de emplear para los médicos. Esto se

debe a que el valor de *SUS score* es superior a 50 en cada una de las encuestas individuales formuladas a cada uno de los usuarios del sistema.

7.5. Comparativa de diferentes versiones de la aplicación

7.5.1. Tiempos de respuesta de la aplicación empleando modelos de colas

En este apartado, calculamos el tiempo de respuesta del sistema con los diferentes modelos de bases de datos empleando modelos de colas. En la Tabla 10 se pueden ver los tiempos de respuesta del sistema en relación al nº de HCEs/hora intercambiados empleando las bases de datos comentadas con anterioridad.

Nº DE HCES / HORA	20	100	200	350	550	750	1000	5000
R_{CA}	0,631	0,640	0,651	0,669	0,719	0,723	0,761	4,765
R_{GW1}	0,134	0,134	0,135	0,135	0,137	0,137	0,139	0,164
R_{LAN1}	0,134	0,134	0,135	0,135	0,137	0,137	0,139	0,164
R_{WAN}	0,134	0,134	0,135	0,135	0,137	0,137	0,139	0,164
R_{LAN2}	0,134	0,134	0,135	0,135	0,137	0,137	0,139	0,164
R_{GW2}	0,631	0,640	0,651	0,669	0,719	0,723	0,761	4,765
R_{ORACLE}	0,059	0,065	0,074	0,069	0,070	0,071	0,075	0,377
R_{DBXML}	0,034	0,226	0,765	0,967	1,254	1,565	2,121	8,617
$R_{XINDICE}$	0,145	0,476	0,765	1,754	2,11	2,312	2,541	9,789
R_{EXIST}	0,052	0,061	0,066	0,067	0,068	0,0687	0,072	0,299
$R_{TORACLE}$ (1 Mb)	1,857	1,883	1,911	1,952	2,060	2,070	2,155	10,975
R_{TDBXML} (1 Mb)	1,832	2,045	2,642	2,95	3,663	4,319	7,222	
$R_{TXINDICE}$ (1 Mb)	1,943	2,3	2,644	3,997	5,64	6,452	10,66	
R_{TEXIST} (1 Mb)	1,85	1,879	1,909	1,95	2,05	2,062	2,153	10,698
$R_{TORACLE}$ (100 Kb)	4,488	4,576	4,686	4,859	5,350	5,394	5,804	
R_{TDBXML} (100 Kb)	4,464	4,739	5,417	5,857	6,953	7,643	10,87	
$R_{TXINDICE}$ (100 Kb)	4,575	4,993	5,417	6,904	8,93	9,776	14,309	
R_{TEXIST} (100 Kb)	4,482	4,572	4,684	4,857	5,347	5,391	5,802	

Tabla 10. Tiempos de respuesta con las diferentes BDs. Fuente: propia.

El tiempo de respuesta medio total para el sistema empleando cada una de las bases de datos es la suma de los tiempos de respuesta asociados a las siete colas del sistema (Cliente A, GW1, LAN1, WAN, LAN2, GW2, BD). Se puede observar a partir de la Tabla 10 que con Xindice 1.2 se obtienen los mayores tiempos de respuesta del sistema tanto para 100 Kb como para 1 Mb. Los menores tiempos se obtienen empleando la base de datos eXist 1.1.1, seguida de Oracle 10g. La elección de una base de datos u otra dependerá de factores como: la disponibilidad, el mantenimiento, la facilidad de instalación, el precio, entre otros.

Además podemos afirmar que se necesita como mínimo un ancho de banda de 1Mb para poder intercambiar 5000 HCEs utilizando Oracle 10g o eXist 1.1.1.

Gráficamente, se pueden observar en la figura 98 los tiempos de respuesta en función del nº de HCEs intercambiados por hora en el caso de 1 Mb y la figura 99 para 100 Kb.

Se puede apreciar a través de la figura 98 que a partir de 200 HCEs intercambiados por hora los tiempos de respuesta del sistema empleando la base de datos Xindice incrementan de forma considerable hasta unos 1300 HCEs por hora (donde el sistema deja de ser estable). En el caso de dbXML 2.0, dichos tiempos empiezan a incrementar a partir de 350 HCEs por hora hasta unos 1500 HCEs por hora. Para el caso de Oracle 10g, los tiempos comienzan a incrementar a partir de 1500 HCEs por hora hasta 7500 HCEs por hora y para eXist 1.1.1 a partir de 1550 HCEs hasta 7550 HCEs por hora.

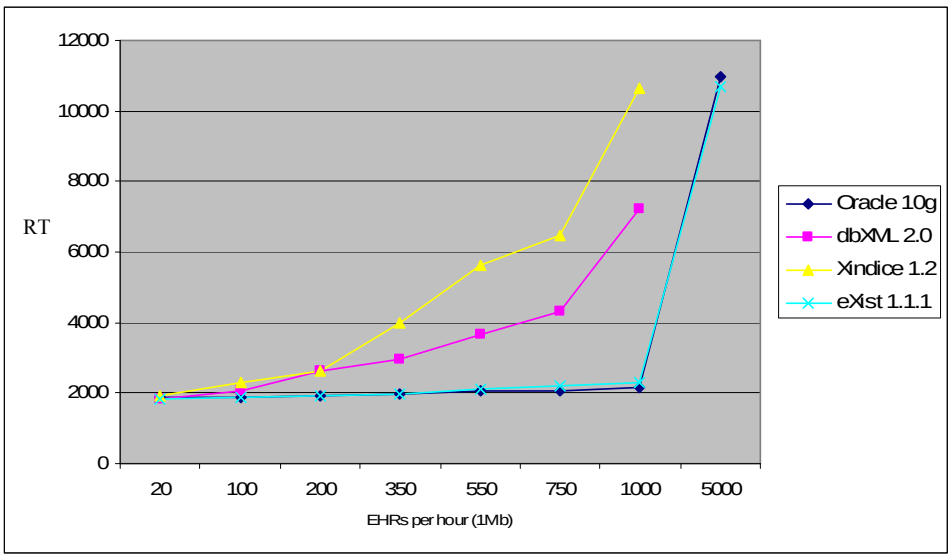


Figura 98. Tiempos de respuesta para 1 Mb con diferentes modelos de BDs en ms. Fuente: propia.

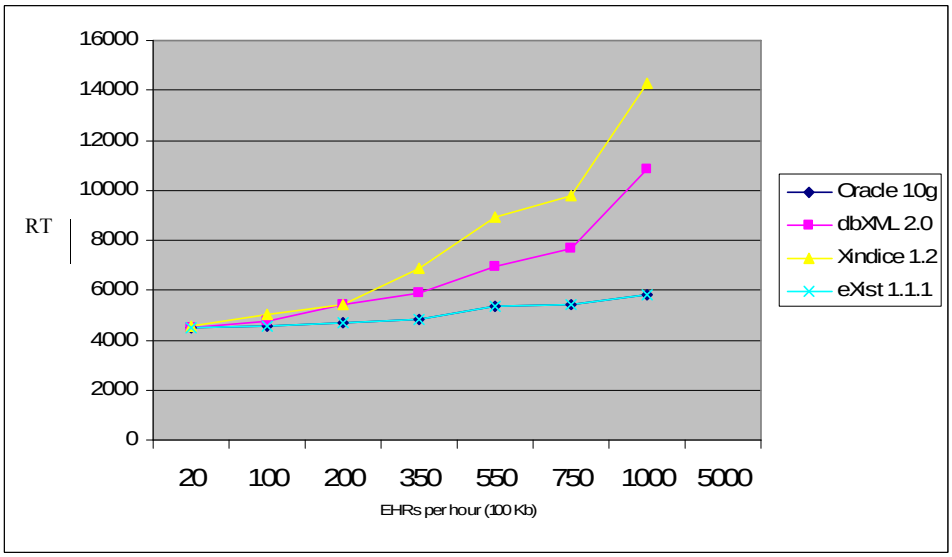


Figura 99. Tiempos de respuesta para 100 Kb con diferentes modelos de BDs. en ms. Fuente: propia.

7.5.2. Tiempos de respuesta de la aplicación. Pruebas propias

Se han desarrollado cuatro versiones de la aplicación **TeleOftalWeb** empleando las BDs que siguen: dbXML 2.0 + MySQL 5.0, eXist 1.1.1 + MySQL 5.0, Xindice 1.2 + MySQL 5.0, y por último Oracle 10g con diferente número de usuarios, número de historiales y navegadores (MF e IE), desde dentro y fuera de la red de la Escuela Técnica Superior de Ingenieros de Telecomunicación (ETSIT) de la Universidad de

Valladolid.

A continuación, se muestra una comparativa de las diferentes BDs empleadas en esta Tesis en relación a una serie de características como son el almacenamiento, mantenimiento, precio, entre otras (Zhang 2003). En la Tabla 11, se puede observar que la BD eXist es la más adecuada en cuanto a la combinación de factores como son el almacenamiento de datos, las actualizaciones, funciones básicas, otras herramientas, mantenimiento, precio, instalación y facilidad de uso.

	ALMACENAMIENTO	*UPDATE*	FUNCIONES BÁSICAS	OTRAS HERRAMIENTAS	MANTENIMIENTO	PRECIO	INSTALACIÓN Y FACILIDAD DE USO	TOTAL
Oracle 9i	3	2	3	3	3	1	1	25
Xindice	2	2	1	1	1	3	3	19
dbXML	2	2	1	1	1	3	3	19
eXist	2	2	2	3	1	3	3	26

Tabla 11. Comparativa de bases de datos nativas XML con Oracle (Zhang 2003).

Muy bueno: 3 No malo: 2 No competitivo: 1

Para la realización de las pruebas se ha utilizado la versión 2.0 del navegador MF, y la versión 7.0 de IE. La velocidad media de conexión de los ordenadores de la red fuera de la ETSIT es de 10 Mbps. En el momento en el que se han tomado las medidas se ha supuesto la red no congestionada. Los usuarios acceden simultáneamente al sistema. Además se supone que los historiales almacenados tienen al menos dos imágenes.

Se han realizado 8 pruebas distintas. Cada una de las pruebas se ha llevado a cabo desde diferentes máquinas dentro de la red ETSIT y desde distintos ordenadores (más concretamente 6) fuera de la red en distintos momentos. Las pruebas realizadas son las siguientes (De la Torre *et ál.* 2010):

1. Tiempo medio de carga de la página principal.
2. Tiempo medio de carga de búsqueda de historiales.
3. Tiempo medio de introducción de un historial con imágenes de exploración.
4. Tiempo medio en añadir una imagen JPEG.
5. Tiempo medio en añadir una imagen DICOM.
6. Tiempo medio de carga búsqueda de imágenes.
7. Tiempo medio de carga de creación de un PDF.

8. Tiempo medio en eliminar un historial.

Cada una de las pruebas se ha realizado para las cuatro BDs, en los dos navegadores Web. Las pruebas se realizan variando el nº de HCE almacenados en la BD, desde 25 hasta más de 1000, y el nº de usuarios accediendo al sistema (desde 5 hasta 100, aunque en el presente trabajo sólo se muestran hasta 20). Los tiempos medios globales aparecen explicados detalladamente en (De la Torre *et ál.* 2010)). Se añadieron más de 20 usuarios al sistema, pero se observó que los resultados obtenidos para más de 20 usuarios eran similares a los de 20, por lo que no aparecen en esta memoria. Así se tienen una serie de variables, en función de las cuáles se van a comparar los diferentes modelos de BDs empleados en la aplicación **TeleOftalWeb**. Las medidas se muestran en tablas (de la Tabla 12 a la Tabla 59), en las que se indica a qué prueba corresponden. Los tiempos medios aparecen expresados en segundos.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	0,710	0,195	0,692	0,187	0,625	0,250	0,766	0,261
50	0,735	0,287	1,980	0,297	1,050	0,313	2,262	0,321
100	1,070	0,310	4,701	0,343	1,469	0,328	4,656	0,401
250	2,825	0,333	9,083	0,345	2,902	0,344	10,203	0,452
500	4,431	0,665	15,273	0,734	4,593	0,672	18,594	0,683
> 1000	8,124	1,162	33,293	1,189	8,182	1,125	36,265	1,209

Tabla 12. Tiempo medio de carga de la página principal desde un ordenador de la red ETSIT y 5 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	0,760	0,203	0,781	0,301	0,828	0,250	0,828	0,298
50	0,820	0,295	1,991	0,387	1,040	0,359	2,344	0,365
100	1,113	0,365	4,760	0,456	1,484	0,328	4,828	0,501
250	2,837	0,355	10,080	0,498	2,953	0,360	10,219	0,600
500	4,602	0,670	15,265	0,876	4,625	0,641	18,937	0,876
> 1000	8,431	1,193	33,785	1,245	8,281	1,094	36,985	1,322

Tabla 13. Tiempo medio de carga de la página principal desde un ordenador de la red ETSIT y 10 usuarios en el sistema. Fuente: propia.

La Tabla 13 muestra el tiempo de carga de la página principal desde un ordenador de la red ETSIT y 10 usuarios conectados simultáneamente. Se puede apreciar que los menores tiempos de carga se obtienen con las BDs: Exist + MySQL. Cuando el número de HCE aumenta el tiempo es mayor, sobretodo ésto se puede apreciar a partir de 1000 HCE en el sistema. En la Tabla 14 se puede observar algo similar con 20 usuarios.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	0,780	0,215	0,786	0,323	0,984	0,250	0,870	0,302
50	1,156	0,309	2,115	0,456	1,160	0,328	2,365	0,456
100	1,184	0,370	4,788	0,598	1,578	0,328	4,703	0,498
250	2,840	0,360	10,125	0,786	2,906	0,375	10,273	0,503
500	4,587	0,688	16,082	0,897	4,657	0,695	18,723	0,698
> 1000	8,062	1,185	33,888	1,234	8,225	1,015	36,870	1,113

Tabla 14. Tiempo medio de carga de la página principal desde un ordenador de la red ETSIT y 20 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	1,279	0,930	1,654	1,012	1,199	0,912	1,404	1,102
50	2,006	1,000	1,931	1,108	1,202	0,940	2,056	1,132
100	2,212	1,010	3,807	1,203	1,744	0,996	4,106	1,234
250	3,086	1,102	8,903	1,453	2,953	1,092	8,966	1,298
500	4,703	1,455	18,094	1,546	4,762	1,366	18,108	1,509
> 1000	8,765	1,962	32,275	2,067	8,402	1,717	34,848	1,988

Tabla 15. Tiempo medio de carga de la página principal desde un ordenador de fuera de la red ETSIT y 5 usuarios en el sistema. Fuente: propia.

La Tabla 15 muestra el tiempo de carga de la página principal desde un ordenador fuera de la red ETSIT y 5 usuarios conectados simultáneamente. Se puede ver que los menores tiempos de carga se obtienen con las BDs: Exist + MySQL. Cuando el número de HCE aumenta el tiempo es mayor, sobretodo ésto se puede apreciar a partir de 1000 HCE en el sistema. En la Tabla 16 se puede observar algo similar con 10 usuarios.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	1,311	0,955	1,654	1,023	1,163	0,915	1,506	1,109
50	2,130	1,002	1,931	1,145	1,336	1,017	2,088	1,199
100	2,208	1,003	3,807	1,235	1,799	0,942	4,245	1,307
250	2,945	1,155	8,903	1,454	2,946	1,031	8,985	1,365
500	4,625	1,472	18,094	1,667	4,761	1,399	18,331	1,687
> 1000	8,903	2,234	32,235	2,156	8,418	2,127	34,470	2,098

Tabla 16. Tiempo medio de carga de la página principal desde un ordenador de fuera de la red ETSIT y 10 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	1,403	0,968	1,690	1,067	1,366	0,922	1,532	1,200
50	2,156	1,016	1,947	1,187	1,641	0,997	2,092	1,234
100	2,243	1,018	3,992	1,356	2,081	1,003	4,140	1,389
250	3,075	1,062	8,972	1,578	3,258	1,006	8,990	1,411
500	4,831	1,496	18,305	1,759	4,780	1,403	18,160	1,721
> 1000	8,952	2,339	32,902	2,234	8,477	2,266	34,782	2,190

Tabla 17. Tiempo medio de carga de la página principal desde un ordenador de fuera de la red ETSIT y 20 usuarios en el sistema. Fuente: propia.

Al igual que con las medidas anteriores, se puede observar a partir de las Tablas 17 y 18 que los mejores tiempos medios se obtienen con las BDs eXist junto con MySQL. A continuación, se muestran el resto de medidas de tiempos medios calculados.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	0,515	0,156	0,890	0,233	0,188	0,219	0,816	0,300
50	0,953	0,296	2,156	0,304	0,203	0,391	2,296	0,389
100	1,370	0,390	4,281	0,457	1,797	0,454	4,375	0,434
250	3,546	0,921	10,265	1,002	3,703	0,891	10,109	0,982
500	6,953	1,640	20,203	1,765	6,875	2,093	20,223	1,987
> 1000	13,415	3,390	40,812	3,401	13,108	3,907	41,672	4,008

Tabla 18. Tiempo medio de carga búsqueda de historiales desde un ordenador de la red ETSIT y 5 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	0,546	0,178	0,906	0,245	0,234	0,219	0,922	0,345
50	0,984	0,315	2,281	0,365	0,217	0,360	2,297	0,433
100	1,640	0,396	4,171	0,481	1,781	0,438	4,400	0,498
250	3,578	0,965	10,625	1,004	3,656	0,953	10,750	1,083
500	6,925	1,756	20,484	1,878	6,781	1,859	20,547	2,084
> 1000	13,402	3,475	40,920	3,345	13,313	3,781	41,725	4,143

Tabla 19. Tiempo medio de carga búsqueda de historiales desde un ordenador de la red ETSIT y 10 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	0,562	0,182	0,910	0,249	0,219	0,250	0,890	0,398
50	1,000	0,363	2,303	0,398	0,228	0,375	2,295	0,478
100	1,609	0,403	4,290	0,677	1,766	0,516	4,408	0,533
250	3,846	1,015	10,880	1,145	3,734	1,031	10,982	1,245
500	6,987	1,988	20,675	1,977	6,812	1,937	20,922	2,356
> 1000	13,406	3,699	41,013	3,704	13,266	3,995	41,823	4,253

Tabla 20. Tiempo medio de carga búsqueda de historiales desde un ordenador de la red ETSIT y 20 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	0,718	0,249	1,031	0,252	0,833	0,217	0,954	0,276
50	1,287	0,453	2,131	0,489	1,133	0,312	2,267	0,378
100	2,734	0,660	4,125	0,789	1,933	0,566	4,124	0,456
250	4,187	1,165	10,218	0,988	3,810	0,982	10,214	1,083
500	6,711	2,522	20,296	2,022	7,161	2,237	20,461	2,143
> 1000	12,921	5,002	40,531	4,988	13,689	4,436	41,276	4,563

Tabla 21. Tiempo medio de carga búsqueda de historiales desde un ordenador de fuera de la red ETSIT y 5 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	0,859	0,263	1,046	0,344	0,852	0,200	1,083	0,298
50	1,250	0,508	2,140	0,672	1,213	0,468	2,262	0,402
100	3,093	0,665	4,170	0,799	1,840	0,552	4,166	0,494
250	3,671	1,203	10,315	1,032	3,965	1,097	10,225	1,243
500	6,718	2,675	20,587	2,290	7,157	2,004	20,438	2,345
> 1000	13,291	4,935	40,702	4,786	13,689	4,101	41,285	4,643

Tabla 22. Tiempo medio de carga búsqueda de historiales desde un ordenador de fuera de la red ETSIT y 10 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	0,937	0,277	1,053	0,405	0,883	0,206	1,004	0,302
50	1,297	0,595	2,156	0,699	1,262	0,580	2,270	0,443
100	3,234	0,766	4,199	0,873	1,900	0,753	4,203	0,602
250	3,593	1,211	10,352	1,197	3,807	1,016	10,273	1,534
500	6,830	2,596	20,605	2,487	7,203	2,132	20,465	2,456
> 1000	13,338	5,015	40,882	5,121	13,870	4,325	41,403	4,734

Tabla 23. Tiempo medio de carga búsqueda de historiales desde un ordenador de fuera de la red ETSIT y 20 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	0,578	0,220	0,406	0,243	0,481	0,234	0,906	0,287
50	0,621	0,346	1,093	0,354	0,797	0,375	1,190	0,321
100	0,625	0,187	2,015	0,274	1,484	0,218	4,266	0,434
250	0,640	0,203	4,890	0,398	1,483	0,281	9,328	0,465
500	0,650	0,355	15,580	0,698	1,484	0,313	18,062	0,567
> 1000	0,650	0,401	29,296	0,899	1,484	0,375	35,578	0,578

Tabla 24. Tiempo medio de introducción de un historial con imágenes de exploración desde un ordenador de la red ETSIT y 5 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	0,593	0,287	0,405	0,298	0,812	0,237	0,938	0,298
50	0,625	0,350	1,109	0,401	0,871	0,406	1,266	0,354
100	0,625	0,260	2,093	0,345	1,454	0,339	4,532	0,493
250	0,645	0,215	4,843	0,493	1,500	0,315	9,797	0,512
500	0,625	0,360	15,640	0,723	1,453	0,335	18,281	0,573
> 1000	0,672	0,412	29,290	0,988	1,485	0,395	35,602	0,599

Tabla 25. Tiempo medio de introducción de un historial con imágenes de exploración desde un ordenador de la red ETSIT y 10 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	0,593	0,320	0,452	0,304	0,782	0,234	0,921	0,302
50	0,640	0,373	1,266	0,522	0,875	0,515	1,295	0,392
100	0,625	0,305	2,066	0,598	1,453	0,415	4,685	0,502
250	0,656	0,295	4,965	0,603	1,515	0,395	9,666	0,543
500	0,660	0,377	15,785	0,879	1,502	0,399	18,290	0,591
> 1000	0,685	0,425	29,780	1,003	1,485	0,398	35,580	0,603

Tabla 26. Tiempo medio de introducción de un historial con imágenes de exploración desde un ordenador de la red ETSIT y 20 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	3,041	1,596	2,171	1,698	2,991	1,534	2,336	1,601
50	3,328	1,609	2,560	1,701	2,995	1,564	3,216	1,655
100	3,118	1,703	3,343	1,867	3,237	1,695	4,976	1,711
250	3,359	2,564	5,812	2,256	3,182	2,212	9,781	2,234
500	3,310	2,807	9,921	2,768	2,980	2,703	18,767	2,843
> 1000	3,312	3,965	28,593	4,009	3,042	3,845	35,527	3,910

Tabla 27. Tiempo medio de introducción de un historial con imágenes de exploración desde un ordenador de fuera de la red ETSIT y 5 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	3,046	1,580	1,875	1,700	2,977	1,570	2,460	1,698
50	3,356	1,690	2,562	1,745	2,992	1,595	3,367	1,709
100	3,118	1,715	3,350	1,889	3,105	1,695	4,980	1,793
250	3,203	2,362	5,911	2,298	2,969	2,233	9,795	2,254
500	3,301	2,855	10,022	2,890	2,974	2,805	18,802	2,934
> 1000	3,412	3,996	29,982	4,112	2,971	3,903	35,732	4,098

Tabla 28. Tiempo medio de introducción de un historial con imágenes de exploración desde un ordenador de fuera de la red ETSIT y 10 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	3,046	1,635	2,105	1,711	3,010	1,588	2,472	1,712
50	3,303	1,695	2,584	1,766	3,059	1,609	3,371	1,823
100	3,234	1,765	3,390	1,901	2,986	1,706	4,995	1,834
250	3,481	2,414	6,003	2,309	3,595	2,265	9,803	2,365
500	3,320	2,915	10,150	2,987	3,015	2,898	18,815	3,032
> 1000	3,339	4,032	29,990	4,203	3,120	3,955	35,862	4,123

Tabla 29. Tiempo medio de introducción de un historial con imágenes de exploración desde un ordenador de fuera de la red ETSIT y 20 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	1,109	0,896	1,632	0,911	0,344	1,078	1,796	1,133
50	2,015	1,187	2,654	1,124	0,472	1,250	2,532	1,245
100	2,921	1,099	4,780	1,133	3,266	1,578	4,875	1,533
250	4,655	1,577	10,875	1,476	5,175	1,766	11,343	1,722
500	7,968	2,709	20,843	2,890	8,219	2,828	20,981	2,434
> 1000	13,904	5,484	41,255	4,998	14,497	4,781	41,938	4,531

Tabla 30. Tiempo medio en añadir imagen JPEG desde un ordenador de la red ETSIT y 5 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	2,015	0,956	1,623	1,099	0,344	1,099	2,188	1,192
50	2,078	1,196	2,718	1,198	0,437	1,288	2,859	1,302
100	2,640	1,175	4,678	1,203	3,234	1,596	5,047	1,601
250	4,578	1,603	10,437	1,501	5,218	1,788	11,125	1,834
500	7,828	2,715	20,843	2,346	8,422	2,865	21,000	2,476
> 1000	13,912	5,431	41,870	5,504	14,131	4,886	41,980	4,643

Tabla 31. Tiempo medio en añadir imagen JPEG desde un ordenador de la red ETSIT y 10 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	2,156	0,922	1,654	1,101	0,344	1,103	1,813	1,221
50	2,015	1,175	2,814	1,245	0,687	1,293	2,872	1,353
100	2,937	1,209	4,864	1,308	3,219	1,603	4,999	1,723
250	4,952	1,611	10,765	1,544	5,203	1,796	11,414	1,902
500	8,265	2,888	21,001	2,434	8,219	2,903	21,015	2,524
> 1000	13,937	5,505	41,722	5,689	14,547	4,875	41,960	4,706

Tabla 32. Tiempo medio en añadir imagen JPEG desde un ordenador de la red ETSIT y 20 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	6,827	5,390	6,249	5,356	6,900	5,483	6,110	5,423
50	6,939	5,765	6,899	5,674	7,231	5,567	7,709	5,674
100	7,530	5,953	8,856	5,980	8,154	5,719	9,834	6,023
250	10,249	6,022	14,921	6,188	10,115	5,017	14,976	6,049
500	12,663	8,153	25,078	8,324	12,902	7,512	25,499	7,023
> 1000	18,842	11,932	50,765	10,343	19,603	11,742	45,912	10,345

Tabla 33. Tiempo medio en añadir imagen JPEG desde un ordenador de fuera de la red ETSIT y 5 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	6,949	5,403	5,978	5,425	6,982	5,499	5,909	5,567
50	6,999	5,788	6,984	5,754	7,301	5,560	7,164	5,787
100	7,640	6,030	8,893	6,140	8,148	5,865	9,870	6,253
250	10,078	6,120	15,001	6,218	9,999	6,032	14,983	6,453
500	12,665	8,252	25,122	8,532	12,904	7,892	25,503	7,623
> 1000	18,857	12,165	50,232	11,132	19,563	11,822	45,960	10,951

Tabla 34. Tiempo medio en añadir imagen JPEG desde un ordenador de fuera de la red ETSIT y 10 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	7,124	5,411	6,024	5,515	6,995	5,487	5,937	5,674
50	7,593	5,790	6,991	5,874	7,304	5,577	7,172	5,983
100	7,702	6,015	8,931	6,242	7,912	5,895	9,883	6,455
250	10,092	6,099	15,103	6,310	10,675	6,015	14,990	6,698
500	12,802	8,159	25,234	8,623	12,991	8,035	25,507	7,782
> 1000	19,001	12,223	50,222	12,112	19,870	11,865	46,080	11,210

Tabla 35. Tiempo medio en añadir imagen JPEG desde un ordenador de fuera de la red ETSIT y 20 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	2,125	0,824	1,761	0,877	0,344	1,218	1,875	1,243
50	2,249	1,076	2,630	0,898	0,500	1,078	2,521	1,324
100	2,859	1,256	4,722	1,022	3,250	1,328	5,079	1,465
250	4,937	1,709	10,992	1,199	5,109	1,860	11,375	1,832
500	8,593	2,562	20,265	2,129	7,968	2,766	21,375	2,353
> 1000	13,503	5,530	41,995	4,988	14,392	4,844	41,906	4,721

Tabla 36. Tiempo medio en añadir imagen DICOM desde un ordenador de la red ETSIT y 5 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	2,530	0,965	1,796	0,913	0,625	1,223	1,969	1,423
50	2,443	1,184	2,640	0,994	0,700	1,241	2,616	1,511
100	3,078	1,299	4,764	1,133	3,343	1,416	5,234	1,598
250	5,187	1,712	10,937	1,289	5,125	1,875	11,328	1,903
500	8,437	2,666	20,937	2,345	8,291	2,766	21,281	2,653
> 1000	13,609	5,565	42,003	5,324	14,469	4,906	41,906	4,932

Tabla 37. Tiempo medio en añadir imagen DICOM desde un ordenador de la red ETSIT y 10 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	2,343	1,002	1,793	1,032	0,656	1,219	1,970	1,622
50	2,409	1,212	2,815	1,245	0,703	1,323	2,898	1,750
100	3,109	1,306	4,905	1,354	3,391	1,457	5,098	1,632
250	5,281	1,765	10,998	1,578	5,256	1,899	11,495	2,043
500	8,562	2,803	20,688	2,564	8,375	2,905	21,293	2,831
> 1000	13,937	5,565	41,998	5,423	14,531	4,995	41,995	5,024

Tabla 38. Tiempo medio en añadir imagen DICOM desde un ordenador de la red ETSIT y 20 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	43,297	37,937	40,021	38,887	40,184	39,564	39,826	38,754
50	44,396	40,465	39,702	39,789	40,318	40,373	40,444	40,342
100	44,749	41,060	42,465	41,253	42,525	40,940	45,351	41,235
250	47,952	41,980	48,432	41,689	44,265	41,525	48,455	41,985
500	47,830	44,065	59,171	45,673	46,330	43,622	58,656	42,644
> 1000	52,412	50,105	78,609	53,987	52,721	47,875	78,882	46,976

Tabla 39. Tiempo medio en añadir imagen DICOM desde un ordenador de fuera de la red ETSIT y 5 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	43,532	38,031	39,115	38,924	40,348	39,666	39,336	38,922
50	44,399	40,488	39,827	40,924	41,657	40,388	40,537	40,731
100	45,671	41,222	43,003	41,868	42,676	40,980	45,440	41,674
250	47,671	41,975	48,570	42,253	44,213	41,602	48,506	42,045
500	48,688	44,122	58,203	45,987	46,156	43,782	57,515	42,921
> 1000	52,437	50,222	78,703	54,154	52,705	48,152	78,882	47,093

Tabla 40. Tiempo medio en añadir imagen DICOM desde un ordenador de fuera de la red ETSIT y 10 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	43,532	38,115	39,723	39,143	40,494	39,703	39,548	39,099
50	44,343	40,765	40,092	41,122	41,612	40,401	40,561	40,924
100	44,843	41,302	43,188	41,979	42,248	41,002	43,367	41,832
250	47,827	41,996	48,666	42,898	48,756	41,646	48,773	42,246
500	48,910	44,203	59,015	46,322	49,801	43,889	58,412	43,542
> 1000	52,462	50,125	78,755	54,121	52,993	49,322	78,991	48,687

Tabla 41. Tiempo medio en añadir imagen DICOM desde un ordenador de fuera de la red ETSIT y 20 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	1,156	0,156	1,040	0,167	0,360	0,391	1,250	0,345
50	1,883	0,294	2,405	0,353	0,713	0,515	2,409	0,465
100	2,655	0,412	4,421	0,541	2,265	0,594	4,719	0,622
250	4,467	0,850	10,823	0,879	4,901	1,062	11,031	1,098
500	8,051	1,965	20,663	2,033	8,140	2,093	21,010	2,134
> 1000	13,870	3,688	41,662	3,769	14,392	3,984	41,734	4,021

Tabla 42. Tiempo medio de carga búsqueda de imágenes desde un ordenador de la red ETSIT y 5 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	1,577	0,288	1,124	0,198	0,640	0,375	1,297	0,394
50	1,890	0,335	2,343	0,378	0,764	0,531	2,563	0,434
100	2,718	0,560	4,359	0,654	2,328	0,603	4,891	0,723
250	4,468	0,896	10,203	0,956	4,909	0,938	10,969	1,245
500	8,031	1,996	20,765	2,134	8,250	2,105	21,000	2,353
> 1000	13,869	3,722	41,520	3,976	14,500	3,991	41,802	4,342

Tabla 43. Tiempo medio de carga búsqueda de imágenes desde un ordenador de la red ETSIT y 10 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	1,984	0,403	1,132	0,211	0,672	0,422	1,297	0,412
50	1,983	0,466	2,467	0,398	0,797	0,546	2,548	0,454
100	3,229	0,620	4,455	0,855	3,406	0,615	4,892	0,831
250	4,499	0,903	10,675	1,244	4,984	1,072	11,062	1,321
500	7,953	2,090	20,995	2,355	8,485	2,116	21,112	2,532
> 1000	13,874	3,795	41,821	4,032	14,406	4,032	41,825	4,398

Tabla 44. Tiempo medio de carga búsqueda de imágenes desde un ordenador de la red ETSIT y 20 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	3,233	0,855	1,327	0,934	3,025	1,125	1,859	1,223
50	3,562	0,987	2,446	1,023	3,331	1,200	2,934	1,243
100	4,171	1,623	4,031	1,542	4,114	1,846	4,278	1,768
250	6,553	3,091	10,265	2,987	6,811	3,205	10,183	3,098
500	8,841	4,032	20,295	4,244	9,002	3,766	20,448	3,765
> 1000	14,202	4,959	41,360	5,044	15,492	3,847	41,689	4,098

Tabla 45. Tiempo medio de carga búsqueda de imágenes desde un ordenador de fuera de la red ETSIT y 5 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	3,365	0,810	1,593	0,993	3,037	1,203	1,830	1,245
50	3,686	0,990	2,467	1,234	3,477	1,215	2,929	1,289
100	4,549	1,670	4,095	1,763	4,071	1,865	4,288	1,901
250	6,906	3,103	10,357	3,098	6,999	3,162	10,279	3,134
500	8,859	4,050	20,346	4,456	9,108	3,803	20,560	3,986
> 1000	14,171	4,962	41,355	5,245	15,495	4,009	41,722	4,143

Tabla 46. Tiempo medio de carga búsqueda de imágenes desde un ordenador de fuera de la red ETSIT y 10 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	3,748	0,886	1,565	1,031	3,119	1,170	1,861	1,343
50	3,981	1,013	2,537	1,354	3,421	1,216	3,013	1,398
100	4,561	1,705	4,103	1,873	4,149	1,876	4,299	2,093
250	7,078	3,101	10,363	3,345	6,832	3,206	10,285	3,345
500	8,872	4,165	20,472	4,567	9,203	3,999	20,572	4,034
> 1000	14,680	5,002	41,403	5,321	15,505	4,537	41,670	4,234

Tabla 47. Tiempo medio de carga búsqueda de imágenes desde un ordenador de fuera de la red ETSIT y 20 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	1,953	1,652	1,879	1,767	0,141	0,130	1,159	1,712
50	1,980	1,722	1,900	1,954	0,145	0,222	1,879	1,789
100	2,031	1,856	2,065	1,912	2,297	1,706	2,120	1,834
250	1,995	1,976	2,134	1,988	2,200	1,809	2,075	1,912
500	1,975	1,932	2,256	1,976	2,047	1,934	2,134	2,015
> 1000	2,005	1,951	2,240	1,991	2,047	1,912	2,567	2,098

Tabla 48. Tiempo medio de carga creación PDF desde un ordenador de la red ETSIT y 5 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	1,986	1,659	1,903	1,765	0,188	0,141	1,200	0,210
50	1,975	1,708	1,934	1,798	0,185	0,245	1,908	0,346
100	1,973	1,888	2,803	1,901	2,157	1,756	2,165	1,254
250	1,996	1,898	2,546	1,992	2,146	1,876	2,147	1,476
500	2,075	1,932	2,987	2,123	2,016	1,918	2,199	1,793
> 1000	2,118	1,976	2,995	2,314	2,084	1,923	2,606	1,912

Tabla 49. Tiempo medio de carga creación PDF desde un ordenador de la red ETSIT y 10 usuarios en el sistema.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	1,959	1,703	2,015	1,812	0,219	0,199	1,333	0,213
50	1,940	1,865	2,365	1,876	0,187	0,234	1,967	0,276
100	2,031	1,972	2,405	1,965	2,195	1,789	2,180	1,435
250	2,090	1,952	2,672	1,978	2,156	1,892	2,156	1,523
500	2,101	1,994	3,015	2,023	2,000	1,912	2,203	1,845
> 1000	2,096	1,929	3,001	2,097	2,172	1,974	2,598	2,011

Tabla 50. Tiempo medio de carga creación PDF desde un ordenador de la red ETSIT y 20 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	2,203	1,565	2,401	1,612	2,134	1,256	2,403	1,301
50	2,125	1,876	2,465	1,843	2,195	1,768	2,560	1,811
100	2,163	1,875	2,506	1,894	2,224	1,803	2,805	1,843
250	2,271	2,013	3,123	2,003	2,240	2,056	3,025	1,914
500	2,203	2,076	3,185	2,034	2,224	2,165	3,121	2,045
> 1000	2,203	2,054	3,174	2,061	2,193	2,199	3,154	2,135

Tabla 51. Tiempo medio de carga creación PDF desde un ordenador de fuera de la red ETSIT y 5 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	2,171	1,600	2,402	1,712	2,232	1,306	2,543	1,412
50	2,156	1,831	2,508	1,892	2,108	1,678	2,609	1,734
100	2,171	1,967	2,865	2,024	2,169	1,814	2,939	1,854
250	2,203	2,123	3,201	2,092	2,165	2,010	3,125	1,984
500	2,203	2,145	3,299	2,187	2,235	2,198	3,196	2,054
> 1000	2,198	2,175	3,287	2,276	2,219	2,200	3,202	2,197

Tabla 52. Tiempo medio de carga creación PDF desde un ordenador de fuera de la red ETSIT y 10 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	2,046	1,611	2,507	1,710	2,193	1,302	2,560	1,423
50	2,162	1,872	2,876	1,893	2,148	1,700	2,571	1,823
100	2,234	2,015	2,866	2,045	2,151	1,865	2,960	1,945
250	2,292	2,061	3,099	2,094	2,182	2,001	3,099	2,053
500	2,234	2,176	3,303	2,193	2,240	2,176	3,203	2,142
> 1000	2,231	2,188	3,315	2,204	2,240	2,198	3,214	2,201

Tabla 53. Tiempo medio de carga creación PDF desde un ordenador de fuera de la red ETSIT y 20 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	0,890	0,109	0,875	0,113	0,110	0,234	0,969	0,255
50	1,323	0,218	2,203	0,243	0,156	0,359	2,190	0,544
100	1,984	0,406	4,328	0,512	2,156	0,515	4,204	0,634
250	3,921	0,968	10,671	1,023	4,015	1,140	10,822	1,245
500	6,937	1,828	20,630	1,879	7,203	2,313	20,750	2,532
> 1000	13,302	4,265	40,312	4,343	13,502	4,547	41,062	4,655

Tabla 54. Tiempo medio en eliminar historial desde un ordenador de la red ETSIT y 5 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	0,952	0,125	0,859	0,143	0,188	0,234	1,046	0,298
50	1,328	0,265	2,281	0,299	0,190	0,361	2,281	0,601
100	2,203	0,550	4,460	0,609	2,094	0,520	4,562	0,698
250	3,921	0,990	10,171	1,289	4,097	1,140	10,828	1,465
500	7,031	1,965	20,625	1,978	7,172	2,365	20,953	2,876
> 1000	13,428	4,333	40,988	4,523	13,596	4,682	41,089	4,768

Tabla 55. Tiempo medio en eliminar historial desde un ordenador de la red ETSIT y 10 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	0,953	0,130	0,863	0,165	0,219	0,237	1,002	0,312
50	1,328	0,299	2,266	0,324	0,203	0,359	2,295	0,693
100	2,100	0,565	4,479	0,876	2,094	0,517	4,688	0,897
250	3,875	1,017	10,585	1,356	4,078	1,165	10,865	1,501
500	7,203	2,112	21,012	2,245	7,312	2,378	21,014	2,967
> 1000	13,312	4,582	41,023	4,867	13,620	4,691	41,102	4,875

Tabla 56. Tiempo medio en eliminar historial desde un ordenador de la red ETSIT y 20 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	1,015	0,218	0,937	0,303	1,065	0,228	0,972	0,301
50	1,531	0,265	2,165	0,365	1,479	0,330	2,137	0,332
100	2,009	0,662	4,046	0,699	2,197	0,754	4,257	0,654
250	3,906	1,202	10,031	1,103	4,299	1,297	10,210	0,988
500	7,262	2,615	19,921	2,567	7,760	2,608	20,544	1,988
> 1000	14,121	4,982	40,531	5,022	14,603	4,963	41,322	4,877

Tabla 57. Tiempo medio en eliminar historial desde un ordenador de fuera de la red ETSIT y 5 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	1,031	0,230	1,468	0,389	1,089	0,228	0,989	0,301
50	1,484	0,301	2,171	0,378	1,497	0,331	2,138	0,354
100	2,017	0,673	4,103	0,766	2,138	0,766	4,280	0,702
250	4,078	1,215	10,188	1,199	4,474	1,365	10,273	1,002
500	7,250	2,712	20,202	2,678	7,789	2,706	20,620	2,243
> 1000	13,937	5,002	40,670	5,122	14,670	4,965	41,330	4,901

Tabla 58. Tiempo medio en eliminar historial desde un ordenador de fuera de la red ETSIT y 10 usuarios en el sistema. Fuente: propia.

	NAVEGADOR IE				NAVEGADOR MF			
Nº HCE	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle	dbXML + MySQL	Exist + MySQL	Xindice + MySQL	Oracle
20	1,000	0,295	1,262	0,411	1,169	0,302	0,990	0,423
50	1,437	0,323	2,180	0,433	1,495	0,337	2,156	0,445
100	2,062	0,695	4,115	0,789	2,126	0,765	4,273	0,788
250	4,091	1,362	10,165	1,243	4,238	1,404	10,303	1,200
500	7,321	2,831	20,255	2,781	7,933	2,705	20,587	2,302
> 1000	14,203	5,013	40,685	5,150	14,791	4,990	41,360	5,002

Tabla 59. Tiempo medio en eliminar historial desde un ordenador de fuera de la red ETSIT y 20 usuarios en el sistema. Fuente: propia.

En todas las Tablas mostradas anteriormente, se ha podido comprobar que la versión más rápida es la que emplea el doble modelo de datos: eXist más MySQL.

A continuación, se muestran las gráficas obtenidas durante la realización de las

pruebas a las 4 versiones de la aplicación, en las cuales se verifica que eXist más MySQL es el modelo de datos más eficiente.

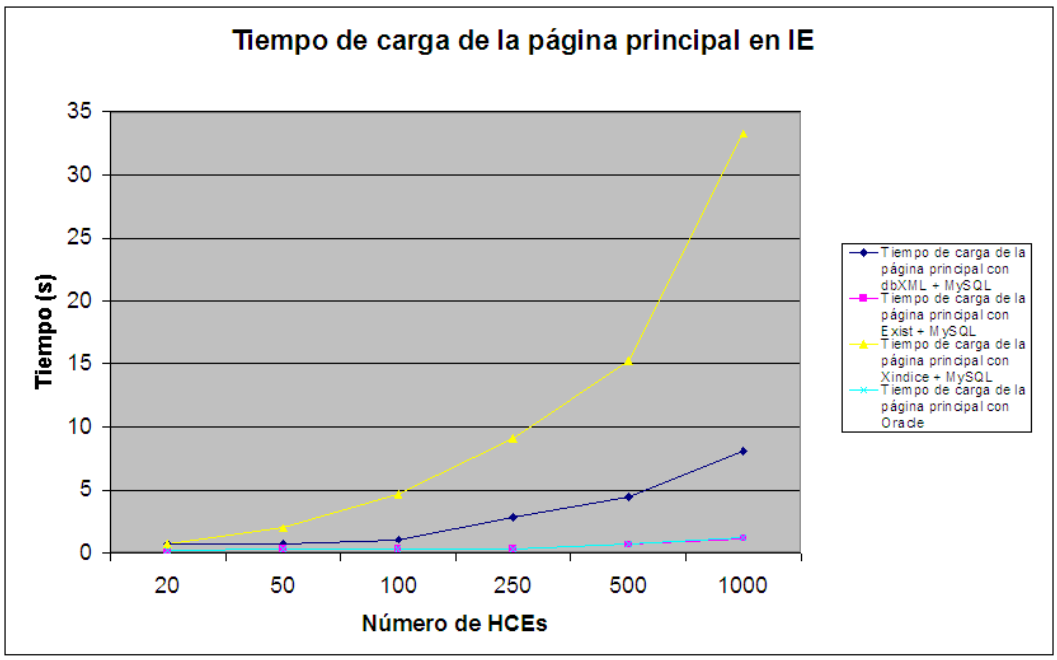


Figura 100. Tiempo medio de carga de la página principal en IE. Fuente: propia.

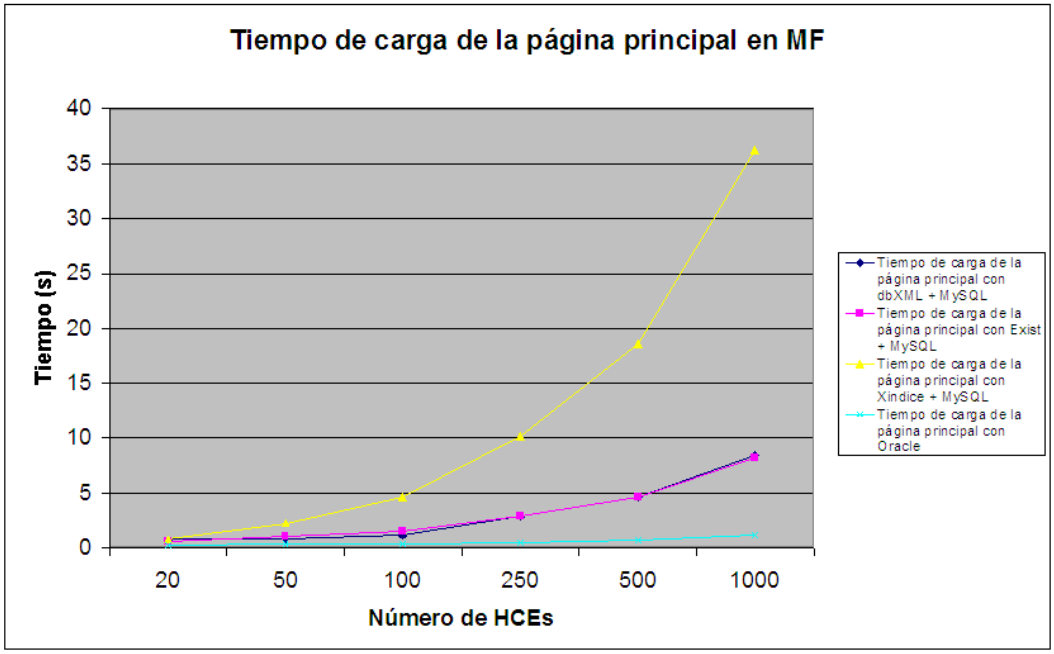


Figura 101. Tiempo medio de carga de la página principal en MF. Fuente: propia.

La figura 100 muestra el tiempo medio de carga de la página principal en IE y la Figura 101 en MF. Se puede observar que los menores tiempos de carga se obtienen con la BDs Oracle 10g en MF, y con eXist + MySQL empleando el navegador IE.

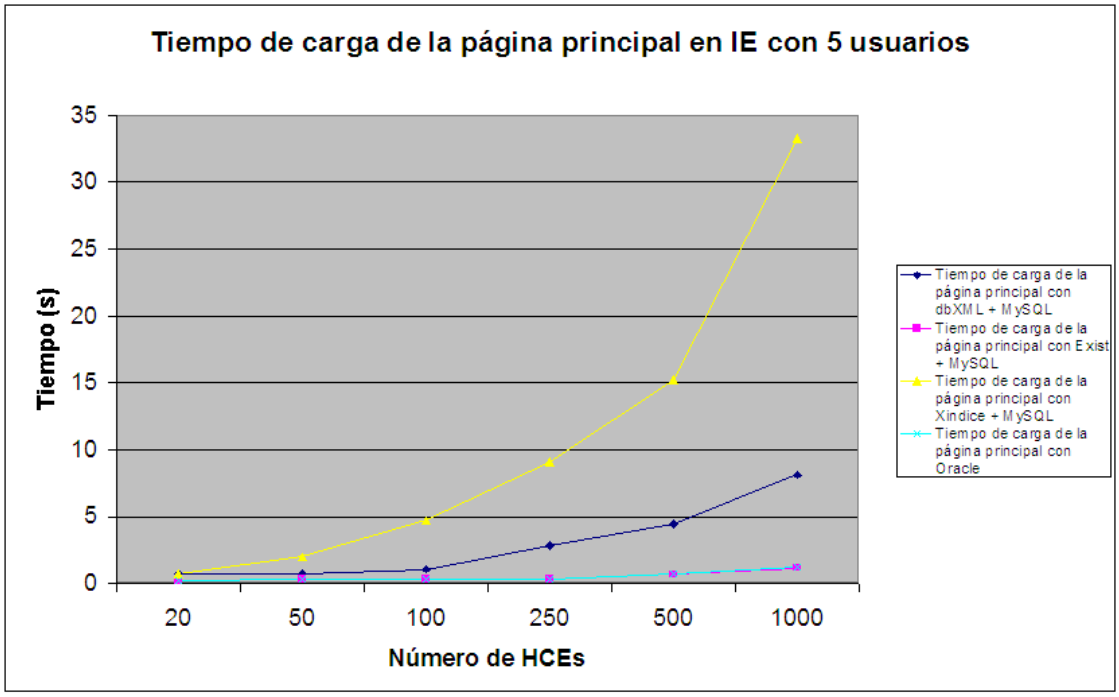


Figura 102. Tiempo medio de carga de la página principal en IE con 5 usuarios. Fuente: propia.

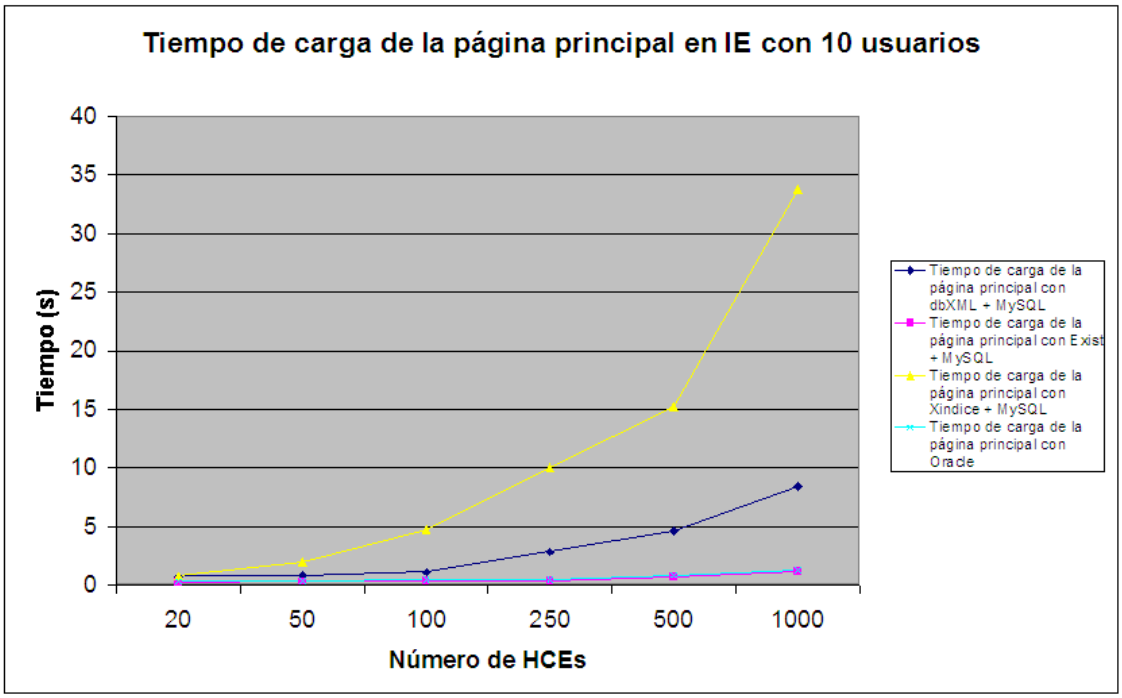


Figura 103. Tiempo medio de carga de la página principal en IE con 10 usuarios. Fuente: propia.

La figura 102 muestra el tiempo medio de carga de la página principal en IE con 5 usuarios, la figura 103 con 10 usuarios y la figura 104 con 20 usuarios. Se puede

observar que los menores tiempos de carga se obtienen con eXist + MySQL en los tres casos.

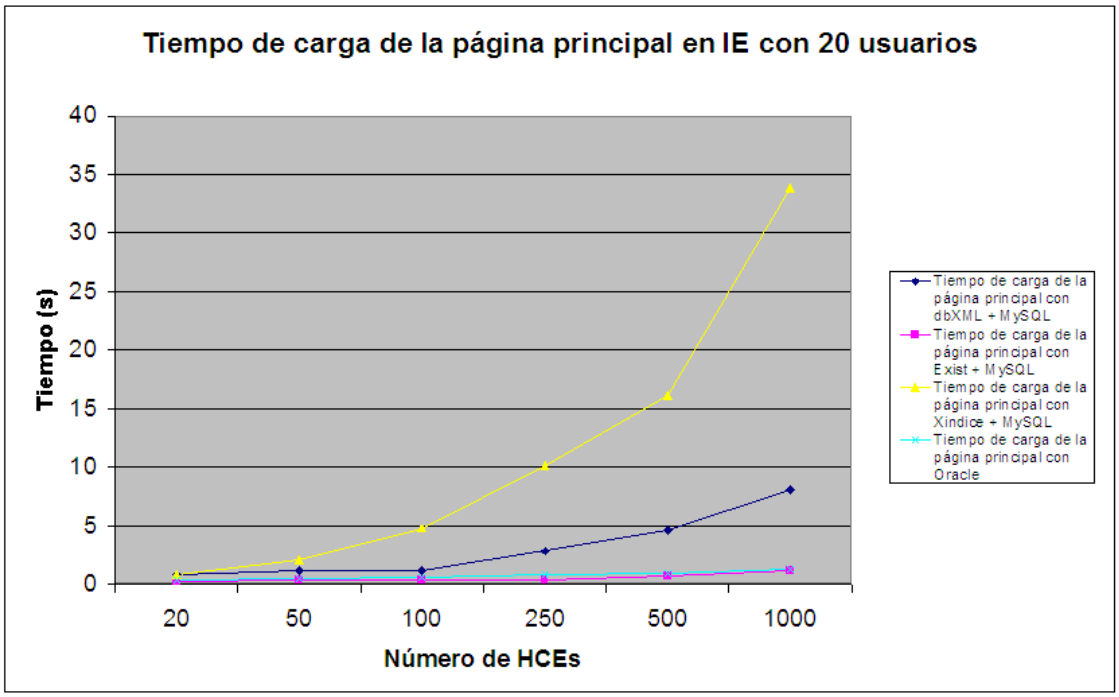


Figura 104. Tiempo medio de carga de la página principal en IE con 20 usuarios. Fuente: propia.

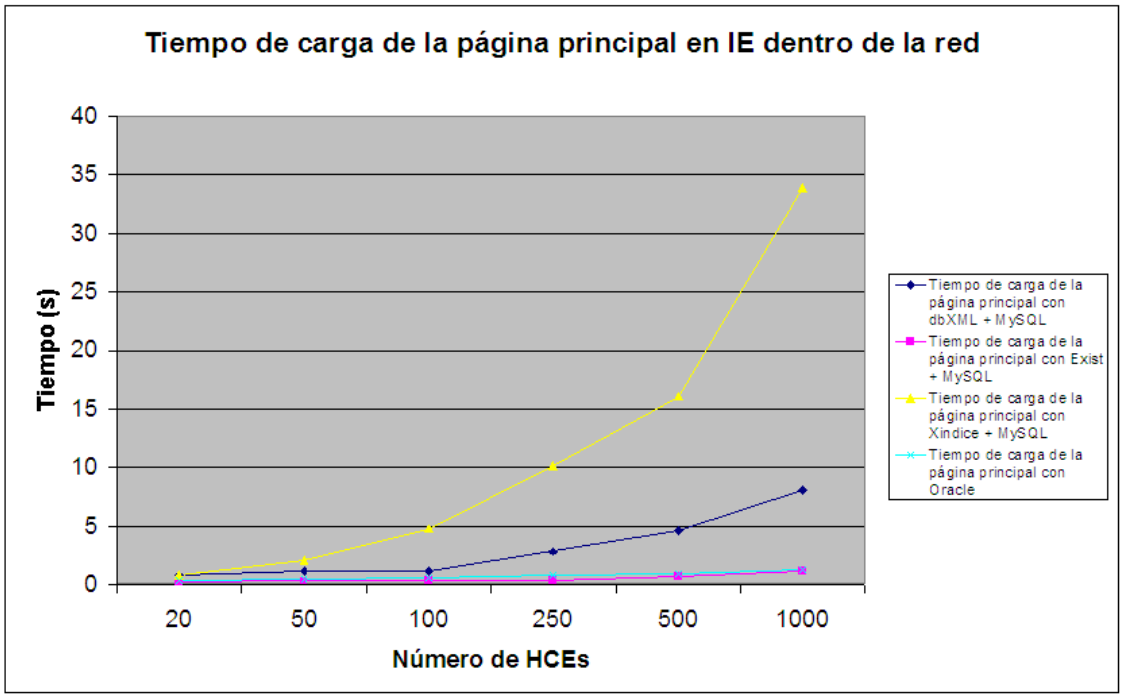


Figura 105. Tiempo medio de carga de la página principal en IE dentro de la red. Fuente: propia.

La figura 105 muestra el tiempo medio de carga de la página principal en IE dentro de la red y la figura 106 el mismo tiempo fuera de la red ETSIT. Se puede observar que los menores tiempos de carga se obtienen empleando eXist + MySQL.

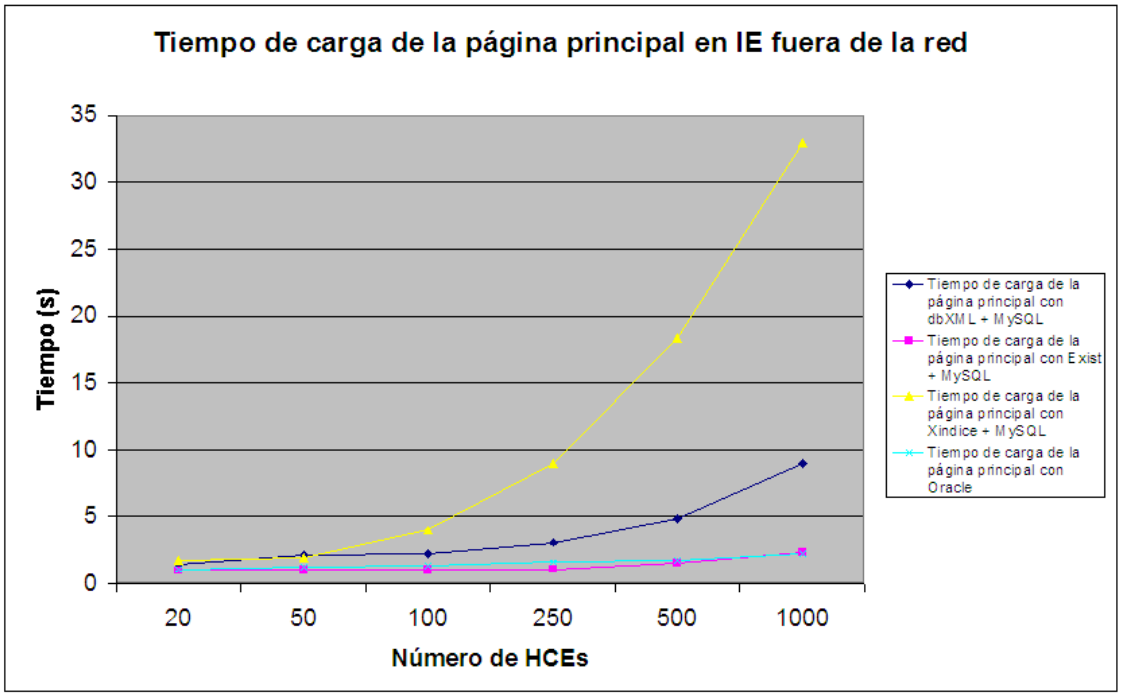


Figura 106. Tiempo medio de carga de la página principal en IE fuera de la red. Fuente: propia.

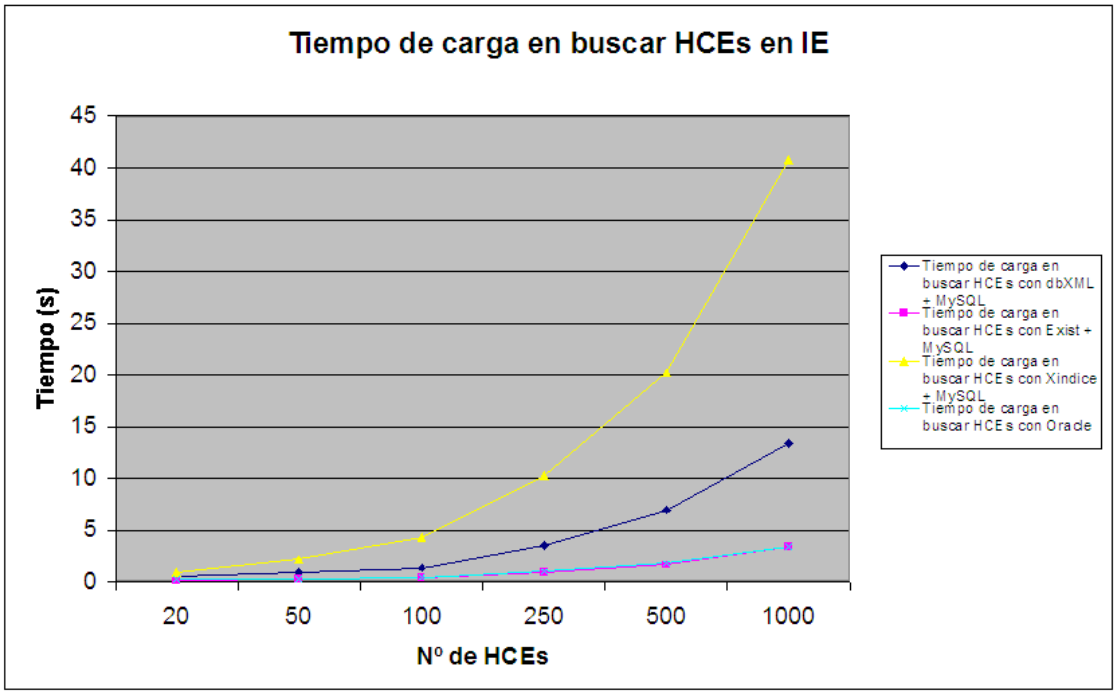


Figura 107. Tiempo medio de carga en buscar HCE en IE. Fuente: propia.

La figura 107 muestra el tiempo medio de carga en buscar HCE en IE y la figura 108 en MF. Se puede observar que los menores tiempos de carga se obtienen con la BD Oracle 10g en MF, y con eXist + MySQL empleando el navegador IE.

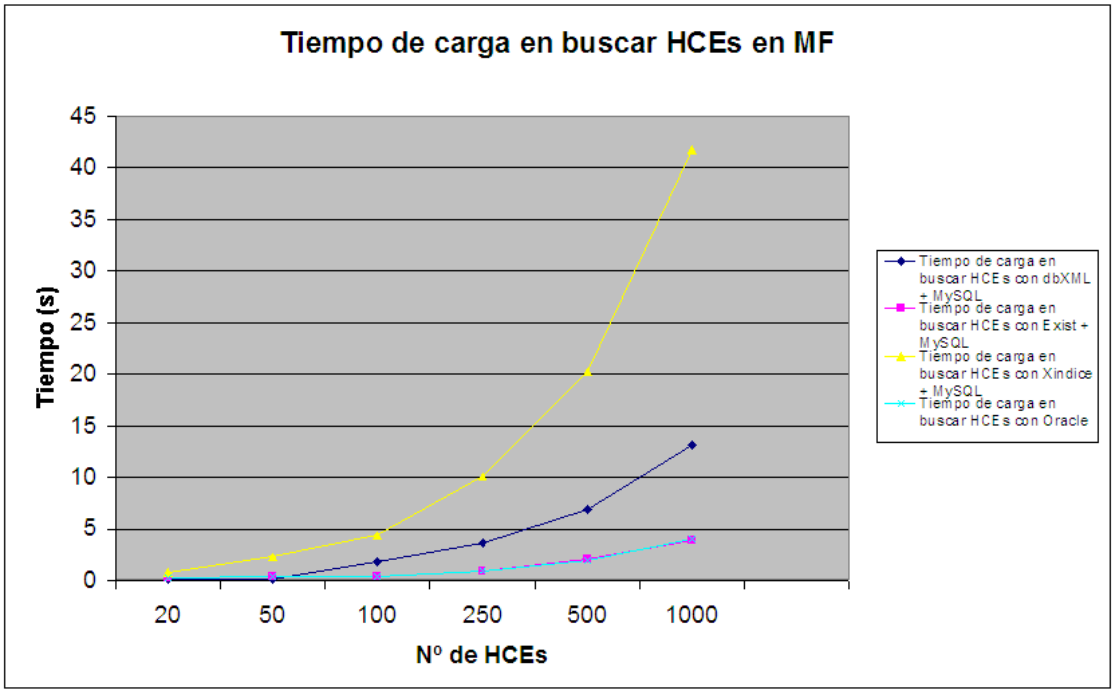


Figura 108. Tiempo medio de carga en buscar HCE en MF. Fuente: propia.

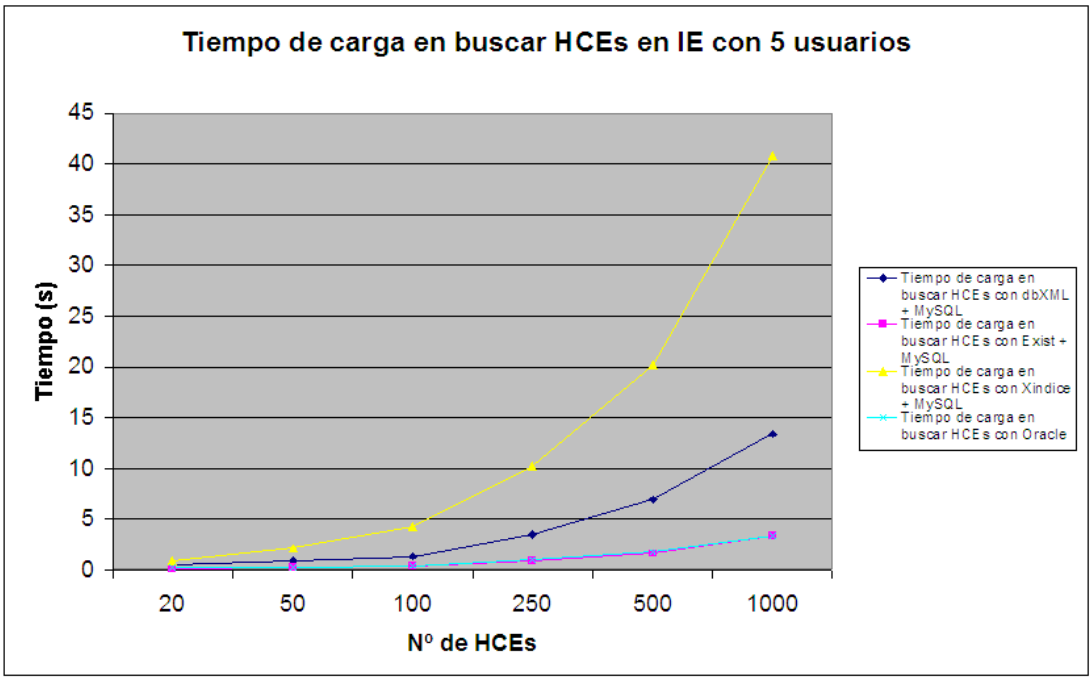


Figura 109. Tiempo medio de carga en buscar HCE en IE con 5 usuarios. Fuente: propia.

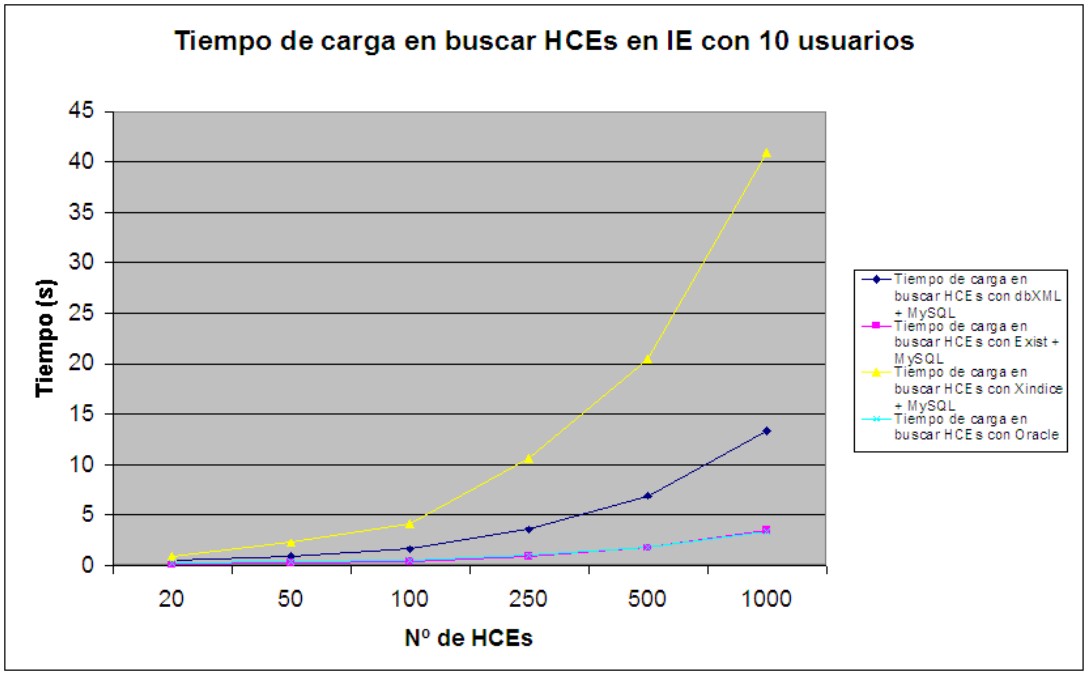


Figura 110. Tiempo medio de carga en buscar HCE en IE con 10 usuarios. Fuente: propia.

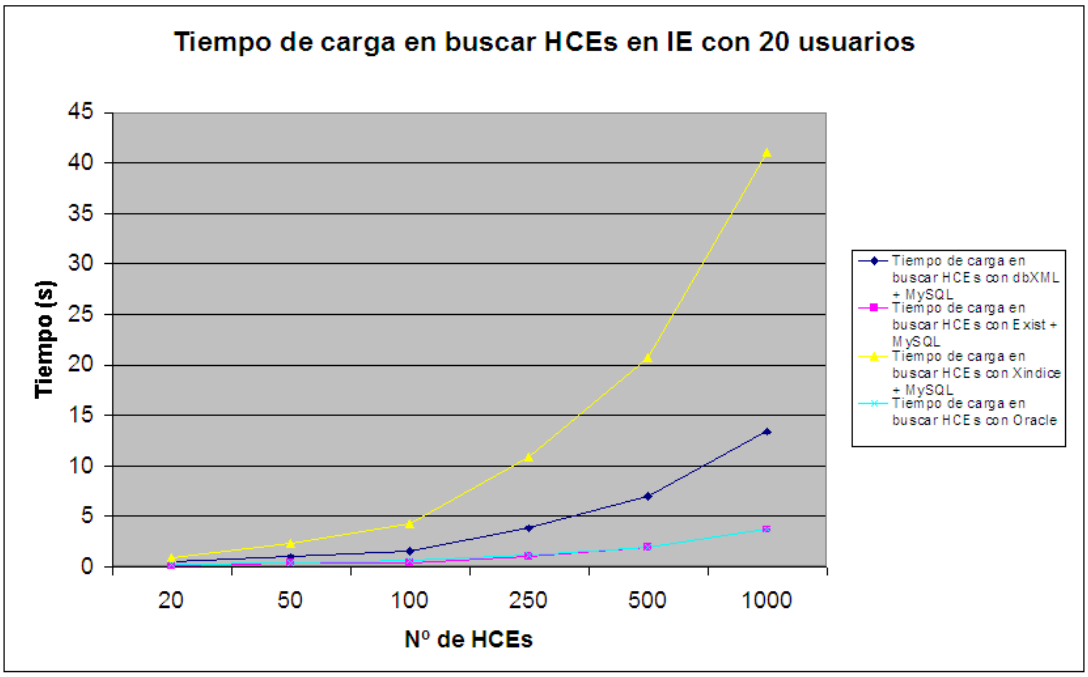


Figura 111. Tiempo medio de carga en buscar HCE en IE con 20 usuarios. Fuente: propia.

La figura 109 muestra el tiempo medio de carga en buscar HCE en IE con 5 usuarios, la figura 110 con 10 usuarios y la figura 111 con 20 usuarios. Se puede

observar que los menores tiempos de carga se obtienen con eXist + MySQL en los tres casos.

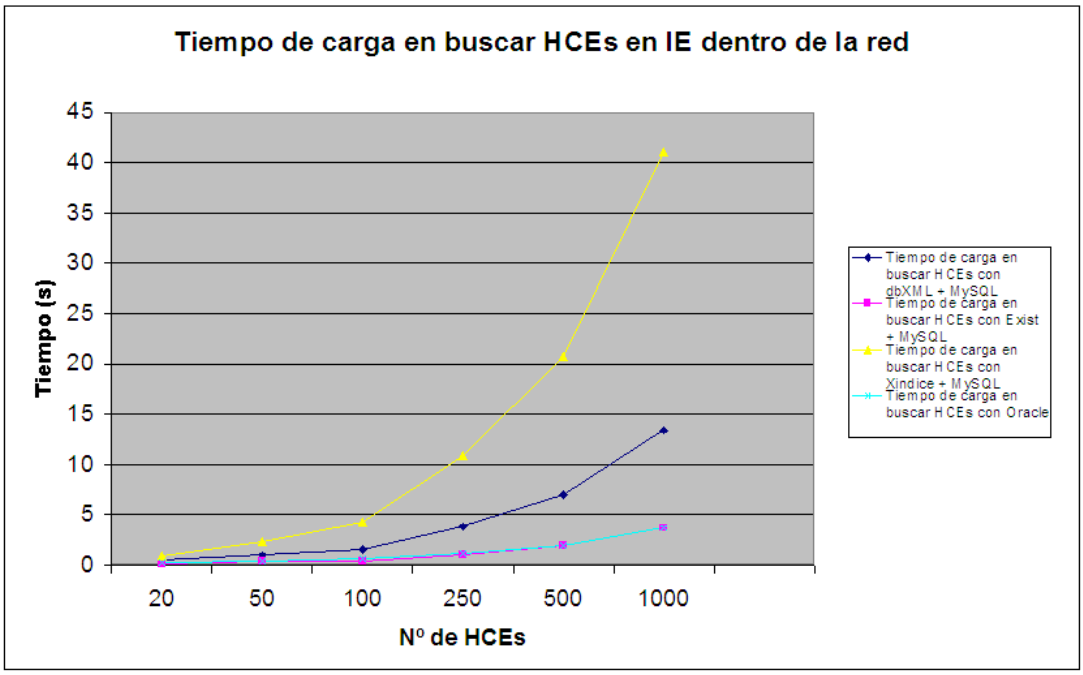


Figura 112. Tiempo medio de carga en buscar HCE dentro de la red. Fuente: propia.

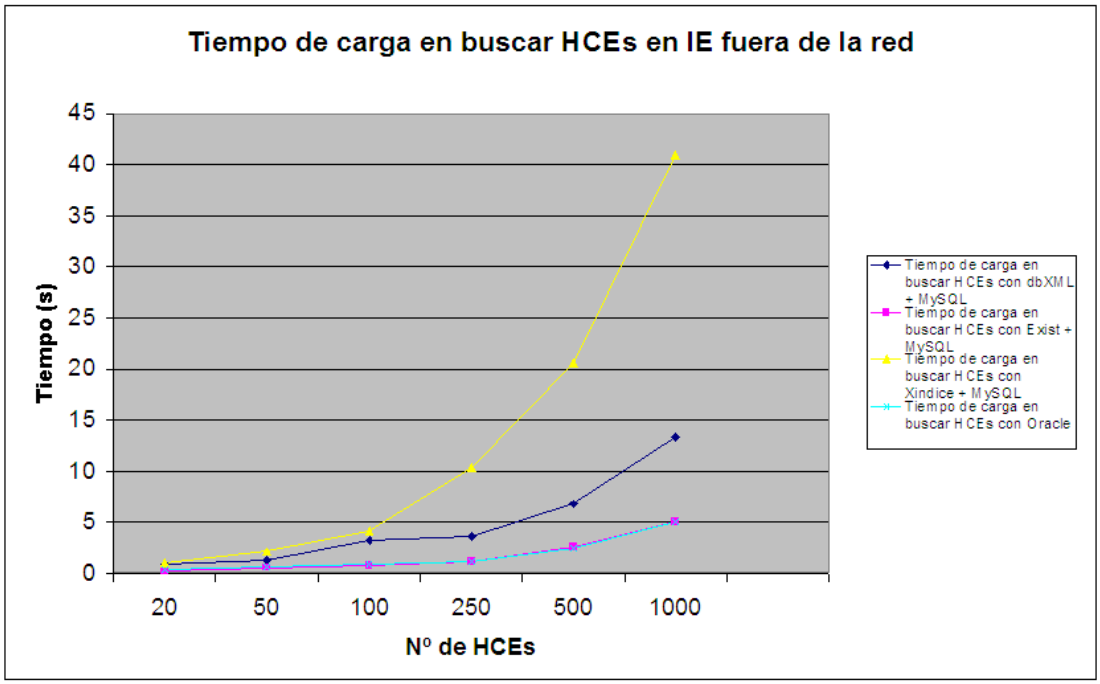


Figura 113. Tiempo medio de carga en buscar HCE fuera de la red. Fuente: propia.

La figura 112 muestra el tiempo medio de carga en buscar HCE dentro de la red y la figura 113 el mismo tiempo fuera de la red ETSIT. Se puede observar que los

menores tiempos de carga se obtienen empleando eXist + MySQL.

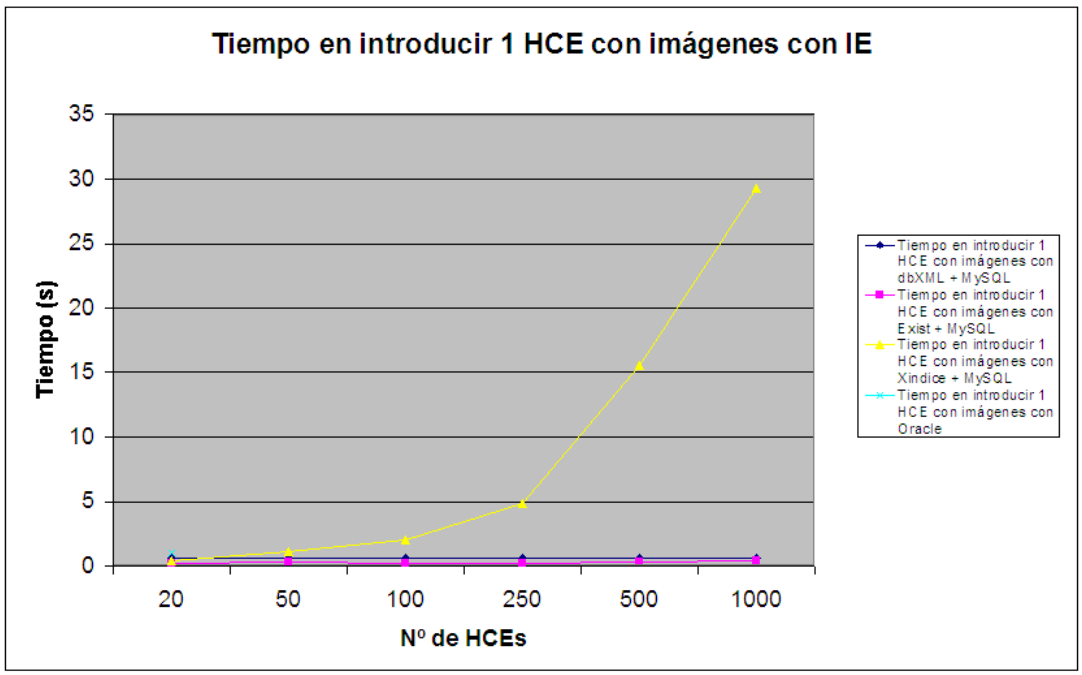


Figura 114. Tiempo medio en introducir 1 HCE con imágenes en IE. Fuente: propia.

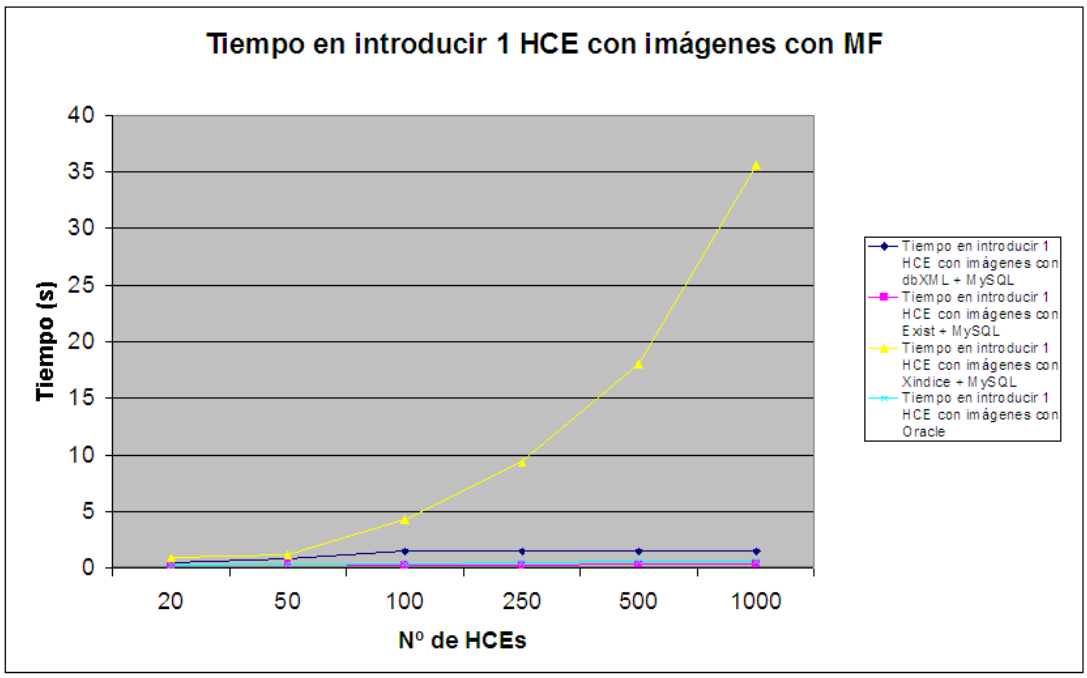


Figura 115. Tiempo medio en introducir 1 HCE con imágenes en MF. Fuente: propia.

La figura 114 muestra el tiempo medio en introducir 1 HCE con dos imágenes en IE y la figura 115 en MF. Se puede observar que los menores tiempos de carga se obtienen con la BD Oracle 10g en MF, y con eXist + MySQL empleando el navegador IE.

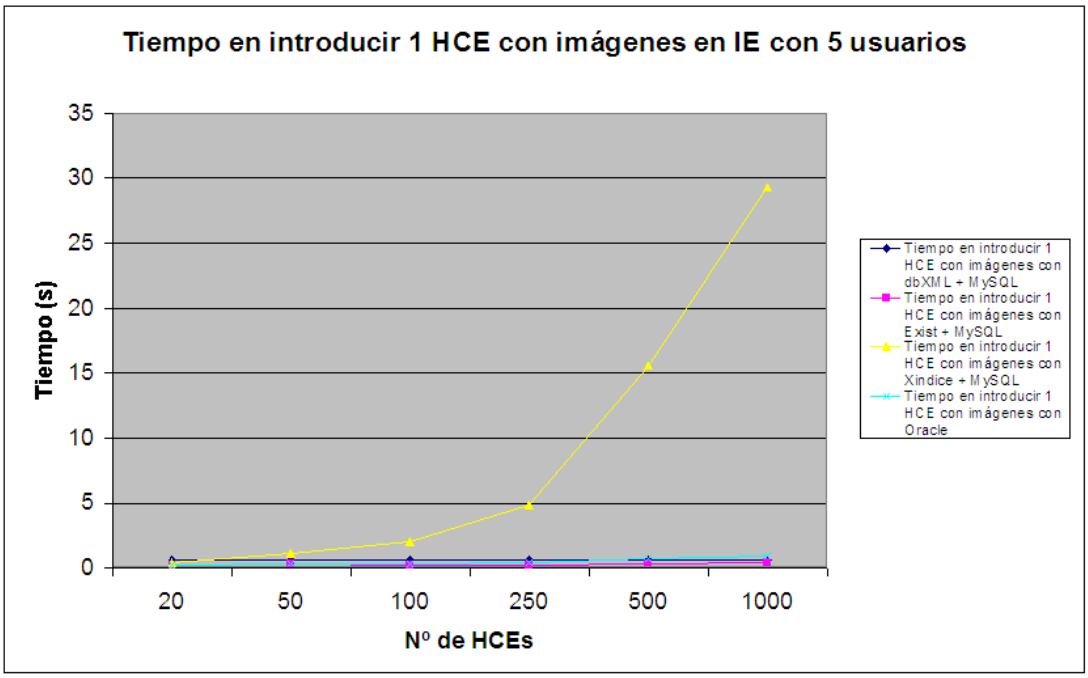


Figura 116. Tiempo medio en introducir 1 HCE con imágenes en IE con 5 usuarios. Fuente: propia.

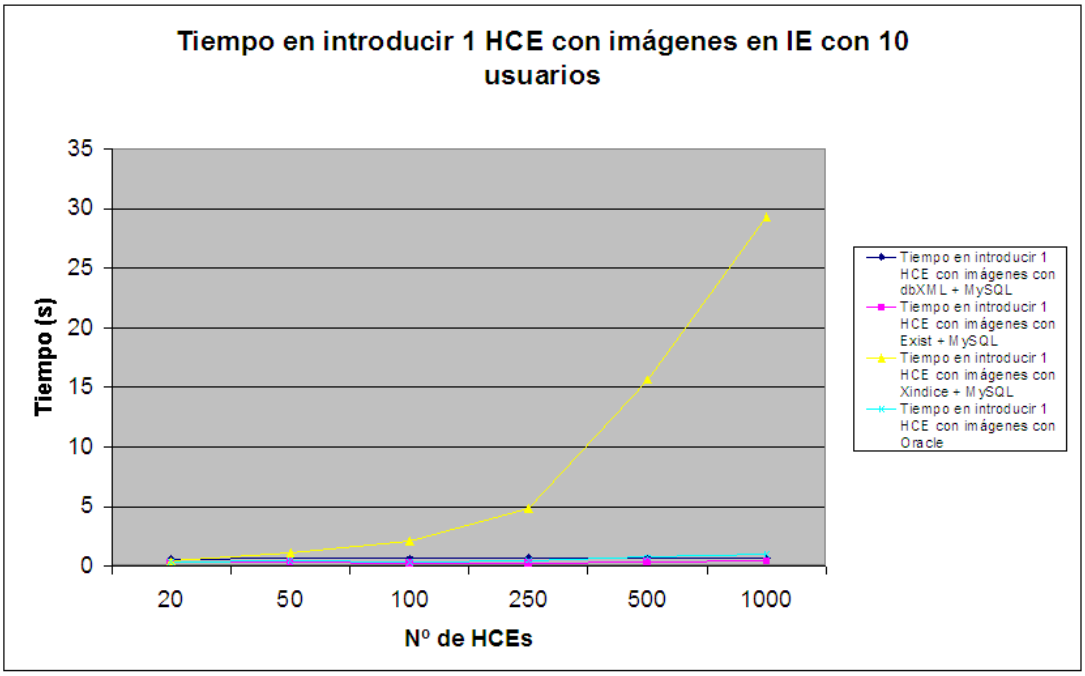


Figura 117. Tiempo medio en introducir 1 HCE con imágenes en IE con 10 usuarios. Fuente: propia.

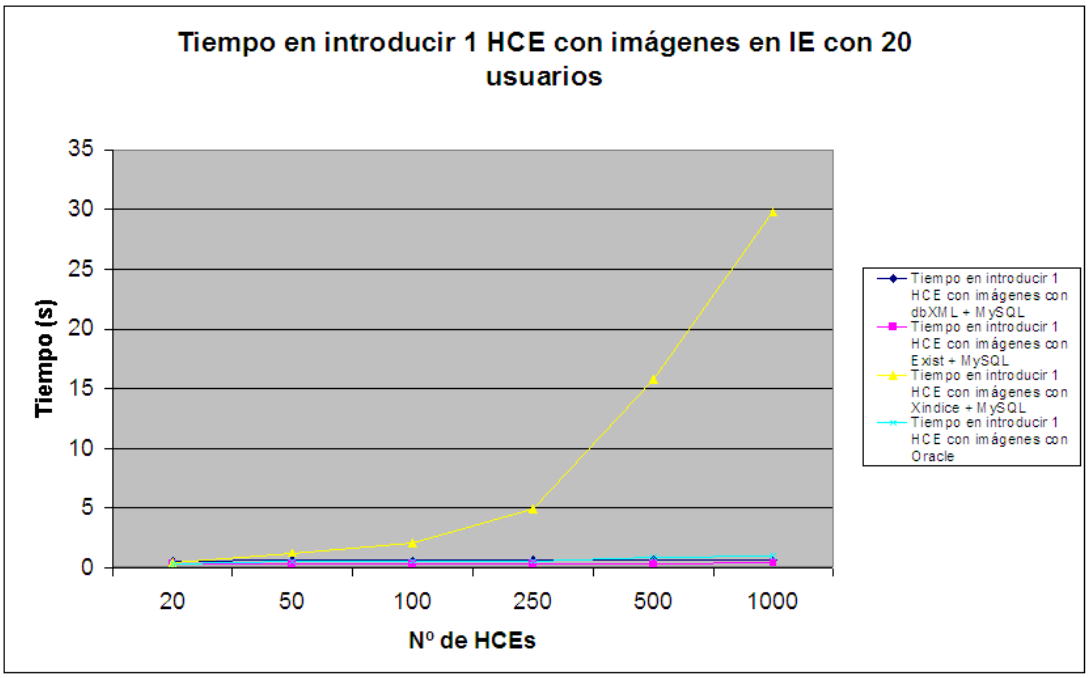


Figura 118. Tiempo medio en introducir 1 HCE con imágenes en IE con 20 usuarios. Fuente: propia.

La figura 116 muestra el tiempo medio en introducir 1 HCE con dos imágenes en IE con 5 usuarios, la figura 117 con 10 usuarios y la figura 118 con 20 usuarios. Se puede observar que los menores tiempos de carga se obtienen con eXist + MySQL en los tres casos.

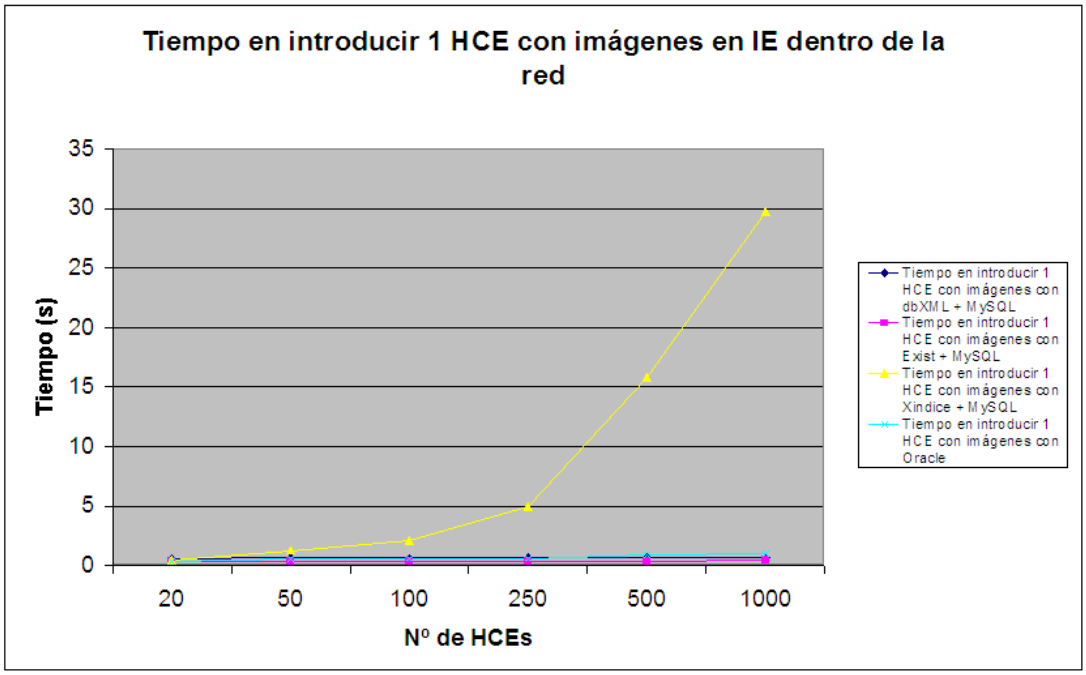


Figura 119. Tiempo medio en introducir 1 HCE con imágenes en IE dentro de la red. Fuente: propia.

La figura 119 muestra el tiempo medio en introducir 1 HCE con dos imágenes dentro de la red y la figura 120 el mismo tiempo fuera de la red ETSIT. Se puede observar que los menores tiempos de carga se obtienen empleando eXist + MySQL.

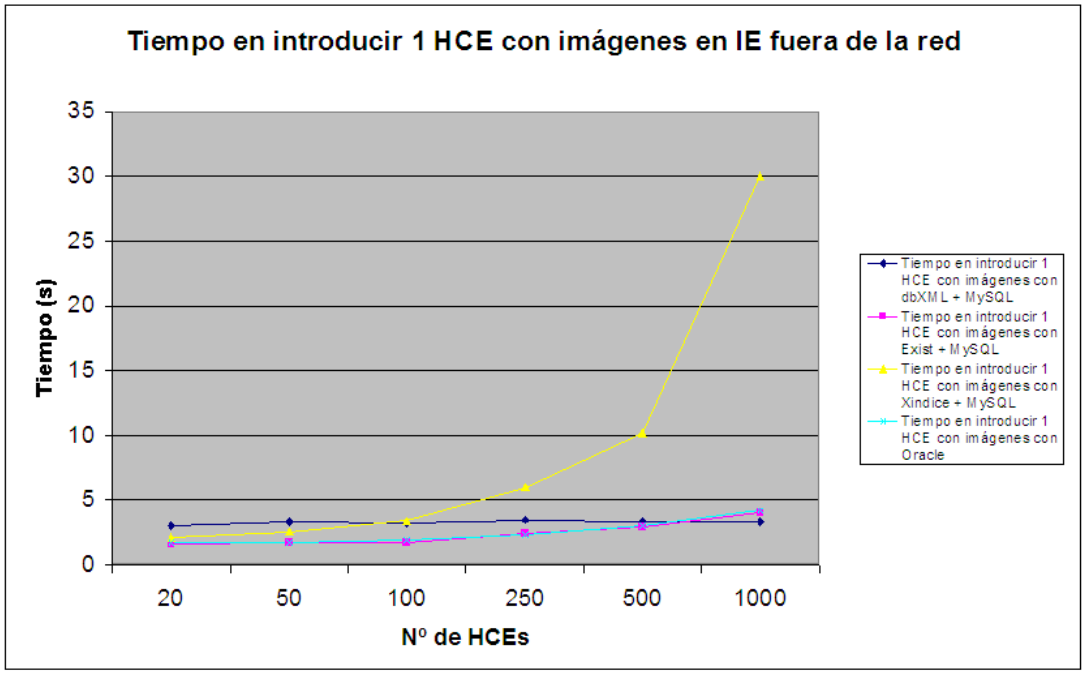


Figura 120. Tiempo medio en introducir 1 HCE con imágenes en IE fuera de la red. Fuente: propia.

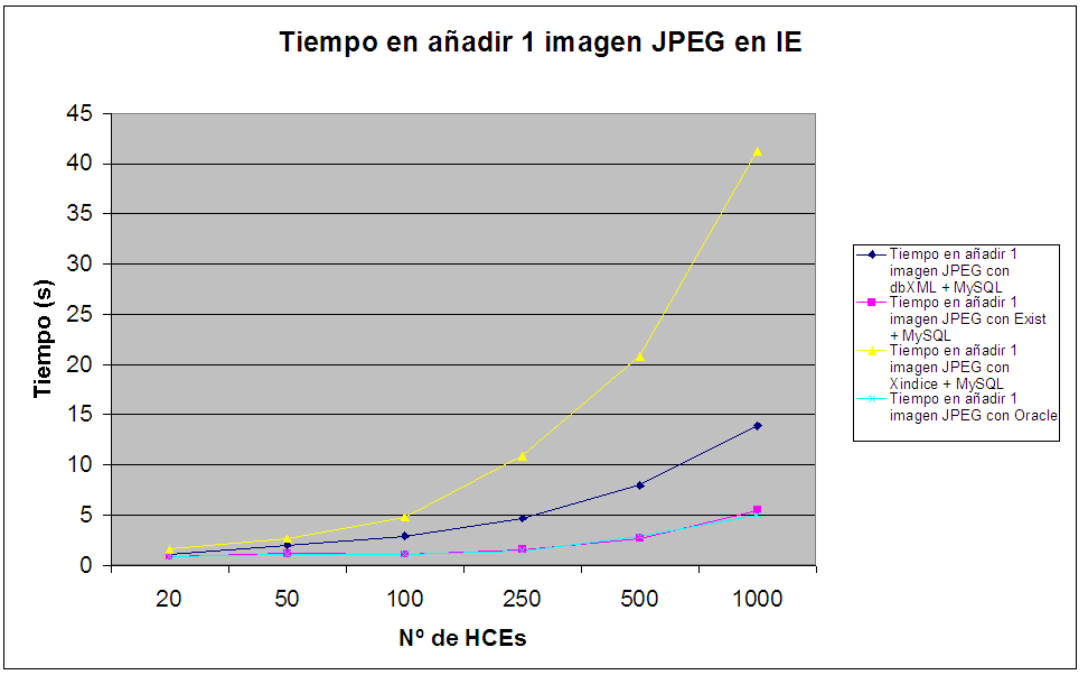


Figura 121. Tiempo medio en añadir 1 imagen JPEG en IE. Fuente: propia.

La figura 121 muestra el tiempo medio en añadir 1 imagen JPEG en IE y la figura 122 en MF. Se puede observar que los menores tiempos de carga se obtienen con la BD Oracle 10g en MF, y con eXist + MySQL empleando el navegador IE.

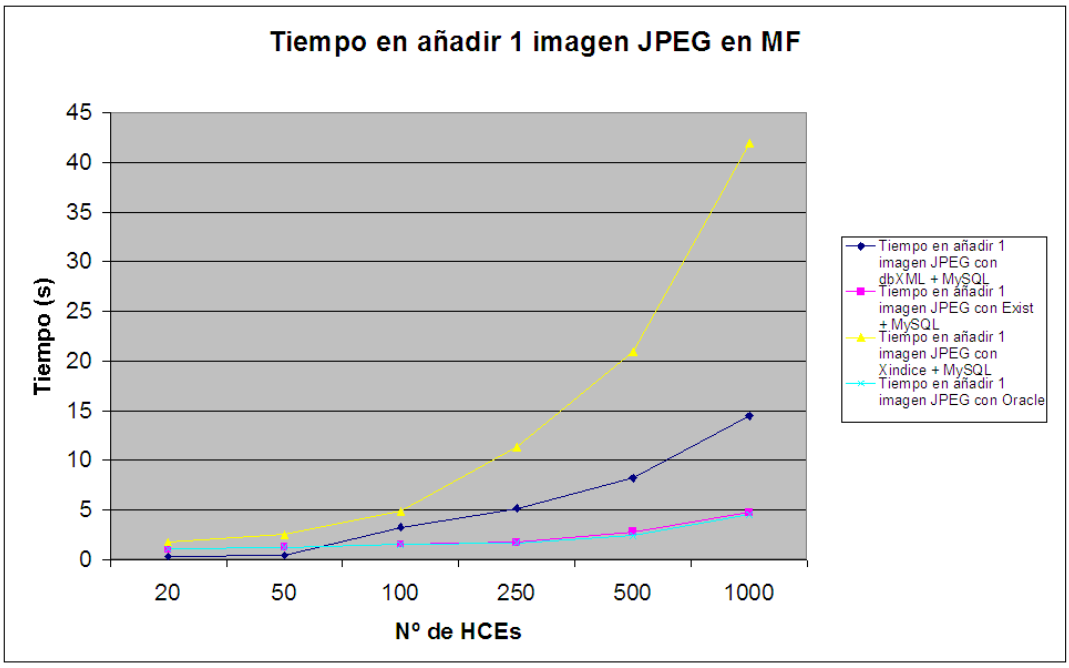


Figura 122. Tiempo medio en añadir 1 imagen JPEG en MF. Fuente: propia.

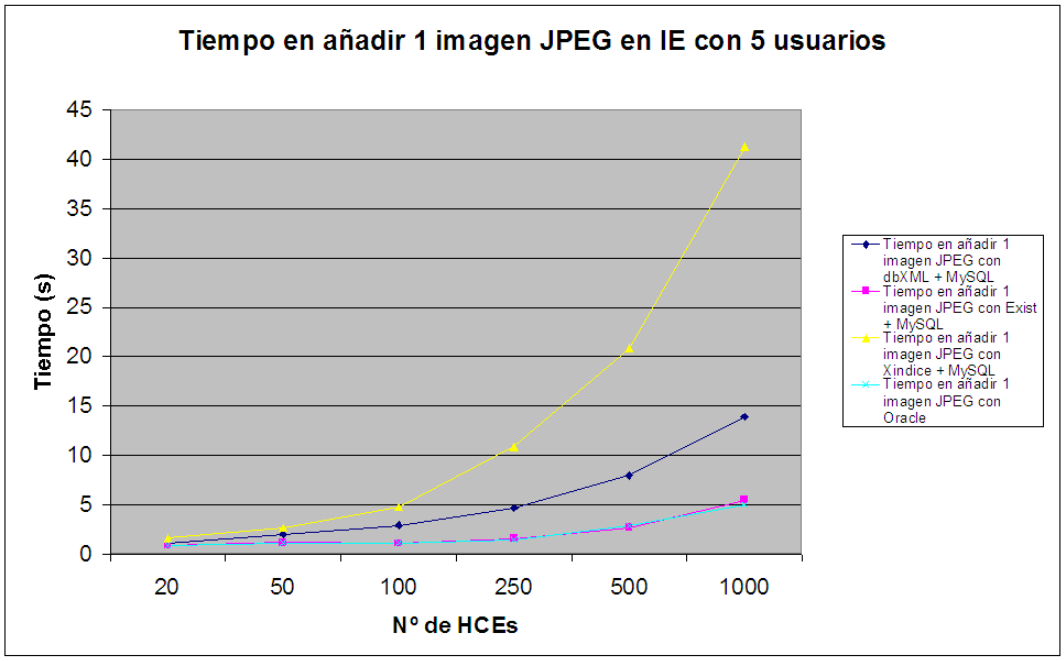


Figura 123. Tiempo medio en añadir 1 imagen JPEG en IE con 5 usuarios. Fuente: propia.

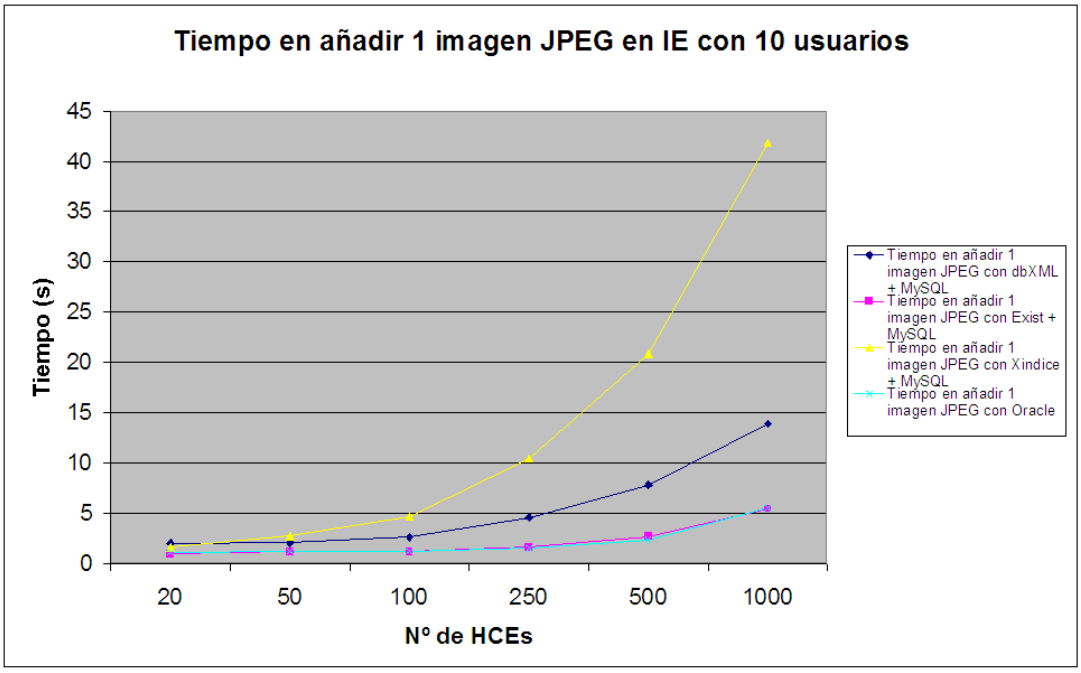


Figura 124. Tiempo medio en añadir 1 imagen JPEG en IE con 10 usuarios. Fuente: propia.

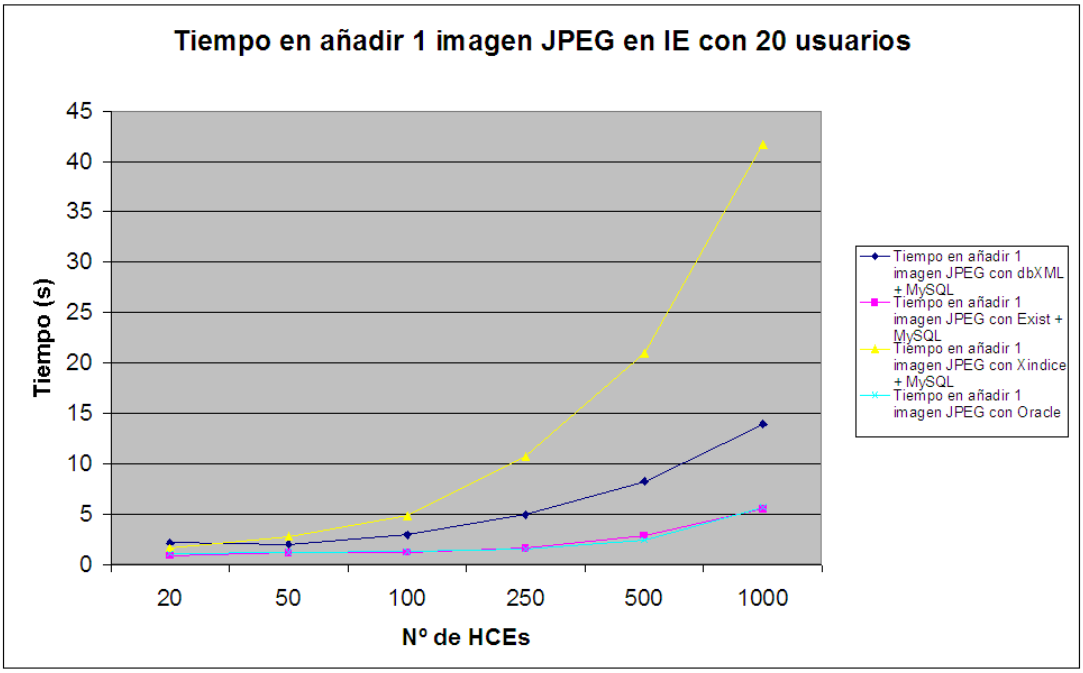


Figura 125. Tiempo medio en añadir 1 imagen JPEG en IE con 20 usuarios. Fuente: propia.

La figura 123 muestra el tiempo medio en añadir 1 imagen JPEG con 5 usuarios, la figura 124 con 10 usuarios y la figura 125 con 20 usuarios. Se puede observar que los menores tiempos de carga se obtienen con eXist + MySQL en los tres casos.

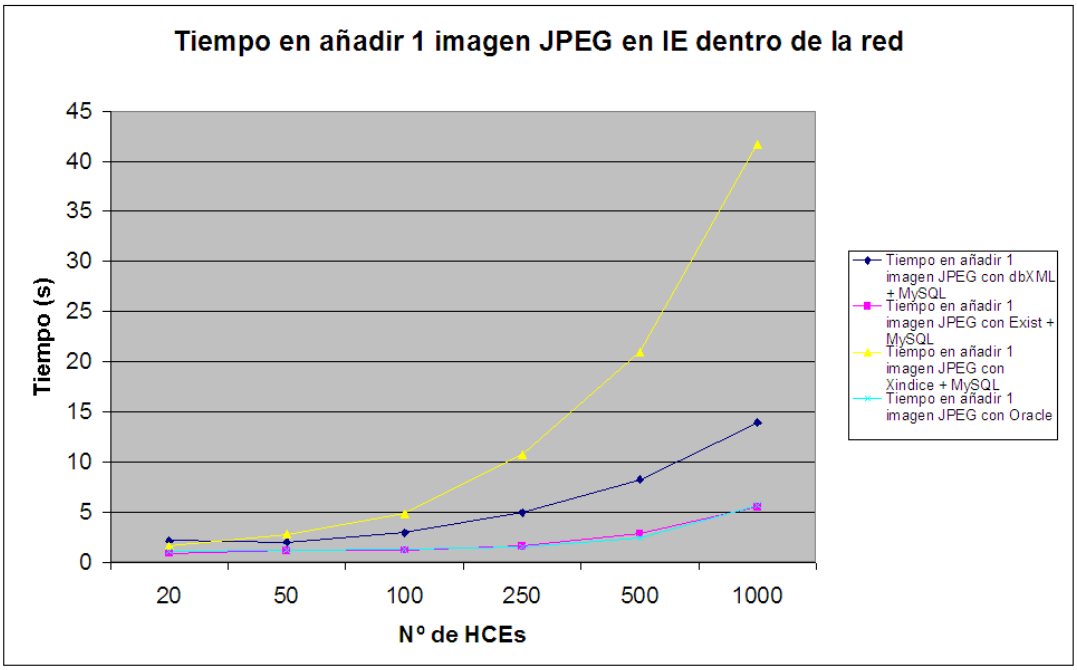


Figura 126. Tiempo medio en añadir 1 imagen JPEG en IE dentro de la red. Fuente: propia.

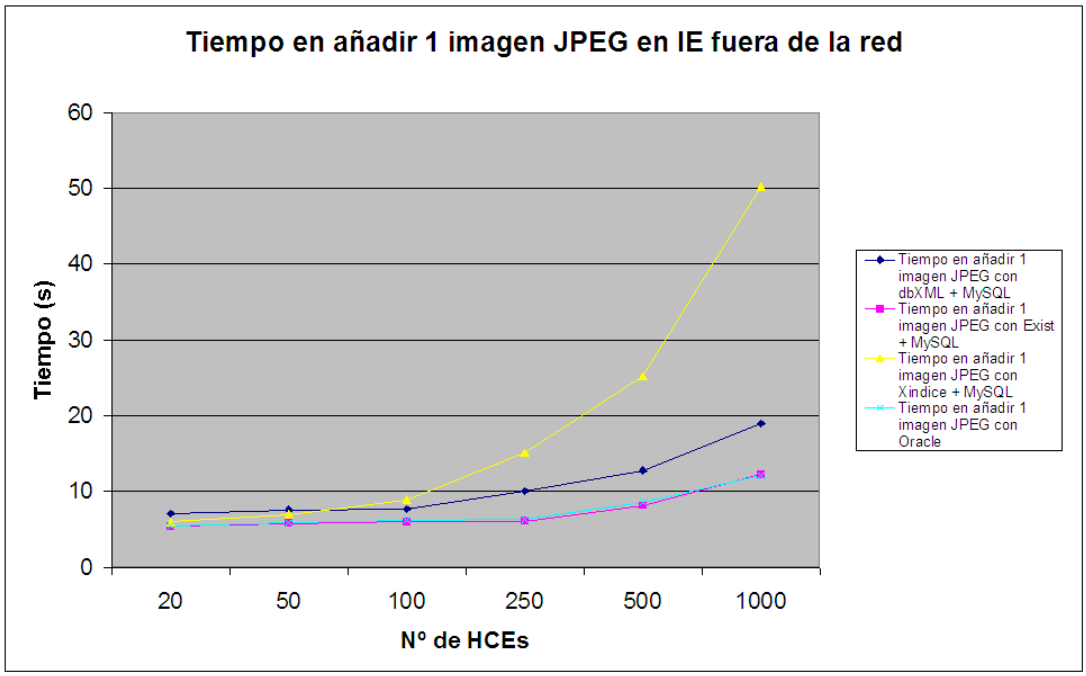


Figura 127. Tiempo medio en añadir 1 imagen JPEG en IE fuera de la red. Fuente: propia.

La figura 126 muestra el tiempo medio en añadir 1 imagen JPEG dentro de la red y la figura 127 el mismo tiempo fuera de la red ETSIT. Se puede observar que los menores tiempos de carga se obtienen empleando eXist + MySQL.

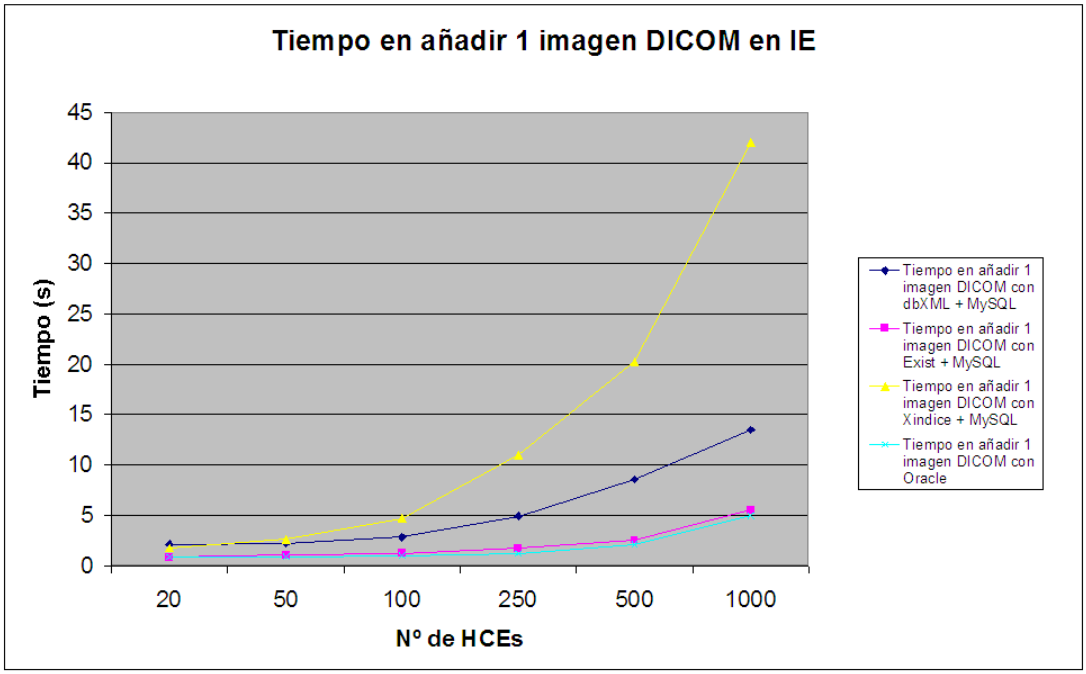


Figura 128. Tiempo medio en añadir 1 imagen JPEG en IE. Fuente: propia.

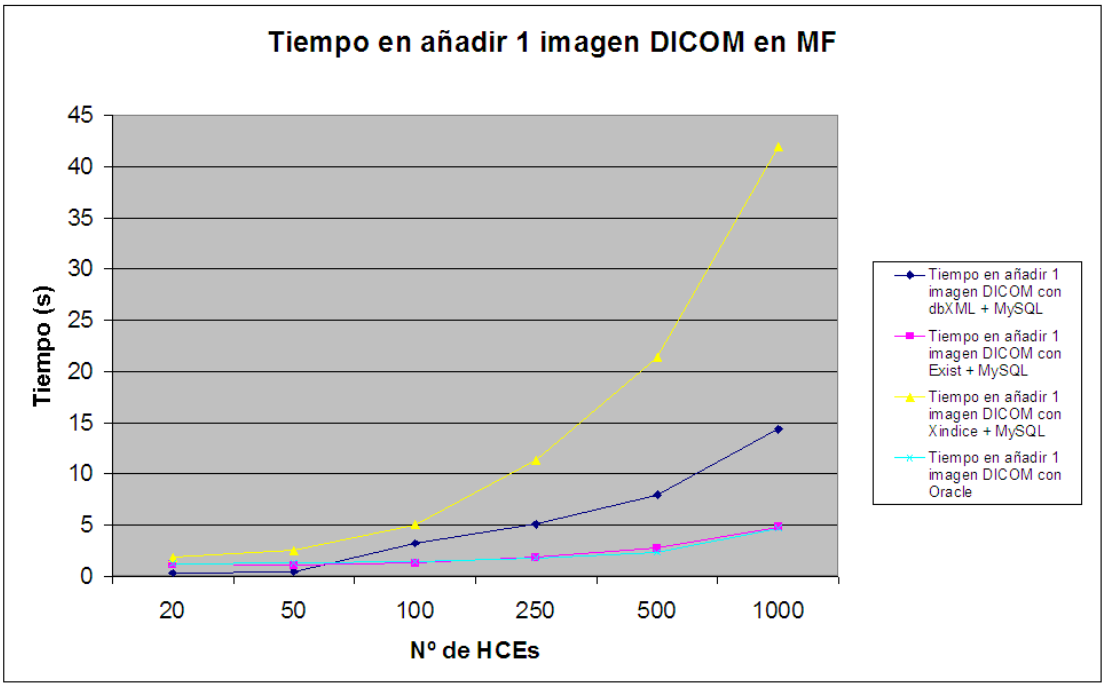


Figura 129. Tiempo medio en añadir 1 imagen DICOM en MF. Fuente: propia.

La figura 128 muestra el tiempo medio en añadir 1 imagen DICOM en IE y la figura 129 en MF. Se puede observar que los menores tiempos de carga se obtienen con la BD Oracle 10g en MF, y con eXist + MySQL empleando el navegador IE.

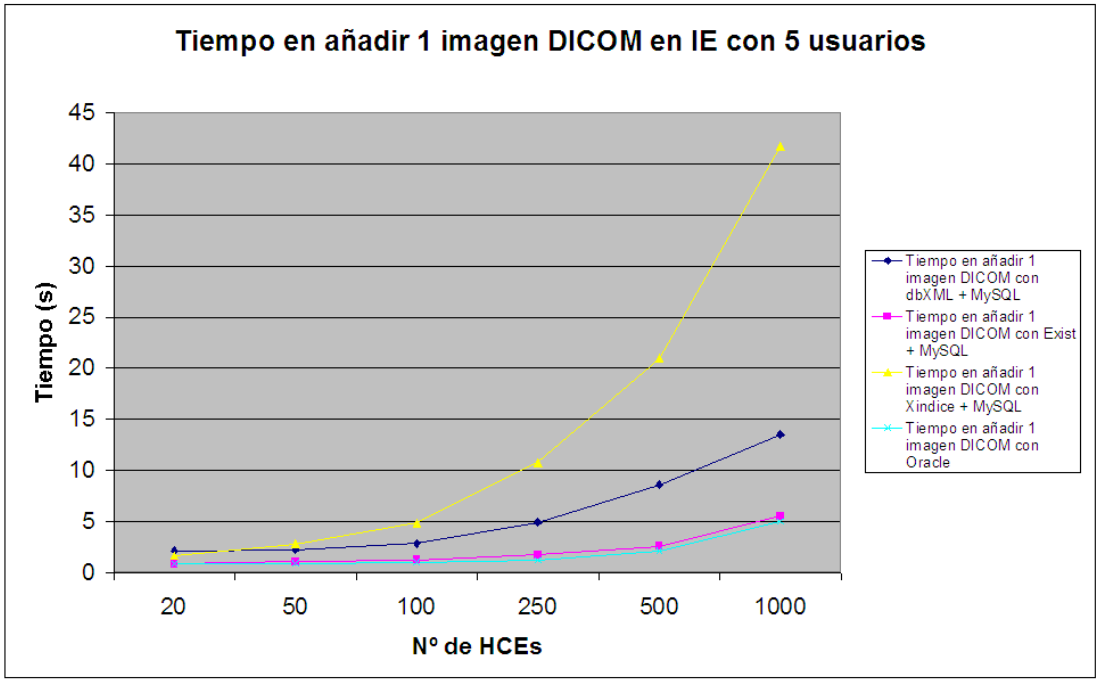


Figura 130. Tiempo medio en añadir 1 imagen DICOM en IE con 5 usuarios. Fuente: propia.

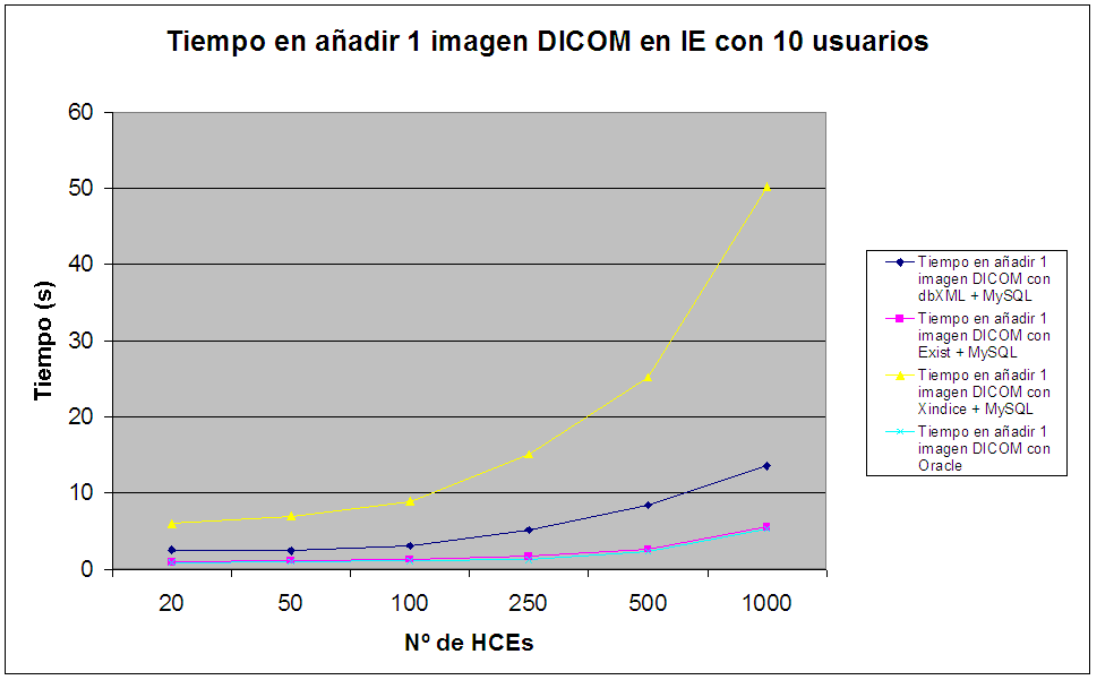


Figura 131. Tiempo medio en añadir 1 imagen DICOM en IE con 10 usuarios. Fuente: propia.

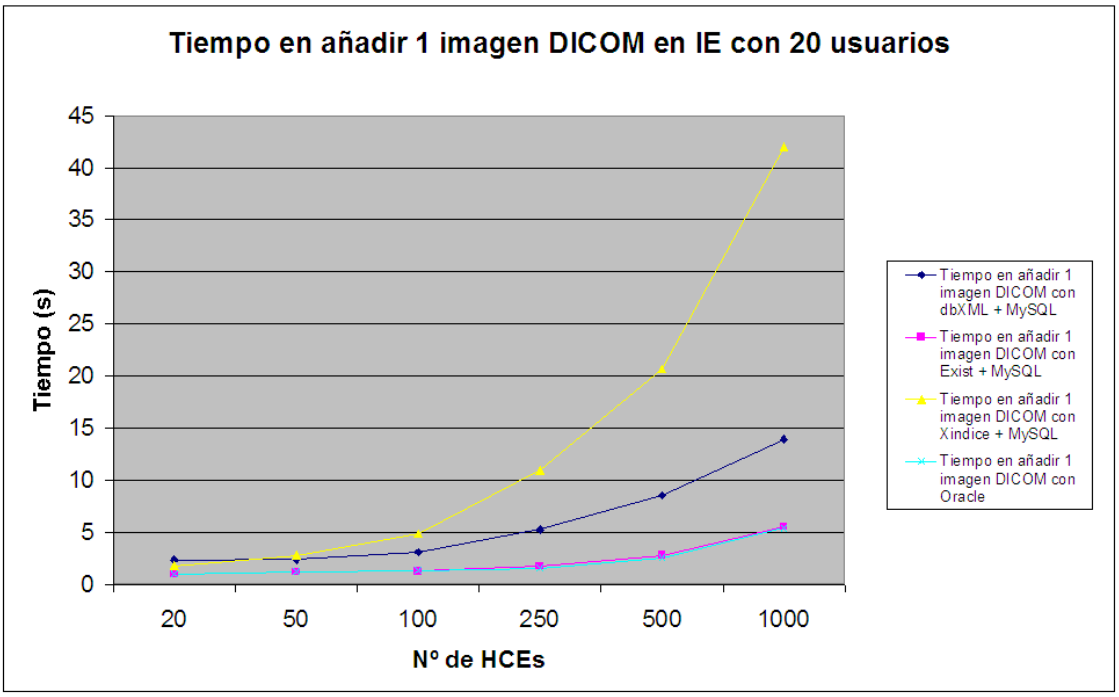


Figura 132. Tiempo medio en añadir 1 imagen DICOM en IE con 20 usuarios. Fuente: propia.

La figura 130 muestra el tiempo medio en añadir 1 imagen DICOM con 5 usuarios, la figura 131 con 10 usuarios y la figura 132 con 20 usuarios. Se puede observar que los menores tiempos de carga se obtienen con la BD eXist + MySQL en los tres casos.

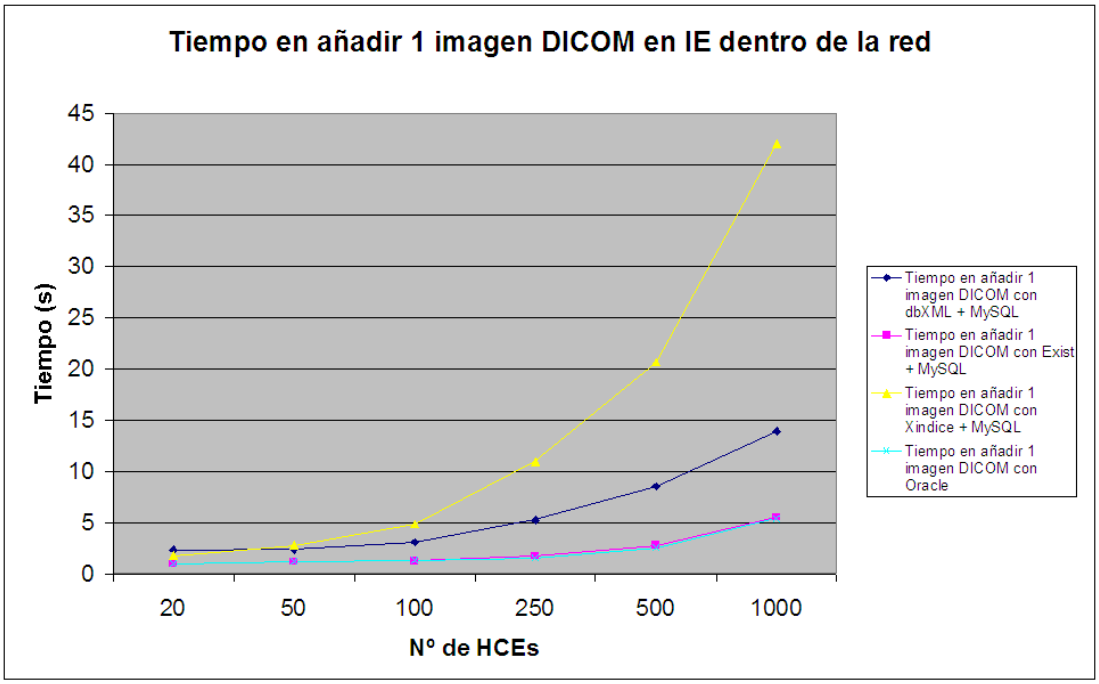


Figura 133. Tiempo medio en añadir 1 imagen DICOM en IE dentro de la red. Fuente: propia.

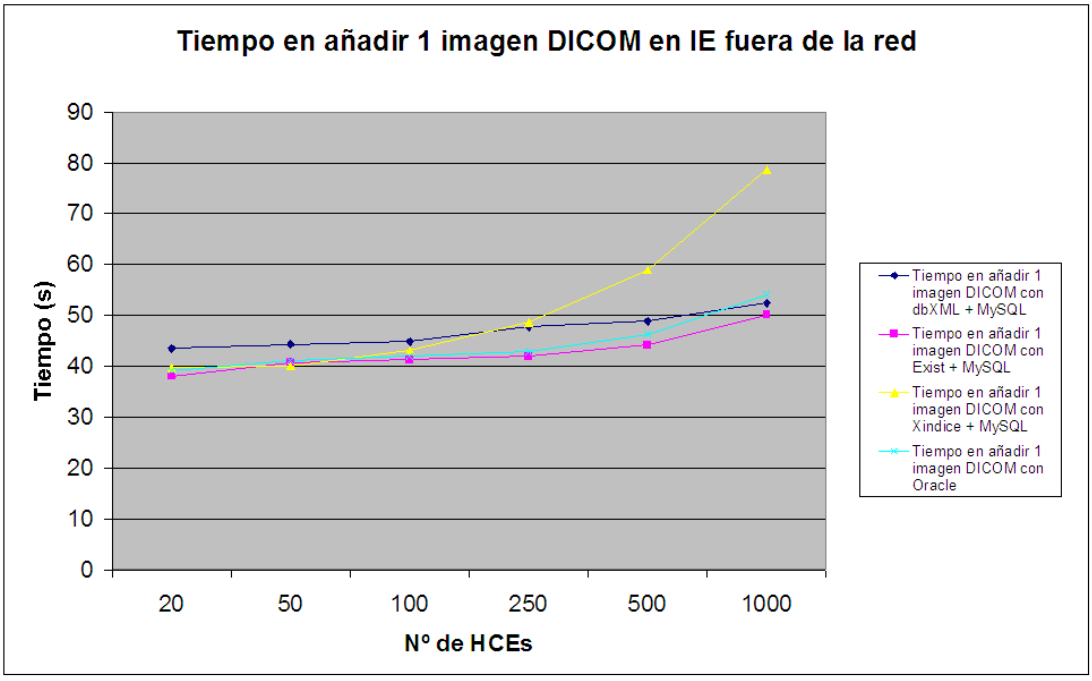


Figura 134. Tiempo medio en añadir 1 imagen DICOM en IE fuera de la red. Fuente: propia.

La figura 133 muestra el tiempo medio en añadir 1 imagen DICOM dentro de la red y la figura 134 el mismo tiempo fuera de la red ETSIT. Se puede observar que los menores tiempos de carga se obtienen empleando eXist + MySQL.

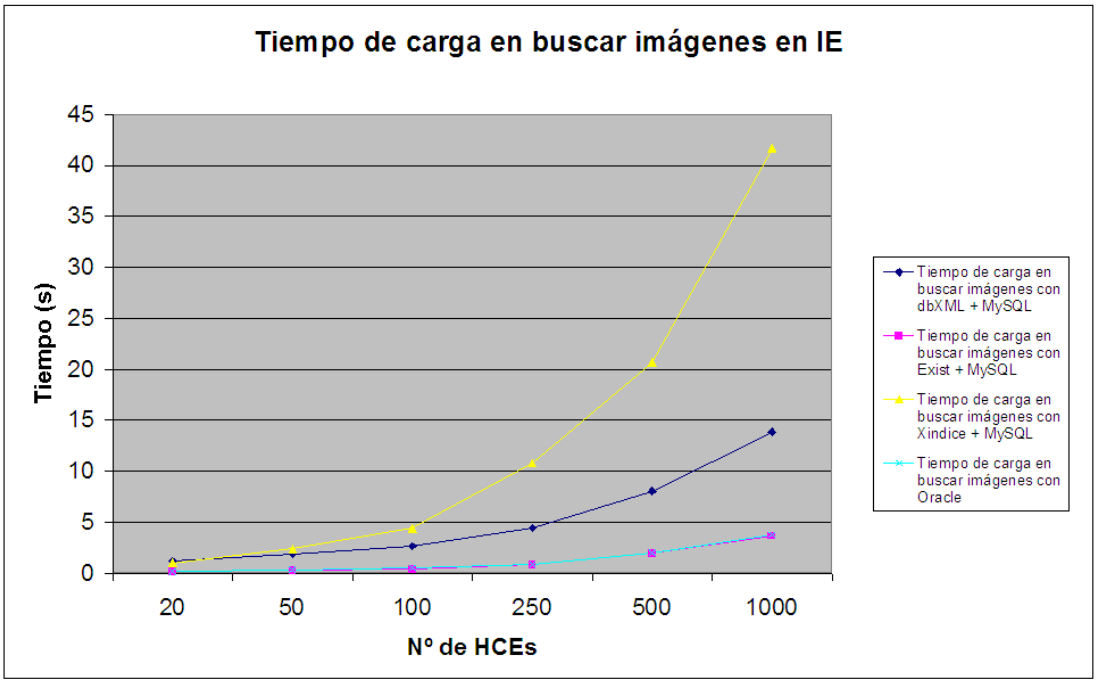


Figura 135. Tiempo medio de carga en buscar imágenes en IE. Fuente: propia.

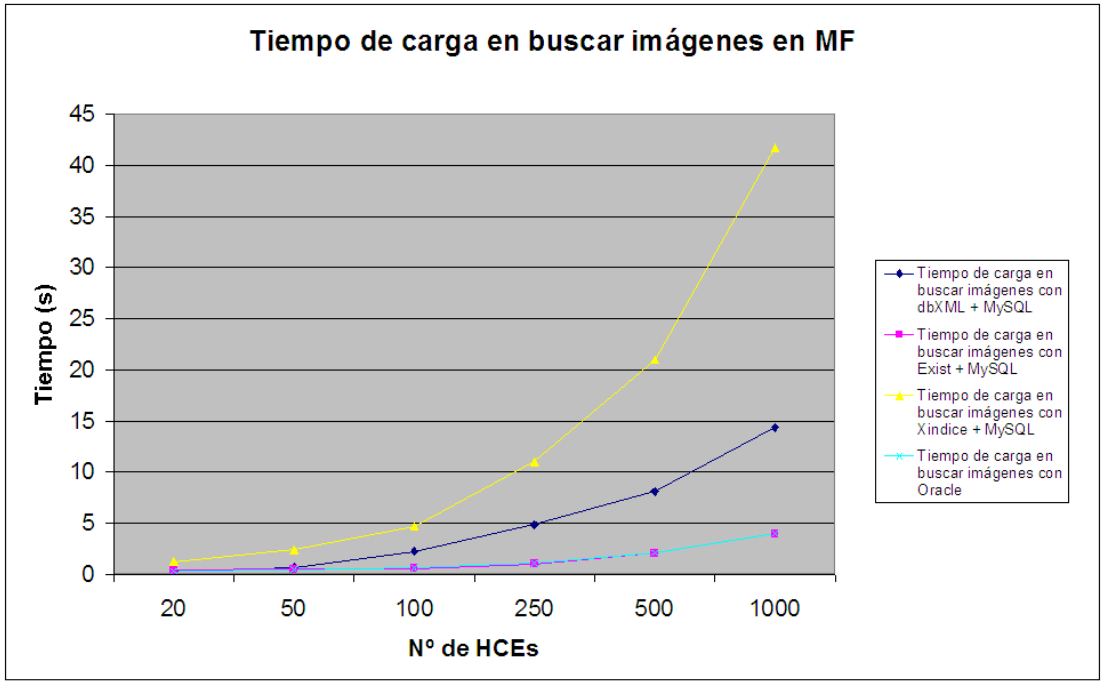


Figura 136. Tiempo medio de carga en buscar imágenes en MF. Fuente: propia.

La figura 135 muestra el tiempo medio de carga en buscar imágenes en IE y la figura 136 en MF. Se puede observar que los menores tiempos de carga se obtienen con la BD Oracle 10g en MF, y con eXist + MySQL empleando el navegador IE.

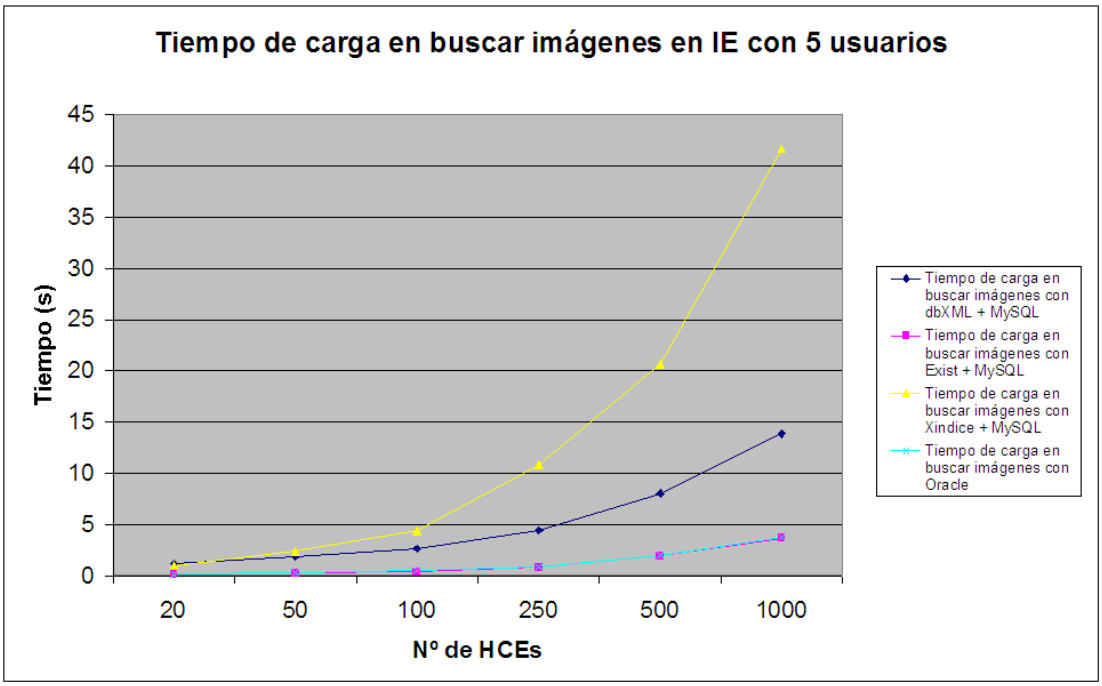


Figura 137. Tiempo medio de carga en buscar imágenes en IE con 5 usuarios. Fuente: propia.

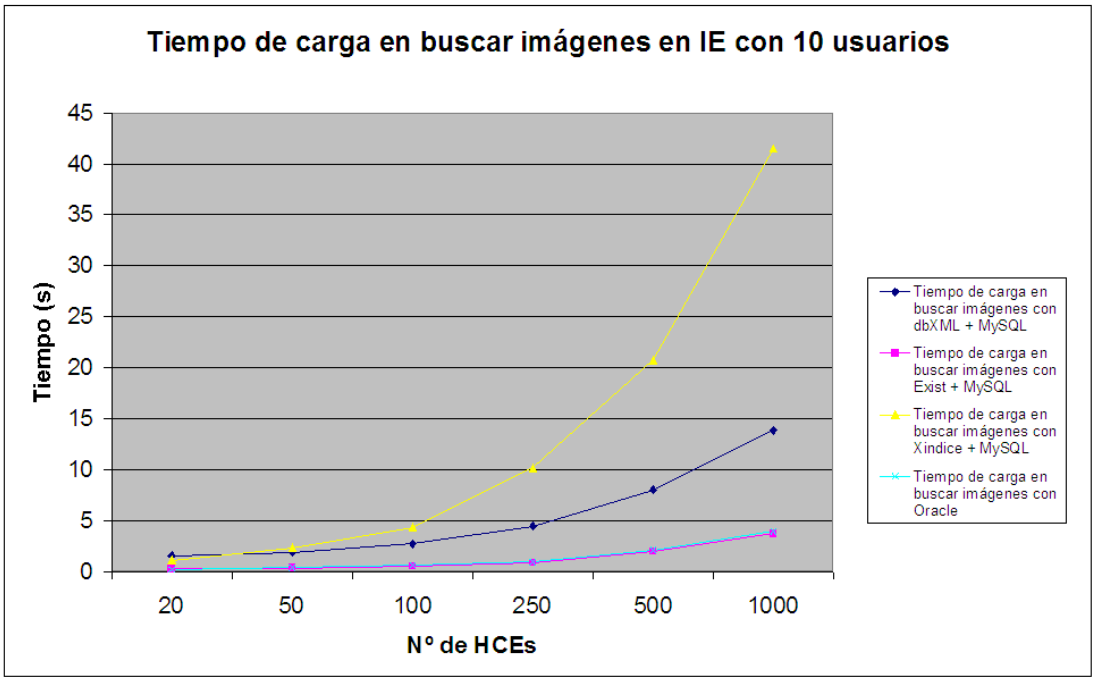


Figura 138. Tiempo medio de carga en buscar imágenes en IE con 10 usuarios. Fuente: propia.

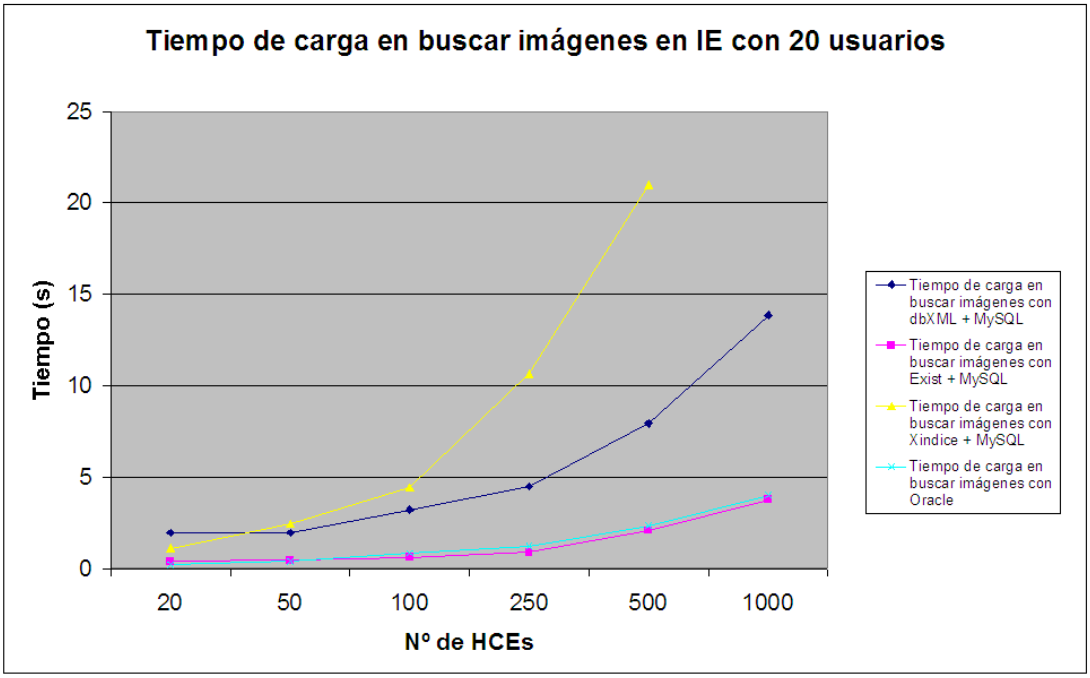


Figura 139. Tiempo medio de carga en buscar imágenes en IE con 20 usuarios. Fuente: propia.

La figura 137 muestra el tiempo medio en buscar imágenes con 5 usuarios, la figura 138 con 10 usuarios y la figura 139 con 20 usuarios. Se puede observar que los

menores tiempos de carga se obtienen con eXist + MySQL en los tres casos.

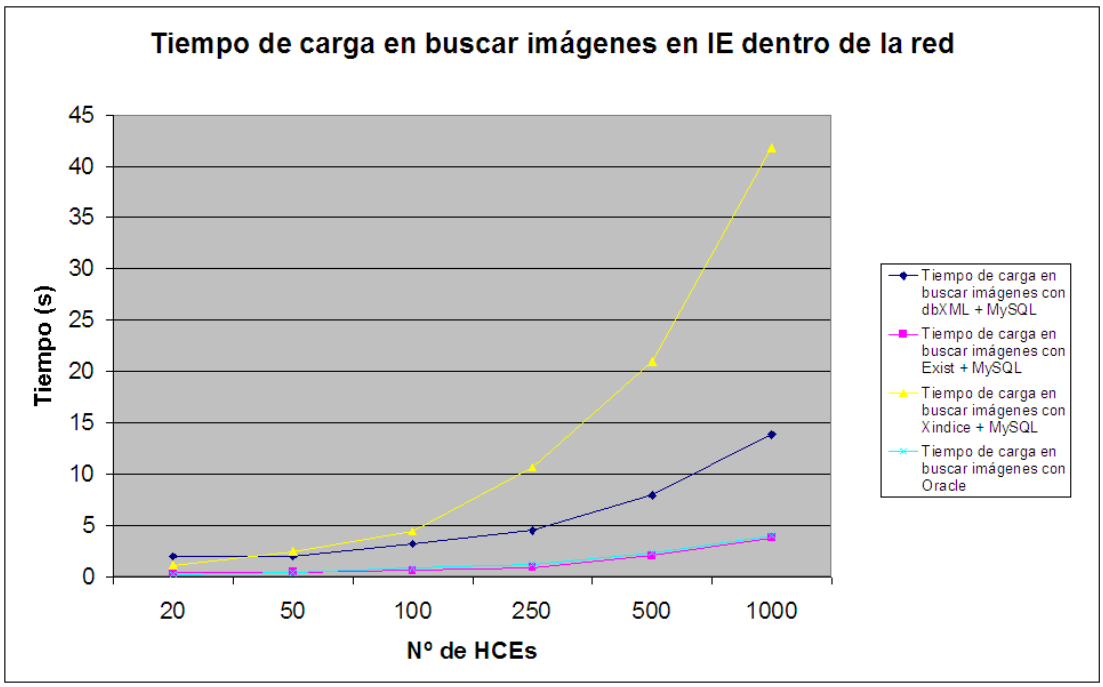


Figura 140. Tiempo medio de carga en buscar imágenes en IE dentro de la red. Fuente: propia.

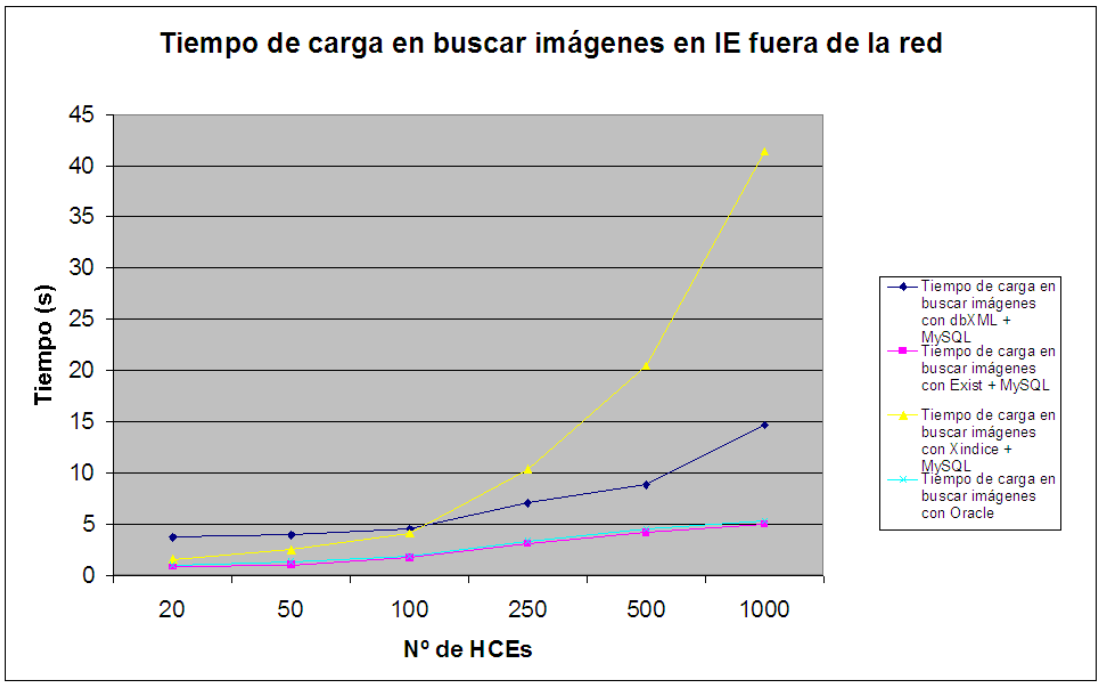


Figura 141. Tiempo medio de carga en buscar imágenes en IE fuera de la red. Fuente: propia.

La figura 140 muestra el tiempo medio de carga en buscar imágenes dentro de la red y la figura 141 el mismo tiempo fuera de la red ETSIT. Se puede observar que los

menores tiempos de carga se obtienen empleando eXist + MySQL.

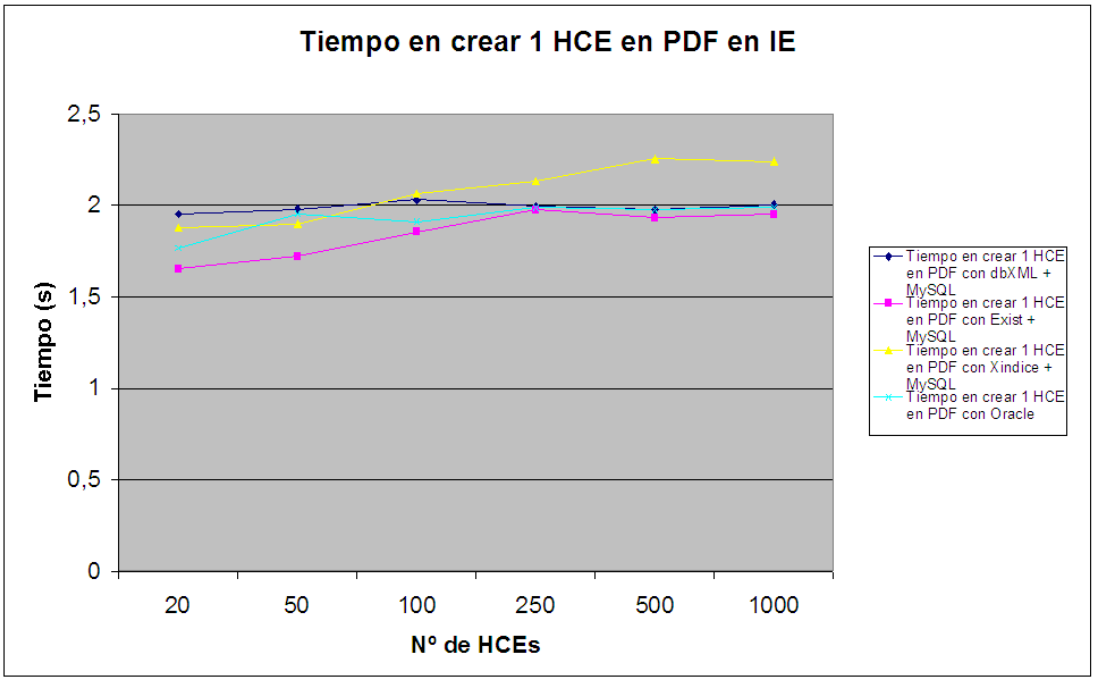


Figura 142. Tiempo medio en crear 1 HCE en PDF en IE. Fuente: propia.

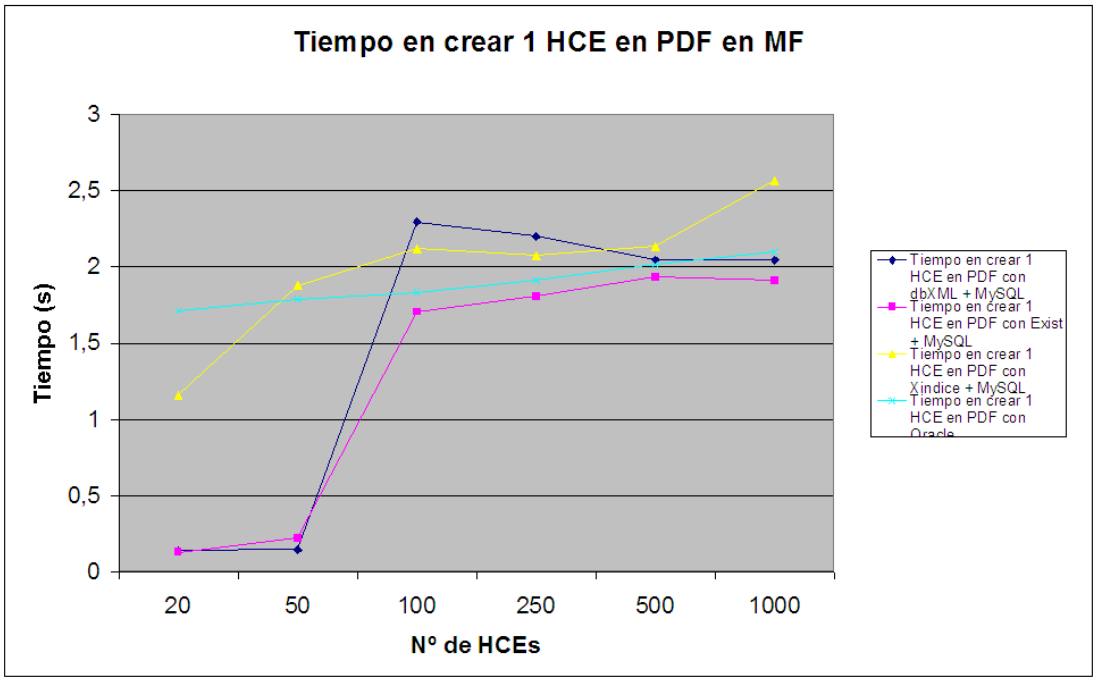


Figura 143. Tiempo medio en crear 1 HCE en PDF en MF. Fuente: propia.

La figura 142 muestra el tiempo medio en crear 1 HCE en PDF en IE y la figura 143 en MF. Se puede observar que los menores tiempos de carga se obtienen con la BD

Oracle 10g en MF, y con eXist + MySQL empleando el navegador IE.

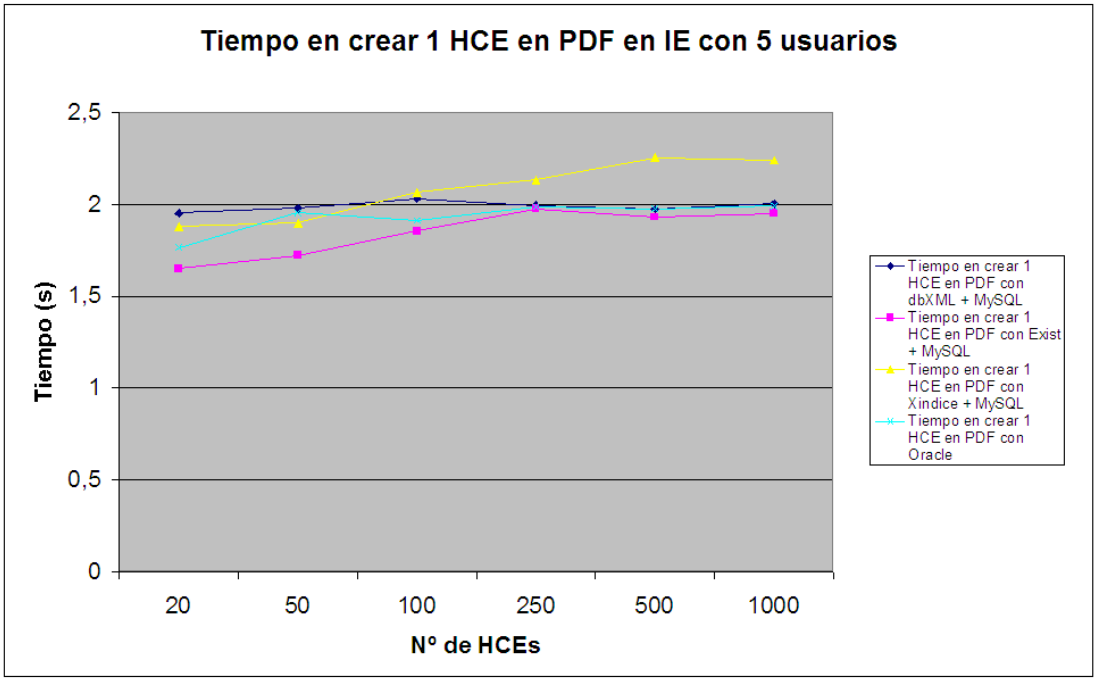


Figura 144. Tiempo medio en crear 1 HCE en PDF en IE con 5 usuarios. Fuente: propia.

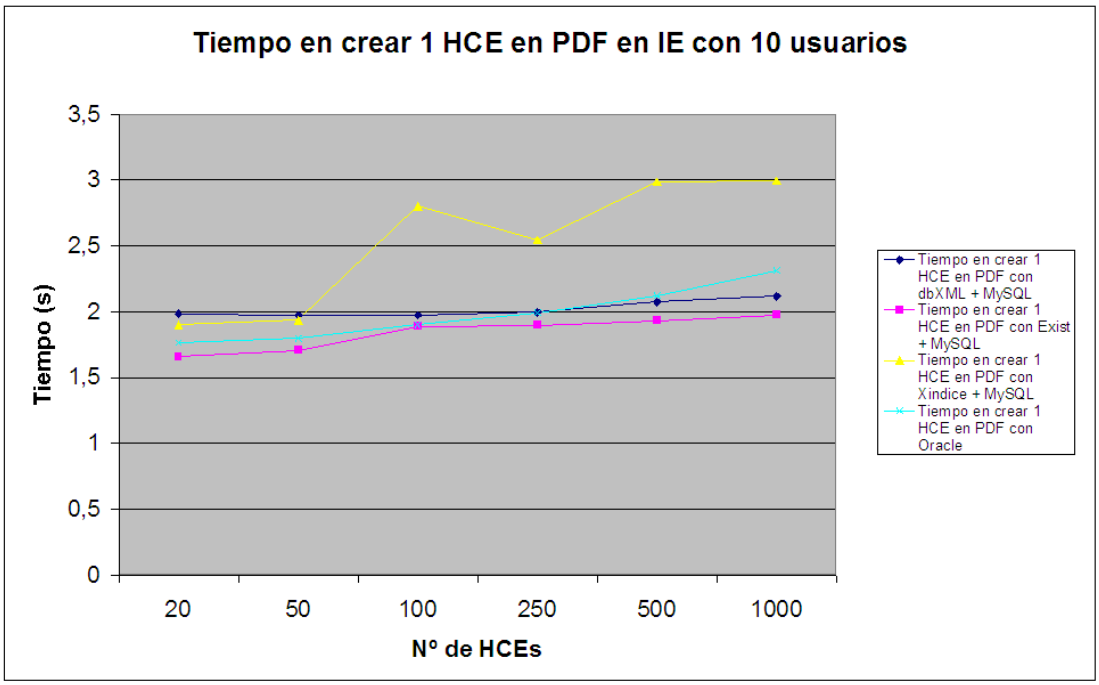


Figura 145. Tiempo medio en crear 1 HCE en PDF en IE con 10 usuarios. Fuente: propia.

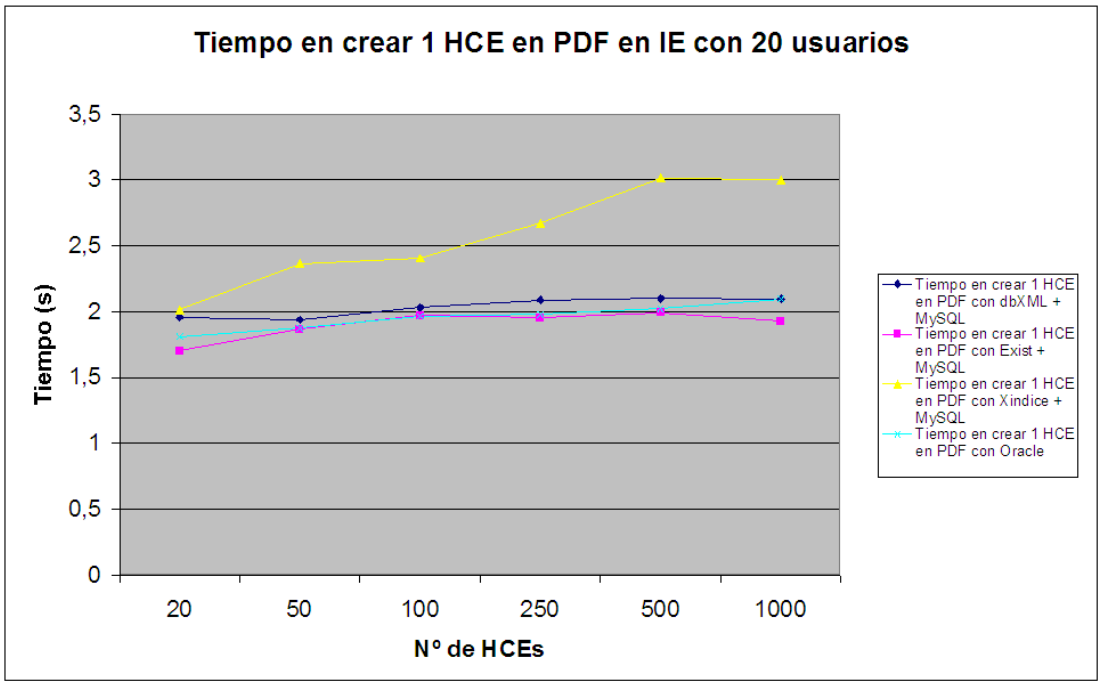


Figura 146. Tiempo medio en crear 1 HCE en PDF en IE con 20 usuarios. Fuente: propia.

La figura 144 muestra el tiempo medio en crear 1 HCE en PDF con 5 usuarios, la figura 145 con 10 usuarios y la figura 146 con 20 usuarios. Se puede observar que los menores tiempos de carga se obtienen con eXist + MySQL en los tres casos.

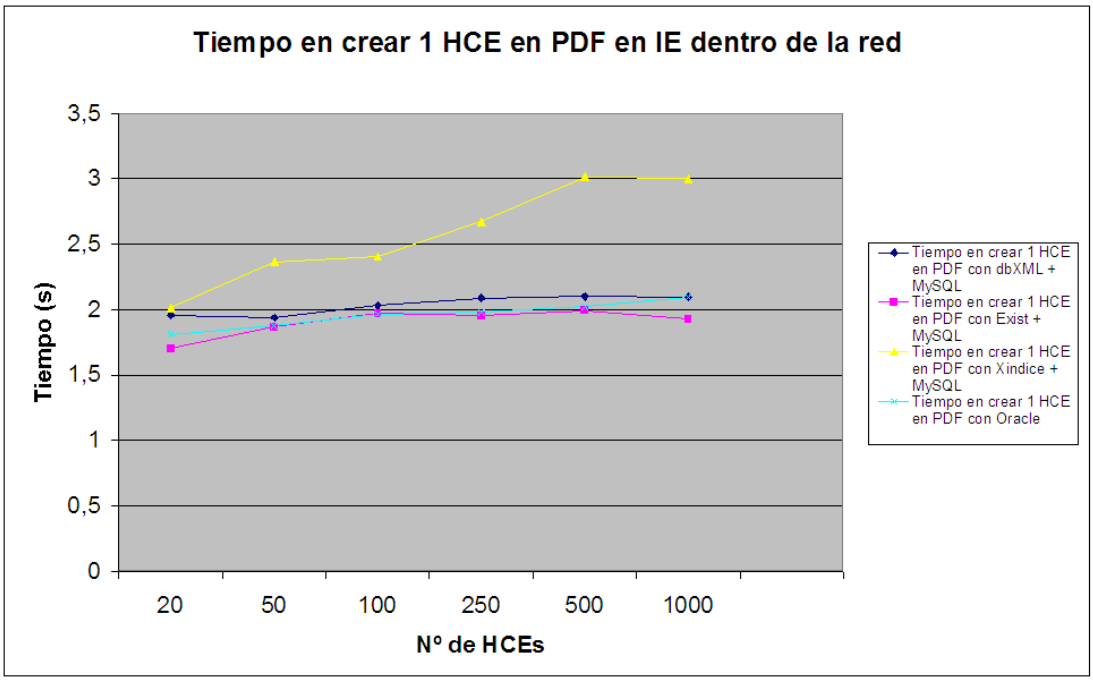


Figura 147. Tiempo medio en crear 1 HCE en PDF en IE dentro de la red. Fuente: propia.

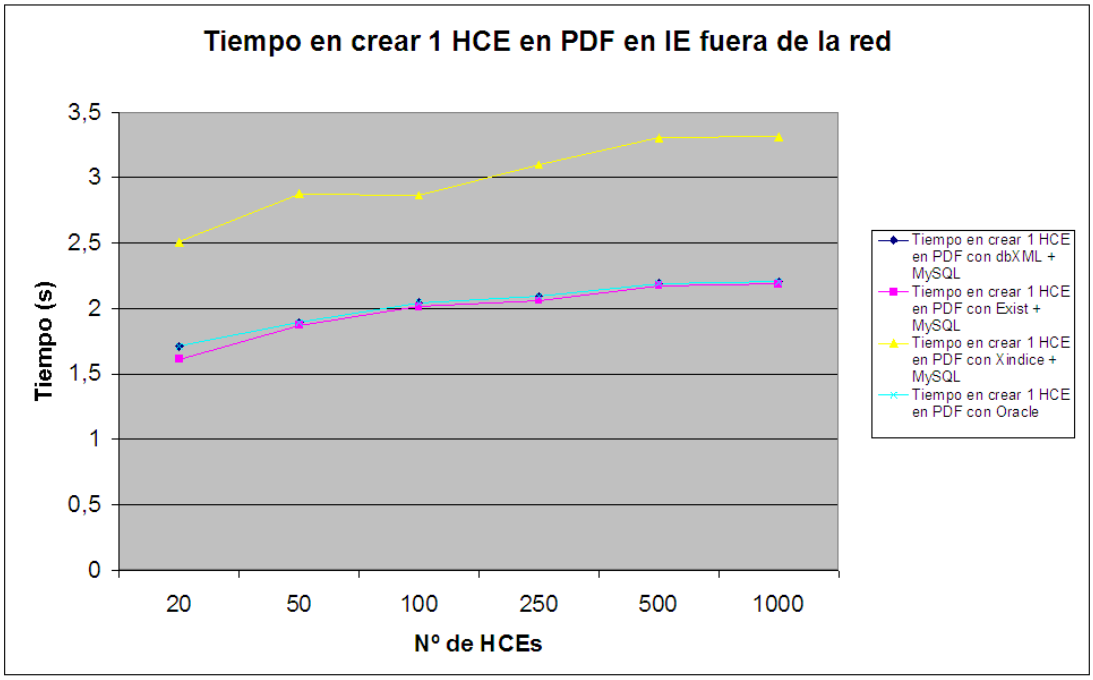


Figura 148. Tiempo medio en crear 1 HCE en PDF en IE fuera de la red. Fuente: propia.

La figura 147 muestra el tiempo medio en crear 1 HCE en PDF dentro de la red y la figura 148 el mismo tiempo fuera de la red ETSIT. Se puede observar que los menores tiempos de carga se obtienen empleando eXist + MySQL.

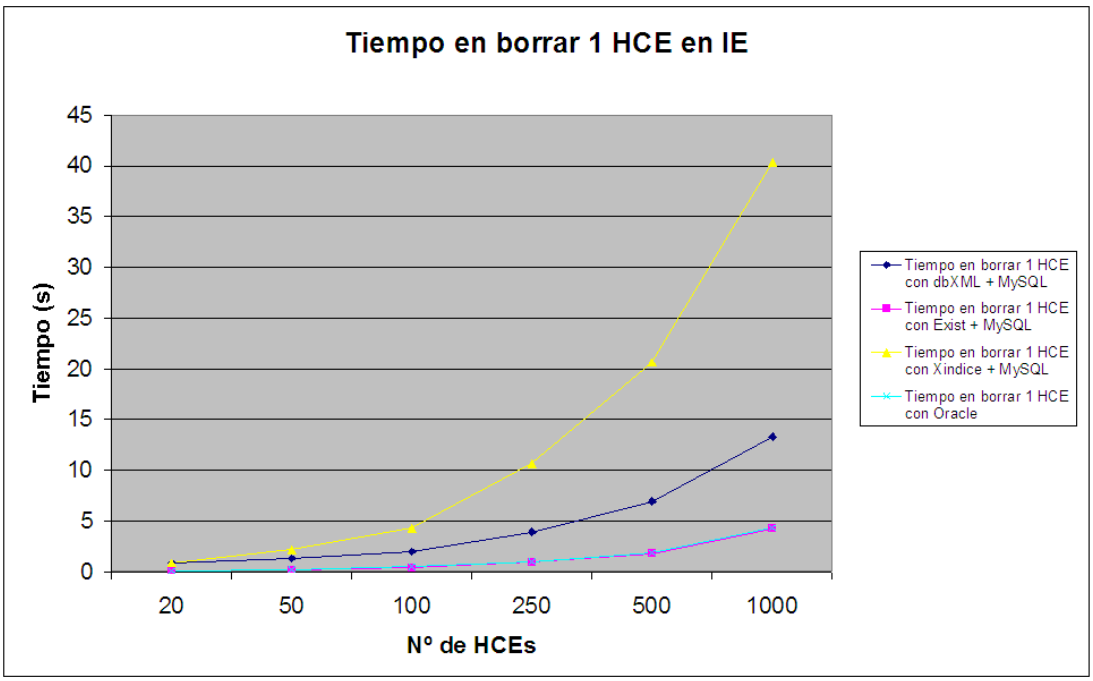


Figura 149. Tiempo medio en borrar 1 HCE en IE. Fuente: propia.

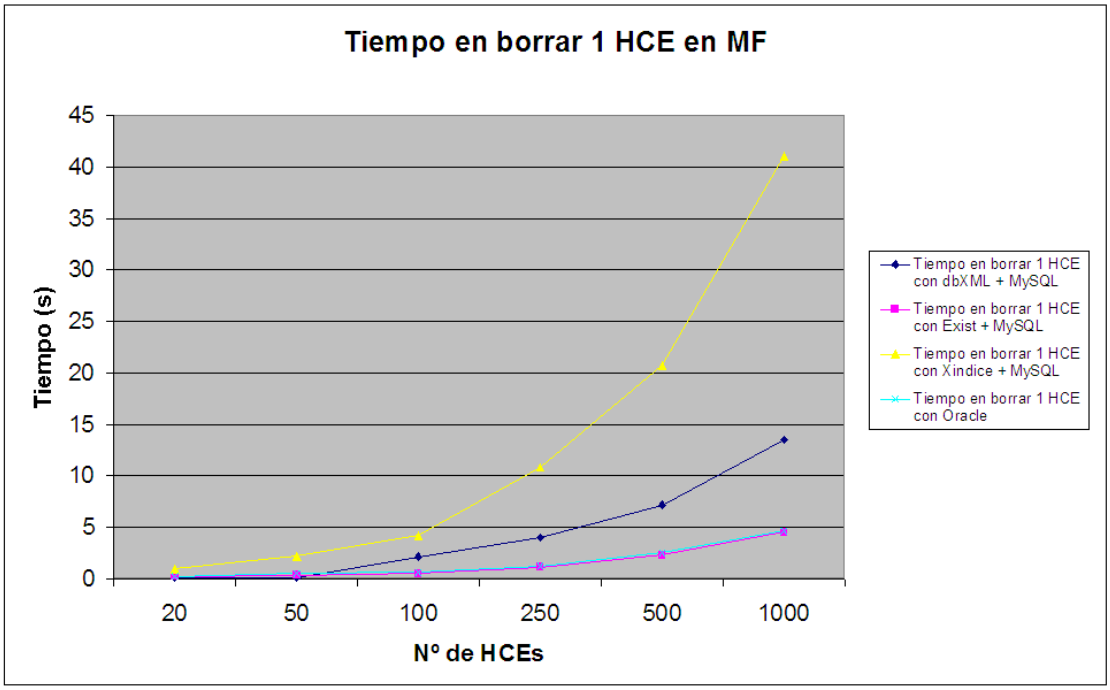


Figura 150. Tiempo medio en borrar 1 HCE en MF. Fuente: propia.

La figura 149 muestra el tiempo medio en borrar 1 HCE en IE y la figura 150 en MF. Se puede observar que los menores tiempos de carga se obtienen con la BD Oracle 10g en MF, y con eXist + MySQL empleando el navegador IE.

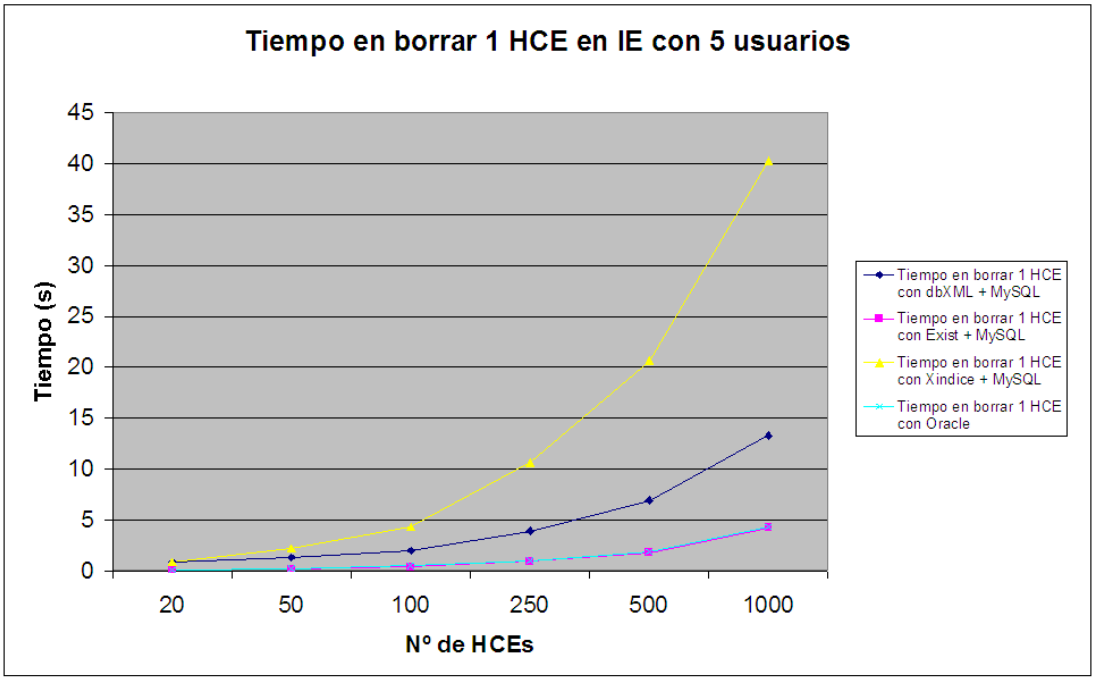


Figura 151. Tiempo medio en borrar 1 HCE en IE con 5 usuarios. Fuente: propia.

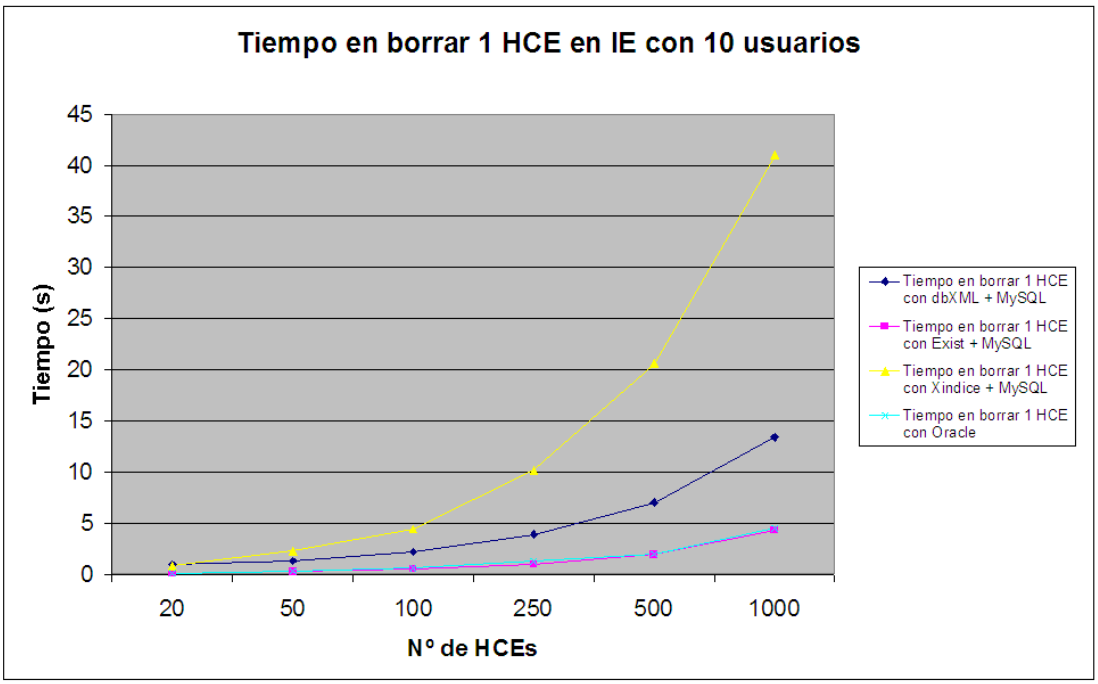


Figura 152. Tiempo medio en borrar 1 HCE en IE con 10 usuarios. Fuente: propia.

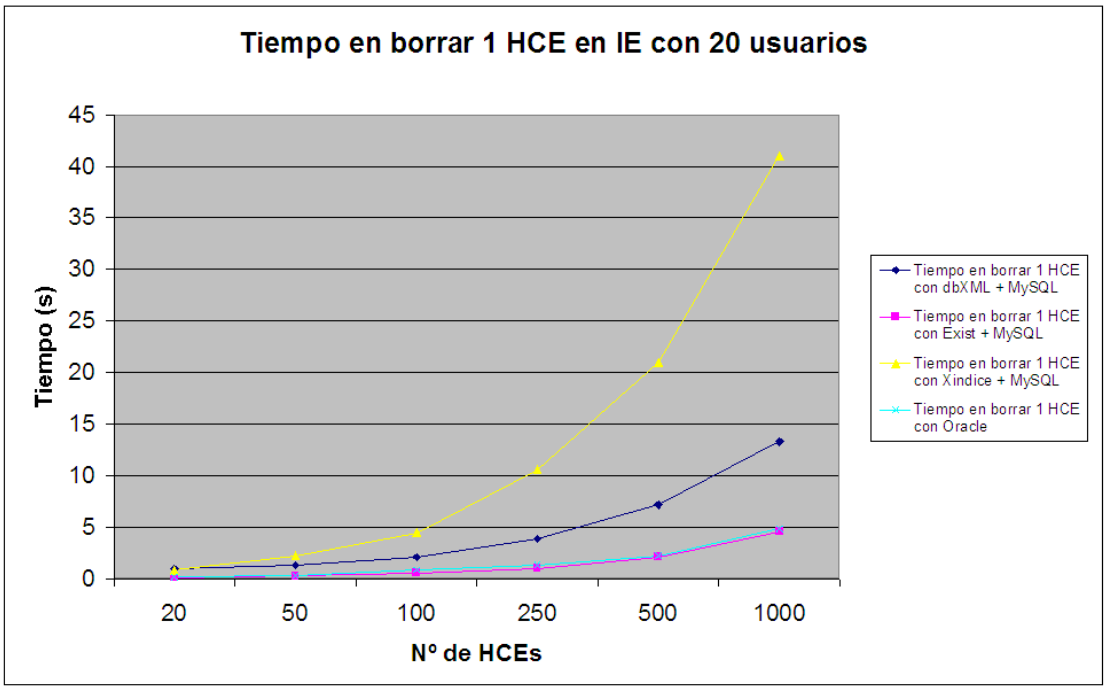


Figura 153. Tiempo medio en borrar 1 HCE en IE con 20 usuarios. Fuente: propia.

La figura 151 muestra el tiempo medio en borrar 1 HCE con 5 usuarios, la figura 152 con 10 usuarios y la figura 153 con 20 usuarios. Se puede observar que los menores tiempos de carga se obtienen con eXist + MySQL en los tres casos.

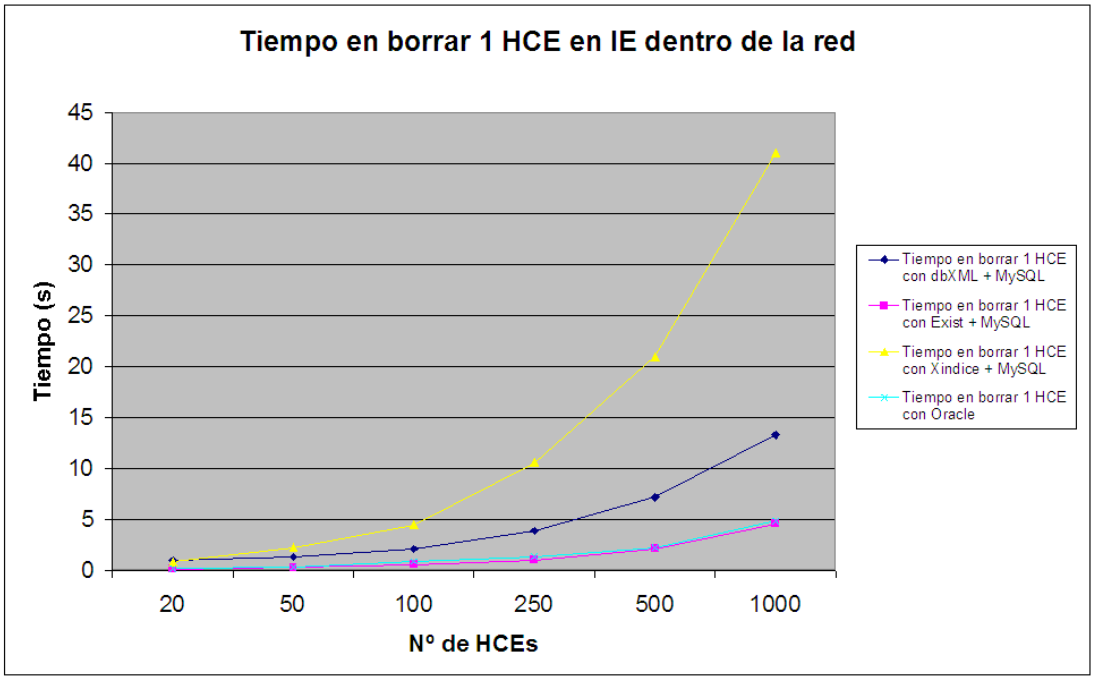


Figura 154. Tiempo medio en borrar 1 HCE en IE dentro de la red. Fuente: propia.

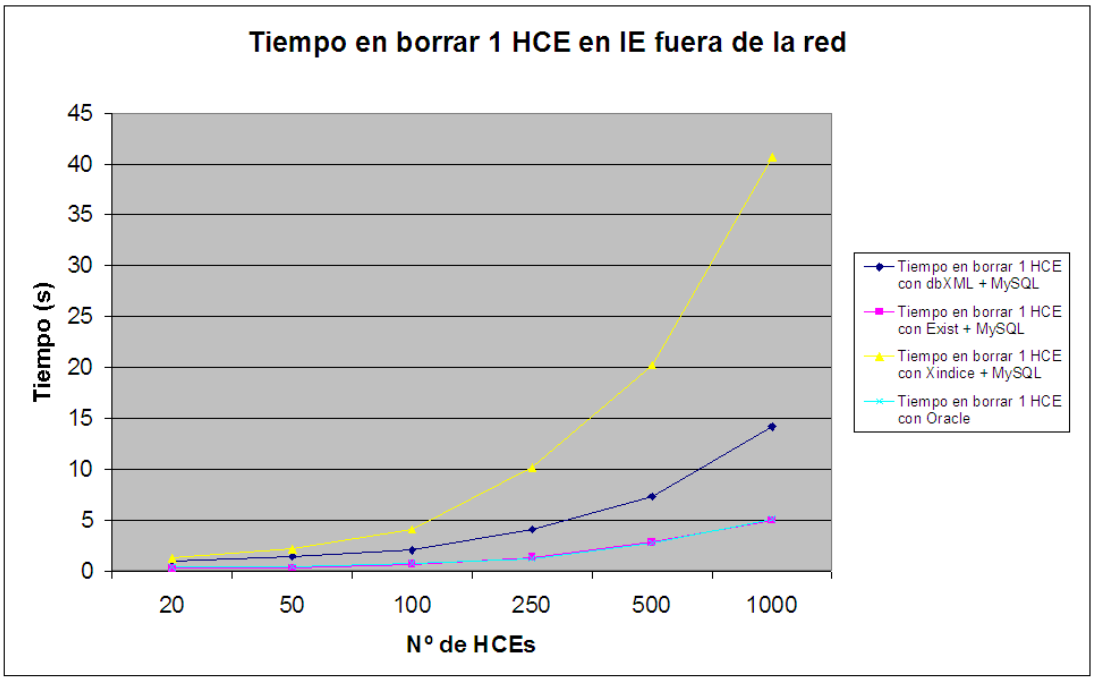


Figura 155. Tiempo medio en borrar 1 HCE en IE fuera de la red. Fuente: propia.

La figura 154 muestra el tiempo medio en borrar 1 HCE en IE dentro de la red y la figura 155 el mismo tiempo fuera de la red ETSIT. Se puede observar que los menores tiempos de carga se obtienen empleando eXist + MySQL.

Las conclusiones que se pueden extraer observando las figuras anteriores son las siguientes:

- Se ha comprobado que la aplicación es más eficiente empleando las BDs MySQL 5.0 y eXist 1.1.1.
- Aplicación más rápida con el navegador Web MF.
- El número de usuarios no influye casi nada en los tiempos de carga.
- Cuando mayor es el número de HCE, más lenta es la aplicación.
- Tiempo de carga de imágenes DICOM mayor que de las JPEG.

8

Discusión

8.1. INTRODUCCIÓN 338
8.2. APLICACIÓN TELEOFTALWEB VS. OTRAS APLICACIONES DE HCE 338
8.3. TIEMPOS DE RESPUESTA OBTENIDOS 350

8.1. Introducción

En este capítulo se van a discutir los resultados obtenidos en el diseño, desarrollo, puesta en marcha y evaluación de la aplicación **TeleOftalWeb**, así como la estandarización del sistema siguiendo los estándares HL7/CDA, DICOM e ISO/TC 215.

Para el diseño de **TeleOftalWeb** fue necesario dividir la aplicación en dos módulos bien diferenciados: gestor y usuario. El módulo gestor hace uso de la BD MySQL 5.0 en las tres primeras versiones de la aplicación (y de Oracle 10g en la cuarta), además se encarga de almacenar los datos personales de los médicos y pacientes dados de alta en sistema. El **manual de usuario** de dicho módulo aparece en el **Apéndice B**.

A continuación, se va a llevar a cabo una comparativa entre diferentes aplicaciones de HCEs publicadas en revistas de impacto, conferencias, etc. con la aplicación desarrollada en la presente Tesis, **TeleOftalWeb**.

8.2. Aplicación TeleOftalWeb *vs. otras aplicaciones de HCE*

Son numerosas las dificultades que surgen cuando se intenta implementar sistemas de HCE como también escasa la literatura que relate estas experiencias a nivel mundial (Vazquez *et ál.* 2008).

A través de la aplicación **TeleOftalWeb**, los médicos pueden analizar los HCE desde cualquier lugar. Ellos sólo necesitan un ordenador con conexión a Internet. La aplicación verifica los estándares relativos a privacidad y confidencialidad. No obstante, la velocidad de la aplicación depende del número de usuarios en el sistema. Cuando hay muchos usuarios, la velocidad es menor.

El estándar DICOM es empleado en muchos campos de la medicina como son: cardiología, radiología, endoscopía, oftalmología, dermatología, etc. En un principio fue muy exitoso en las especialidades de cardiología y radiología, pero está empezando a tener éxito en otras especialidades. El Departamento de *Veterans Affairs* realizó un trabajo en el que empleó DICOM en otras especialidades como la oftalmología (Kuzmak & Dayhoff 2000). En (Kuzmak & Dayhoff 2003), emplean DICOM para

aplicaciones de imagines en color en las especialidades de endoscopia gástrica y oftalmología.

El estándar HL7/CDA está presente en algunas especialidades médicas como cardiología (Marcheschi *et ál.* 2004), entre otras.

Tras la revisión de más de 200 artículos sobre sistemas de HCE en uso desde 1995 hasta la actualidad. Analizamos HCE en diferentes especialidades como pediatría (Ginsburg 2007), departamento de urgencias (Amouh *et ál.* 2005), oncología (James *et ál.* 2001), etc. En el sistema telemático para oncología, ellos usan un *datawarehouse* como servidor de HCE. Los autores no muestran la estandarización del sistema. El sistema Web de HCE de urgencias es un sistema multiplataforma, multiusuario, que emplea Java y usa el estándar *openEHR*. En el sistema de HCE en pediatría, los autores emplean estándares abiertos. Chew *et ál.* (1998) desarrollaron la aplicación OphthWeb dentro de un proyecto multidisciplinario de HCE en Singapore. Este sistema almacena tanto el texto como las imágenes asociadas con una completa exploración oftalmológica, como el caso de **TeleOftalWeb**.

En (Chen *et ál.* 2007), los autores diseñan y desarrollan un sistema basado en una plantilla (llamado Julius). El sistema ha sido implementado, testeado y desarrollado en *Stockholm*, *Sweden*. En la aplicación OpenSDE, los autores han expandido la metodología de modelo tradicional en filas con columnas adicionales que permiten una representación de narrativa médica (Los *et ál.* 2004).

En (Taddei *et ál.* 1997) diseñan y desarrollan un sistema Web de HCE para la especialidad de cardiología en un instituto de salud italiano. El sistema emplea un doble modelo de BD: una BD Oracle y otra DB2 de IBM. El desarrollo de este sistema se incluye dentro del proyecto SPERIGEST.

Existen otras aplicaciones Web como CareWeb™ que emplea el estándar HL7 (Halamka *et ál.* 1999). Otras aplicaciones de HCE son: PHIMS y CipherMe. PHIMS es un repositorio Web de información de salud de los pacientes, que suministra interfaces para el almacenamiento de información de salud (Kim 2006). La arquitectura de CipherMe permite almacenar de forma segura los HCE y gestionarlos de acuerdo a una serie de necesidades y obligaciones legales. Todos los datos en el sistema son XML (Hansen 2006).

Becker & Sewell (2004) presentan un sistema de HCE basado en reglas específicas definidas por el Servicio de Salud de Reino Unido. Dicho sistema se denomina Cassandra. En (Brelstaff *et ál.* 2001) construyen un prototipo de cliente para

HCE, InfoDOM, basado en tecnologías Web. Emplea el protocolo SSL y maneja componentes de *software* JavaBeen. Desarrollan un visualizar de imágenes médicas. El sistema fue testeado por médicos especialistas en neurología, radiología y epilepsia de tres hospitales en Cagliari, Italia. Otros autores como (Siika *et ál.* 2005) describen el desarrollo y estructura de un sistema de HCE para pacientes con el Virus de la Inmunodeficiencia Humana (VIH) en Kenia. Cho & Park (2003) desarrollan un sistema de HCE basado en la versión coreada de la versión beta de *International Classification for Nursing Practice* (ICNP). El sistema fue evaluado en 2 hospitales coreanos por 20 enfermeras y 57 pacientes. Hannan *et ál.* (2000) presenta una investigación formativa de un sistema de HCE en el Centro de Salud Mosoriot, en Kenia. Los autores muestran el desarrollo e implementación del sistema de HCE en un área rural de Kenia, orientado a personas con VIH.

En (Karagiannis *et ál.* 2007), desarrollan un sistema Web de HCE (pEHR) que se implata en tres hospitales europeos. Participan 22 médicos especialistas y 150 pacientes. El sistema fue desarrollado para atender las necesidades de los pacientes con enfermedad cardiaca congénita, Parkinson y Diabetes tipo 2. Tanto los pacientes como los médicos quedaron satisfechos con el empleo del sistema. Cimino *et ál.* (2002) desarrollan el sistema *Patient Clinicial Information System* (PatCIS). El periodo de evaluación del mismo fue de 3 años y fue utilizado tanto por los especialistas como por pacientes. Se verificó que había mucha diversidad en el empleo del sistema por parte de dichos pacientes. Lo empleaban desde una vez al mes o menos hasta varias veces por día. En (Hanzlicek *et ál.* 2005) desarrollan una arquitectura de HCE en el Centro EuroMise basándose en proyectos y estándares europeos. El sistema desarrollado se denomina MUDR. Tiene una arquitectura de tres capas y emplea el lenguaje XML para la comunicación entre el servidor de aplicación y el cliente. Los autores evalúan su efectividad. De Clercq et ál. (2001) describen y discuten sobre el diseño y empleo de un sistema de HCE personalizado para gestionar enfermedades crónicas. El sistema se basa en tecnologías Web y permite la interacción entre los médicos especialistas y los pacientes. Como ejemplo se presenta un sistema en el área de la Diabetes.

La talasemia es un trastorno hereditario que afecta la producción de hemoglobina normal (un tipo de proteína presente en los glóbulos rojos cuya función es transportar oxígeno a los tejidos del cuerpo). La talasemia incluye varias formas diferentes de anemia. La gravedad y el tipo de anemia dependen del número de genes que estén afectados. La beta-talasemia es causada por mutaciones en la cadena beta de

la molécula de hemoglobina y se trata de una enfermedad hereditaria. En (Deftereos *et ál.* 2001) desarrollan un sistema de HCE (JAnaemia) dirigido a pacientes con la enfermedad de beta-talasemia. El sistema Java se basa en el estándar ENV 13606:1999 de CEN/TC 251. El sistema se evaluó entre pacientes con dicha enfermedad en cuatro hospitales de Grecia.

Eyers *et ál.* (2006) muestran un prototipo de HCE, CBCL OASIS. Describen los componentes de su arquitectura de control de acceso OASIS. El prototipo cumple los niveles de seguridad requeridos por el Sistema de Salud de Reino Unido.

En la Tabla 60, comparamos diferentes aplicaciones de HCE en relación a la especialidad en la que están implantados, si aplican los estándares HL7/CDA y DICOM.

Nombre	Especialidad	Emplea tecnologías Web	Aplica HL7/CDA	Aplica DICOM
CareWeb™	Todas	✓		
CipherMe	Todas			
InfoDOM	Neurología, Epilepsia y Radiología	✓		
JAnaemia	Beta-Talasemia			
Julius	Todas	✓		
MUDR	Todas	✓		
OpenSDE	Todas	✓		
Patois	Todas			
PedOne System	Pediatría	✓		
pEHR	Cardiología, Diabetes y Parkinson	✓		
PHIMS	Geriatría	✓		
TeleOftalWeb	Oftalmología	✓	✓	✓

Tabla 60. Comparativa entre diferentes sistemas de HCE. Fuente: propia.

Nombre del sistema (o localización)	País	Número de publicaciones	Fechas de publicación
CareHaven	China	1	2001
CareWeb	Estados Unidos	2	[1998, 2000]
Chili	Alemania	1	2004
Clicks	Israel	1	2003
Clinical Management System	China	1	2005
Daegu – Korea	Corea	1	2003
GDGHA – General Hospital of Athens	Grecia	4	[2002, 2004]
GP *Software* Integration Project	Australia	1	2003
H:S	Dinamarca	1	2005
health@net	Austria	2	[2004, 2005]
Heilderberg	Alemania	1	2001
MUDR	República Checa	1	2004
Munster	Alemania	5	[2001, 2005]
National Tech Univ. Athens	Grecia	1	2002
OACIS	Australia	1	2003
old@home	Suiza	1	[2004, 2005]
Oxford Clinical Intranet	Reino Unido	2	[1999, 2004]
RMRS – Regenstrief	Estados Unidos	3	[1999, 2001]
Shanghai Hospital	China	1	2005
SPERIGEST	Italia	5	[1997, 2002]
SUP	Dinamarca	1	2005
Synapses	Irlanda	4	[1997, 2001]
TACWeb	Italia	1	2003
Thessaloniki	Grecia	1	2003
V-Net Med	Alemania	1	2005
Web-EPR	Brasil	1	2001

Tabla 61. Sistemas de Información médica relacionados con HCE publicados desde 1995 hasta 2005.

Fuente: (Cruz-Correia *et ál.* 2007)

Cruz-Correia *et ál.* (2007) hicieron una revisión bibliográfica de las BDs: Medline (Medline 2008), IEEE (IEEE 2008) e *ISI* Web *of Knowledge* (ISI 2008) desde 1995 hasta 2005 sobre los SI en telemedicina. En la Tabla 61, se ha realizado un listado de los sistemas relacionados con HCE, el país donde se han desarrollado, el número de publicaciones generadas y el año de publicación de las mismas.

En la Tabla 62, se muestran aplicaciones de tipo comercial orientadas a la especialidad de oftalmología.

APLICACIÓN	PROVEEDOR	CARACTERÍSTICAS
Ifa Ophthalmology *Software*	ifa systems AG	Soporta SNOMED/ ICD 9/ ICD 10/ ICPM/ DRGs/ LOINC
AllMeds for Eye	AllMeds, Inc.	Empleado en óptica y oftalmología. Soporta ICD 9. Privacidad y seguridad
ChartLogic™ Ophthalmology Electronic Medical Office	ChartLogic	Para TabletPC. BD: MSDE (2GB) o servidor SQL
Compulink's Ophthalmology Advantage	Compulink Business Systems, Inc.	Integrado con *Microsoft Word.* BD: SQL
MaximEyes Optometric & Ophthalmology *Software*	First Insight Corporation	Soporta códicos CPT y ICD 9. Permite visualizar imágenes en 3D.
ophthalmic*suite*™	Medflow Inc.	HL7, DICOM, IHE. Base de datos: *Microsoft SQL Server 2000.* TabletPC
ophthalmic*suite*ASC	Medflow Inc.	HL7, DICOM, IHE. Base de datos: *Microsoft SQL Server 2000.* TabletPC
Optical Shop *Software* Suite	Medflow Inc.	HL7, DICOM, IHE. *Windows XP Professional SP2*
medisoft Ophthalmology	Medisoft Limited	Códigos OPCS4.3

Tabla 62. Soluciones de HCE comerciales en oftalmología. Fuente: propia.

Otras aplicaciones de HCE similares a TeleOftalWeb.

RETISALUD

En 2001, la consejería de Sanidad de la Comunidad Autónoma de Canarias lanza el proyecto piloto DRAGO para la integración de la Historia Clínica Electrónica. La

existencia de esta historia informatizada, única por paciente en toda la comunidad posibilita la transferencia de información entre los distintos profesionales que intervienen en ella y facilita la interrelación entre la red de Atención Primaria, la de Atención Especializada y la Hospitalaria (Gobierno de Canarias 2009). Poco después, en 2005, sus avances son aplicados al mundo de la oftalmología, para la aparición del programa RETISALUD. En él destaca la incorporación de los médicos generalistas de Atención Primaria como parte del primer escalón. Su filtrado de pacientes diabéticos llega a la atención oftalmológica en los centros hospitalarios, en este caso, a la sección de Retina del Servicio de Oftalmología del Hospital Universitario de Canarias.

Los resultados arrojados por la implantación del sistema RETISALUD son un éxito. A fecha del 31 de Mayo de 2009, se han atendido 31.279 pacientes con diabetes desde el 1 de Junio de 2007 y se da cobertura a un total de 60 zonas de Canarias.

YKONOS

España puede presumir de ser la cuna de uno de los proyectos más laureados a nivel internacional en el ámbito de los HCEs. La Alianza Mundial de Tecnología y Servicios de Información *World Information Technologies Services Alliance* (WITSA), galardonó en 2008 al proyecto castellanomanchego Ykonos como el mejor proyecto de tecnologías de la información (WITSA Global ICT Excellence Awards 2009).

Gracias al sistema Ykonos de imagen médica digital, que ya está operativo en todos los hospitales de la región y en los Centros de Especialidades, las radiografías, ecografías y otras pruebas de modalidades diagnósticas ya no se entregan en formato papel al paciente si no que son enviadas junto con su correspondiente informe a una red interna en soporte digital a la que se tiene acceso desde cualquier otro centro de la región. El éxito de este sistema lo ha hecho extensible a otras áreas como áreas como la Anatomía Patológica, Medicina Nuclear, Dermatología, Aparato digestivo y Oftalmología, haciendo de ésta última una de las especialidades punteras en todo el panorama nacional. Algunas de las ventajas que proporciona este sistema al ciudadano son:

- Las pruebas radiológicas anteriores están siempre disponibles y en cualquier momento, con lo cual es posible comparar con las últimas pruebas realizadas.

- No es necesario la repetición de pruebas cuando un paciente se desplaza de un centro a otro.
- Es posible acceder a estas pruebas incluso desde los centros de atención primaria, con lo cual los médicos de cabecera tienen también conocimiento y acceso inmediato a la evolución del paciente.
- Se reduce el tiempo de espera para la obtención de las pruebas radiológicas ya que se elimina la etapa de revelado de la película convencional donde el técnico especialista en radiodiagnóstico

En la figura 156 se puede observar la interfaz del sistema. Si bien por el momento, dado su estado de madurez, no es comparable a **TeleOftalWeb**, si que sirve para fundar las expectativas de las funcionalidades que éste puede llegar a aunar.

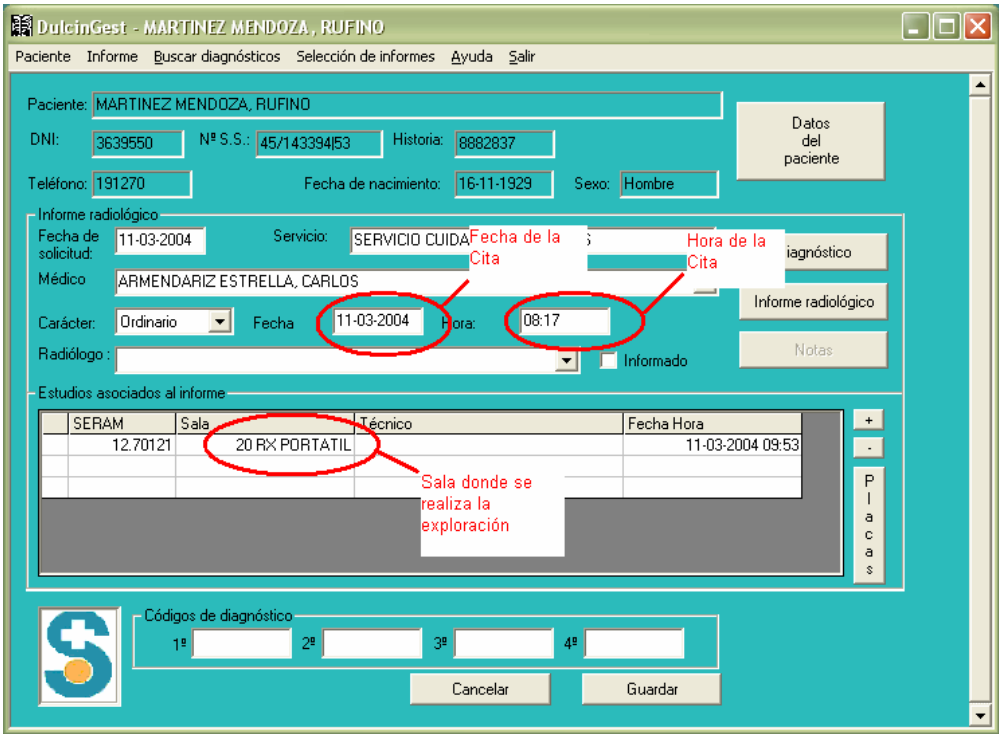


Figura 156. Interfaz del sistema Ykonos (WITSA Global ICT Excellence Awards 2009).

Con todo ello, en la actualidad, este sistema está instalado en los siguientes hospitales:

- Hospital Virgen de La Salud (Toledo).
- Hospital Virgen del Valle (Toledo).

- Hospital Nacional de Parapléjicos (Toledo).
- Hospital La Mancha Centro (Alcázar de San Juan).
- Hospital Virgen de Altagracia (Manzanares).
- Hospital Ntra Sra. Del Prado (Talavera de la Reina).
- Hospital Universitario de Guadalajara.
- Hospital Virgen de la Luz (Cuenca).
- Hospital Gutierrez Ortega (Valdepeñas).
- Hospital General de Ciudad Real.
- Hospital General de Albacete (Albacete) (En despliegue).
- Hospital del Perpetuo Socorro (Albacete) (En despliegue).

El listado de centros que lo incorporan da una idea del gran éxito del sistema. Por otra parte, desde su implantación en el año 2003 y hasta la actualidad, se han realizado un total de 1.132.764 estudios repartidos en los diferentes hospitales (SESCAM 2009).

OFTALCLINIC

Las empresas tecnológicas privadas españolas también han encontrado su mercado en el sector de la sanidad. Una muestra es la herramienta OftalClinic, ofertada por DISA S.L. (DISA S.L. 2009).

OftalClinic es uno de los sistemas integrales para la gestión de clínicas oftalmológicas más completos del mercado. Su software no se limita a la funcionalidad de gestión de HCE sino que además permite el control de la administración y dirección financiera. Este sistema resulta comparativamente interesante respecto a TeleOftalWeb al basarse al igual que ésta en un sistema cliente-servidor sobre una base de datos. Por otra parte, el rango de funcionalidades y ventajas que incluye son análogas a las que propone la herramienta **TeleOftalWeb** a evaluar:

- Sustitución del papel por la HCE.
- Agilización de los procesos internos de trabajo.
- Comunicaciones internas.

- Informes automatizados.
- Mínimo uso del teclado mediante la adquisición automática de datos e imágenes.
- Selección de información mediante listas desplegables de datos más frecuentemente usados,
- Realización de dibujos o esquemas sobre plantillas o imágenes reales.

La figura 157 trata de ilustrar la interfaz que proporciona OftalClinic:

Figura 157. Interfaz de la herramienta OftalClinic (DISA S.L. 2009).

En la Tabla 63, aparece el listado de aplicaciones de HCE de *software* libre. En la primera columna se encuentra el nombre de la aplicación, en la segunda sus características más significativas y en la tercera la dirección Web donde se puede encontrar toda la información detallada de la propia aplicación.

APLICACIÓN	CARACTERÍSTICAS	PÁGINA WEB
CHITS	Dirigido a entornos rurales. Modular, tecnologías Web, soporta Linux y Windows	http://www.chits.ph
ClearHealth	HL7, vía Web	http://www.clear-health.com
CottageMed™	Aplicación *FileMaker Pro*, cumple los estándares de seguridad HIPAA	http://www.cottagemed.org
FFEHR	Web. Soporta Linux y Windows	http://www.iosn.net/Members/amarcelo/FFEHR_HEADWAY
HOSxp	Vía Web de información cliente/servidor empleado en 150 hospitales en Tailandia. Soporta Windows	http://hosxp.net
IndivoHealth	Vía Web, distribuido, multiplataforma	http://www.indivohealth.org
MedClipse	Nativo y multiplataforma	http://sourceforge.net/projects/medclipse
Med'In Tux	Soporta Windows, Mac y Linux	http://medintux.org/medintux
MirrorMed	Basado en PHP. Emplea plataformas Web y soporta Linux	http://www.mirrormed.org
OpenEMR	Web, soporta Mac y Linux. Estándares de seguridad y privacidad HIPAA	http://www.oemr.org
OpenMRS	Soporta Windows, Mac y Linux	http://openmrs.org/wiki/OpenMRS
OpenVista	Seguro, vía Web y multiplataforma.	http://sourceforge.net/projects/openvista
OSCARMcMaster	Vía Web multiplataforma	http://www.oscarmcmaster.org
PatientOS	Nativo multiplataforma	http://www.patientos.org
Tolven Health Record	Emplea Java 5, *Faces*, *Facelets*, AJAX, base de datos relacional y *LDAP* para seguridad. Sistema Web multiplataforma	http://www.tolven.org
Ultimate EMR	Vía Web. Soporta Linux y Windows	http://www.uemr.com
WorldVistA	Está basado en el sistema VistA del Departamento *Veterans Affairs* de USA. Soporta Linux y Windows	http://worldvista.org

Tabla 63. Aplicaciones de HCE de *software* libre. Fuente: propia.

Finalmente en la Tabla 64, aparece el listado de aplicaciones de HCE a nivel de investigación, comerciales y de *software* libre comparadas con la aplicación Web **TeleOftalWeb**.

Aplicación	Características
CareWeb™	HL7
CipherMe	Todas las especialidades
InfoDOM	Neurología, Epilepsia y Radiología. Tecnologías Web
JAnaemia	Sistema Web de HCEs para la enfermedad de la Beta-Talasemia
Julius	Tecnologías Web
MUDR	Tecnologías Web
OpenSDE	Tecnologías Web
PatCIS	Tecnologías Web
PedOne System	Pediatría
pEHR	Cardiología, Diabetes y Parkinson. Tecnologías Web
PHIMS	Geriatría. Tecnologías Web
TeleOftalWeb	Oftalmología. Tecnologías Web. HL7/CDA, DICOM.
CHITS	Dirigido a entornos rurales. Modular, tecnologías Web, soporta Linux
ClearHealth	HL7, vía Web
CottageMed™	Aplicación *FileMaker Pro*, cumple los estándares de seguridad HIPAA
FFEHR	Web. Soporta Linux y Windows
HOSxp	Vía Web de información cliente/servidor empleado en 150 hospitales en Tailandia. Soporta Windows
IndivoHealth	Vía Web, distribuido, multiplataforma
MedClipse	Nativo y multiplataforma
Med'In Tux	Soporta Windows, Mac y Linux
MirrorMed	Basado en PHP. Emplea plataformas Web y soporta Linux
OpenEMR	Web, soporta Mac y Linux. Estándares de seguridad y privacidad HIPAA
OpenMRS	Soporta Windows, Mac y Linux
OpenVista	Seguro, vía Web y multiplataforma.
OSCARMcMaster	Vía Web multiplataforma
PatientOS	Nativo multiplataforma
Tolven Health Record	Emplea Java 5, *Faces*, *Facelets*, AJAX, base de datos relacional
Ultimate EMR	Vía Web. Soporta Linux y Windows
WorldVistA	Está basado en el sistema VistA del Departamento *Veterans Affairs* de USA. Soporta Linux y Windows
Ifa Ophthalmology *Software*	Soporta SNOMED/ ICD 9/ ICD 10/ ICPM/ DRGs/ LOINC
AllMeds for Eye	Empleado en óptica y oftalmología. Soporta ICD 9. Privacidad y seguridad
ChartLogic™ Ophthalmology Electronic Medical Office	Para TabletPC. BD: MSDE (2GB) o servidor SQL
Compulink's Ophthalmology Advantage	Integrado con *Microsoft Word.* Base de datos: SQL
MaximEyes Optometric & Ophthalmology *Software*	Soporta códicos CPT e ICD 9. Permite visualizar imágenes en 3D.
ophthalmic*suite*™	HL7, DICOM, IHE. Base de datos: *Microsoft SQL Server 2000.* TabletPC
ophthalmic*suite*ASC	HL7, DICOM, IHE. Base de datos: *Microsoft SQL Server 2000.* TabletPC
Optical Shop *Software* Suite	HL7, DICOM, IHE. *Windows XP Professional SP2*
medisoft Ophthalmology	Códigos OPCS4.3
TeleOftalWeb	**HL7/CDA, DICOM. BD más eficiente: eXist. Entorno Windows**

Tabla 64. Comparativa de sistemas de HCE. Fuente: propia.

8.3. Tiempos de respuesta obtenidos

El modelo de colas M/M/1 propuesto para calcular los diferentes tiempos de respuesta, ha dado como resultado la detección de cuellos de botella en el *backbone* de la red y en la base de datos empleada en el servidor. A medida que el ancho de banda de la red aumenta, se puede observar que los tiempos de respuesta del sistema son menores. Para tasas de llegadas mayores dichos tiempos decrecen. Con el fin de decrementar los tiempos de respuesta del sistema, aconsejamos aumentar el ancho de banda de la red y emplear la base de datos con menores tiempos de servicio.

En nuestro caso de estudio la elección estaba entre las BDs eXist 1.1.1 y Oracle 10g ya que eran las más eficientes para **TeleOftalWeb,** empleando tanto teoría de colas como *benchmarking* propio. Los resultados obtenidos al emplear el modelo de colas y al realizar las pruebas orientadas a la aplicación aportan una pequeña diferencia ventajosa a eXist, que unida a factores como su gratuidad, su fácil puesta en marcha, eficiente actualización y evolución nos hicieron decantarnos por ella.

Se han detectado tras el análisis una serie de limitaciones. Los tiempos de servicio asociados a las CPUs y a los GWs 1, 2 son estimados de forma teórica. Con el cliente y el servidor empleados para realizar todas las pruebas, cuyas características aparecen en el apartado de Métodos, hemos obtenidos los resultados citados en este artículo. Con el uso de otro tipo de máquinas obtendríamos resultados diferentes.

Los modelos empleados en este trabajo, tanto el de colas como el *benchmarking* propio, pueden ser utilizados para calcular tiempos de respuesta de cualquier sistema Web con una arquitectura similar al sistema evaluado. La aplicación se desarrolló inicialmente para ser empleada por especialistas pero la idea es que los pacientes también pudieran acceder a su propio HCE.

9

Conclusiones y líneas futuras

9.1. INTRODUCCIÓN 352
9.2. CONTRIBUCIONES ORIGINALES 353
9.3. LÍNEAS FUTURAS DE INVESTIGACIÓN 356

9.1. Introducción

En la presente Tesis Doctoral se ha llevado a cabo el diseño, desarrollo, puesta en marcha y evaluación de la aplicación **TeleOftalWeb** con el fin de almacenar e intercambiar HCEs y retinografías en oftalmología siguiendo los estándares internacionales: HL7/CDA, DICOM e ISO 18308. Los médicos pueden acceder a la misma a través de cualquier navegador Web.

Son múltiples las ventajas que brinda una HCE en relación con la tradicional: está siempre disponible; puede ser consultada por más de un especialista simultáneamente; es siempre legible; permite un ingreso estandarizado de los datos; evita reiteración de estudios y tratamientos; es más fácil de archivar y además permite la interacción con programas que brindan asistencia para el manejo del paciente, etc. Una de las principales barreras para su implementación es la falta de estándares en múltiples áreas, especialmente en lo referente al vocabulario médico, identificación, intercambio de datos, privacidad, seguridad y confidencialidad.

Las principales ventajas de la aplicación **TeleOftalWeb** son: su adaptación a los estándares HL7/CDA y DICOM; facilita la interoperabilidad entre aplicaciones e instituciones, y las transacciones son seguras. No obstante, la aplicación tiene algunas limitaciones sobretodo en cuanto a los tiempos de respuesta de las diferentes secciones de la misma. Estos tiempos dependen del número de usuarios del sistema, de la conexión a Internet empleada y del navegador Web usado, principalmente.

Los médicos también pueden visualizar el HCE de los pacientes en formato PDF y cualquier tipo de imagen digital asociada a dicho HC. Además, nuestra aplicación trata de resolver algunas de las barreras de adopción de los HCE en oftalmología. Los historiales y las retinografías pueden ser actualizadas por cualquier médico registrado en el sistema.

Después del cálculo de los diferentes tiempos de respuesta de la aplicación empleando diferentes BDs, se seleccionó aquella que ofrecía una serie de "mejoras" a la aplicación en cuento a gratuidad y efectividad (rapidez en tiempos de acceso) para ser implantada. La BD seleccionada fue **eXist 1.1.1**. Una vez puesta en marcha la aplicación empleando la BD eXist, se desarrolló una encuesta Web que fue contestada por los médicos que emplearon la plataforma. A través de dicha encuesta se verificó que la aplicación **TeleOftalWeb** era fácil de usar y útil para los médicos que la utilizaron.

9.2. Contribuciones originales

Una de las principales tareas de este trabajo ha sido el **desarrollo íntegro de TeleOftalWeb** (desde el diseño inicial hasta la posterior evaluación pasando por su puesta en marcha). La selección de la versión más efectiva de la aplicación (mediante el cálculo de los tiempos de respuesta **TeleOftalWeb** con diferentes modelos de BDs) supone un modelo a seguir para aquellos programadores que deseen implantar un sistema similar al nuestro, siguiendo el modelo de colas propuesto o realizando un *benchmarking* análogo al que proponemos.

Se puede concluir que **TeleOftalWeb** (empleando el modelo de colas) puede procesar la HCE de hasta 7500 pacientes por hora empleando la BD Oracle 10g con una conexión de 1 Mb y de hasta 7550 pacientes por hora empleando eXist 1.1.1 con esa misma velocidad. Con dbXML 2.0 puede procesar hasta 1500 HCEs en una hora y con Xindice 1.2 hasta 1300 HCEs en una hora con una conexión de 1Mb. Para 100 Kb el sistema sólo puede intercambiar hasta 1250 HCEs en una hora empleando cualquiera de las bases de datos citadas anteriormente. Con la comparativa realizada en este trabajo se concluye que la mejor base de datos para nuestra aplicación es eXist 1.1.1.

Actualmente el sistema Web con la base de datos más eficiente (eXist 1.1.1 que ofrece mejores tiempos de respuesta y sus características en cuanto a coste, disponibilidad, mantenimiento son favorables) está siendo utilizado por especialistas del Instituto de Oftalmobiología Aplicada de la Universidad de Valladolid (IOBA).

Un objetivo a largo plazo es la incorporación de servicios móviles que permitan a los especialistas manipular la HCE de sus pacientes en cualquier lugar desde su terminal móvil.

Además, se ha llevado a cabo la revisión bibliográfica de artículos pertenecientes a revistas sobre informática médica y Telemedicina con alto índice de impacto (*IEEE-ITB, JAMIA, Telemedicine and e-Health Journal, Journal of Telemedicine and Telecare, Methods of Information in Medicine, Medical Informatics and the Internet in Medicine, Journal of Biomedical Informatics, Journal of Medical Systems, International Journal of Medical Informatics*). Se han analizado los artículos que hacían referencia a HCE y los estándares (HL7, DICOM, XML, OpenEHR, GEHR, CORBA, IHE, de ISO/ TC 215 y CEN/ TC 251). El trabajo de la presente tesis ha sido publicado en la revista Journal of Medical Systems (De la Torre *et ál.* 2010).

Se ha prestado bastante énfasis en la revista IEEE-ITB. En la Tabla 65 se pueden observar el número de conferencias, revistas de la IEEE analizadas en el presente trabajo sobre telemedicina, HCE, y estándares. Para realizar la búsqueda bibliográfica se han introducido los términos: "telemedicine", "teleophthalmology", "EHR", "EPR", "HL7", "DICOM", "ISO/ TC 215", "OpenEHR", "GEHR", "CORBA", "IHE" y "CEN/TC 251". Se han buscado dichos términos en los distintos títulos, "*keywords*" y *abstracts* de los artículos de las diferentes revistas citadas anteriormente, desde su fecha de inicio hasta la actualidad.

ARTÍCULOS Y/O CONFERENCIAS SOBRE	CONFERENCIAS	REVISTAS	EN LA REVISTA IEEE-ITB
Telemedicine	32	10	5
EHR	43	8	5
EPR	3	17	10
HL7	42	13	4
DICOM	66	43	7
ISO/TC 215	0	0	0
OpenEHR	2	0	0
GEHR	0	1	1
CORBA	0	1	1
IHE	0	2	0
CEN/TC 251	2	0	1
Total	**193**	**98**	**34**

Tabla 65. Publicaciones de la IEEE analizadas. Fuente: propia.

En la Tabla 66 se pueden observar el número total de artículos analizados en revistas con índice de impacto elegidas.

Términos clave en *abstract, title, keywords*	*International Journal of Medical Informatics*	*IEEE Transactions on Information Technology in Biomedicine*	*Journal of the American Medical Informatics Association*	*Methods of Information in Medicine*	*Telemedicine and e-health Journal*	*Journal of Biomedical Informatics*	*Medical Informatics and the Internet in Medicine*	*Journal of Telemedicine and Telecare*	*Journal of Medical Systems*	*Journal of Medical Internet Research*
Telemedicine	62	42	19	20	11	22	13	12	10	7
Teleophthalmology	0	0	0	0	6	0	1	2	0	0
EHR	54	51	61	17	2	11	6	1	3	2
EPR	163	20	66	20	20	3	8	11	10	1
HL7	4	55	12	9	0	2	3	0	0	0
DICOM	1	109	6	5	3	1	4	5	0	0
ISO/TC 215	0	0	0	0	0	0	0	0	0	0
XML	0	3	0	3	0	0	4	0	0	1
OpenEHR	4	2	0	2	0	2	0	0	0	0
GEHR	3	1	0	1	0	0	1	0	0	0
CORBA	9	1	2	2	1	0	1	0	0	0
IHE	0	5	0	0	0	1	1	0	0	0
CEN/TC 251	1	2	2	0	0	0	0	0	0	0
Total	301	291	168	79	43	42	42	31	23	11

Tabla 66. Artículos analizados en revistas de impacto de telemedicina. Fuente: propia.

Se han seleccionado artículos desde el primer número de la revista hasta la actualidad (IEEE-ITB e *International Journal of Medical Informatics* desde 1997, JAMIA desde 1994, *Methods of Information in Medicine, Telemedicine and e-health Journal, Medical Informatics and the Internet in Medicine, Journal of Telemedicine and Telecare, Journal of Medical Systems, Journal of Medical Internet Research* desde 1999 y *Journal of Biomedical Informatics* desde 2001). Se puede observar fijándonos en las tablas anteriores que el número de artículos concernientes a estándares e iniciativas sobre HCEs es muy reducido. DICOM y HL7 son los dos estándares más difundidos y de los que existen un mayor número de publicaciones científicas.

9.3. Líneas futuras de investigación

A continuación, se citan las posibles líneas futuras de investigación que surgen de la presente Tesis Doctoral:

- Desarrollo de la aplicación con **otras BDs** tanto de *software* libre como comerciales.
- Diseño y desarrollo de **nuevas técnicas de *benchmarking*** para el cálculo de tiempos de respuesta de sistemas Web.
- Incorporación de **TeleOftalWeb** como parte de un sistema de HCE, por ejemplo, dentro de un sistema global de Teleoftalmología.
- Integración con sistemas de procesado de imágenes, con el fin de detectar enfermedades como la RD, cataratas, glaucoma, etc.
- Empleo de la plataforma por parte de pacientes reales.
- Posibilidad de desarrollar el sistema en otras plataformas: Televisión Digital (TD), móvil, etc.
- Verificación de otros estándares dentro de **TeleOftalWeb**.
- *Benchmarking* más detallado a nivel funcional de las aplicaciones de HCEs enunciadas en este Trabajo y de otras.
- Desarrollo de un **plan de *marketing* y negocio** con el fin poder llevar a cabo transferencia tecnológica de la Plataforma Web.

Apéndice A

Publicaciones a las que ha dado lugar la realización de la Tesis Doctoral

A.1. REVISTAS INTERNACIONALES INDEXADAS EN EL JOURNAL CITATION REPORT **¡ERROR! MARCADOR NO DEFINIDO.**
A.2. REVISTAS NACIONALES **¡ERROR! MARCADOR NO DEFINIDO.**
A.3. CAPÍTULOS DE LIBRO **¡ERROR! MARCADOR NO DEFINIDO.**
A.4. CONGRESOS INTERNACIONALES **¡ERROR! MARCADOR NO DEFINIDO.**
A.5. CONGRESOS NACIONALES **¡ERROR! MARCADOR NO DEFINIDO.**

A.1. Revistas internacionales indexadas en el Journal Citation Report

Isabel de la Torre, Francisco Javier Díaz, Miriam Antón, José Fernando Díez, Beatriz Sainz, Miguel López, Roberto Hornero and María Isabel López, "Choosing the most efficient database for a Web-based system to store and exchange Ophthalmologic Health Records", **Journal of Medical Systems (Factor de Impacto JCR: 0.674 (2008)), 2010, Aceptado.**

Isabel de la Torre, Roberto Hornero, Miguel López, María Isabel López, "Ophthalmologic Electronic Health Records using HL7 and DICOM standards", *Thirteenth Annual International Meeting & Trade Show (ATA 2008*), **Telemedicine and e-Health Journal, (Factor de Impacto JCR: 1.389 (2008)),** pp. 100, 2008.

Isabel de la Torre, Francisco Javier Díaz, Miriam Antón, Miguel López, Roberto Hornero, Beatriz Sainz and María Isabel López, "Performance evaluation of a Web-based system to exchange Electronic Health Records", **Journal of Web Engineering (Impact Factor: 0.412 (2008)),** En revisión.

Isabel de la Torre, Miriam Antón, Francisco Javier Díaz, Miguel López, Roberto Hornero, María Isabel López and Beatriz Sainz, "Response time estimation of a Web-based Electronic Health Record (EHR) system using the M/M/1 queuing model", **Journal of Systems and Software (Impact Factor: 1.241 (2008)),** En revisión

A.2. Revistas nacionales

Isabel de la Torre, Beatriz Sainz, Francisco J. Díaz, Miriam Antón, Miguel López, Guillermo Gónzalez, Fernando Rabanal, "Bases de datos nativas XML en la Telemedicina: comparación y evolución del tratamiento de la información en contextos clínicos", Revista Informática y Salud, ISSN: 1579-8070, nº. 76, pp. 96-106, Septiembre 2009.

Isabel de la Torre, Beatriz Sainz, Francisco Javier Díaz, María Isabel López, Miguel López, Roberto Hornero, Miriam Antón, "Desarrollo y evaluación de una aplicación de Historiales Clínicos Electrónicos empleando Software libre", Revista eSalud electrónica, ISSN: 1698-7969, vol. 5, nº. 20, Noviembre 2009. Acceso Web: http://www.revistaesalud.com/index.php/revistaesalud/article/view/355/710

A.3. Capítulos de libro

Isabel de la Torre, Roberto Hornero, Miguel López, Jesús Poza, María Isabel López, "A Web-based Application to exchange Electronic Health Records and Medical Images in Ophthalmology*",* Editors: Athina A. Lazakidou, Ph.D and Konstantinos M. Siassiakos, Ph.D, University of Piraeus, Greece, Libro: "Handbook of Research on Distributed Medical Informatics and E-Health", ISBN: 978-1-60566-002-8, pp. 152-164, Publisher: Medical Information Science Reference, Agosto 2008.

Isabel de la Torre, Roberto Hornero, Miguel López, María Isabel López, "Electronic Health Records System using HL7 and DICOM in Ophthalmology", Editors: Wayne Pease, University of Southern Queensland, Australia, Professor Malcolm Cooper, Ritsumeikan Asia Pacific University, Japan, Assoc. Professor Raj Gururajan, University of Southern Queensland, Australia, Libro: "Biomedical Knowledge Management: Infrastructures and Processes for E-Health Systems", ISBN: 1-60566-266-6, 2007. Aceptado.

Isabel de la Torre, Roberto Hornero, Miguel López, María Isabel López, "Electronic Health Records in a teleophthalmologic application with Oracle 10g*",* Editors: Athina A. Lazakidou, Ph.D and Konstantinos M. Siassiakos, Ph.D, University of Piraeus, Greece, Libro: "Handbook of Research on Biocomputation and Biomedical Informatics", pp. 89-105, ISBN: 978-1-60566-768-3, 2009.

Isabel de la Torre, Roberto Hornero, Miguel López, María Isabel López, "A Web-based Application to Exchange Ophthalmologic Health Records using open-source databases*",* Editors: Athina A. Lazakidou, Ph.D and Konstantinos M. Siassiakos, Ph.D, University of Piraeus, Greece, Libro: "Handbook of Web-based Applications in Health Care & Biomedicine", pp. 59-75, ISBN: 978-1-4419-1273-2, 2010.

A.4. Congresos internacionales

Isabel de la Torre, Roberto Hornero, Miguel López, María Isabel López, "Ophthalmologic Electronic Health Records using HL7/CDA and DICOM: TeleOftalWeb", *International Conference on Health Informatics*, HealthInf 2009, Porto, Portugal, 2009.

Isabel de la Torre, Roberto Hornero, Miguel López, María Isabel López, "HL7-CDA and DICOM standards in Electronic Health Records System in Ophthalmology", *Fourth IASTED International Conference on Telehealth and Assistive Technologies*, pp. 162-167, Publication Code: 619, ACTA Press, Baltimore, USA, 2008.

Isabel de la Torre, Roberto Hornero, Jesús Poza, Miguel López, María Isabel López, "TeleOftalWeb 3.2: A Web-based Application to Exchange Electronic Health Records in Ophthalmology", *The Third IASTED International Conference on Telehealth*, pp. 564-004, ISBN: 978-0-88986-667-6, Montreal, Canadá, 2007.

A.5. Congresos nacionales

Isabel de la Torre, Roberto Hornero, Miguel López, María Isabel López, "Desarrollo e implantación de un Sistema de Historiales Clínicos Electrónicos estandarizados en Oftalmología", CASEIB 2008, *Libro de Actas del XXVI Congreso Anual de la Sociedad Española de Ingeniería Biomédica*, ISBN: 978–84–691–3640–9, Valladolid, España, 2008.

Apéndice B

Manual de usuario

B.1. MODULO DE ADMINISTRACIÓN DE USUARIOS 362
B.2. MÓDULO DE GESTIÓN DE HISTORIALES 375

B.1. Modulo de administración de usuarios

Mediante dicho módulo se pueden realizar las siguientes operaciones:

- Abrir la ficha de un usuario. Permite consultar los datos personales así como inicializar la contraseña de un usuario registrado en el sistema.
- Eliminar un usuario. Permite eliminar definitivamente un usuario del sistema.
- Modificar datos de usuario. Se podrán modificar todos los datos de un usuario a excepción de su DNI.
- Creación de usuarios. Mediante este módulo se podrán registrar nuevos usuarios en el sistema ya sea con perfil de usuario o de administrador.
- Consulta de historiales. Se podrán ver los historiales asociados a un usuario, tanto los creados por él como aquellos en los que tiene permisos de lectura. El historial podrá se eliminado de forma permanente en caso de haber sido creado por error. Por último se podrán ver los participantes asociados a cada uno de estos historiales.
- Consulta de estadísticas. Se abrirá una nueva ventana en la que aparece información sobre los historiales creados por el usuario, cuántos de ellos están compartidos y en cuántos historiales aparece el usuario como compartido, es decir, con permisos de lectura.

Pantalla de entrada

Para entrar en el módulo de administración mediante una conexión segura es necesario introducir la siguiente URL en la barra de direcciones del navegador:

https://localhost/teleoftalWeb/admin (si la aplicación se quiere ejecutar desde local).

https://dir_IP_servidorWeb/teleoftalWeb/admin (si la aplicación se quiere ejecutar de forma remota desde otro ordenador. Aquí dir_IP_servidorWeb es la dirección IP del servidor Web donde está alojada la aplicación o el nombre del servidor).

Figura 158. Pantalla de acceso al módulo de administración. Fuente: propia.

En la figura 158 se puede observar cómo aparece la imagen de un candado tanto en la barra de direcciones como en la parte inferior derecha de la ventana, lo cual es la confirmación de que nos encontramos en una conexión segura y por lo tanto todos los datos transmitidos y recibidos se realizarán por un canal encriptado.

Pantalla inicial

Una vez introducidos un nombre y contraseña de usuario válidos para el módulo de administración, aparecerá la pantalla inicial de la herramienta, desde donde se podrá acceder a todas las funcionalidades de la misma. La pantalla inicial se puede ver en la figura 159.

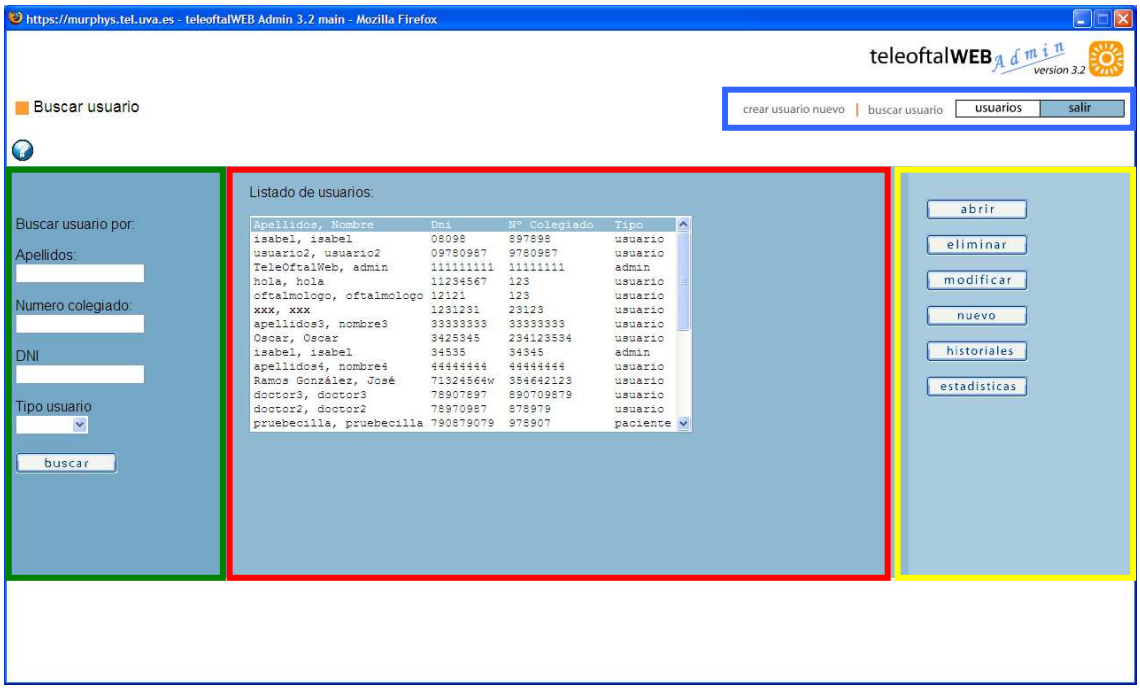


Figura 159. Pantalla inicial de la herramienta de administración. Fuente: propia.

Como se puede observar en la figura 159 la pantalla inicial está dividida en cuatro partes importantes:

- Buscador (verde). Mediante el buscador se puede buscar usuarios atendiendo a las características más importantes, como son apellidos, número de colegiado, DNI o el tipo de usuario. En el apartado siguiente se profundizará más en la metodología de búsqueda.
- Información de usuarios (rojo). En la zona central de la pantalla aparece el listado con todos los usuarios registrados en el sistema. Para cada usuario se proporcionan algunos datos de interés, que coinciden con aquellos parámetros por los que se puede realizar la búsqueda.
- Botones de función (amarillo). En la parte derecha de la pantalla aparece una columna de botones desde los cuales se puede realizar cualquiera de las operaciones posibles sobre los usuarios. Cada una de estas operaciones se describirán con detalle en los siguientes apartados. Según el orden en el que aparecen los botones, las operaciones correspondientes son:
    - Abrir la ficha de un usuario.
    - Eliminar de forma permanente un usuario.
    - Modificar los datos de un usuario.
    - Crear un nuevo usuario.

  - Información sobre los historiales a los que tiene acceso el usuario.
  - Estadísticas del usuario.

- Menú superior (azul). En la parte superior de la pantalla se pueden observar dos botones, "salir" y "usuarios". El primero permite abandonar la aplicación de forma segura, mientras que al pinchar sobre el segundo se despliega un submenú con las funciones más habituales:
  - Buscar usuario. Lleva de nuevo a esta pantalla de la aplicación en caso de encontrarse en alguna de las pantallas de función.
  - Crear usuario nuevo. Mediante esta opción se podrá acceder desde cualquier punto de la aplicación a la pantalla de creación de nuevos usuarios. Tiene el mismo efecto que el botón "nuevo" antes mencionado.

En todas las acciones que se realizan sobre los usuarios, el procedimiento será el mismo, se seleccionará el usuario en el listado central y se pulsará el botón correspondiente a la acción deseada. En caso de tener un listado muy amplio de usuarios se podrá hacer uso del buscador para facilitar la tarea. Para ver el manual de usuario hay que hacer doble clic en el símbolo que aparece en la esquina superior izquierda .

Búsqueda de usuarios

La búsqueda de usuarios se puede realizar en base a cuatro parámetros: apellidos, número de colegiado, DNI y tipo de usuario. Estos criterios de búsqueda pueden utilizarse bien de forma aislada o combinando varios de ellos. Para iniciar la búsqueda es suficiente rellenar el campo deseado y pulsar el botón "buscar". Como puede observarse en las siguientes figuras, el listado de usuarios resultante aparecerá en la parte central de la pantalla. En caso de dejar todos los campos en blanco la búsqueda dará como resultado todos los usuarios del sistema. En las figuras 160 y 161 se puede ver cómo se realiza una búsqueda de usuarios en base los parámetros citados anteriormente.

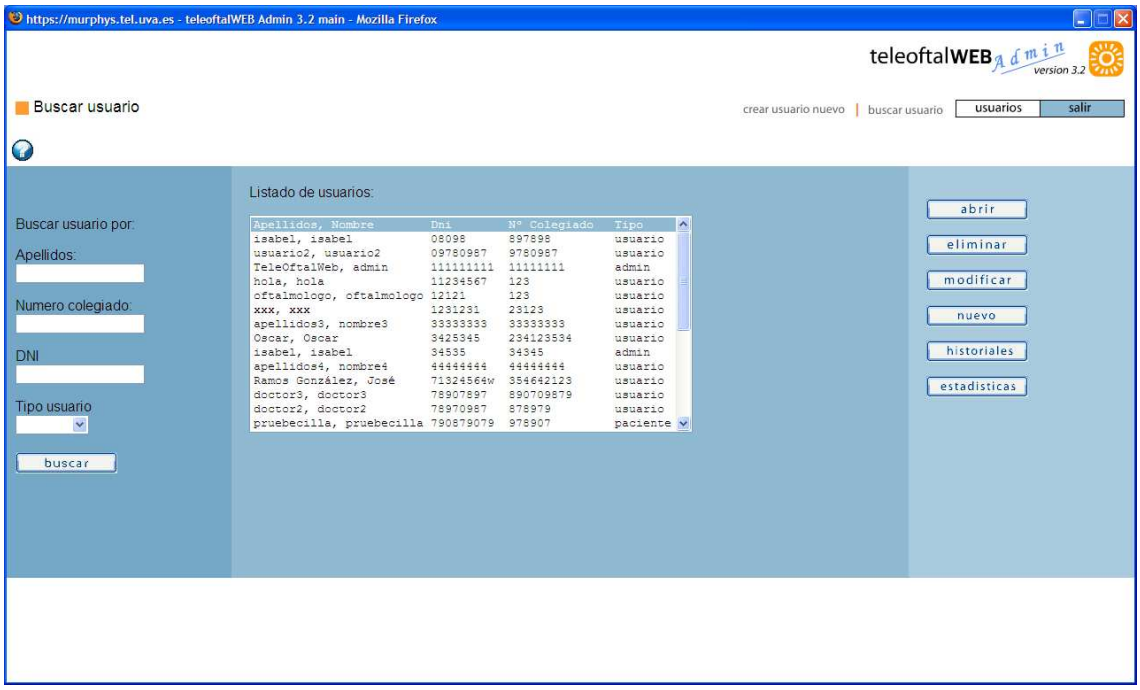


Figura 160. Listado completo de usuarios. Fuente: propia.

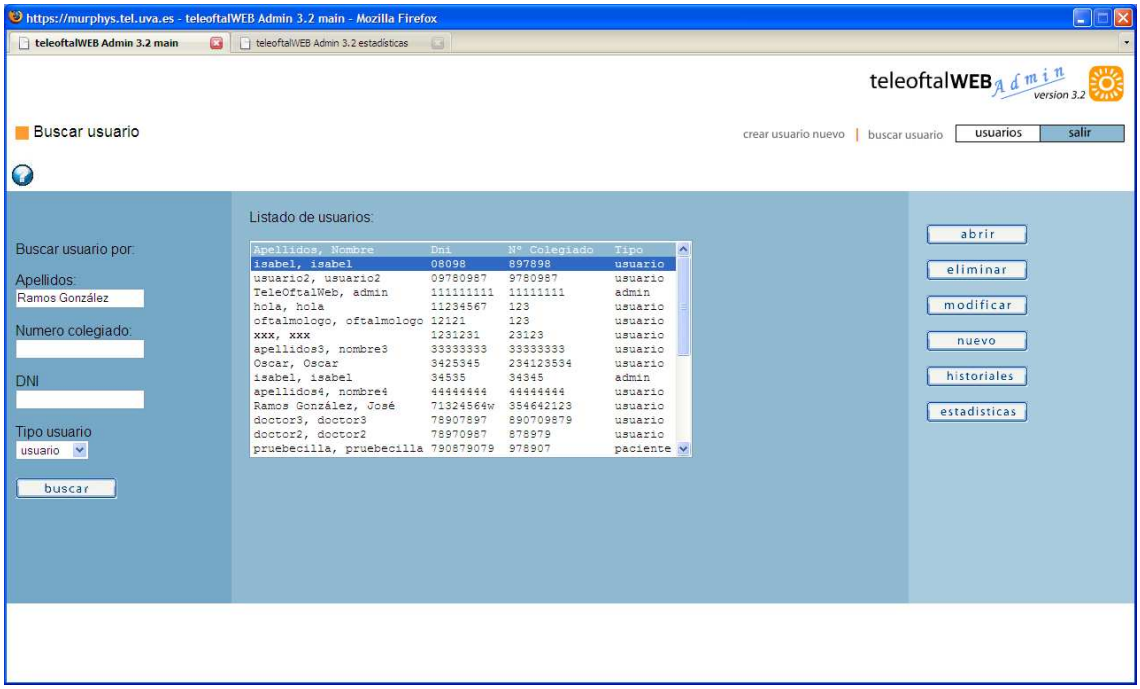


Figura 161. Búsqueda de usuarios por apellidos y tipo de usuario. Fuente: propia.

Abrir ficha de usuario

Se puede acceder a los datos de un usuario de dos formas:

- Haciendo doble *click* sobre el usuario deseado en el listado.
- Seleccionando al usuario en el listado y pulsando el botón “abrir”.

Como puede observarse en la figura 162 los datos de usuario se abrirán en modo de lectura, por lo que no podrán ser modificados. Para poder modificar dichos datos es necesario acceder a los mismos mediante el botón “modificar”

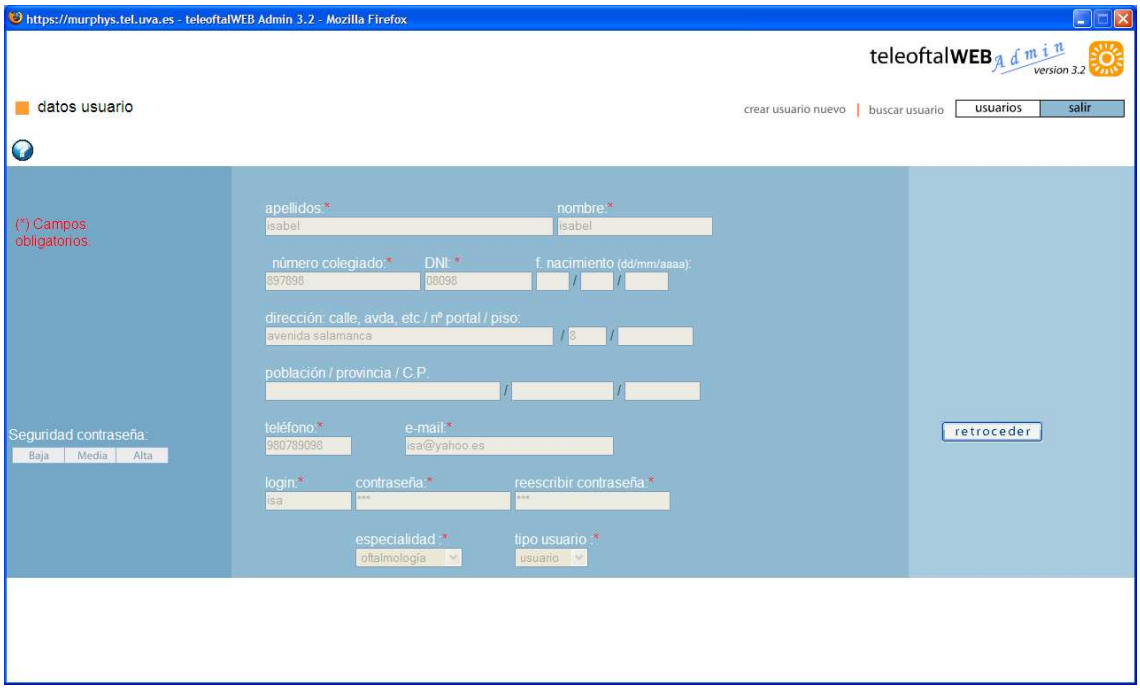


Figura 162. Abrir datos de usuario. Fuente: propia.

Eliminar un usuario

Se podrá eliminar un usuario de forma definitiva del sistema sólo con seleccionarle y pulsando en el botón “eliminar”. Una vez realizado esto se abrirá una ventana como la de la figura 163 en la que se pide la confirmación para eliminar al usuario. Si se pulsa la opción “Aceptar” el usuario será eliminado del sistema.

Figura 163. Confirmación de eliminación de usuario. Fuente: propia.

Modificar datos de usuario

Los datos de un usuario pueden ser modificados mediante la opción “modificar”. Para ello es suficiente seleccionar el usuario deseado y pulsar dicho botón. Una vez seleccionada la acción de modificar, aparecerá una pantalla como la siguiente (véase figura 164):

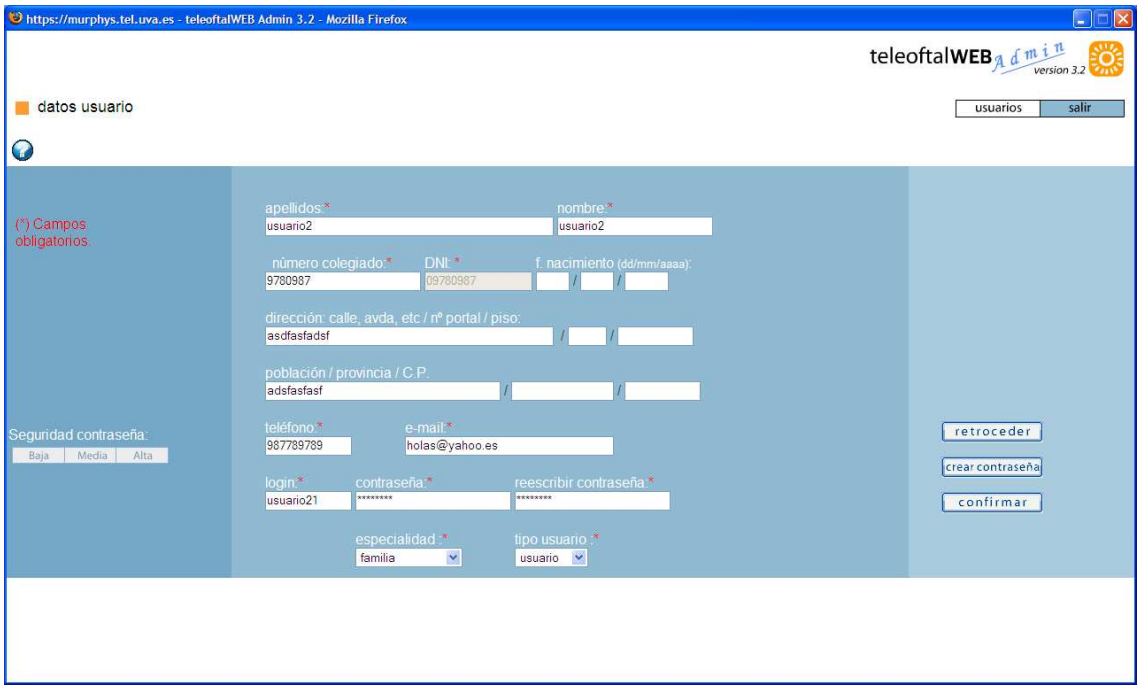


Figura 164. Modificación de datos de usuario. Fuente: propia.

En esta pantalla se podrá modificar cualquiera de los datos de usuario excepto el DNI. Una vez modificados los datos, para guardar los cambios habrá que pulsar el botón “confirmar” o bien “retroceder” para descartarlos.

Crear un nuevo usuario

Para crear un nuevo usuario es suficiente con pinchar en el botón "nuevo" de la pantalla inicial, o bien mediante la opción correspondiente del menú superior. Una vez pulsada la opción deseada, aparecerá una ventana como la de la figura 165:

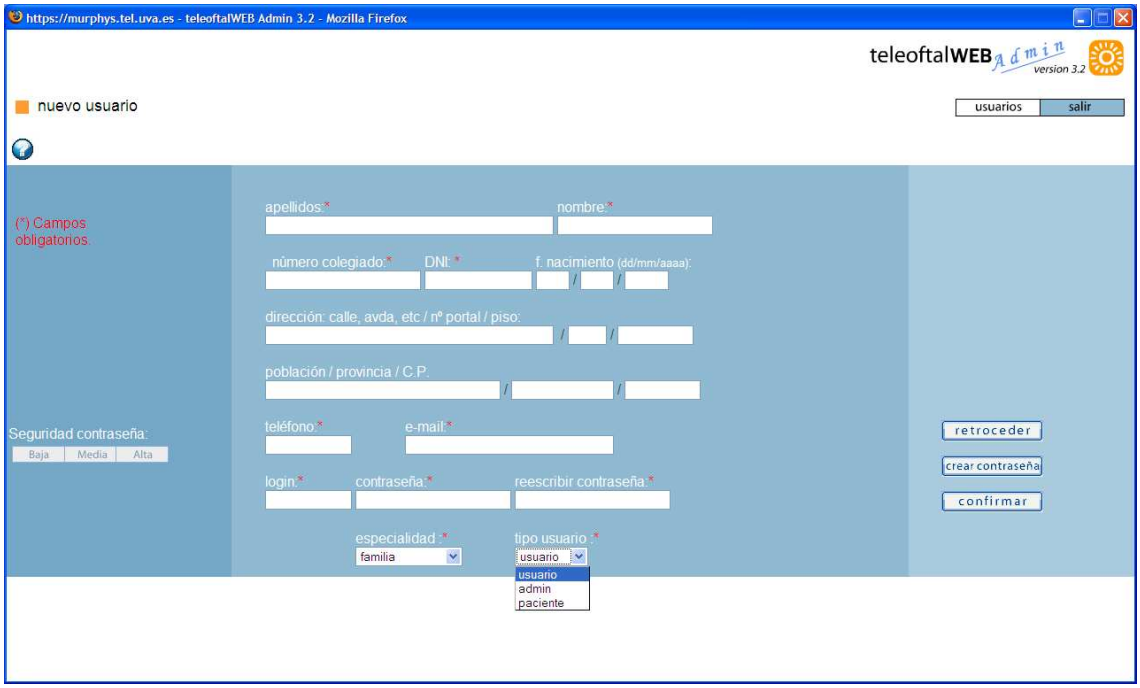


Figura 165. Ventana de creación de usuarios. Fuente: propia.

En esta pantalla se irán introduciendo los datos personales del usuario, teniendo en cuenta que los marcados con un asterisco rojo son de carácter obligatorio. Los datos que se deben introducir son los siguientes:

- Apellidos.
- Nombre.
- Número de colegiado. Debe ser numérico y con un máximo de 9 dígitos.
- DNI. Debe tener un máximo de 9 caracteres.
- Fecha de nacimiento.
- Datos del domicilio.
- Teléfono.
- Correo electrónico.
- *Login*.
- Contraseña.

- Especialidad. Pudiendo elegir entre: familia, oftalmología, endocrinología, otras.
- Tipo de usuario. Actualmente sólo están dos perfiles de uso para la aplicación, los administradores con permisos para la gestión de usuarios, y los usuarios propiamente dichos que serán los que introducirán nuevos historiales en la herramienta.

Como se puede observar en la figura 165, en la parte inferior izquierda de la ventana aparece un pequeño indicador que señala el grado de seguridad de la contraseña introducida, en función de aspectos como su longitud, caracteres, etc. Por otro lado el administrador tiene la opción de generar la contraseña de forma automática mediante el botón "crear contraseña", para ello es necesario haber introducido anteriormente los apellidos, nombre y DNI del usuario. Una vez generada la contraseña y confirmada la creación del usuario el sistema mostrará por pantalla al administrador dicha contraseña. Una vez introducidos los datos del usuario, para guardar dichos datos es necesario pulsar el botón "confirmar". Se puede cancelar en cualquier momento la creación del usuario pulsando en el botón "retroceder"

Consultar historiales de un usuario

Mediante el botón "historiales" se accede a un listado de todos los historiales visibles para un determinado usuario. En el listado se proporciona información relativa al historial, como el nombre del paciente, DNI del mismo, número de la seguridad social, fecha de creación del historial y carácter del mismo, es decir si ha sido creado por el usuario o por el contrario es un historial compartido.

Al igual que ocurre en el listado de usuarios, a la izquierda de la pantalla (figura 166) hay un buscador que permite realizar búsquedas de historiales de acuerdo a diversos parámetros.

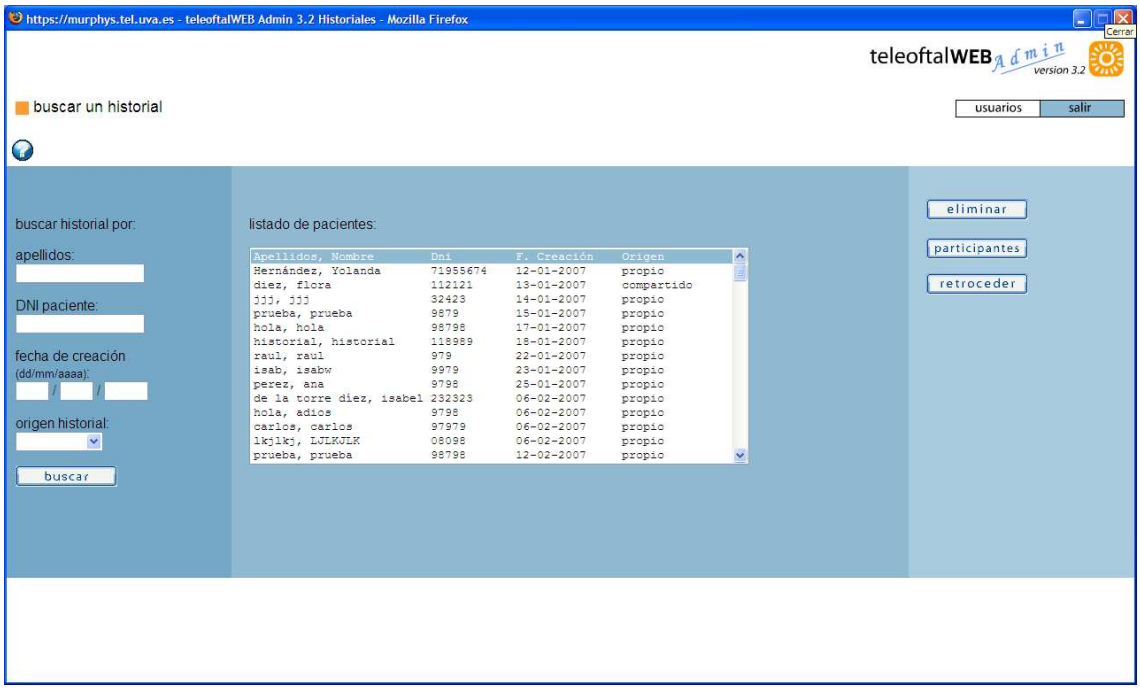


Figura 166. Listado de historiales de un usuario. Fuente: propia.

En esta pantalla se puede realizar varias operaciones sobre un historial seleccionado:

- Eliminar. Mediante el botón “eliminar” se puede borrar un historial de forma definitiva de la BD en caso de que se desee. Al pulsar dicho botón se abrirá una ventana como la de la figura 167, en la que se pide confirmación para realizar la operación.

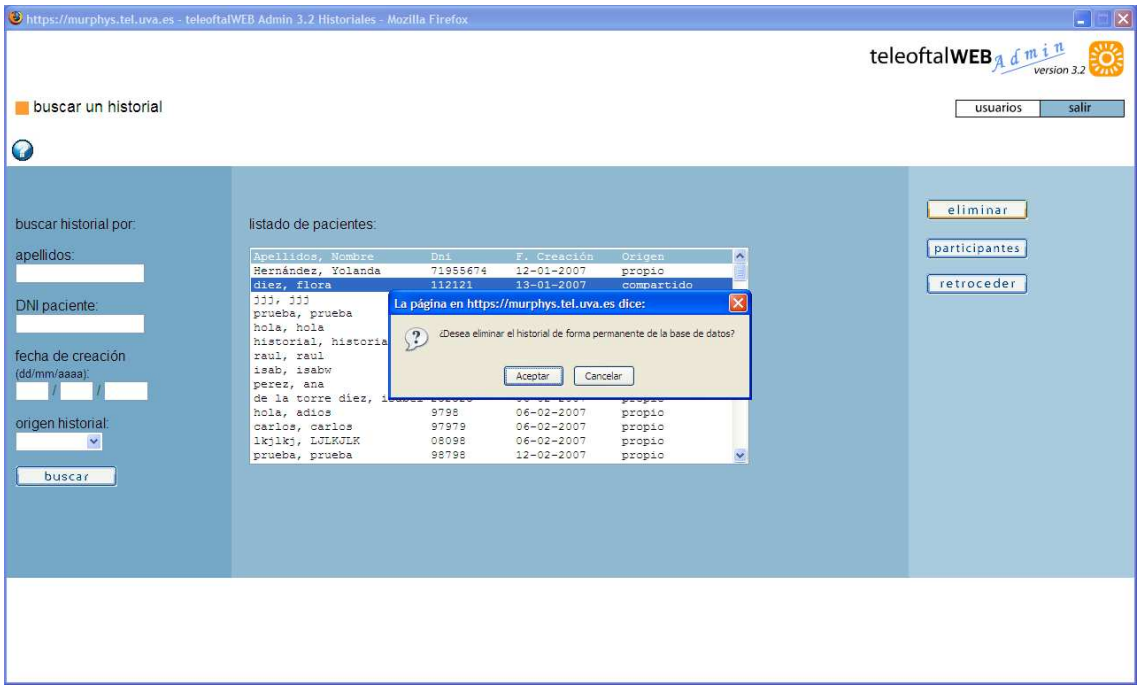


Figura 167. Ventana de confirmación para eliminar un historial. Fuente: propia.

- Consultar participantes. El botón "participantes" permite acceder a un listado con todos los participantes de un determinado historial (figura 168). En esta ventana al igual que ocurre en listados similares dentro de la herramienta se tiene la posibilidad de realizar una búsqueda de participantes según su apellido. Por último en esta pantalla se podrá eliminar un participante de un historial mediante el botón "eliminar", siempre y cuando dicho participante no sea el propietario del mismo, en cuyo caso se mostrará un mensaje de error.

Figura 168. Listado de participantes de un historial. Fuente: propia.

Consultar estadísticas de un usuario

Mediante el botón "estadísticas" (véase figura 169) se abrirá una nueva ventana presentando información sobre los historiales creados por el usuario, cuántos de ellos están compartidos y en cuántos historiales aparece el usuario como compartido, es decir con permisos de lectura.

Figura 169. Estadísticas de usuario. Fuente: propia.

B.2. Módulo de gestión de historiales

En este apartado se va a describir el funcionamiento del módulo de gestión de historiales de la herramienta **TeleOftalWeb**, de manera que un usuario que previamente no esté familiarizado con la misma pueda llegar a utilizarla sin problemas.

Este módulo permite entre otras posibilidades crear de historiales, añadir nuevas revisiones a un historial, consultar revisiones, gestionar y asociar imágenes a un historial, etc. A lo largo de este apartado se irá analizando cada una de estas acciones.

Pantalla de entrada

Para entrar en el módulo de gestión de historiales mediante una conexión segura es necesario introducir la siguiente URL en la barra de direcciones del navegador:

https://localhost/teleoftalWeb (si la aplicación se quiere ejecutar desde local).

https://dir_IP_servidorWeb/teleoftalWeb (si la aplicación se quiere ejecutar de forma remota desde otro ordenador. Aquí *dir_IP_servidorWeb* es la dirección IP del servidor Web donde está alojada la aplicación o el nombre del servidor).

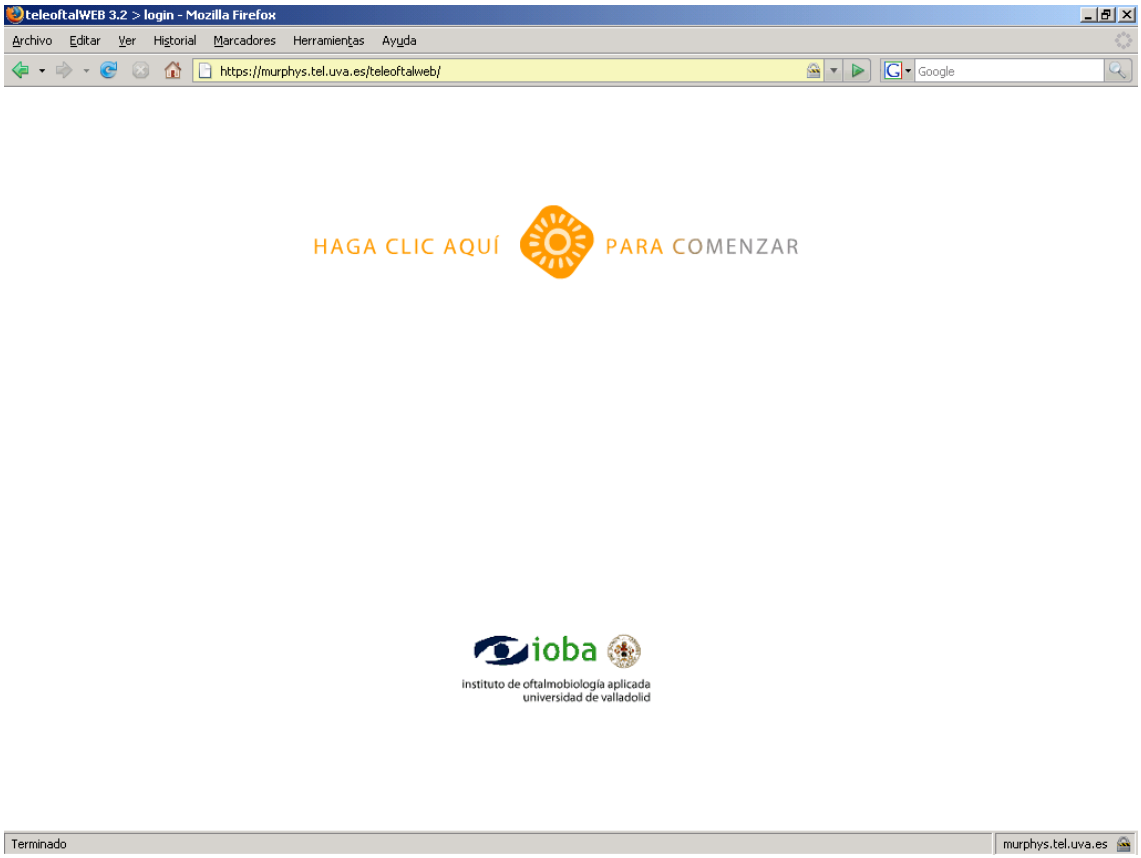


Figura 170. Ventana de acceso a la herramienta TeleOftalWeb. Fuente: propia.

Figura 171. Ventana de autentificación. Fuente: propia.

Para poder acceder a esta parte de la herramienta (figura 170: ventana de acceso y figura 171: ventana de autentificación) es necesario tener un usuario con permisos de usuario.

Pantalla inicial

Una vez introducidos un nombre y contraseña de usuario válidos aparecerá la pantalla inicial de la herramienta, donde se muestran información sobre estadísticas de los historiales, en lo que se refiere al número de historiales que se comparten, en relación a los totales. En este punto sería posible modificar los datos del usuario, para lo cual se pincharía el botón "modificar", o sería posible consultar o añadir historiales a la BD, pinchando en "historiales", así como añadir o consultar las imágenes disponibles, pinchando en "imágenes". La pantalla inicial de puede observar en la figura 172.

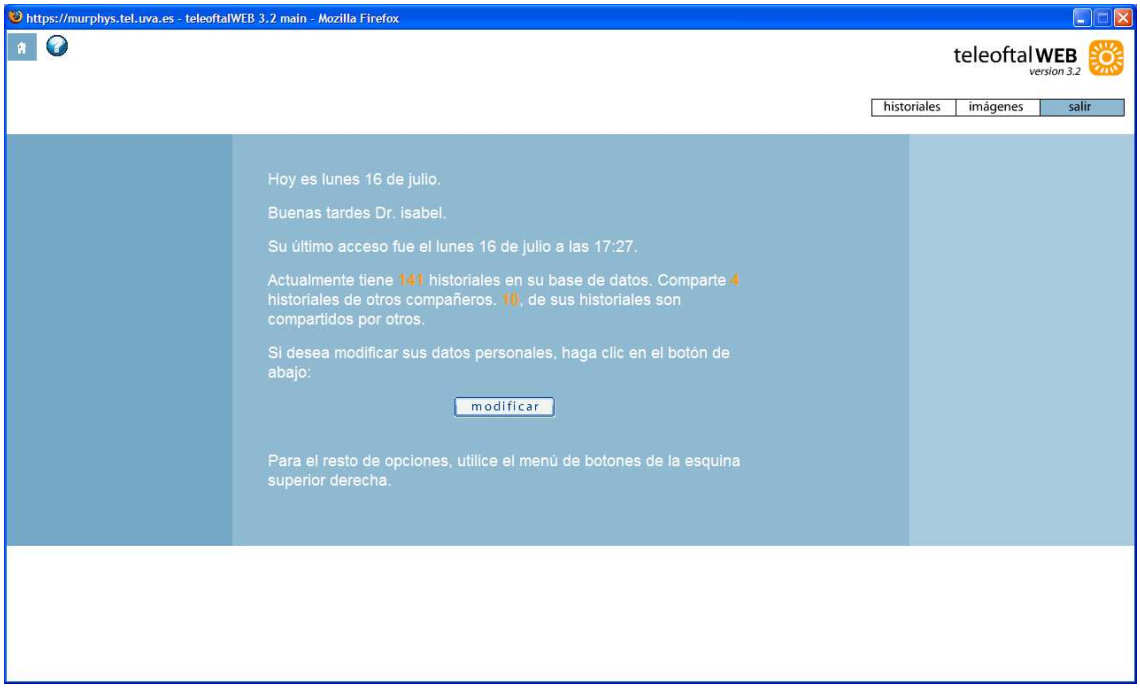


Figura 172. Ventana inicial de presentación de estadísticas. Fuente: propia.

En caso de que se pinche en "modificar" aparecería la figura 173, en la que se mostrarían los datos del usuario, siendo posible realizar cambios y salvarlos, "confirmar", o descartarlos y volver a la pantalla de inicio, "retroceder".

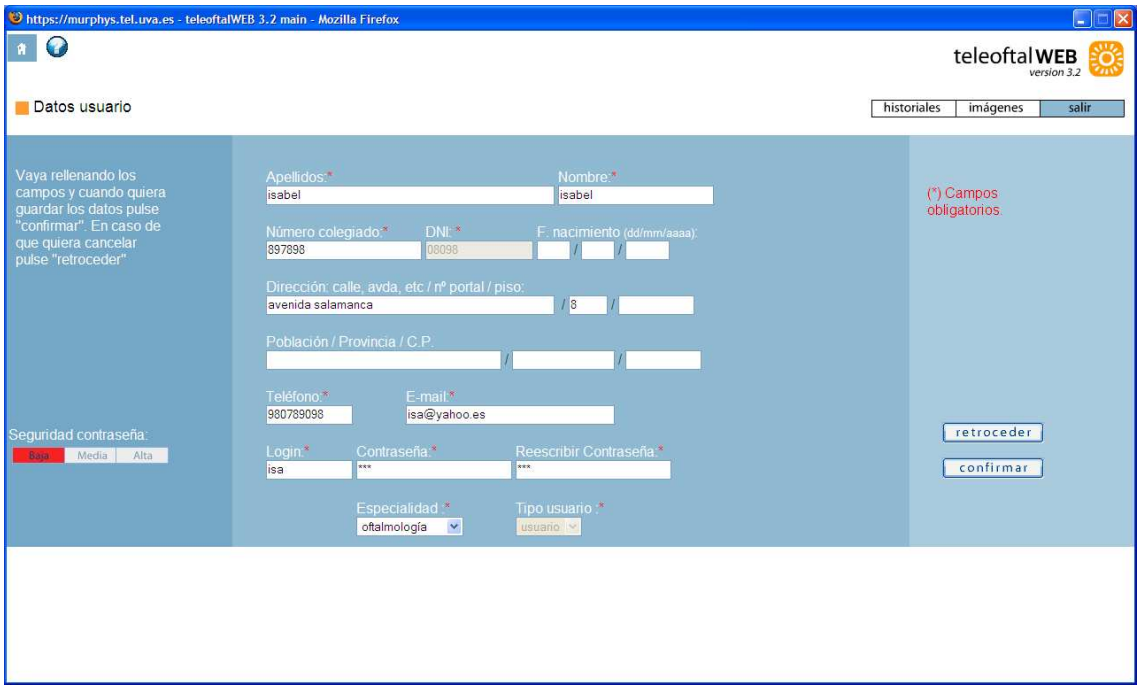


Figura 173. Ventana con los datos de usuario. Fuente: propia.

Partiendo de la pantalla de inicio, si lo que se quiere es añadir un historial nuevo pinchamos en el botón "historiales" y se desplegará un submenú a la izquierda, desde el que se podrá crear un historial nuevo o buscar un historial.

Creación de historiales

Para crear un historial nuevo, se pinchará en el botón "historiales" y posteriormente en crear historial nuevo. A continuación, se deben ir rellenando los sucesivos campos que irán apareciendo, pudiéndose desplazar adelante y atrás mediante los botones "avanzar" y "retroceder" que aparecen en la parte inferior derecha. De esta manera se irá pasando por los bloques: datos de filiación (figura 174), motivo de la consulta (figura 175), antecedentes personales oftalmológicos (figura 176), antecedentes personales generales (figura 177), y antecedentes familiares (Figuras 178-181). Al pulsar en esta última pantalla en el botón "confirmar", si los datos introducidos son correctos, se actualizará la BD. En cuanto a los campos que van apareciendo, estos son cajas de texto, botones, o cajas de selección, y listas desplegables, que intentan adecuarse al tipo de historiales usados en el IOBA en la práctica clínica diaria.

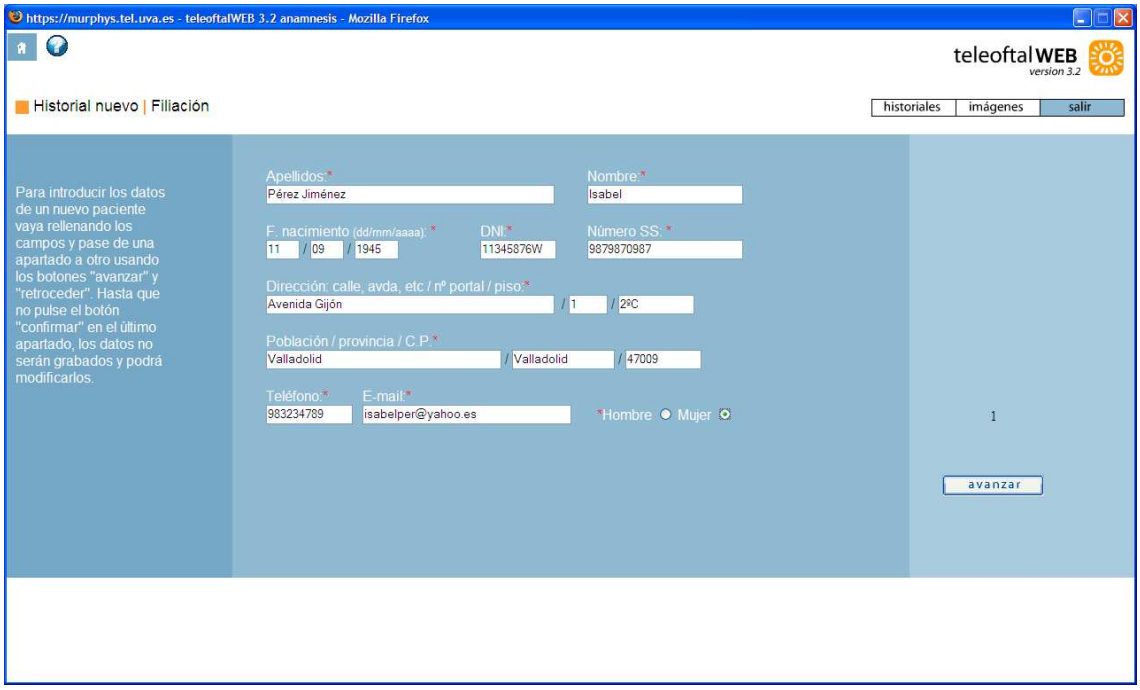


Figura 174. Ventana con los datos de filiación de la revisión de un paciente. Fuente: propia.

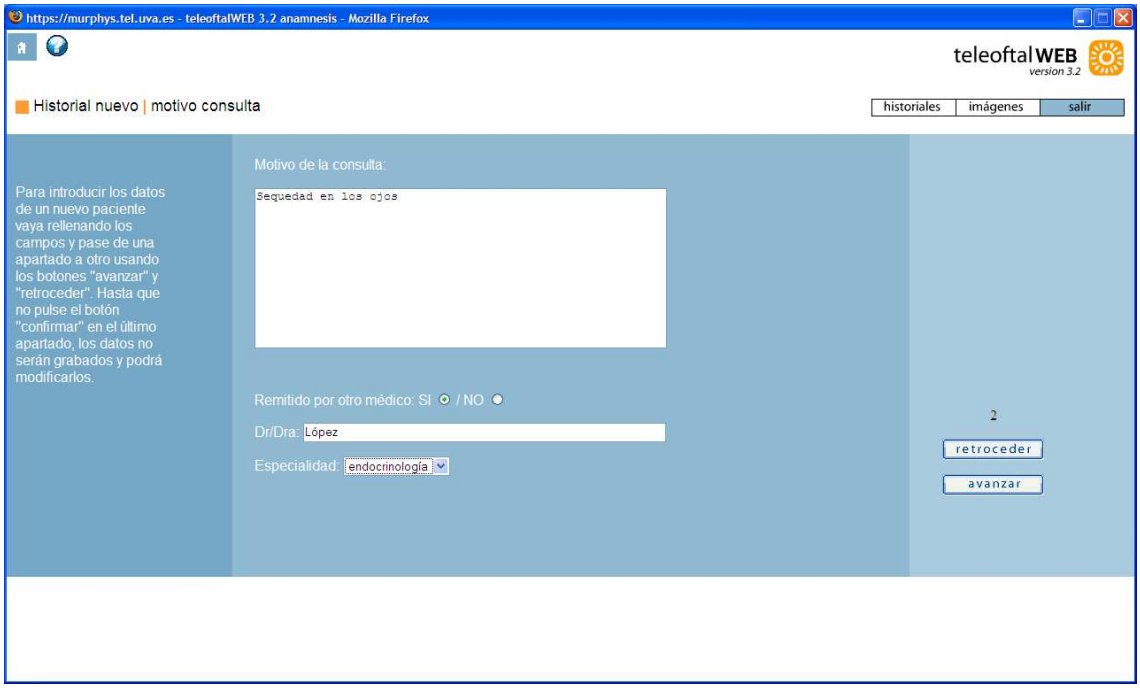


Figura 175. Ventana con el motivo de consulta de una revisión de un paciente. Fuente: propia.

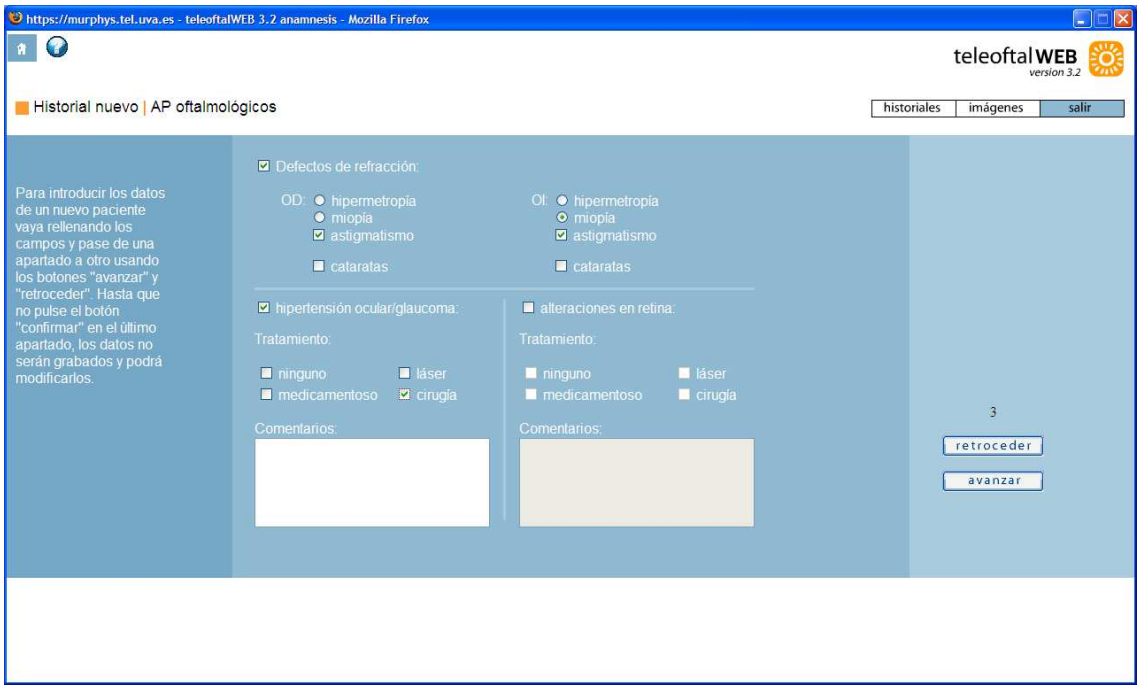


Figura 176. Ventana con los antecedentes personales oftalmológicos de la revisión de un paciente. Fuente: propia.

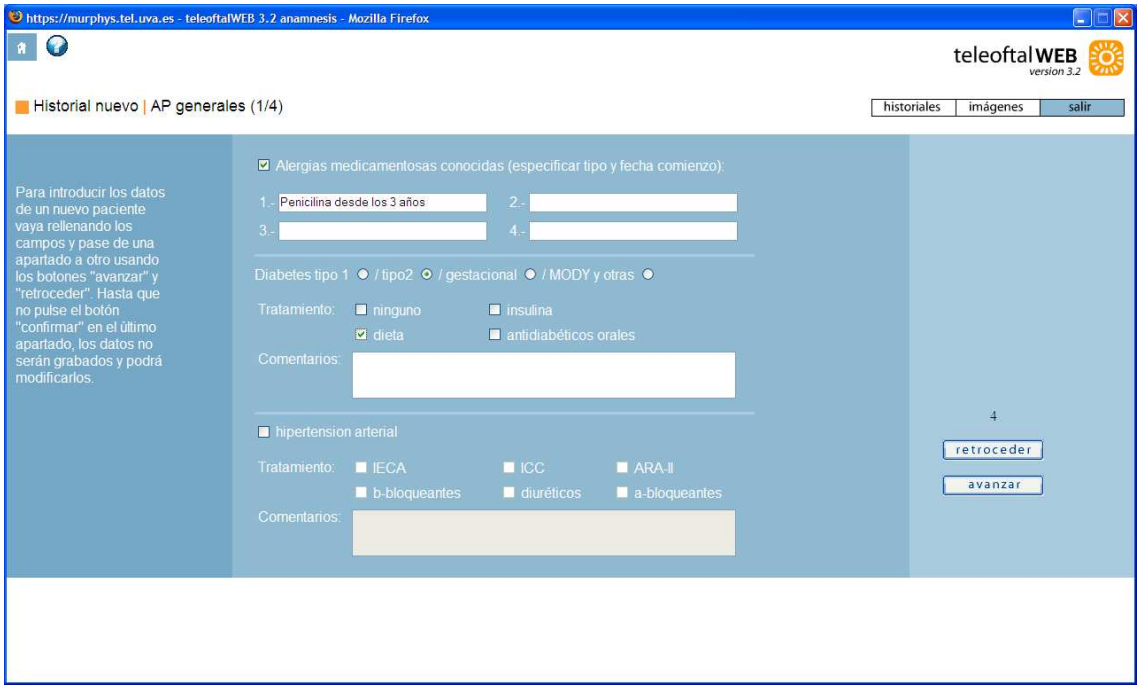


Figura 177. Ventana con la primera parte de los datos de los antecedentes personales generales de la revisión de un paciente. Fuente: propia.

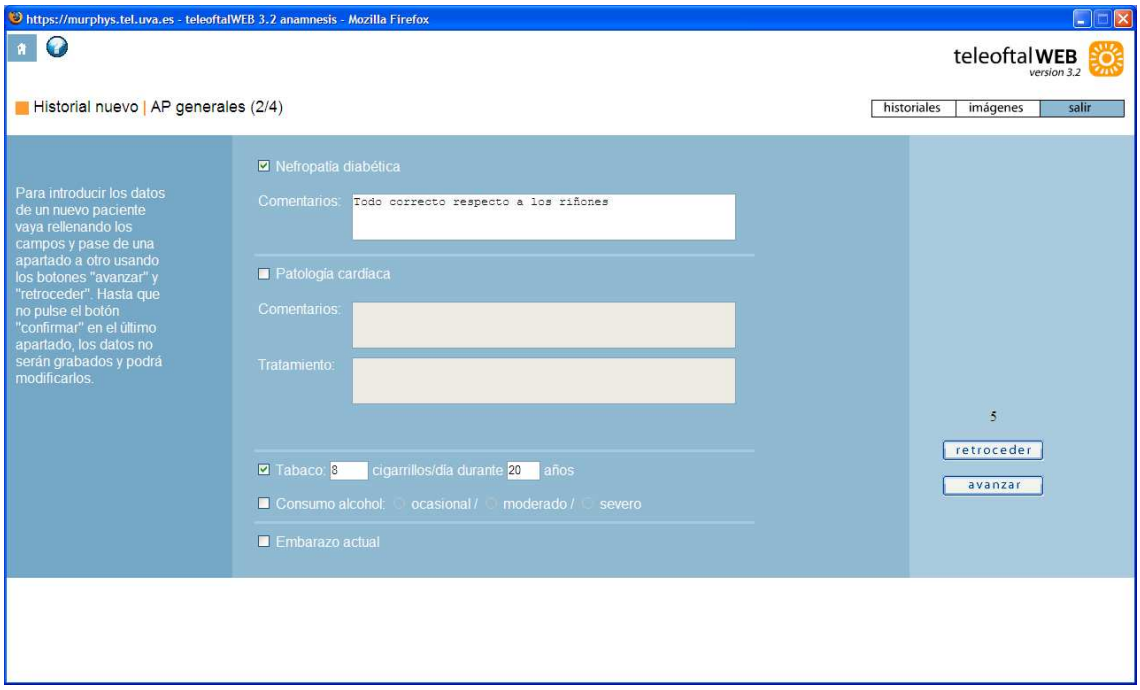


Figura 178. Ventana con la segunda parte de los antecedentes personales de la revisión de un paciente. Fuente: propia.

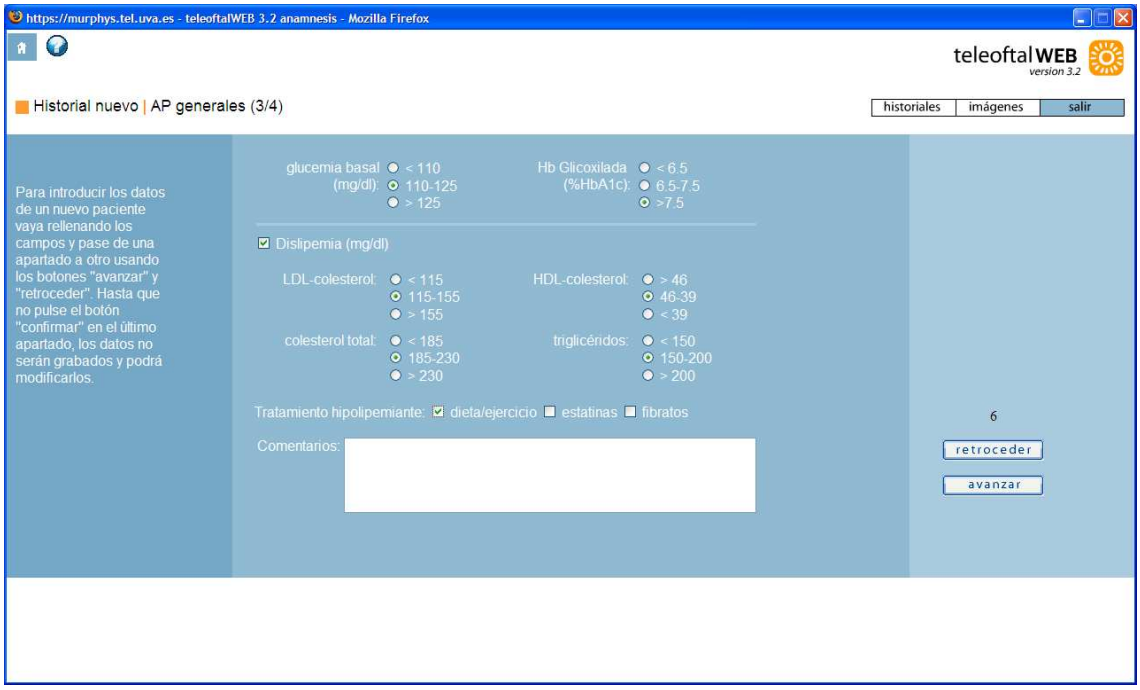


Figura 179. Ventana con la tercera parte de los antecedentes personales generales de la revisión de un paciente. Fuente: propia.

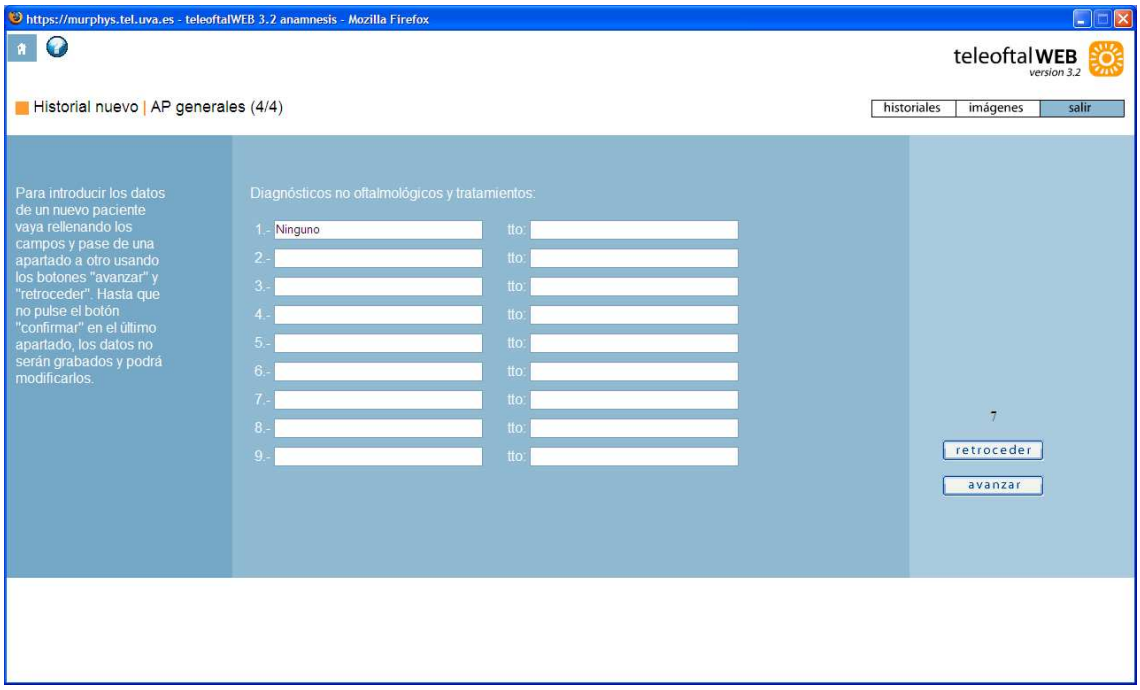


Figura 180. Ventana con la tercera parte de los antecedentes personales generales de la revisión de un paciente. Fuente: propia.

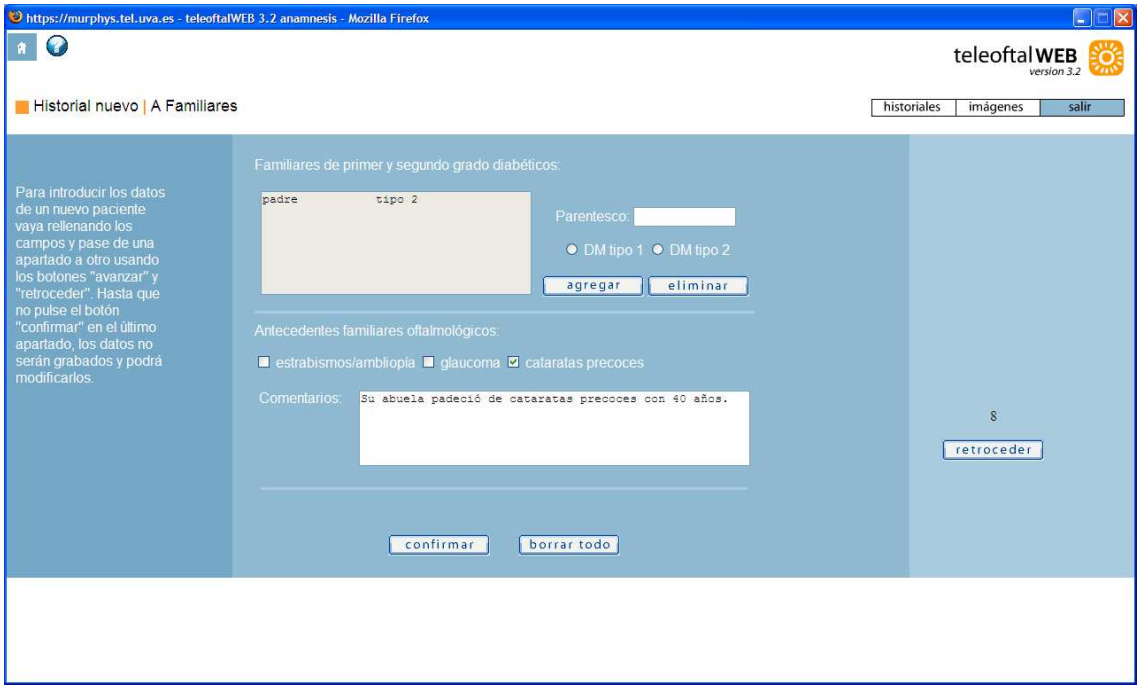


Figura 181. Ventana con los antecedentes familiares de la revisión de un paciente. Fuente: propia.

Una vez confirmada la correcta escritura en la BD, se mostraría la siguiente pantalla (figura 182), en la que se invita a pasar al bloque de exploración del paciente, pulsando "avanzar".

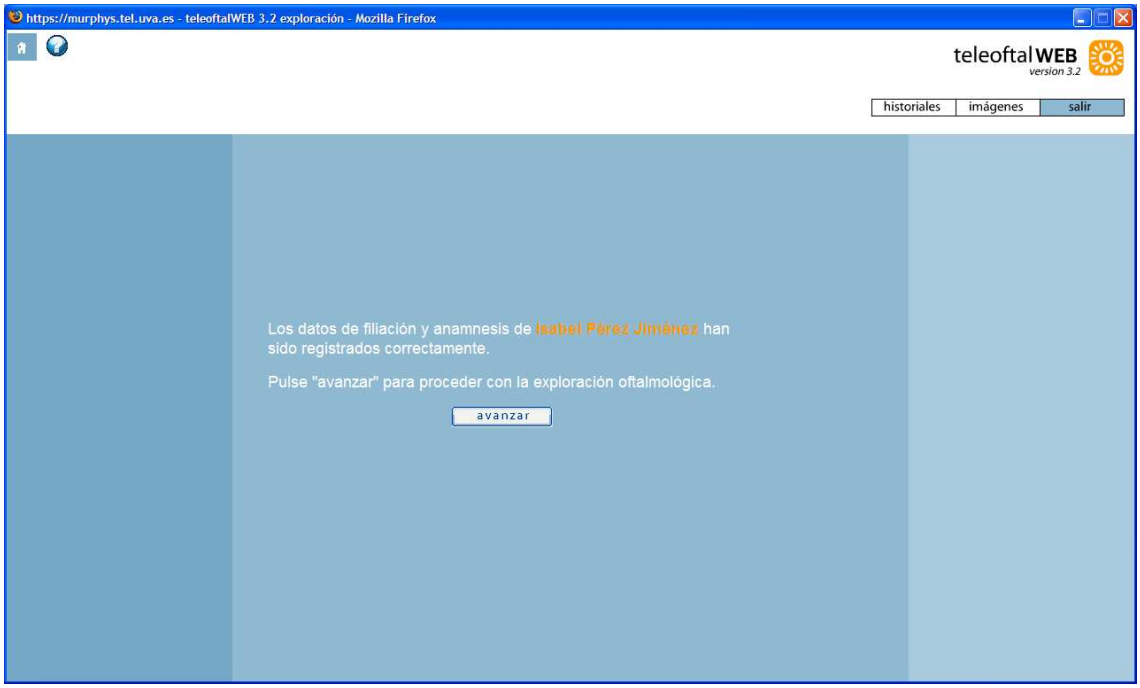


Figura 182. Ventana de indicación de que los datos de filiación y la anamnesis han sido grabados en la base de datos de forma correcta. Fuente: propia.

En este bloque se tienen tres pantallas (figura 183, figura 184, figura 185) en las que se va mostrando campos de texto, listas desplegables, y una serie de esquemas de Biomicroscopía de Polo Anterior (BPA) y de fondo de ojo (figura 185), sobre los cuales se puede dibujar pinchando sobre el dibujo de los mismos, ya que en ese caso se maximizan y aparece un menú de dibujo, que consta de una herramienta para escribir texto libre, de un lápiz de dibujo libre, de una goma, una herramienta para situar flechas indicativas encima del dibujo y una paleta de colores que permite cambiar el color de edición.

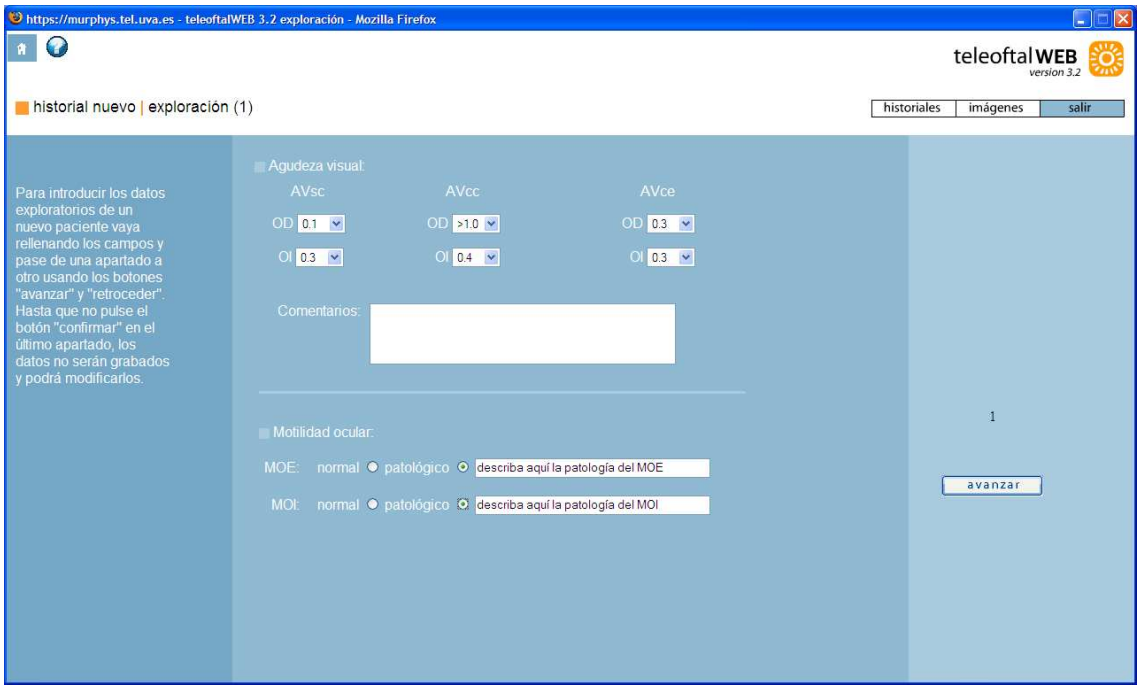


Figura 183. Ventana con la primera parte de la exploración de una revisión del paciente. Fuente: propia.

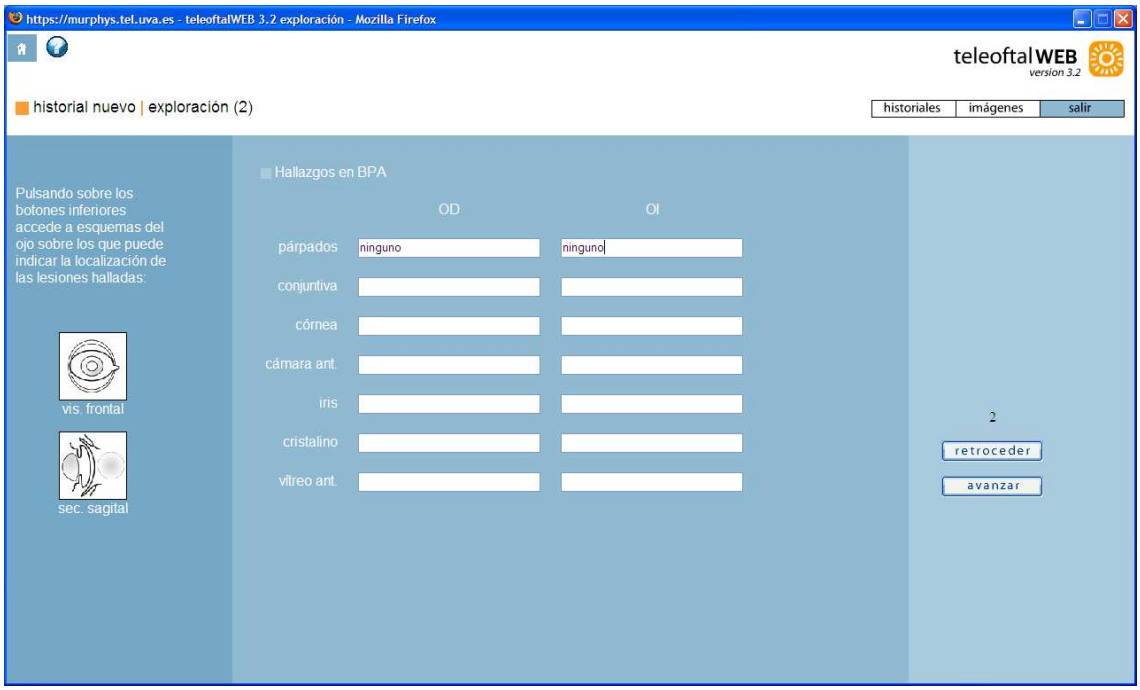


Figura 184. Ventana con la segunda parte de la exploración. Fuente: propia.

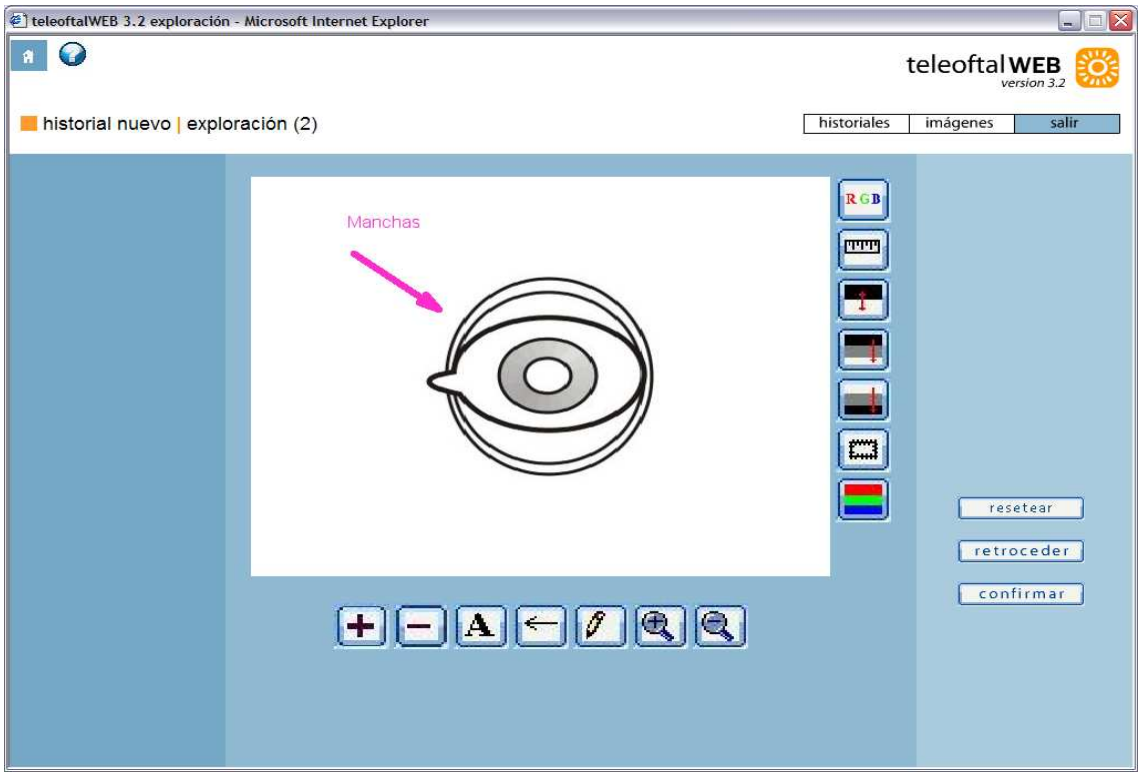


Figura 185. Ventana de edición de los esquemas de ojo de una revisión del paciente. Fuente: propia.

Figura 186. Ventana con la tercera parte de la exploración en la revisión de un paciente. Fuente: propia.

Al igual que en el bloque de filiación y anamnesis, al llegar a la última pantalla del bloque aparece una pantalla en la que se puede confirmar los datos introducidos (pulsando "confirmar") o borrarlos (pulsando "borrar todo"). En caso de que se confirmen y estos sean correctos se actualiza la BD con los nuevos valores, y se pasa al último bloque en el que se recogen otras pruebas complementarias, así como el juicio diagnóstico y el tratamiento sugerido. El inicio del siguiente bloque comienza con un mensaje indicativo, como el de la figura 187, así si se pulsa el botón "avanzar" aparecen las tres últimas pantallas en las que se muestran una serie de pruebas complementarias (figura 188), así como el diagnóstico o diagnósticos realizados (figura 185 y figura 186), y los tratamientos propuestos. Al igual que antes, estas pantallas están constituidas por campos de texto y botones de selección. Es de notar que en la última pantalla, en la que aparece el botón "confirmar", también se muestra unos campos en los que el usuario puede editar la fecha para la siguiente revisión.

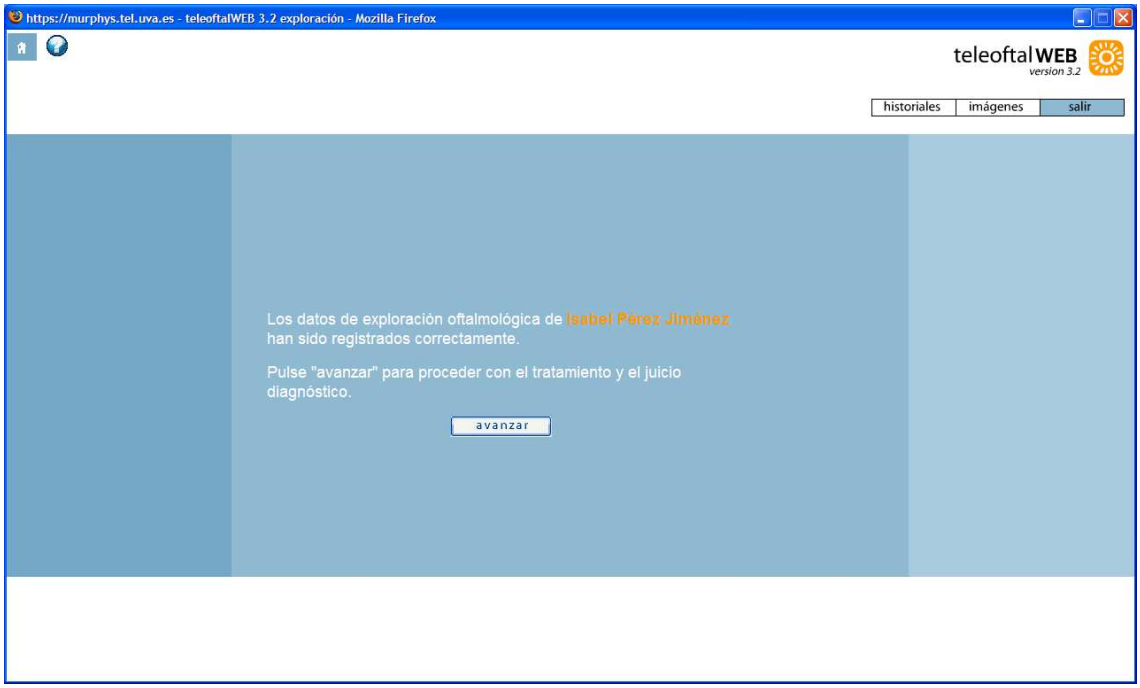


Figura 187. Mensaje de indicación acerca de que los datos de exploración han sido grabados en la base de datos de forma correcta. Fuente: propia.

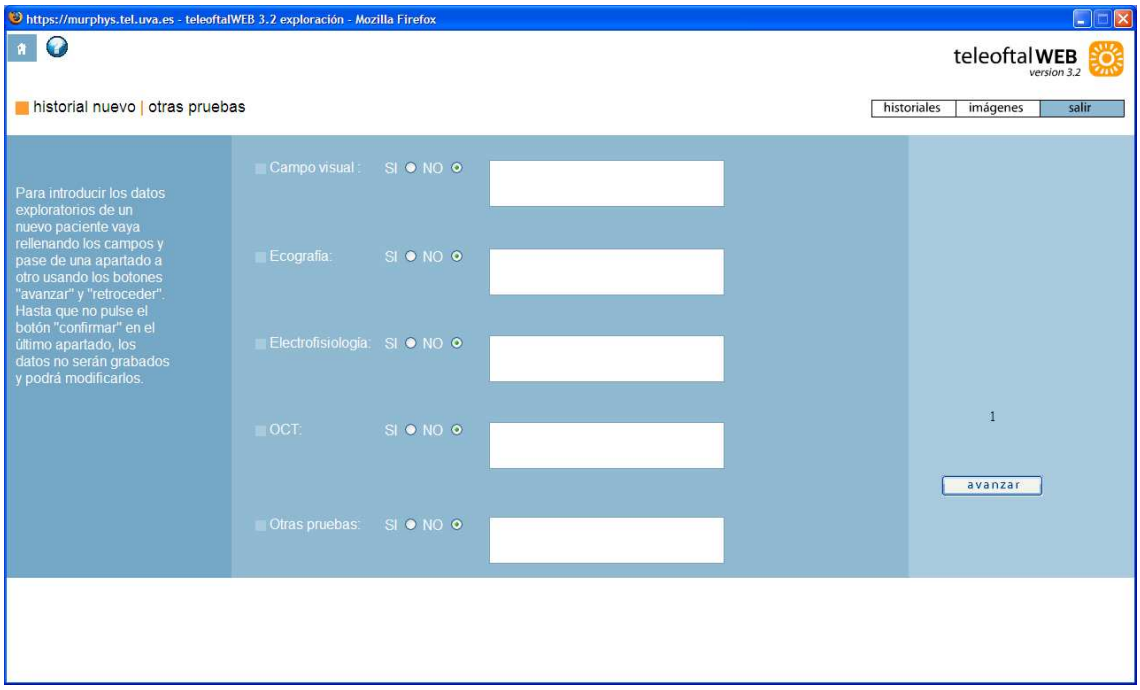


Figura 188. Ventana con los datos de las pruebas adicionales realizadas. Fuente: propia.

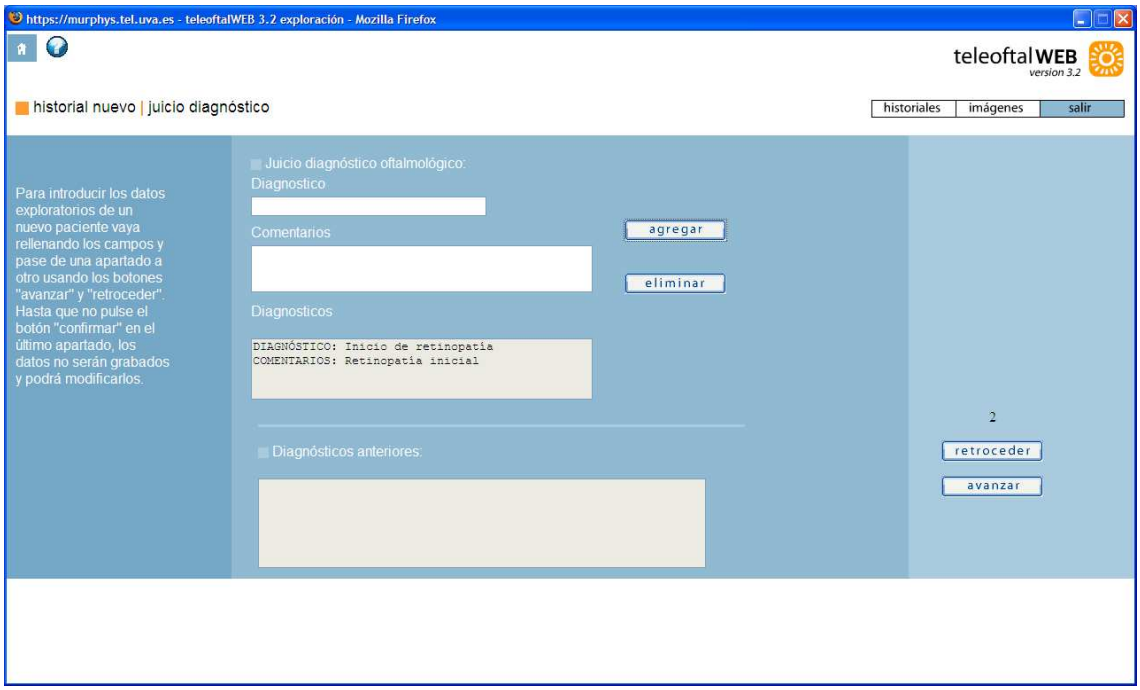


Figura 189. Ventana de diagnósticos tras insertar un nuevo diagnóstico. Fuente: propia.

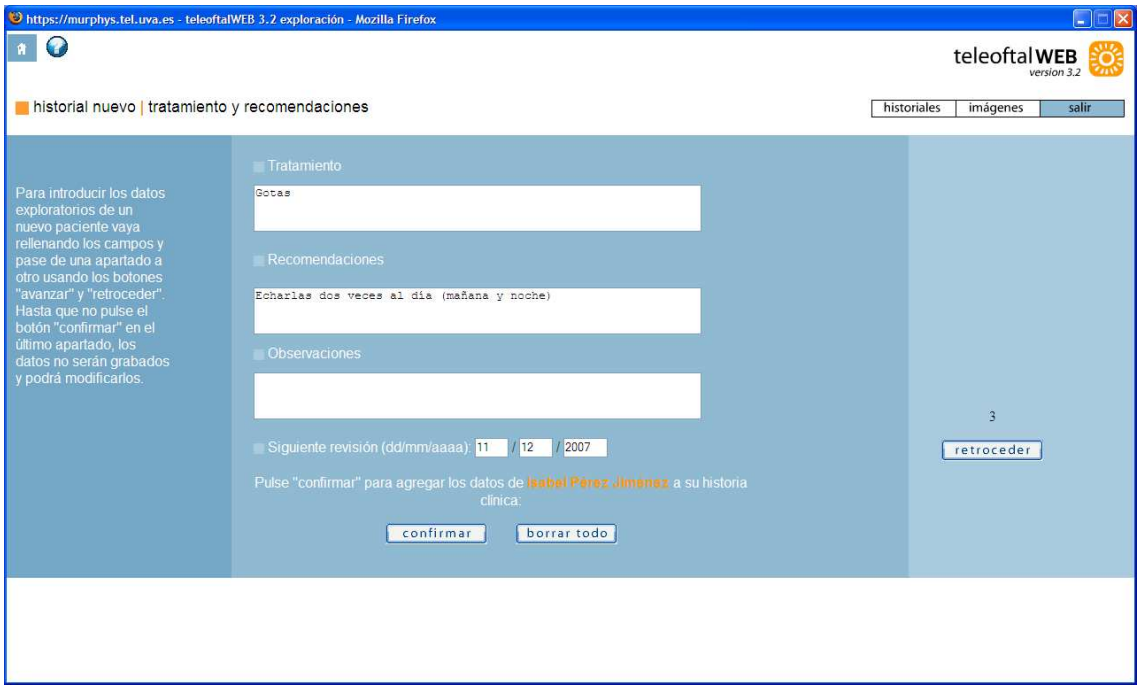


Figura 190. Ventana con el tratamiento y las recomendaciones pertinentes. Fuente: propia.

Una vez que se ha llegado a este punto, si se pulsa el botón "confirmar" y los datos son correctos, se vuelve a actualizar la BD, y se muestra la pantalla inicial con las estadísticas del usuario, frente al resto.

Búsqueda de historiales

Para iniciar la búsqueda de un historial, se debe pinchar sobre el botón "historiales" y luego sobre el botón desplegado a su izquierda buscar historial. Seguidamente aparece una pantalla (figura 191) como la que se muestra a continuación, con una lista en la que se muestran los pacientes a cuyo historial se tiene acceso. Cuando un historial es compartido no se tendrá acceso a los datos personales del paciente por seguridad. Así la primera columna de la lista son los apellidos y nombre del paciente, a continuación aparece el DNI del paciente, la fecha de creación del historial y por último el tipo de acceso al historial. Como esta lista puede ser más o menos grande, en función del número de pacientes, será conveniente poder realizar búsquedas en la misma, siguiendo ciertos criterios.

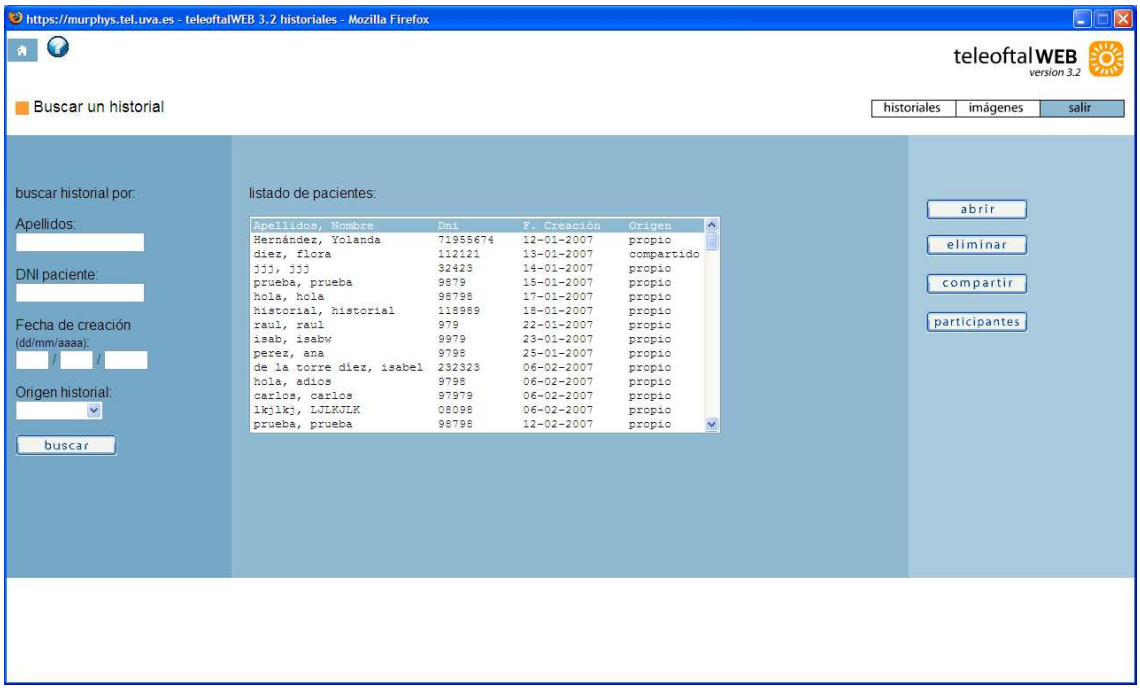


Figura 191. Ventana de búsqueda de un historial. Fuente: propia.

Uno de los criterios mencionados anteriormente, sería realizar una búsqueda por apellidos, para lo cual se introduciría el apellido a buscar en el campo de texto correspondiente de la parte izquierda de la aplicación, y se pulsaría "buscar". Así la aplicación cliente generaría una consulta al servidor, que se comunicaría con la BD para encontrar las posibles ocurrencias del apellido. Los resultados se muestran en la lista en la que antes se mostraban todos los pacientes accesibles. Seleccionando un registro de la lista se puede realizar tres operaciones, según indican los botones de la derecha: abrir el historial (pulsando "abrir"), eliminar el historial (pulsando "eliminar"), y compartir el historial (pulsando "compartir"). Es de notar que en lo referente a la operación eliminar, el historial sólo se eliminará a nivel de vista del usuario ya que se mantendrá una copia en la BD, para poder recuperarlo en caso de que se hubiera eliminado por error. Los otros criterios que se pueden emplear para realizar una búsqueda es utilizar el identificador del paciente, la fecha de creación del historial o el tipo de acceso.

Al realizar una búsqueda, o simplemente al consultar la lista completa de los pacientes, es posible realizar tres operaciones: abrir historial, eliminar historial y compartir historial. En el primer caso, al abrir un historial aparecería una lista con las revisiones que se han realizado al paciente. De estas se puede seleccionar alguna ficha en concreto pulsando "abrir", en cuyo caso aparecería inicialmente la ficha de filiación

con los datos rellenos y luego se podría ir pasando por todos los campos vistos anteriormente. Si se pulsa “nuevo” aparecería la ficha de filiación del paciente rellena, con los datos de la última revisión. Estos datos pueden ser editados en caso de haberse producido algún cambio. A continuación se podrán rellenar los datos de anamnesis, exploración, etc. de la nueva revisión. En el caso de ser un historial compartido los datos personales del paciente no serían visibles. En la figura 192, se puede observar el listado de revisiones de un HCE.

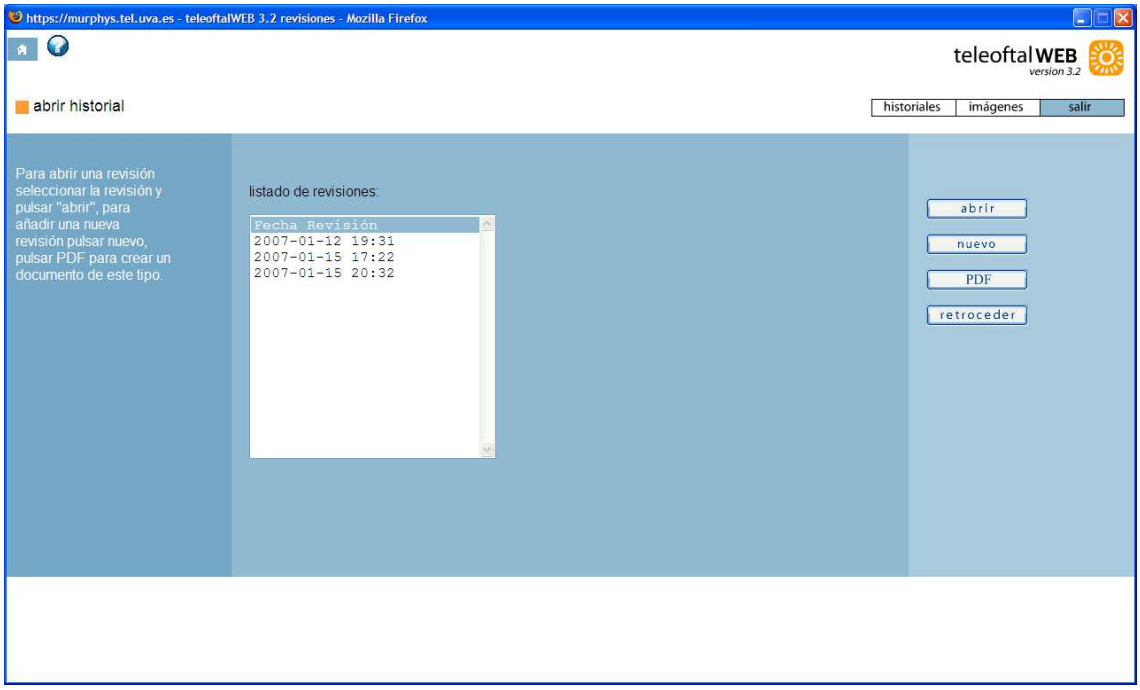


Figura 192. Ventana con las revisiones de un historial. Fuente: propia.

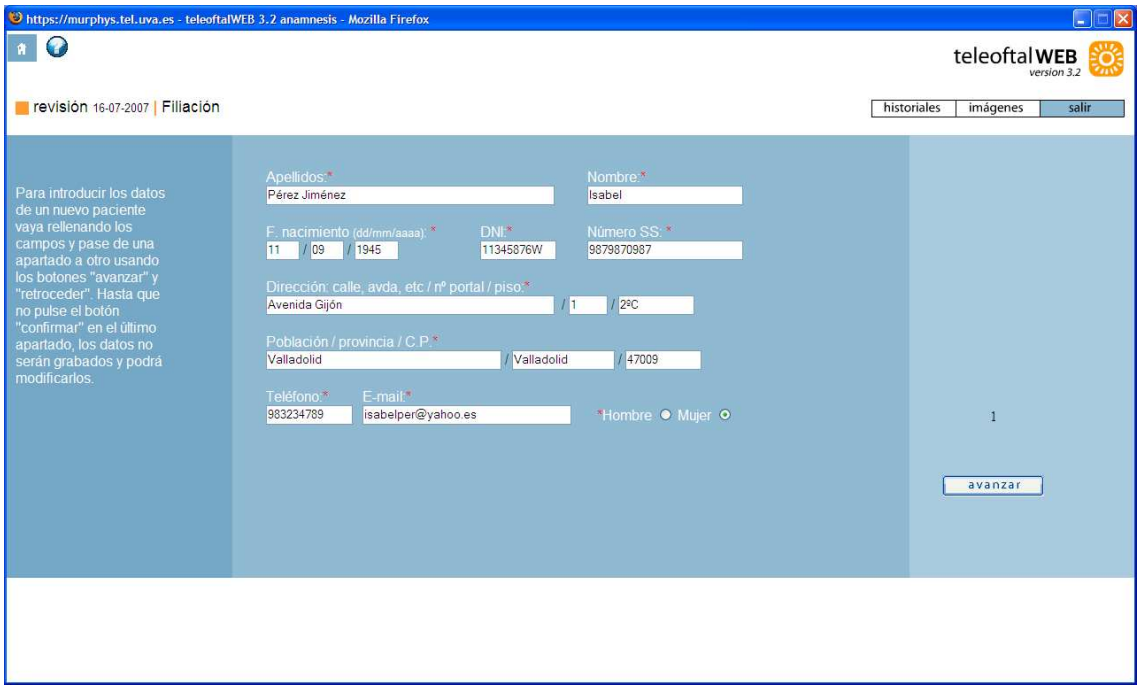


Figura 193. Ventana con los datos personales de una revisión de un historial propio. Fuente: propia.

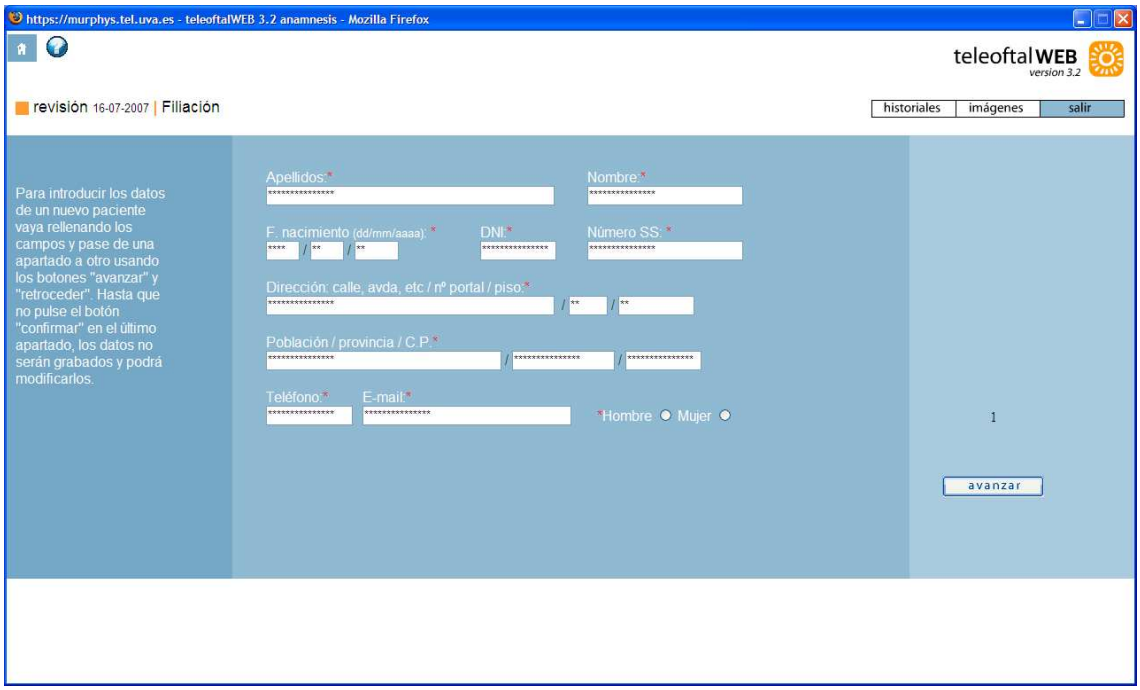


Figura 194. Ventana con los datos personales de una revisión de un historial compartido. Fuente: propia.

Las otras operaciones que se pueden realizar son: la eliminación de un historial, y compartir un historial con otros usuarios. En la eliminación de historiales habrá que confirmar la decisión, por si acaso se hubiera pulsado el botón por error. En cualquier

caso el historial solamente se eliminará de forma lógica para el usuario, pero permanecerá almacenado en la BD. Además se podrá compartir un historial con otros usuarios, para por ejemplo pedir una segunda opinión, o por fines de investigación, si la patología se saliera de lo habitual. En la figura 195, se muestra la pantalla que aparecería con el listado de usuarios disponibles para compartir. Para compartir el historial simplemente se seleccionaría el usuario con el que se quiere compartir el historial y se pulsaría "compartir", en caso de que se quisiera volver a la pantalla de los historiales se pulsaría "retroceder". Sólo se podrán compartir los historiales propios.

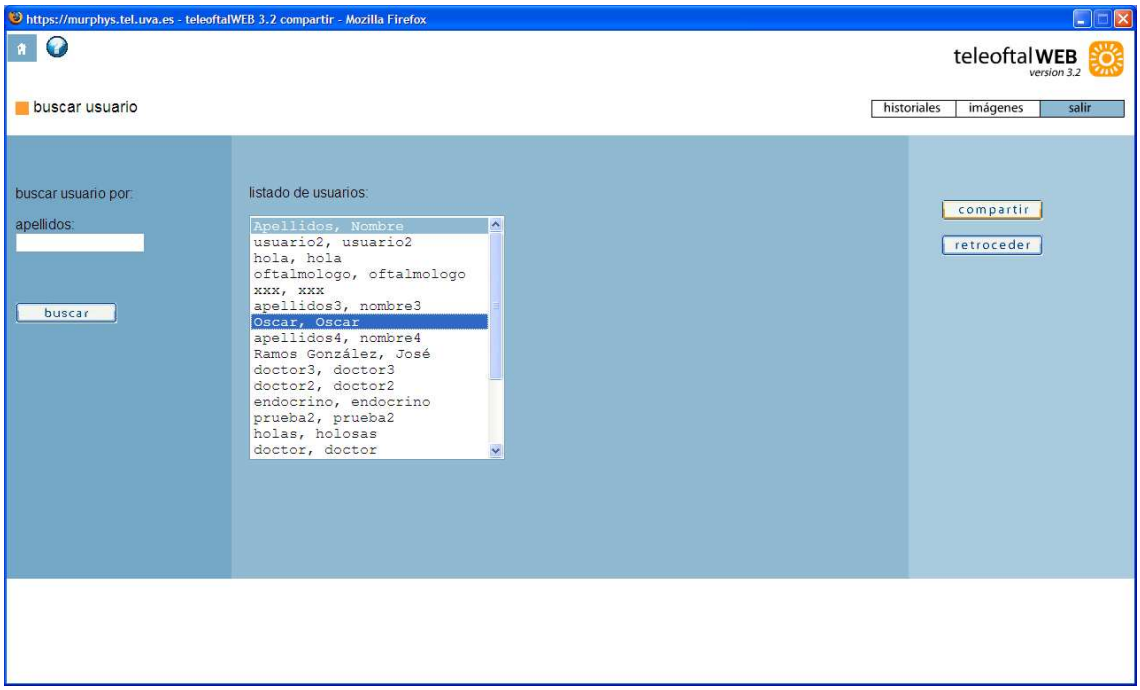


Figura 195. Ventana con los participantes de un historial. Fuente: propia.

Acceder a la aplicación de impresión de HCE

Para acceder a la aplicación de impresión de los HCEs en formato PDF hay que usar la opción 'PDF' que aparece en la ventana abrir historial. En esta ventana se pueden diferenciar varias partes: el cuerpo de la página que es donde se encuentran las revisiones del paciente, la parte izquierda de la pantalla que es una pequeña ayuda de lo que se puede hacer , la parte de la derecha con los diferentes botones de lo que hace la aplicación (aquí se encuentra el nuestro para poder acceder a nuestra herramienta de impresión) y la parte superior en la que se distinguen los diferentes módulos (imágenes, historiales y la posibilidad de salir de la aplicación).

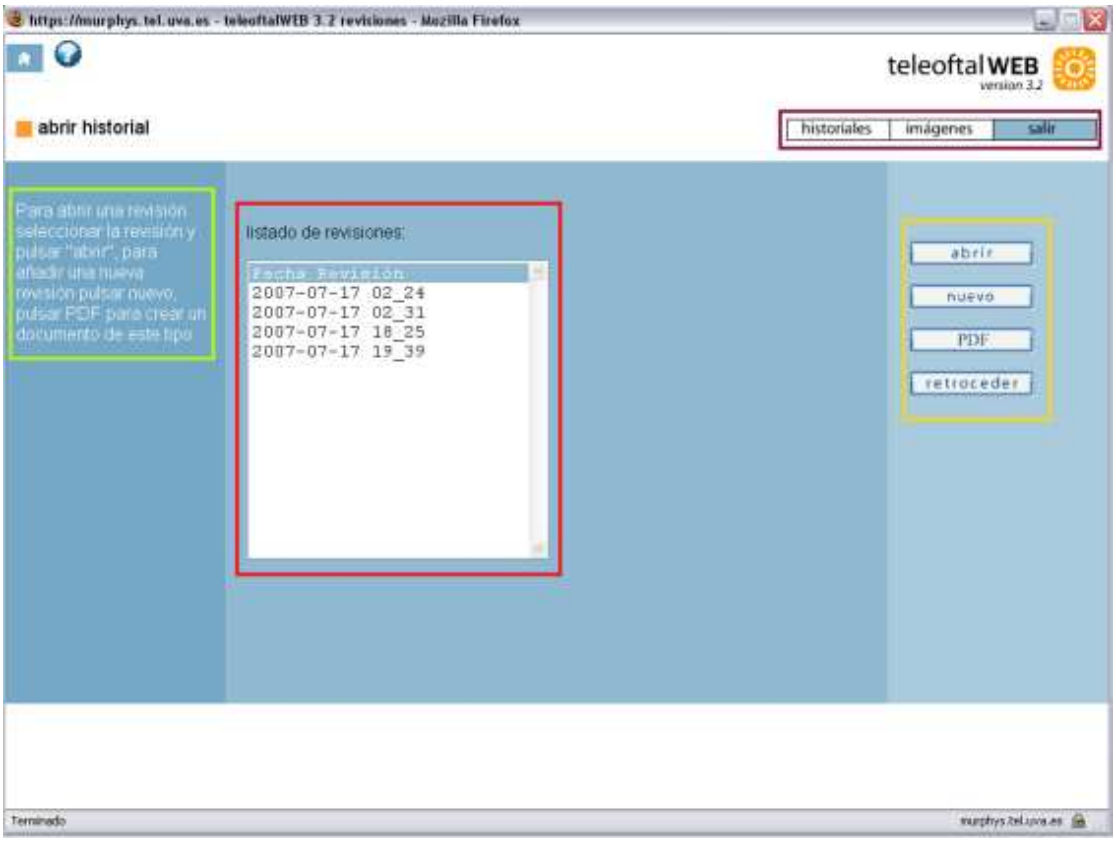


Figura 196. Abrir Historial del paciente. Fuente: propia.

Para poder acceder a la herramienta de impresión de es necesario que se selecciones una revisión de las que aparecen en el cuerpo de la pantalla y a continuación se pulse el botón “PDF” que se encuentra a la derecha de la aplicación. En caso de que no se seleccione una revisión no se podrá acceder a la impresión del historial.

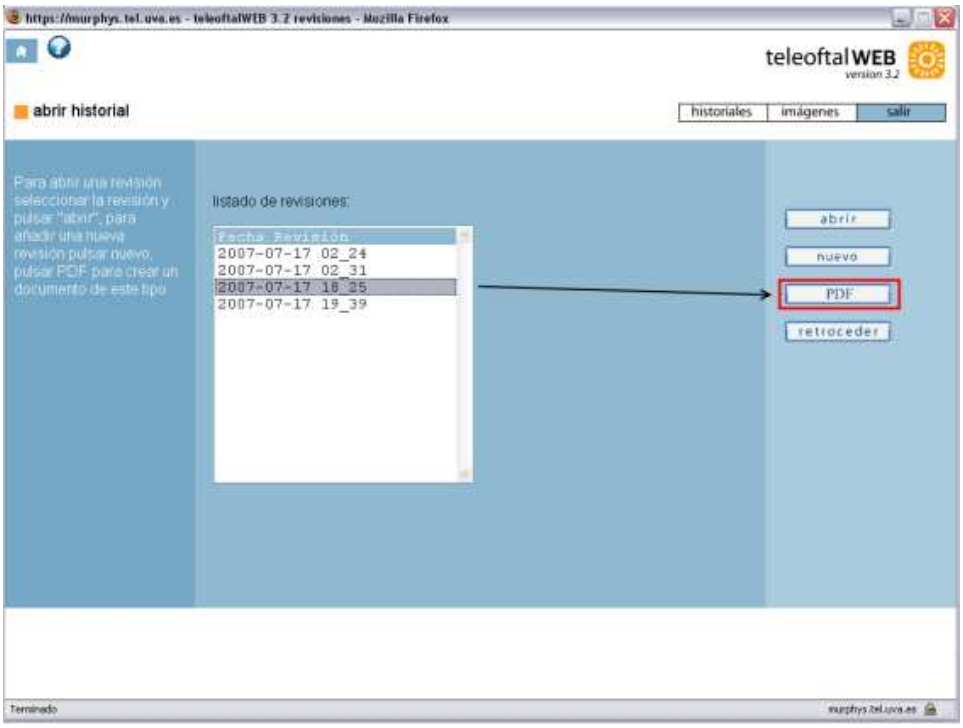


Figura 197. Acceder a la herramienta de impresión. Fuente: propia.

Creación del HCE en formato PDF

Una vez pulsada la opción de "PDF" aparecerá una nueva ventana en la que aparecerá la revisión (en formato año-mes-día y hora) que se va a imprimir en el centro de la pantalla y a la derecha las opciones que se pueden realizar.

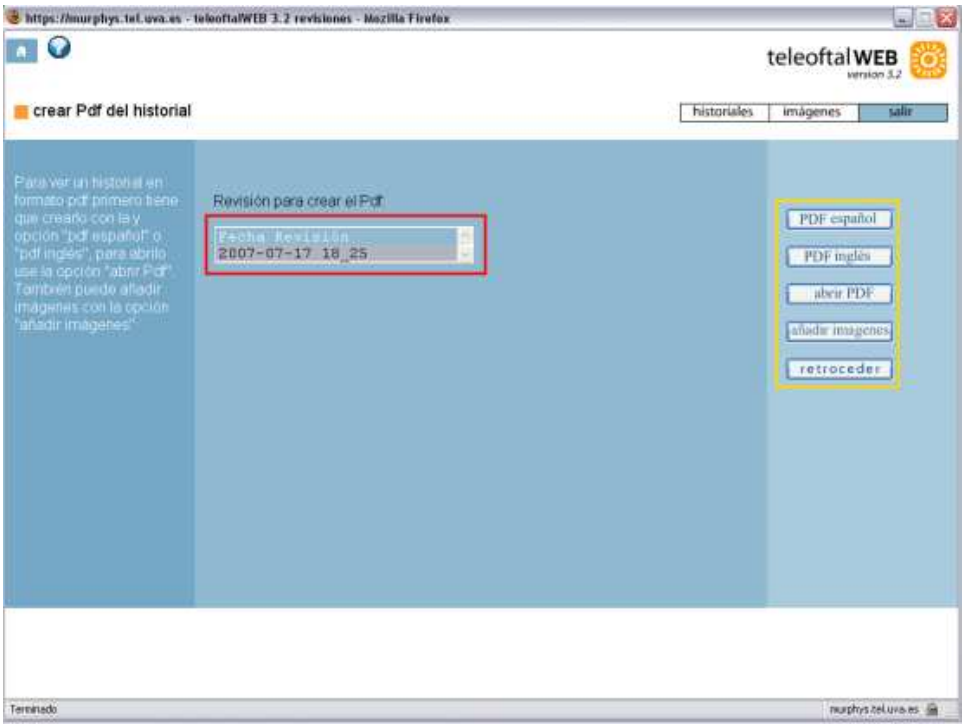


Figura 198. Ventana para la creación del PDF. Fuente: propia.

Creación del documento PDF en español o inglés

Para crear el documento en español de la revisión hay que usar el botón “PDF español”. En caso de que el PDF se haya creado correctamente aparecerá un mensaje indicando que se ha creado sin problemas el documento (“Archivo pdf creado con éxito”).

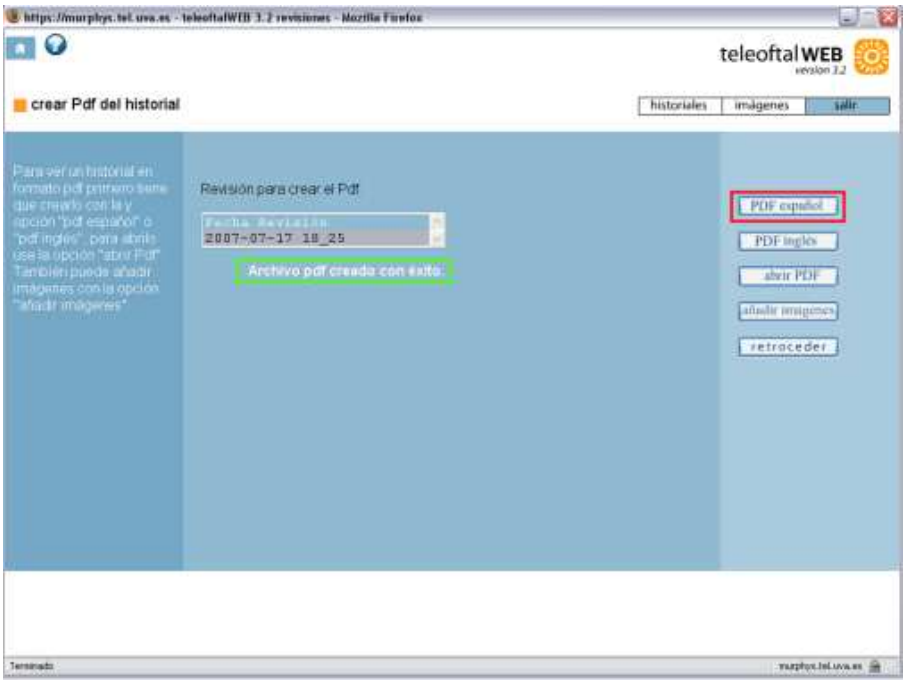


Figura 199. Creación del PDF en español. Fuente: propia.

Para crear el documento en inglés usar la opción “PDF inglés”. En caso correcto aparecerá el mismo mensaje que antes.

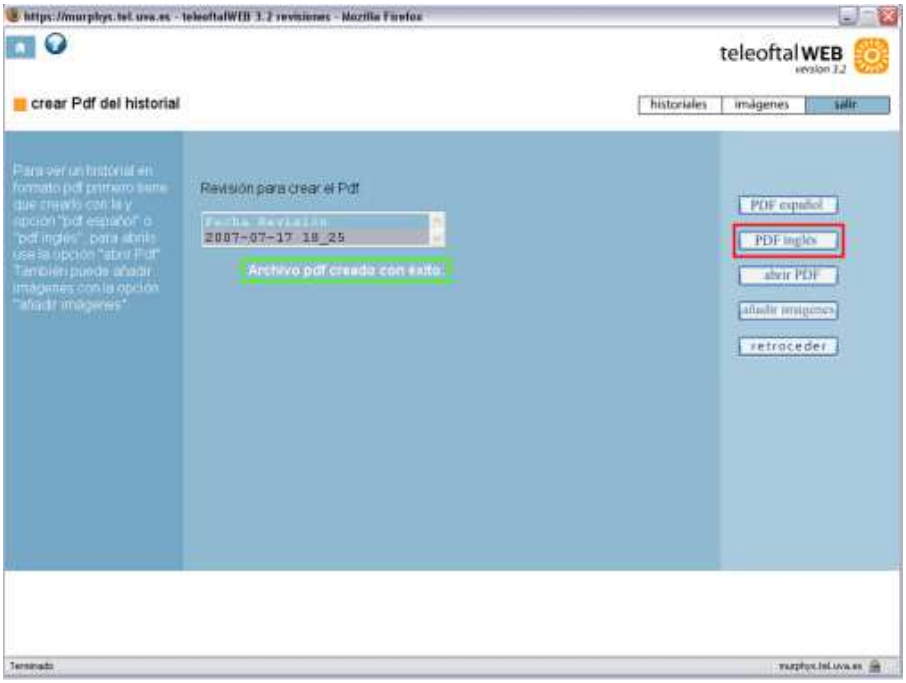


Figura 200. Creación del PDF en inglés. Fuente: propia.

Abrir el documento PDF

Antes de poder abrir el PDF es necesario crearlo con la opción “PDF español” o “PDF inglés”, si no se realiza esta acción previamente aparecerá una ventana con un mensaje de alerta en el que se indicará que el archivo PDF no se ha creado y no se puede abrir.

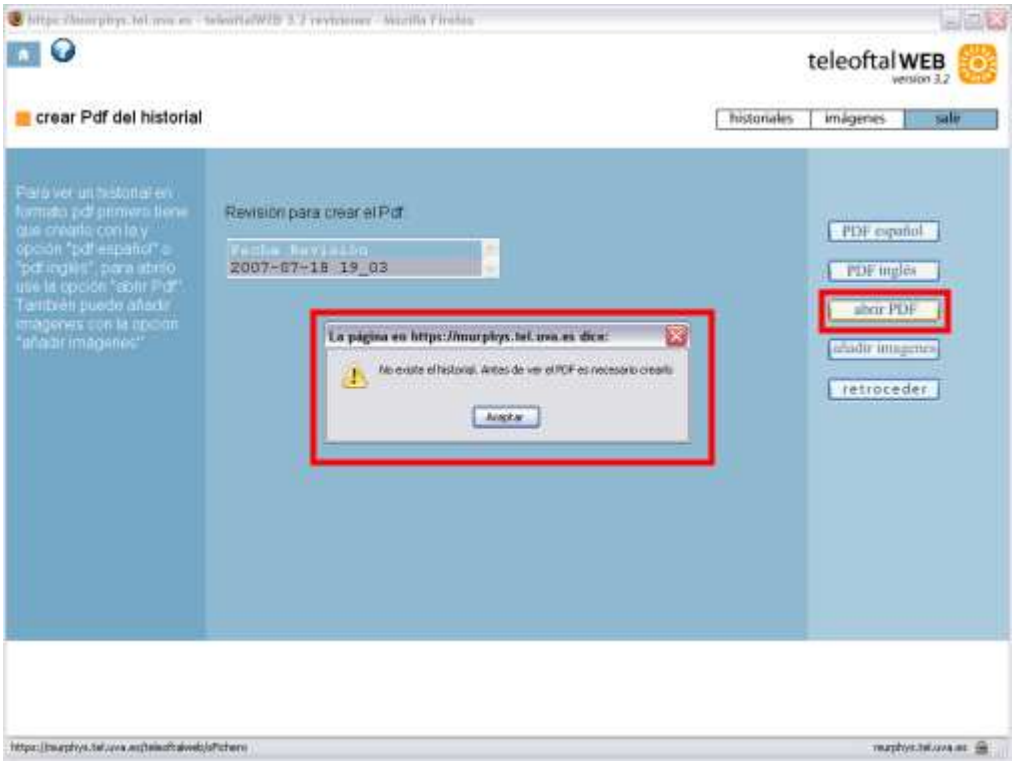


Figura 201. Mensaje de alerta al no poder abrir el PDF. Fuente: propia.

El mensaje de alerta es el siguiente:

Figura 202. Mensaje de alerta. Fuente: propia.

En caso de que el historial exista (porque ya se ha creado) se abrirá un documento PDF con los datos del paciente en esa revisión. En caso de que se vaya a

otra parte de la aplicación una vez creado el documento PDF y durante la sesión se quiera volver a ver la misma revisión del paciente no será necesario crear de nuevo el PDF (a no ser que se realicen modificaciones en el, como puede ser añadir otras imágenes distintas a las anteriores). Vemos un ejemplo del documento creado en español:

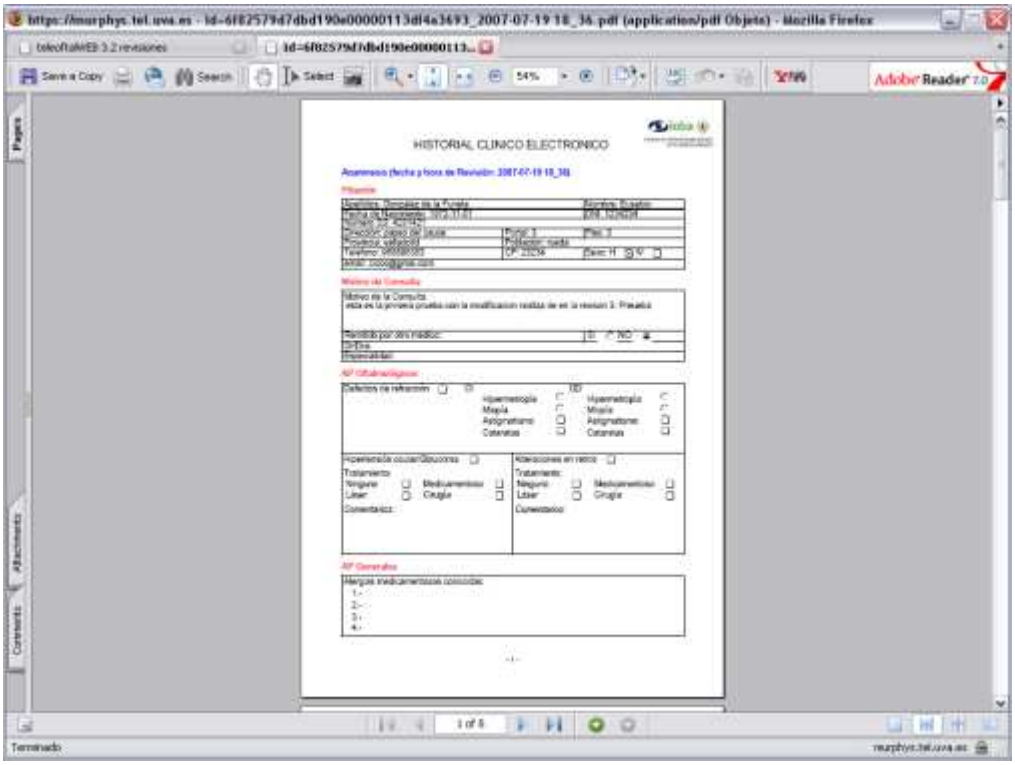


Figura 203. Parte de un documento PDF en español. Fuente: propia.

También se puede crear el documento PDF en inglés:

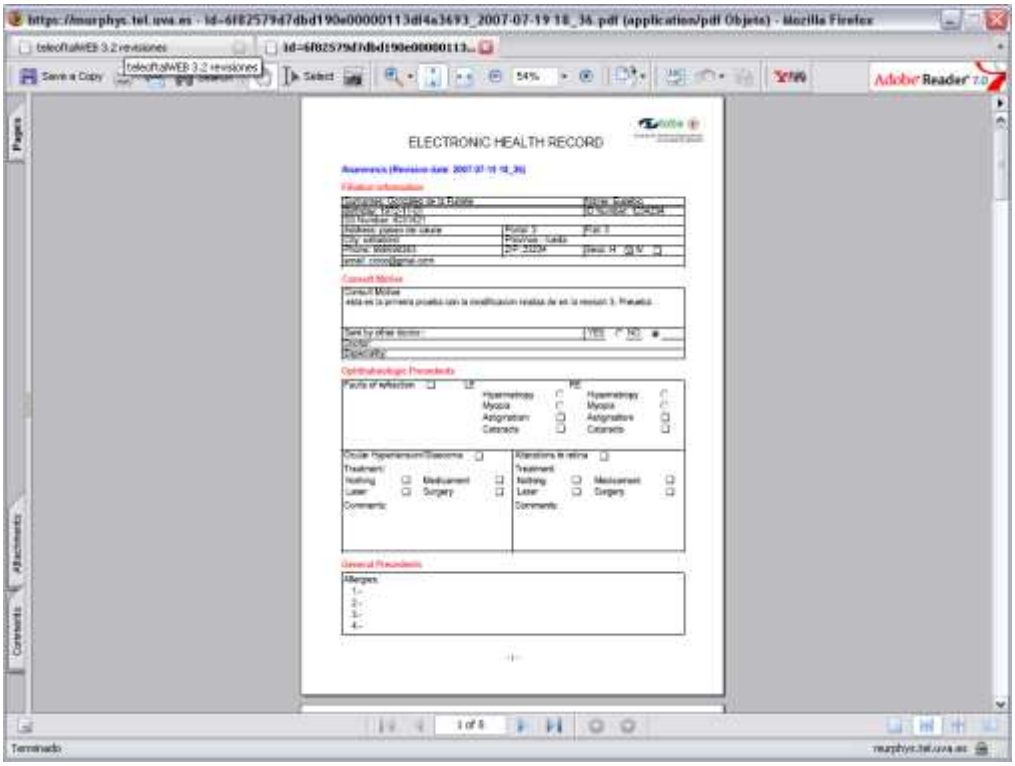


Figura 204. Parte de un documento PDF en inglés. Fuente: propia.

Las otras dos opciones que quedan son "añadir imágenes" y "retroceder". Cuando se pulsa la primera de ellas se irá a otra pantalla (añadir imágenes PDF) en la que aparecerán todas las imágenes asociadas al paciente y en caso de que se pulse retroceder se irá a la pantalla anterior (abrir historial).

Añadir imágenes al documento PDF

En esta pantalla de la aplicación aparecen todas las imágenes asociadas al historial (es decir, del paciente). Se pueden diferenciar tres partes: la de la derecha donde se encuentra un pequeña explicación de lo que se puede hacer en esta pantalla, la parte del centro que es donde se muestran todas las imágenes del paciente y la parte de la derecha que muestra información de la imagen (fecha de creación, nombre y descripción de la imagen) y 2 botones con las opciones que se pueden realizar.

Hay que tener en cuenta que en esta parte no es posible añadir imágenes, puesto que ya existe en la aplicación una parte en la que se pueden añadir y modificar las imágenes (sección imágenes) de los historiales.

Se pueden realizar 3 acciones en esta parte de la aplicación: una da la posibilidad de añadir las imágenes del historial con el botón "añadir imágenes" al documento PDF, otra de ellas es volver a la pantalla anterior sin añadir ninguna imagen con el botón

"retroceder" y la última opción es ver la imagen más ampliada haciendo doble *click* sobre la propia imagen.

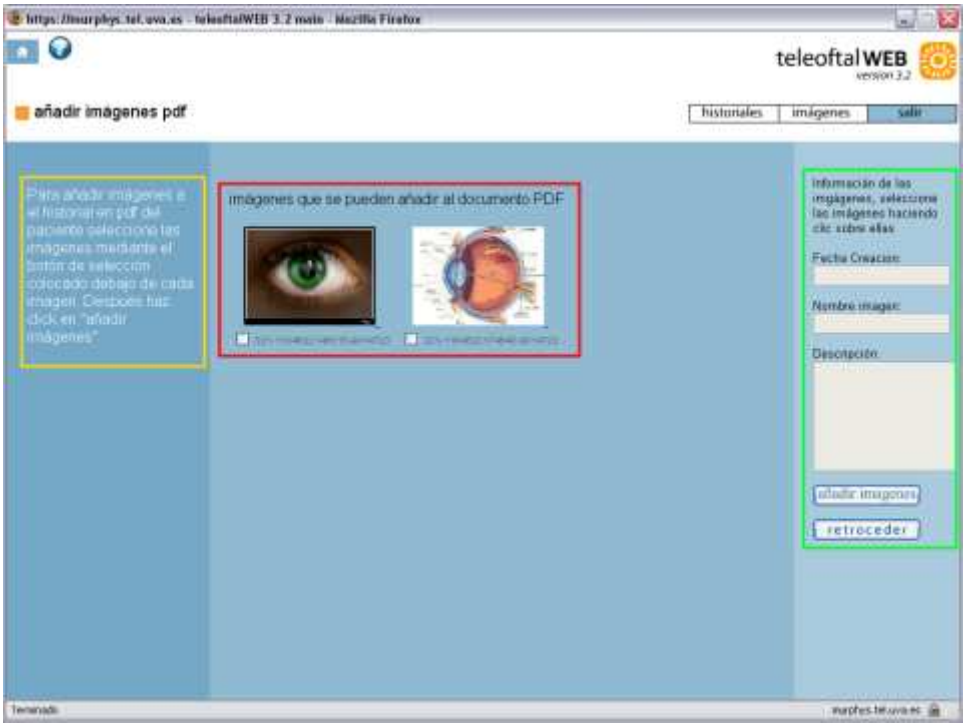


Figura 205. Partes de la sección añadir imágenes a el documento PDF. Fuente: propia.

Si el paciente no tiene ninguna imagen añadida aparecerá en el centro de la pantalla un mensaje indicando que no existen imágenes para ese historial.

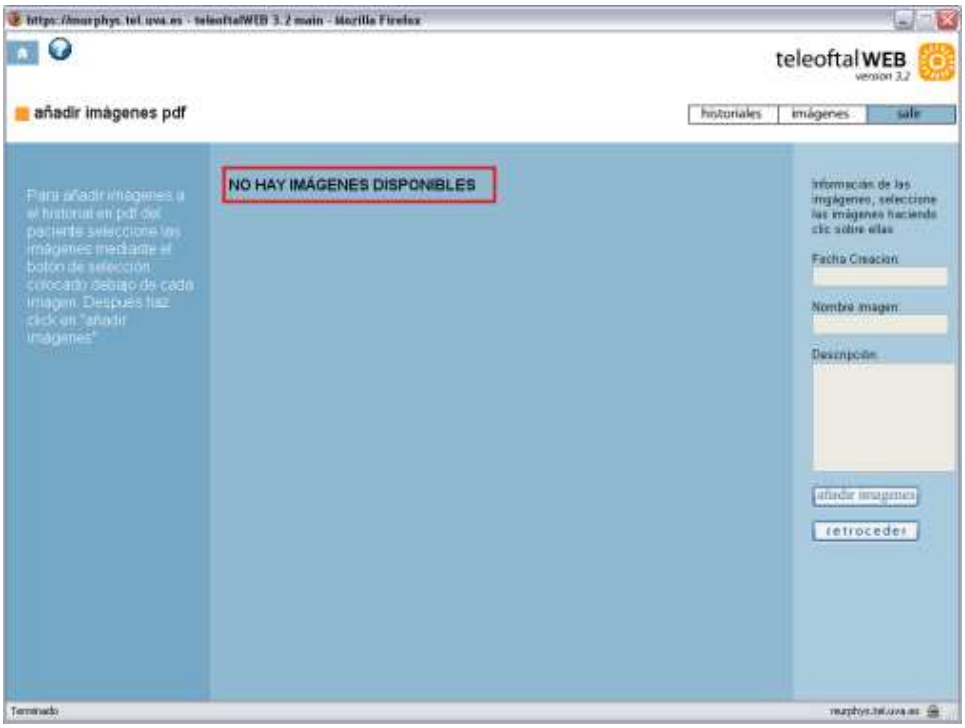


Figura 206. No hay imágenes disponibles para añadir al PDF. Fuente: propia.

Para añadir las imágenes al documento PDF es necesario seleccionar aquellas imágenes que se desean imprimir mediante una casilla de verificación que se encuentra debajo de cada imagen. Cuando se pulsa el botón "añadir imágenes" volvemos a la pantalla de creación del PDF y todas aquellas imágenes que se estaban seleccionadas se imprimirán en el historial del paciente. Esta operación habrá que realizarla siempre que se desee añadir imágenes a la revisión.

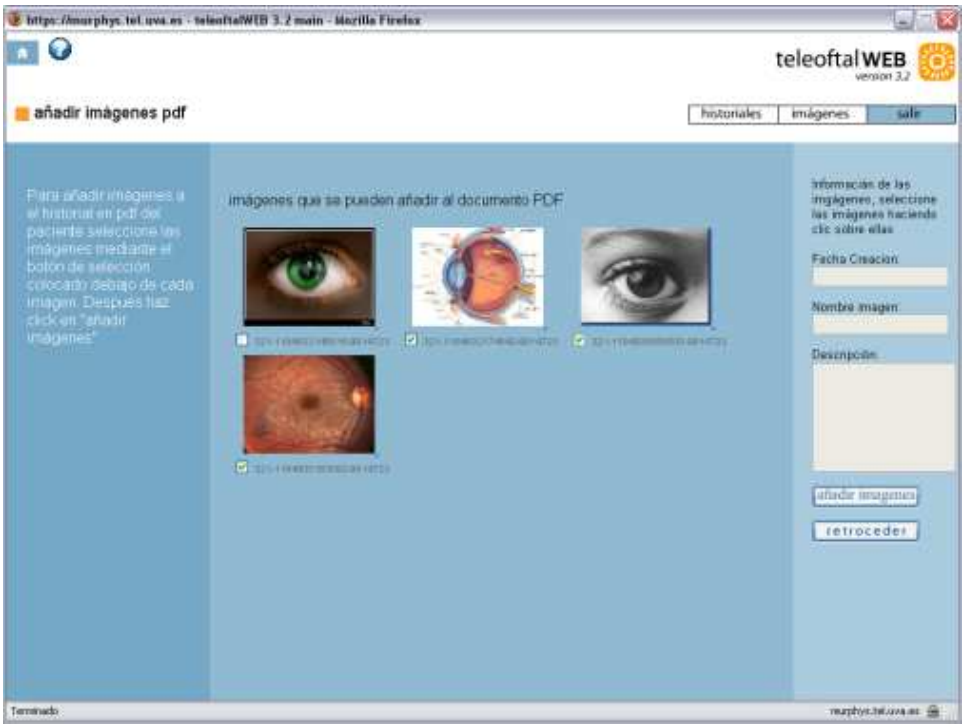


Figura 207. Selección de imágenes para el historial. Fuente: propia.

Figura 208. Imágenes seleccionadas para la impresión. Fuente: propia.

Después de añadirse las imágenes y crear el PDF tendríamos nuestro resultado en la figura 209.

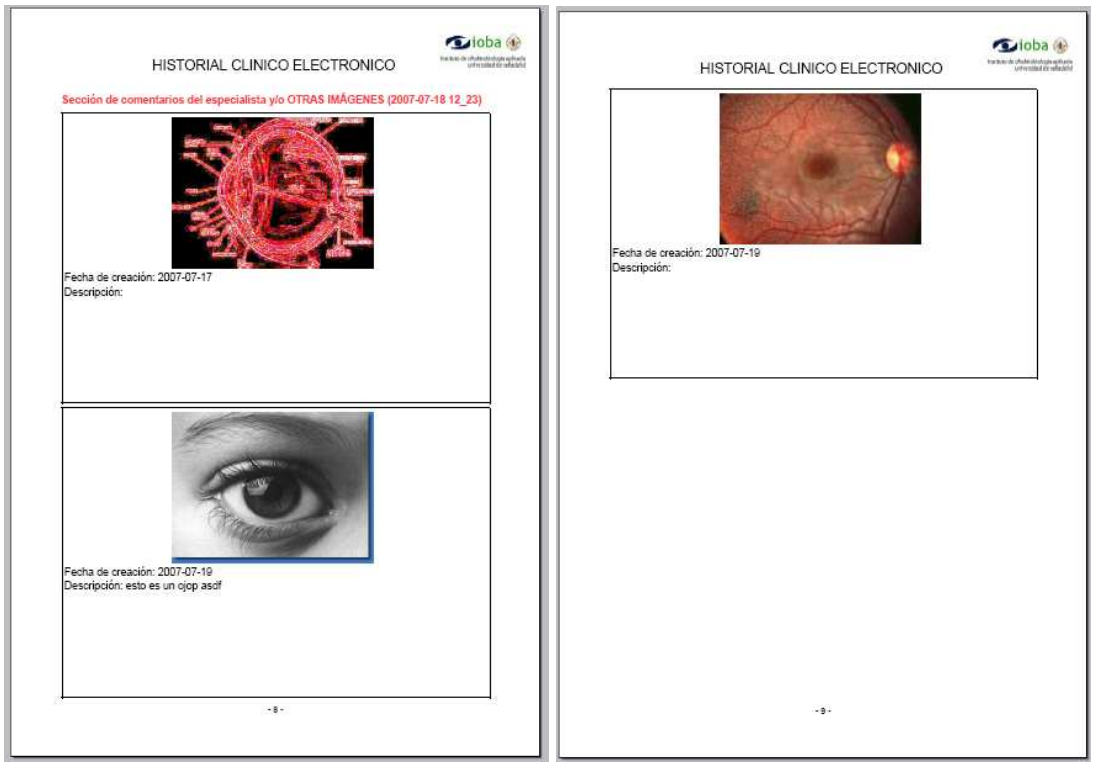


Figura 209. Imágenes que aparecen en el PDF. Fuente: propia.

Puede verse que la primera figura es distinta a la que realmente aparece en la pantalla de la aplicación, esto es así porque lo que realmente se imprime en el historial es la imagen modificada. En caso de que las imágenes no se hayan modificado, aparecerán las imágenes originales.

Otra opción que se puede realizar es ver la imagen original en un tamaño mayor al que aparece en la pantalla, para ello es necesario hacer doble *click* sobre la imagen, cuando esto ocurre aparece en el centro de la pantalla la imagen original y a la derecha se despliega un nuevo menú en el que se muestran dos botones nuevos: uno de ellos es "editada", que en el caso de que se pulse aparecerá la imagen modifica (solo en caso de que esté modificada, sino no ocurrirá nada al pulsar el botón) y el otro es "retroceder" que vuelve a la pantalla anterior, es decir donde se encuentran todas las imágenes en miniatura.

A continuación mostramos una secuencia de figuras en las que se puede ver la explicación de lo que se acaba de decir. Hacemos doble *click* sobre la siguiente imagen:

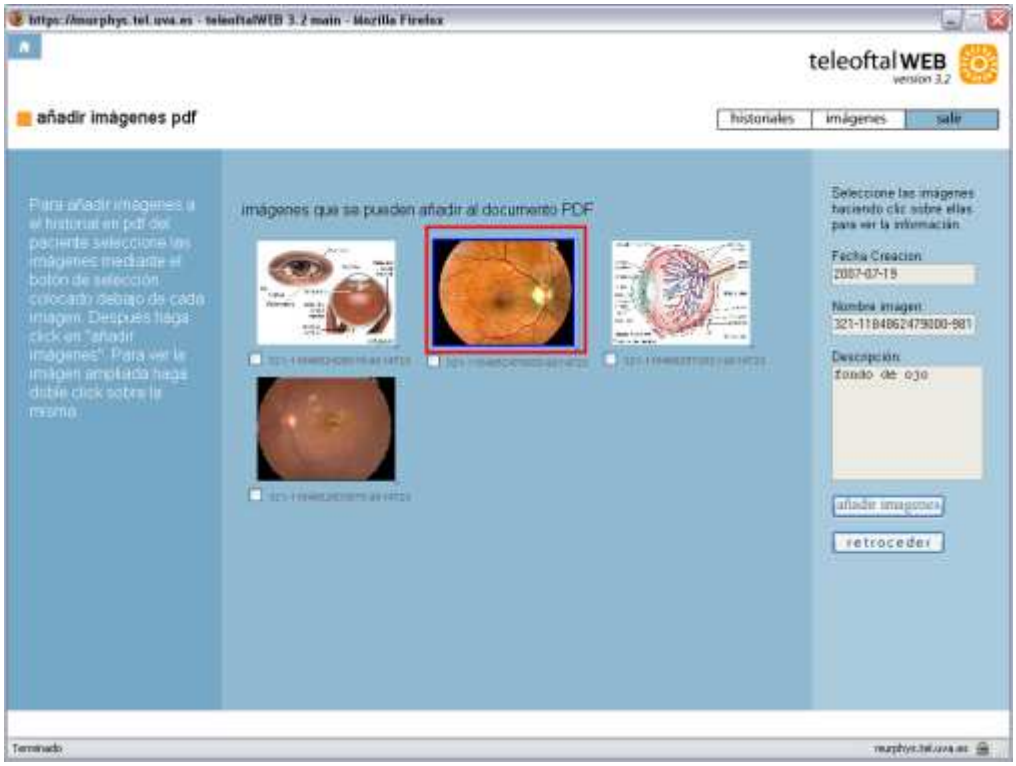


Figura 210. Selección de imagen. Fuente: propia.

Se amplia la imagen que se ha seleccionado y puede verse la imagen original.

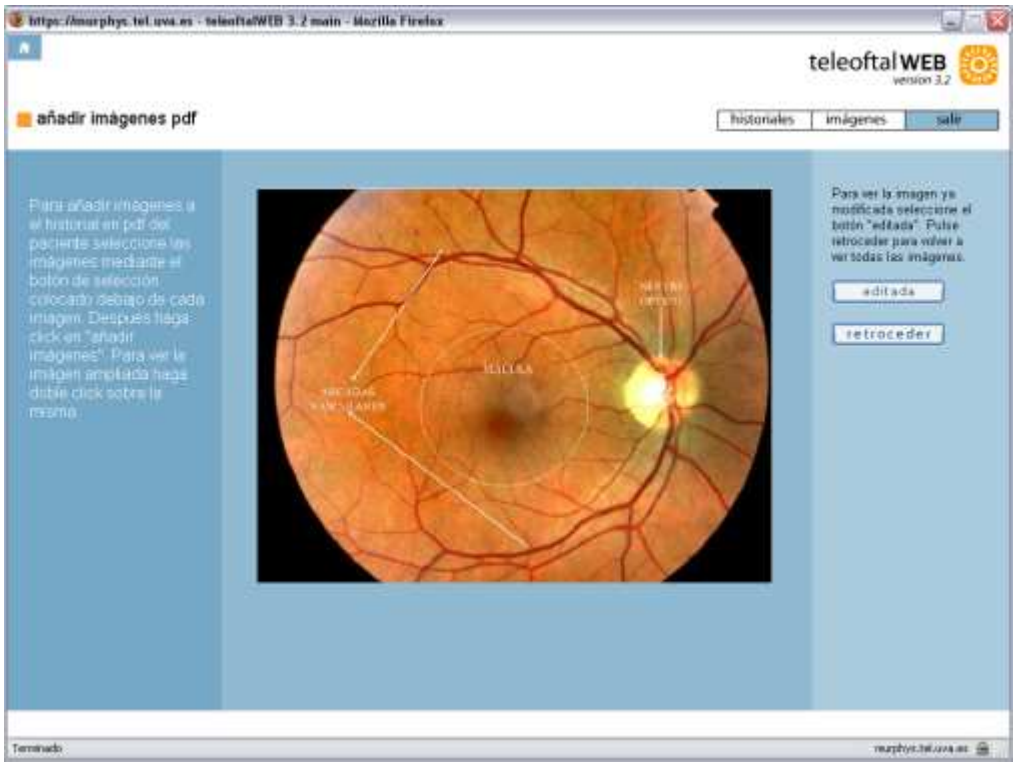


Figura 211. Imagen original. Fuente: propia.

Si se hace *click* sobre el botón "editada" puede verse la imagen modificada (en el caso de que se haya modificado) que es, como se ha indicado anteriormente la que se incluirá en el historial en PDF del paciente.

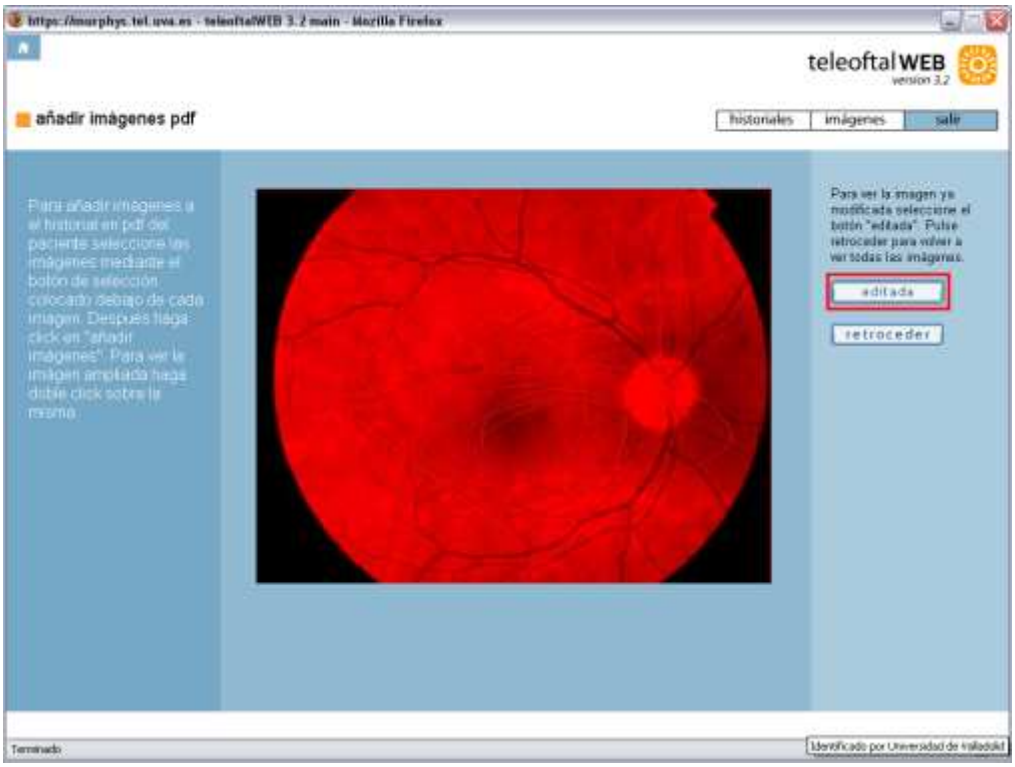


Figura 212. Imagen modificada. Fuente: propia.

Si se pulsa "retroceder" se vuelve a la pantalla anterior.

Es posible que un historial tenga asignada un número elevado de imágenes y en la pantalla no aparezcan todas, para poder ver las imágenes que faltan hay que utilizar las flechas que se encuentran situadas a la izquierda de la pantalla, justo encima de las imágenes. Además en cada momento se nos indicará en que número de página nos encontramos. Vemos un ejemplo de lo que se acaba de explicar:

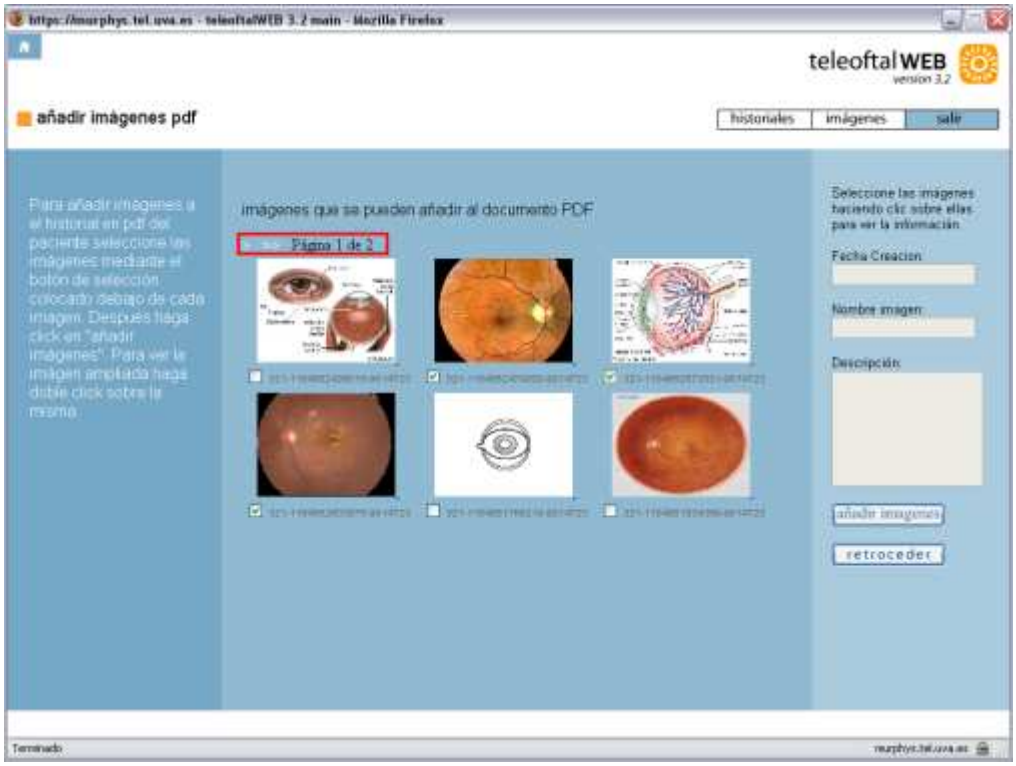


Figura 213. Primera página de imágenes. Fuente: propia.

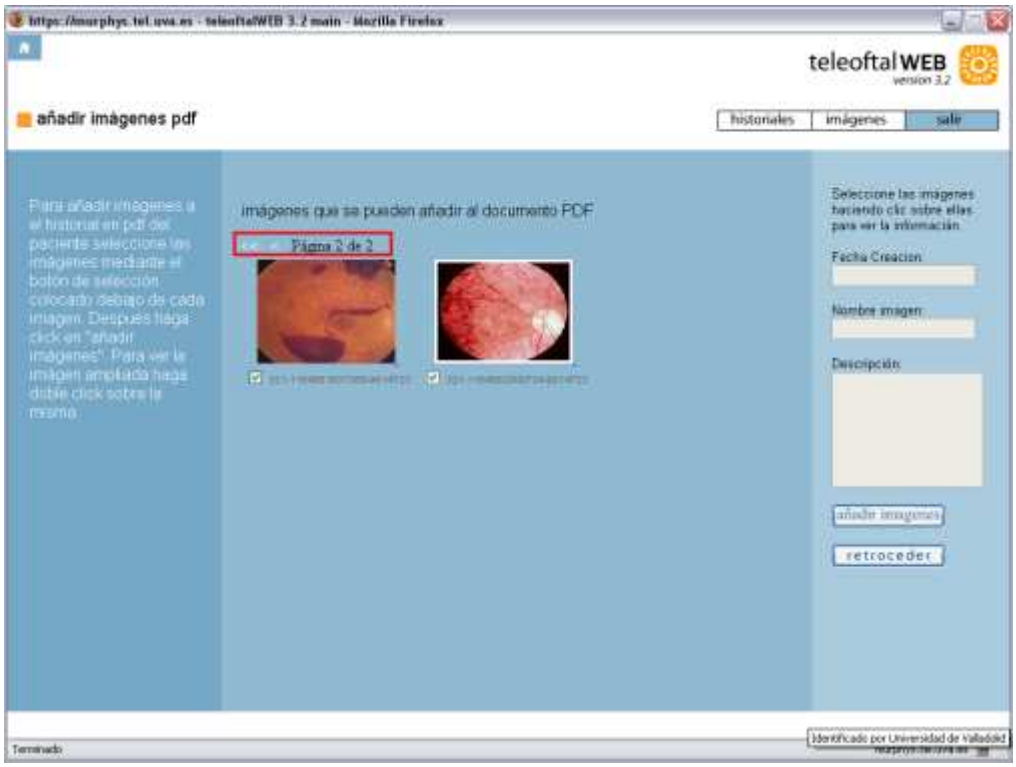


Figura 214. Segunda página de imágenes. Fuente: propia.

Añadir imágenes

El otro módulo de la aplicación estaría formado por la gestión de las imágenes asociadas a los historiales. Para acceder a estas funciones se pulsaría en el botón "imágenes" en la parte superior derecha, desplegándose a continuación dos submenús: añadir imagen y buscar imágenes. En el caso de añadir imágenes a la revisión, aparecería la siguiente ventana (figura 215). En la parte izquierda aparecen cuatro elementos: un cuadro de elección del historial del paciente al que se quiere añadir la imagen; otro cuadro de selección del medio del cual se cargará la imagen; un cuadro en el que añadir un comentario que se asociará con la imagen; y el botón "añadir imagen" mediante el cual se añaden al historial la imagen seleccionada. En la parte central se muestran las imágenes ya cargadas en el historial, y en la parte de la derecha se puede visualizar la información asociada a cualquiera de las imágenes cargadas sólo con seleccionarla en la parte central.

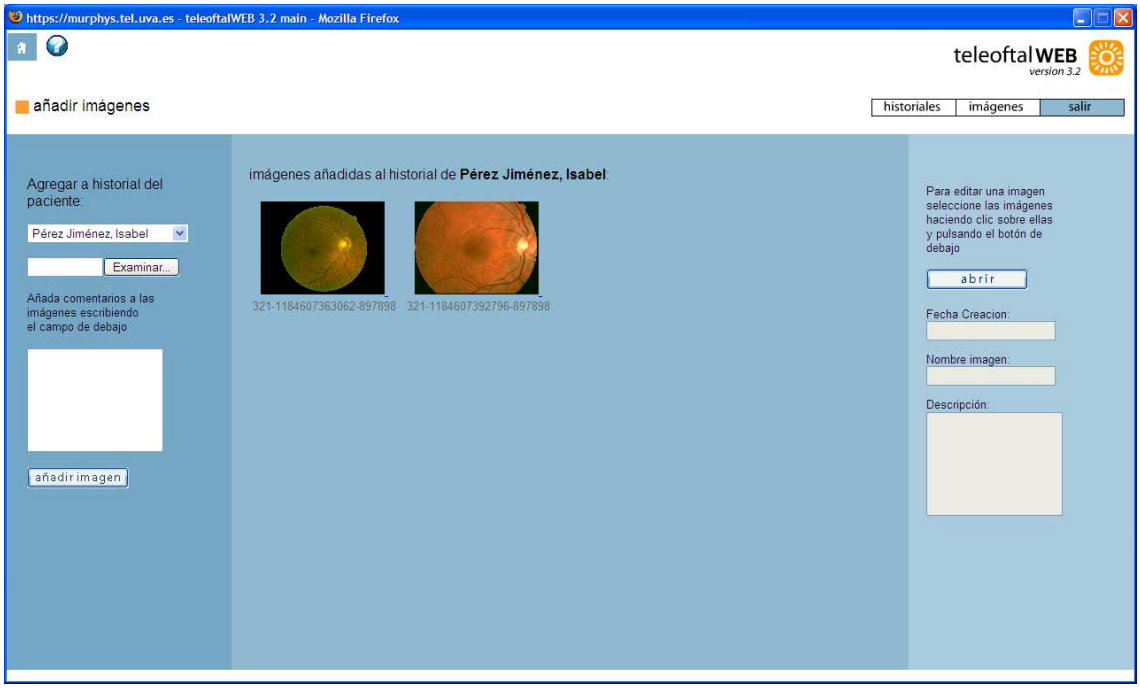


Figura 215. Ventana para añadir imágenes al historial de un paciente. Fuente: propia.

Mediante el botón "abrir" o haciendo doble clic sobre la imagen deseada, podremos acceder a la siguiente pantalla en la que se muestra la imagen ampliada y se pueden realizar varias operaciones sobre la misma. En la parte de la derecha de la

pantalla nos saldrán una serie de botones que nos permitirán: mediante el botón "editada" abrir la imagen editada, para ver los cambios que se han realizado en ella. Podemos eliminar la imagen del historial (pulsando "eliminar"), retroceder a la pantalla anterior (pulsando "retroceder"), o bien podemos deshacer todos los cambios que hayamos hecho a la imagen (original o editada, dependiendo de con cual de las dos estemos trabajando) mediante el botón "resetear", o podemos guardar los cambios realizados pulsando el botón "confirmar".

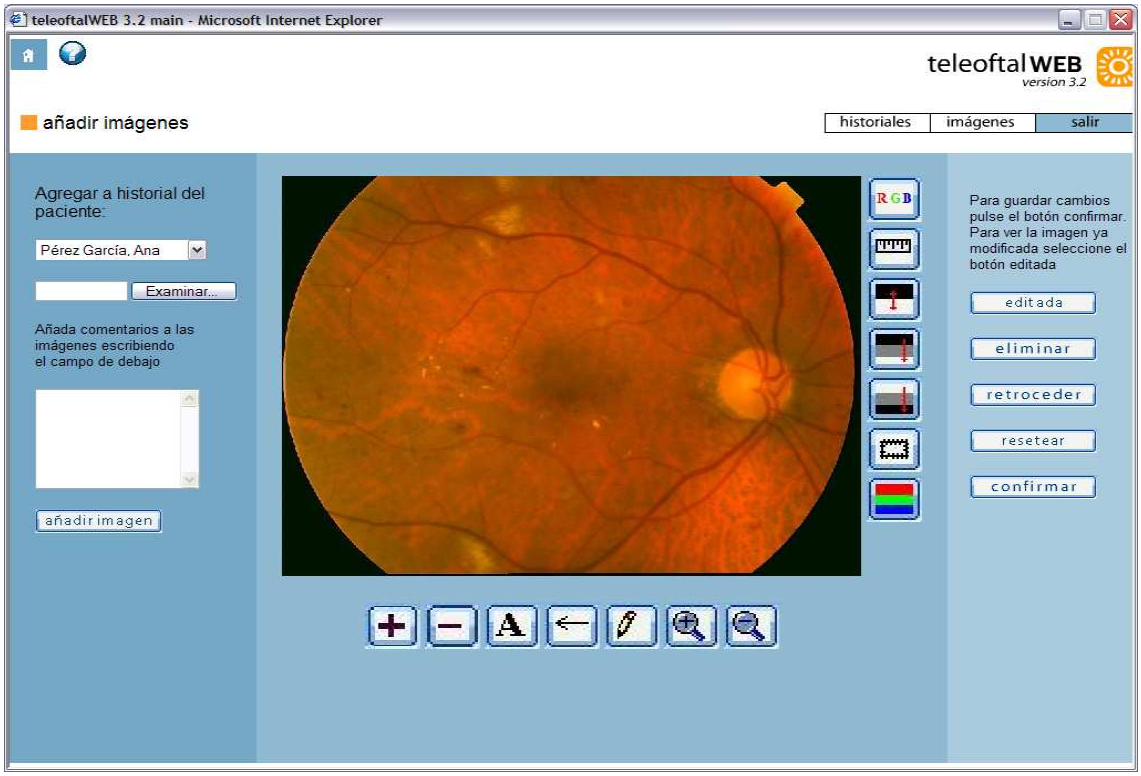


Figura 216. Pantalla con una imagen del historial ampliada. Fuente: propia.

En el centro de la pantalla nos aparece la imagen seleccionada ampliada rodeada de una serie de botones que nos permitirán realizar una serie de modificaciones.

De esta forma a la derecha de la imagen en primer lugar tenemos el botón

(RGB) que lo que hace es aplicar un filtro rojo, verde o azul dependiendo el número de veces que pinchemos sobre él.

A continuación, aparece el botón (regla) que nos permitirá calcular distancias relativas al ancho de la imagen para tener una idea proporcional de la separación que puede haber entre dos puntos de la imagen.

El siguiente botón (invertir) realiza una inversión de los colores de la imagen, lo que nos puede resultar útil para ver mejor ciertos detalles.

Seguidamente nos encontramos con los botones (aclarar) y (oscurecer) que según vamos pinchando sobre ellos obtenemos una imagen más clara u oscura según a cual de ellos demos.

Debajo de estos se encuentra el botón (bordes) que lo que hace es remarcar los límites de la imagen dándonos una mejor idea de cómo es el perfil de la imagen. Esta herramienta puede ser útil en algunas situaciones en las que no se vean bien los detalles de la imagen.

Por último nos encontramos con el botón (selecciona color) que nos abre una paleta de colores para elegir con cual de ellos queremos dibujar sobre la imagen. Una vez pinchado sobre el color, deberemos pinchar en el botón "OK" para que los cambios se realicen. En la figura 217 podemos ver como queda la pantalla tras pinchar en este botón.

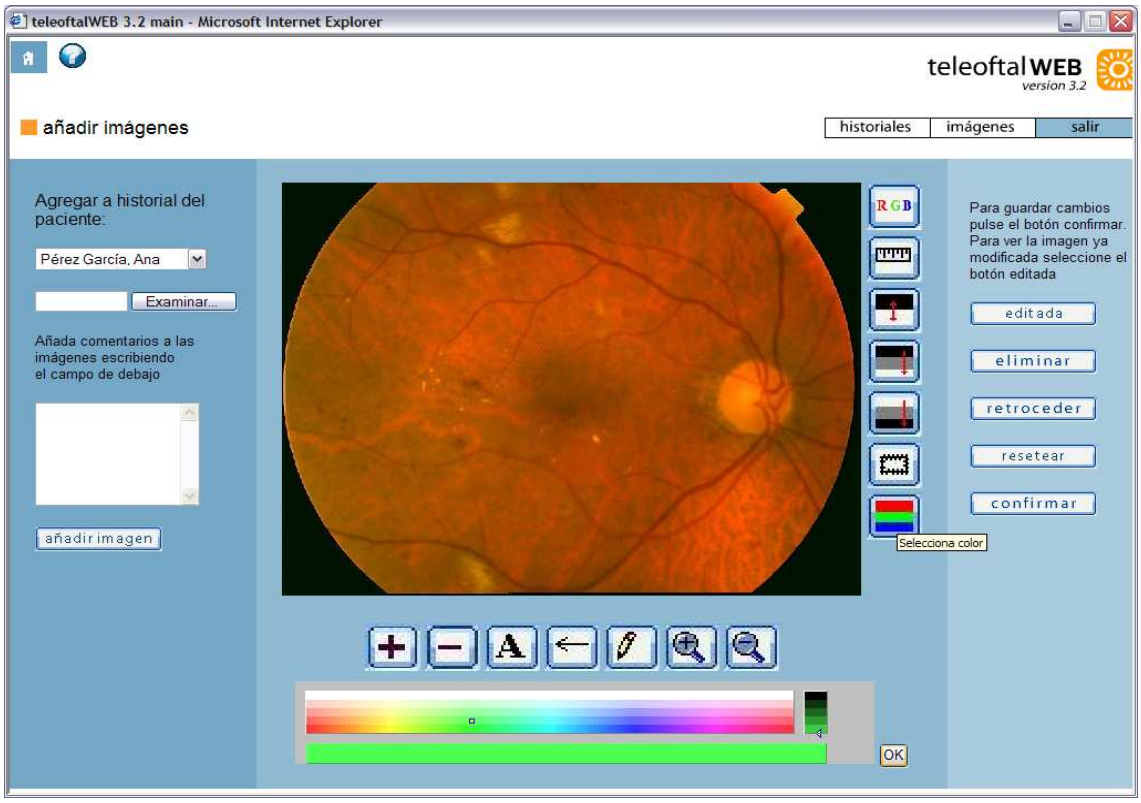


Figura 217. Selección de color. Fuente: propia.

Una vez explicados estos botones pasemos a los que se encuentran en la parte inferior de la pantalla.

En primer lugar encontramos los botones (+ ancho) y (- Ancho), que según pinchemos sobre ellos cambiaremos el grosor de los trazos y textos dibujados sobre la imagen así con el primer botón ensancharemos el grosor y con el segundo lo haremos más fino.

El botón (texto) se encarga de proporcionarnos la herramienta texto, que nos permitirá dibujar anotaciones en la imagen tras pulsar el botón "enter" del teclado, hasta que no se pulse este botón los cambios no tendrán lugar en la imagen.

El siguiente botón, (flecha) nos permite dibujar flechas sobre nuestra imagen, para ello simplemente debemos pinchar sobre un punto de la imagen y arrastrar hasta donde queremos que nuestra flecha apunte.

Seguidamente tenemos el botón (lápiz) que nos proporciona la herramienta adecuada para realizar dibujo libre sobre nuestra imagen permitiéndonos hacer cualquier tipo de anotación sobre ella.

Por último nos encontramos con los botones *(zoom in)* y *(zoom out)* que son los encargados de proporcionarnos una ampliación o reducción de la imagen que estemos viendo. Para ello simplemente tenemos que pinchar en la parte de la imagen que queremos ampliar o reducir. Se recomienda realizar primeramente la exploración necesaria con el zoom in y si se quiere volver a la imagen original pinchar sobre el botón "reset" en lugar de salir con el botón de zoom out, ya que este distorsiona algo la imagen.

Buscar imágenes

Al igual que con las fichas de las revisiones, también es posible realizar búsquedas de las imágenes asociadas a un historial, para lo cual tenemos cuatro posibles criterios de búsqueda: por historial, por identificador, por fechas de creación, y por diagnóstico o comentarios. Así una vez incluidos los términos de búsqueda se pulsaría el botón "buscar" y aparecería una pantalla como la que se muestra a continuación. En la parte de la derecha se pueden visualizar los detalles de una imagen o ir a la pantalla de edición de la misma como se vio en el apartado anterior.

Figura 218. Ventana con los resultados de una búsqueda de las imágenes de un paciente. Fuente: propia.

Fuente: propia.

Apéndice C

Glosario de abreviaturas y acrónimos

ACID: Atomicidad, consistencia, aislamiento y durabilidad.

ACR-NEMA: Colegio Americano de Radiología – Asociación de Manufacturadores Eléctricos Nacional *(American Collage of Radiology - The Nacional Electrical Manufacturers Association).*

AENOR: Asociación Española de Normalización y Certificación.

ANSI: Instituto de Estándares Nacional Americano *(American National Standards Institute).*

AP: Atención primaria.

API: Interfaz de Programación de Aplicación *(Application Programming Interface).*

ASP: Páginas de Servidor Activo *(Active Server Pages).*

ATA: Asociación de Telemedicina Americana *(American Telemedicine Association).*

AV: Agudeza Visual.

BD: Base de Datos.

BDOO: Bases de Datos Orientadas a Objetos.

BDU: Bases de Datos Unificada.

BMP: Mapa de Bits *(BitMaP).*

BPA: Biomicroscopía de Polo Anterior.

CCOW: Grupo de trabajo de Objeto en el Contexto Clínico *(Clinical Context Object Workgroup).*

CDA: Arquitectura de Documento Clínico *(Clinical Document Architecture).*

CEN: Comité Europeo para la Normalización *(European Committee for Standardization*).

CGI: Interfaz de *gateway* común *(Common Gateway Interface).*

CHDATA: Datos de carácter *(Character Data).*

CIAS: Servicio de acceso a información multimedia.

COAS: Servicio de acceso a observaciones clínicas.

COCIR: Comité de Coordinación Europeo para la Industria Radiológica y Electrotécnica.

CORBA: *Common Object Request Broker Architecture.*

CPR: Historial de Paciente basado en Ordenador *(Computer-based Patient Record).*

CPT: Terminología procedimental actual *(Current Procedural Terminology).*

CWE: Código con Extensiones *(Coding With Extensions).*

DA: Fecha *(Date).*

DBA: Administrador de Base de Datos *(DataBase Administrator).*

DDL: Lenguaje de Definición de Datos *(Data Definition Language).*

DHCP: Protocolo de Configuración de *Host* Dinámico *(Dynamic Host Configuration Protocol).*

DHTML: Lenguaje de Marcado de Hipertexto Dinámico *(Dynamic HyperText Markup Language).*

DICOM: Imagen Digital y Comunicaciones en Medicina *(Digital Imaging and Comunicacions in Medicine).*

DIMSE: Elemento de Servicio de Mensaje DICOM *(DICOM Message Service Element).*

DM: Diabetes Mellitus.

DML: Lenguaje de Manipulación de Datos *(Data Manipulation Language).*

DNI: Documento Nacional de Identidad.

DOM: Modelo de Objeto de Documento *(Document Object Model).*

DTD: Definición de Tipo de Documento *(Document Type Definition).*

EHR: Historial Clínico Electrónico *(Electronic Health Record).*

EMR: Historial Médico Electrónico *(Electronic Medical Record).*

EN: Estándar Europeo *(European Standard).*

ENV: Preestándar Europeo *(European Prestandard).*

E-R: Entidad-Relación.

ERP: Planificación de Recursos Empresariales *(Enterprise Resource Planning).*

ETSIT: Escuela Técnica Superior de Ingenieros de Telecomunicación.

FO: Objetos de Formateo *(Formatting Objects).*

FTP: Protocolo de Transferencia de Fichero *(File Transfer Protocol).*

GEHR: *Good European Health Record.*

GNU: GNU No es Unix.

HC: Historial Clínico.

HCE: Historial Clínico Electrónico.

HIPAA: Ley de Portabilidad y Responsabilidad del Seguro Médico *(Health Insurance Portability and Accountability Act).*

HIS: Sistema de Información Hospitalario *(Hospital Information System).*

HL7: Nivel de Salud 7 *(Health Level 7).*

HSDC: Historia de Salud Digital del Ciudadano.

HTML: Lenguaje de Marcado de Hipertexto *(HyperText Markup Language).*

HTTP: Protocolo de Transferencia de Hipertexto *(HyperText Transfer Protocol).*

HTTPS: Protocolo de Transferencia de Hipertexto sobre SSL *(HyperText Transfer Protocol over SSL).*

ICD: Desfibrilador Cardiaco Implantable *(Implantable cardiac defibrillator).*

ICNP: Clasificación Internacional para la Práctica de Enfermería *(International Classification for Nursing Practice)*.

IDE: Ambiente de Desarrollo Integrado *(Integrated Development Environment).*

IE: *Internet Explorer.*

IEEE: Instituto de Ingenieros Eléctricos y Electrónicos *(Institute of Electrical and Electronics Engineers).*

IEEET-ITB: Revista *IEEE Transactions sobre Tecnologías de la Información en Biomedicina (IEEE Transactions on Information Technology in Biomedicine).*

IHE: Integrando la Empresa de Salud *(Integrating the Healthcare Enterprise).*

IOBA: Instituto Universitario de Oftalmobiología Aplicada.

IOD: Definición de Objeto de Información *(Information Object Definition).*

IOM: Instituto de Medicina *(Institute of Medicine).*

IP: Protocolo de Internet *(Internet Protocol).*

ISCL: Capa de Comunicación Segura Integrada *(Integrated Secure Communication Layer).*

ISDN: Red Digital de Servicios Integrados *(Integrated Services Digital Network).*

ISO: Organización de Estándares Internacional *(International Standards Organization).*

ITU: Unión Internacional de Telecomunicaciones *(International Telecommunication Union).*

J2EE: Plataforma Java 2, Edición Empresa *(Java 2 Platform, Enterprise Edition).*

JAMIA: Revista de la Asociación Médica Informática Americana *(Journal of the American Medical Informatics Association).*

JDBC: Conectivad de Base de Datos Java *(Java Database Connectivity).*

JDK: Kit de Desarrollo Java *(Java SE Development Kit).*

JIRA: Asociación de Industrias Japonesas de Sistemas Radiológicos *(Japan Industries Association of Radiological Systems).*

JNDI: Interfaz de Nombrado y Directorio Java *(Java Naming and Directory Interface).*

JPEG: *Joint Photographic Experts Group.*

JSP: Páginas de Servidor Java *(Java Server Pages).*

JTC: *Joint Technical Committee.*

JVM: Máquina Virtual Java *(Java Virtual Machine).*

LDAP: Protocolo de Acceso de Directorio Ordenado *(Lightweight Directory Access Protocol).*

LDL: Lipoproteína de baja densidad *(Low-density lipoprotein)*.

LOINC: Identificadores de Observaciones Lógicos, Nombres, Códigos *(Logical Observations Identifiers, Names, Codes).*

MEDIS-DC: Centro de Desarrollo de Sistema de Información Médica *(Medical Information System Development Center).*

MF: *Mozilla Firefox*

MIME: Extensiones de Correo Internet Multipropósito *(Multipurpose Internet Mail Extensions).*

N3: Red Nacional Nueva *(New National Network).*

NASA: Administración Espacial y Aeronaútica Nacional *(Nacional Aeronautics and Space Administration).*

NFS: Sistema de Archivos de Red *(Network File System).*

NHS: Servicio Nacional de Salud *(National Health Service).*

NIU: Unidad de Interfaz de Red *(Network Interface Unit).*

NPfIT: Programa Nacional de Tecnología de la Información *(National Programme for Information Technology).*

OCT: Tomografía Coherente Óptica *(Optical Coherence Tomography).*

OMA: Arquitectura de Gestión de Objeto *(Object Management Architecture).*

OMG: Grupo de Gestión de Objeto *(Object Management Group).*

OMS: Organización Mundial de la Salud.

OOP: Programación Orientada a Objetos *(Object-Oriented Programming).*

OPTI: Observatorio de Prospectiva Tecnológica Industrial.

OSI: Interconexión de Sistemas Abiertos *(Open Systems Interconnection).*

PACS: Sistemas de Comunicación y Archivo de Imágenes *(Picture Archiving and Communication Systems).*

PatCIS: Sistema de Información Clínica para el Paciente *(Patient Clinicial Information System)*.

PDF: Formato Portátil de Documento *(Portable Document Format).*

PGA: Área Global de Programa *(Program Global Area).*

PIDS: Servicio de Identificación de Paciente.

PN: Nombre de Persona *(Person Name)*.

POCT: Pruebas acerca del Paciente *(Point of Care Testing)*.

PRA: Arquitectura de Historial de Paciente *(Patient Record Architecture)*.

RAC: Clúster de Aplicación Real *(Real Application Cluster)*.

RAID: Conjunto de Discos Redundantes Independientes *(Redundant Array of Independent/Inexpensive Disks)*.

RD: Retinopatía Diabética.

RDBMS: Sistema Gestor de Bases de Datos Relacional *(Relational DataBase Management System)*.

RDNP: Retinopatía Diabética no Proliferativa.

RDP: Retinopatía Diabética Proliferativa.

RGB: Rojo, Verde, Azul *(Red, Green, Blue)*.

RIM: Modelo de Información de Referencia *(Reference Information Model)*.

RIS: Sistemas de Información en Radiología *(Radiology Information Systems)*.

SEIS: Sociedad Española de la Información de la Salud.

SGA: Área Global de Sistema *(System Global Area)*.

SGBD: Sistema Gestor de Bases de Datos.

SGML: Lenguaje de Marcado Generalizado Estándar *(Standard Generalized Markup Language)*.

SI: Sociedad de la Información.

SIP: Protocolo de Iniciación de Sesión *(Session Initiation Protocol)*.

SMP: Protocolo de Gestión Simple *(Simple Management Protocol).*

SMTP: Protocolo Simple de Transferencia de Correo *(Simple Mail Transfer Protocol).*

SNOMED: Nomenclatura Sistemática de Medicina *(Systematized Nomenclature of Medicine).*

SOA: Arquitectura Orientada a Servicio *(Service Oriented Architecture).*

SOP: Par Servicio-Objeto *(Service-Object Pair).*

SQL: Lenguaje de Consulta Estructurado *(Structured Query Language).*

SR: Informe Estructurado *(Structured Report).*

SSI: *Server-Side Includes.*

SSL: Capa de *Sockets* Segura *(Secure Sockets Layer).*

SUS: Escala de Usabilidad del Sistema *(System Usability Scale).*

TAC: Tomografía Axial Computarizada.

TC: Comité Técnico *(Technical Committee).*

TCP/IP: Protocolo de Transmisión de Control / Protocolo de Internet *(Transmission Control Protocol / Internet Protocol).*

TDT: Televisión Digital Terrestre *(Terrestrial Digital Television).*

TIC: Tecnologías de la Información y las Comunicaciones.

TIFF: Formato de archivo de imágenes con etiquetas *(Tagged Image File Format).*

TLS: Seguridad de la Capa de Transporte *(Transport Layer Security).*

UID: Instancia Identificadora Única.

UML: Lenguaje de Modelado Unificado *(Unified Modelling Language).*

URL: Localizador de Recursos Uniforme *(Uniform Resource Locator).*

USB: Bus de Serie Universal *(Universal Serie Bus).*

VIH: Virus de la Inmunodeficiencia Humana.

W3C: Consorcio *World Wide Web.*

WG: Grupo de Trabajo *(Working Group).*

WWW: *World Wide Web.*

xDSL: Líneas de Subscripción Digital *(Digital Subscriber Line).*

XML: Lenguaje de Marcado Extensible *(Extensible Markup Language).*

XSL: Lenguaje de hojas de estilo extensible *(Extensible Stylesheet Language).*

XSL-FO: Objetos de Formateo de Lenguaje de hojas de estilo extensible *(eXtensible Stylesheet Language Formatting Objects).*

XSLT: Transformaciones del Lenguaje de hojas de estilo extensible *(Extensible Stylesheet Language Transformations).*

Bibliografía

Alberta 2008. *Página Web del hospital de Alberta, en el que se ha desarrollado un sistema de HCE.* [Online]. [Accediendo Agosto 2008]. Disponible desde WWW: http://www.albertawellnet.org/content.asp?catid=48&rootid=4

Alschuler, L & Boyer, S 2000, 'XML and the HL7 Document Patient Record Architecture', *Proceedings of the HL7 Spring Working Group Meeting*, Cleveland, OH, USA.

Alschuler, L & Dolin, R 2000, *Clinical Document Architecture Framework Version 1.0 DRAFT.*

American Diabetes Association 2004, 'Retinopathy in diabetes', *Diabetes Care*, vol. 27, suppl. 1, pp. 84–87.

Amouh, T, Gemo, M, Macq, B, Vanderdonckt, J, Gariani, AWE, Reynaert, MS, Stamakis, L & Thys, F 2005, 'Versatile Clinical Information System Design for Emergency Departments'*, IEEE Transactions on Information Technology in Biomedicine*, vol. 9, n.º 2, pp. 174-183.

ANSI/HL7 2007, *The HL7 EHR System Functional Model, Release 1 R1-2007*.

Bai, J, Zhang J & Dai, B 1998, 'Design and Development of an Interactive Medical Teleconsultation System over the World Wide Web', *IEEE Transactions on Information Technology in Biomedicine,* vol. 2, n.º 2, pp. 74–79.

Beaumnot 2008. *Patients in Ireland Benefit from Electronic Health Record System.* [Online]. [Accediendo Agosto 2008]. Disponible desde WWW: http://www.bea.com/content/news_events/white_papers/BEA_Beaumont_Hospital_cs.pdf

Beaumont Hospital 2004, *Patients in Ireland Benefit from Electronic Health Record System*, Case of Study.

Becker, MY & Sewell, P 2004, 'Cassandra: flexible trust management, applied to electronic health records'. *17th IEEE Proceedings Computer Security Foundations Workshop,* pp. 139-154.

Bennett 1998, *Medical Information Protection Act of 1998.* 105th US Congress. [Online]. [Accediendo Abril 2008]. Disponible desde WWW: http://thomas.loc.gov

Blobel, B & Holena, M 1998, 'CORBA: security services for health information system', *International Journal of Medical Informatics,* vol. 52, n.º 1, pp. 29-37.

Blomdahl, S, Calissendorff, B & Jacobsson, U 2002, 'Patient-focused urban tele-ophthalmology services', *Journal of Telemedicine and Telecare*, vol. 8, suppl. 2, pp. 43-44.

Böhme, T. & Rahm, E. (2003). Multi-User Evaluation of XML Data Management Systems with XMach-1. Lecture Notes in Computer Science (LNCS) 2590, pp. 148-159, Springer.

Brelstaff, G, Moehrs, S, Anedda, P, Tuveri, M & Zanetti, G 2001, 'Internet Patient Records: new techniques'. *J Med Internet Res,* vol. 3, nº 1. [Online]. [Accediendo Septiembre 2008]. Disponible desde WWW: http://www.jmir.org/2001/1/e8

Burbano, DJ 2006, *Análisis comparative de bases de datos de código abierto vs. código cerrado,* Quito, Ecuador.

Burns, F 1998, *Information for Health. An Information Strategy for the Modern National Health Service 1998-2005. A National Strategy for local implementation*. Department of Health Publications. [Online]. [Accediendo Septiembre 2008]. Disponible desde WWW: http://www.nhsia.nhs.uk/def/pages/info4health/contents.asp

Cabrera, G 2001, *Sistemas Gestores de Bases de Datos,* Paraninfo.

Calidad y riesgo. La historia clínica online: La red Sinapsis. Datos actualizados a Julio de 2009. Disponible en: http://www.calidadyriesgo.es/calidadasistencial/38-red-sinapsis-historia-clinica-online. Última visita 11 de Agosto de 2009.

Ceballos, FJ 2006, *Java 2: Lenguaje y Aplicaciones,* Editorial Ra-Ma.

CEN 2008. [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: http://www.cen.eu/cenorm/homepage.htm

CEN/TC 251 Published Standard 2008. [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: http://www.cen.eu/CENORM/Sectors/TechnicalCommitteesWorkshops/CENTechnicalCommittees/Standards.asp?param=6232&title=CEN%2FTC+251

Chen, R, Enberg, G & Klein, G 2007, 'Julius--a template based supplementary electronic health record system', *BMC Medical Informatics and Decision Making*, pp. 7-10.

Chew, SJ, Cheng, HM, Lam, DSC, Cheng, ACK, Leung, ATS, Chua, JKH, Yu, CP, Balakrishnan, V & Chan WK 1998, 'OphthWeb-cost-effective telemedicine for ophthalmology', *HKMJ,* vol. 4. pp. 300-304.

Cho, I & Park H 2003, 'Development and evaluation of a terminology-based electronic nursing record system'. *Journal of Biomedical Informatics*, vol. 36, nº 4, pp. 304-312.

Christiaens, J, Van de Walle, R, Lemahieu, I & Taeymans, Y 1999, 'ECG-based selection of angiographic images using DICOM facilities', *Computers in Cardiology*, pp. 447-454.

Chronaki, CE, Lelis, P, Demou, C, Tsiknakis, M & Orphanoudakis, SC 2001, 'An HL7/CDA framework for the design and deployment of telemedicine services*', Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 4, pp. 3504-3507.

Cimino, JJ, Patel, VL & Kushniruk, AW ,'The patient clinical information system (PatCIS): technical solutions for and experience with giving patients access to their electronic medical records'. *International Journal of Medical Informatics,* vol. 68, pp. 113-127.

Corwin, J, Silberschatz, A, Miller, P & Marenco, L 2007, 'Dynamic Tables: An Architecture for Managing Evolving, Heterogeneous Biomedical Data in Relational Database Management Systems', *J. Am. Med. Inform. Assoc*., vol. 14, n.º 1, pp. 86-93.

Croteau, AM & Vieru, D 2002, 'Telemedicine adoption by different groups of physicians', *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, pp. 1985–1993.

Cruz-Correia, RJ, Vieira-Marques, PM, Ferreira, AM, Almeida, FC, Wyatt, JC & Costa-Pereira, AM 2007, 'Reviewing the integration of patient data: how systems are evolving in practice to meet patient needs', *BMC Medical Informatics and Decision Making* 2007, vol. 7, n.º 14. [Online]. [Accediendo Octubre 2008]. Disponible desde WWW: http://www.biomedcentral.com/1472-6947/7/14

Cubero, A & Luna, S 2003, *Programación Orientada a Objetos. Servlets y JSP*. Departamento de Informática y Automática. Universidad de Valladolid.

D'Ambrogio, A. & Iazeolla, G. (2003). Steps towards the automatic production of performance models of Web applications. Computer Networks, 41, 29-39.

DbXML 2008. Página Web oficial de la base de datos dbXML. [Online]. [Accediendo Junio 2008]. Disponible desde WWW: http://www.dbxml.com

De Clerq, PA, Hasman, A & Wolffenbuttel, BH 2001, 'Design of a consumer health record for supporting the patient-centered Management of chronic diseases'. *Stud Health Technol*

Inform, vol. 84, pp. 1445-1449.

De la Torre, I, Díaz, FJ, Antón, M, Díez, F, Sainz, B, López, M, Hornero, R, López, MI 2010, "Choosing the most efficient database for a Web-based system to store and exchange Ophthalmologic Health Records", Journal of Medical Systems. Aceptado

De la Torre, I, Hornero, R, López, M & López, MI 2008, 'Desarrollo e implantación de un Sistema de Historiales Clínicos Electrónicos estandarizados en Oftalmología', CASEIB 2008, *Libro de Actas del XXVI Congreso Anual de la Sociedad Española de Ingeniería Biomédica*, ISBN: 978–84–691–3640–9, Valladolid, España.

De la Torre, I, Hornero, R, López, M & López, MI 2008, 'HL7-CDA and DICOM standards in Electronic Health Records System in Ophthalmology', *Fourth IASTED International Conference on Telehealth and Assistive Technologies*, pp. 162-167, Baltimore, USA.

De la Torre, I, Hornero, R, López, M & López, MI 2008, 'Ophthalmologic Electronic Health Records using HL7 and DICOM standards', *Thirteenth Annual International Meeting & Trade Show*, Seattle, USA. *Telemedicine and e-Health Journal*, pp. 100.

De la Torre, I, Hornero, R, López, M & López, MI 2009, 'Ophthalmologic Electronic Health Records using HL7/CDA and DICOM: TeleOftalWeb', *International Conference on Health Informatics, HealthInf* 2009, Porto, Portugal.

De la Torre, I, Hornero, R, López, M, Poza, J & López, MI 2008, *Web-based Application to exchange Electronic Health Records and Medical Images in Ophthalmology,* Editors: Athina A. Lazakidou, Ph.D and Konstantinos M. Siassiakos, Ph.D, University of Piraeus, Greece, Libro: "Handbook of Research on Distributed Medical Informatics and E-Health", ISBN: 978-1-60566-002-8, pp. 152-164, Publisher: Medical Information Science Reference.

De la Torre, I, Hornero, R, López, M, Poza, J & López, MI 2009, 'Electronic Health Records System using HL7 and DICOM in Ophthalmology', Editors: Wayne Pease, University of Southern Queensland, Australia, Professor Malcolm Cooper, Ritsumeikan Asia Pacific University, Japan, Assoc. Professor Raj Gururajan, University of Southern Queensland, Australia, Libro: "Biomedical Knowledge Management: Infrastructures and Processes for E-Health Systems", ISBN: 1-60566-266-6

De la Torre, I, Hornero, R, López, M, Poza, J & López, MI 2009, 'Electronic Health Records in a teleophthalmologic application with Oracle 10g*',* Editors: Athina A. Lazakidou, Ph.D and Konstantinos M. Siassiakos, Ph.D, University of Piraeus, Greece, Libro:

"Handbook of Research on Biocomputation and Biomedical Informatics", . ISBN: 978-1-60566-768-3.

De la Torre, I, Hornero, R, Poza, J, López, M & López, MI 2007, 'TeleOftalWeb 3.2: A Web-based Application to Exchange Electronic Health Records in Ophthalmology', *The Third IASTED International Conference on Telehealth*, pp. 564-004, ISBN: 978-0-88986-667-6, Montreal, Canadá.

DebugBar 2008, Página *Web* de la aplicación DebugBar. [Online]. [Accediendo Junio 2008]. Disponible desde WWW: http://www.debugbar.com/?langage=en

Deftereos, S, Lambrinoudakis, C, Andriopoulos, P, Farmakis, D & Aessopos A 2001, 'Java-based Electronic Healthcare Record *Software* for Beta-thalassaemia', vol. 3, nº 4. [Online]. [Accediendo Septiembre 2008]. Disponible desde WWW: http://www.jmir.org/2001/4/e33

DICOM Standards Committee 2004, *Suplemento 91, Ophthalmic Photography SOP Classes Standard.*

Digital Imaging and Communications in Medicine (DICOM) 2008. Partes del Estándar. [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: ftp://medical.nema.org/medical/dicom/2008

DISA S.L. Oftalclinic. Disponible en*:* http://tecnomedicacanaria.com/software/folleto.doc. Última visita el 17 de Agosto de 2009.

Dolin, R, Alschuler, L, Boyer, S & Beebe, C 2000, 'An Update on HL7's XML-based Document Representation Standards', *Proc of the AMIA 2000 Annual Symposium*, Los Angeles, California.

Elmasri, R & Navathe, S 2007, *Fundamentos de Sistemas de Bases de Datos*, Quinta Edición, Pearson Addison-Wesley.

EXist 2008. Página Web oficial de la base de datos eXist 2008. [Online]. [Accediendo Junio 2008]. Disponible desde WWW: http://exist.sourceforge.net

Eyers, DM, Bacon, J & Moody, K 2006, 'OASIS role-based access control for electronic health records'. *IEEE Proceedings Software*, vol. 153, nº 1, pp. 16-23.

Falagán, J 2001, 'La condición de médico no supone libre acceso a la historia clínica', Rev Galega Act Sanit, vol. º, n.º 1, pp. 6-9.

Ferreiro, S 2003, *Historia Clínica y exploración en oftalmología*, Oftalmología en Atención Primaria, Formación Alcalá, pp. 69-89.

Ferrer-Roca, O 2001, *La Telemedicina: Situación Actual y Perspectivas*, Fundación Retevisión.

Firebug 2008, Página *Web* de complementos para Firefox. [Online]. [Accediendo Junio 2008]. Disponible desde WWW: https://addons.mozilla.org/es-ES/firefox/addon/1843

Froufe, A 1997. *Tutorial de Java.* [Online]. [Accediendo Junio 2008]. Disponible desde WWW: http://www.ulpgc.es/otros/tutoriales/java/Intro/carac.html

Garbayo, JA, Sanz, J, Carnicero, J & Sánchez, C 2003, *La seguridad, confidencialidad y disponibilidad de la información clínica*, V Informe SEIS.

Gilfillan, I 2003, *MySQL,* Anaya Multimedia.

Ginsburg, M 2007, 'Pediatric Electronic Health Record Interface Design: The PedOne System', *Proceedings of the 40th Hawaii International Conference on System Sciences*, pp. 1-10.

Gobierno de Canarias. Comunicación: Sanidad. Disponible en: http://www2.gobiernodecanarias.org/sanidad/gc/noticias/2009/20090710BalanceDosSanidad.pdf. Última visita el 17 de Agosto de 2009.

González, C 2002, 'Integrated Telemedicine System in Andalusia', *IEEE 2002 28th Annual Conference of the Industrial Electronics Society*, vol. 4, pp. 3398-3402.

Gordon, C 2005, *Oracle Database 10g Release 2: A revolution in database technology*, White paper.

Grimson, J, Grimson, W, Berry, D, Stephens, G, Felton, E, Kalra, D, Toussaint, P & Weier, OW 1998, 'A CORBA-based integration of distributed electronic healthcarerecords using the Synapses approach', *IEEE Transactions on Information Technology in Biomedicine,* vol. 2, n.º 3, pp. 124-138.

Grupo Kybele 2004, *Bases de datos XML. Bases de datos avanzadas*, Universidad Rey Juan Carlos, Madrid, España.

Halamka, JD, Osterland, C & Safran, C 1999, 'CareWeb™, a Web-based medical record for an integrated health care delivery system', *International Journal of Medical Informatics*, vol. 54, pp. 1–8.

Hall, G, Hennessy, M, Barton, J & Coroneo, M 2005, 'Teleophthalmology-Assisted Corneal Foreign Body Removal in a Rural Hospital', *Telemedicine and e-Health*, vol. 11, n.º 1, pp. 79-83.

Hannan, TJ *et ál.* 2000, 'The Mosoriot medical record system: design and initial implementation of an outpatient electronic record system in rural Kenya', *International Journal of Medical Informatics,* vol. 60, pp. 21-28.

Hansen, IB 2006, 'CipherMe: personal Electronic Health Records in the hands of patients-owners', *Proceedings of the 1st Distributed Diagnosis and Home Healthcare (D2H2) Conference,* pp. 148-151.

Hanzlicek, P, Spidlen, J, Heroutova, H & Nagy, M 2005, 'User interface of MUDR electronic health record'. *International Journal of Medical Informatics,* vol. 74, pp. 221-227.

Health Level Seven (HL7) 2008. [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: http://www.hl7.org

Hijar, CA, Moreno, J & Jiménez, L 1998, 'Prevalencia y riesgo de pérdida de visión por retinopatia diabética en Soria', *Archivos de la Sociedad Española de Oftalmología*, pp. 103–107.

Hilbel, T, Reiter, MA, Brockmeier, K, Kuecherer, HF, Haass, M 2000, 'Advantages of a cardiac DICOM network server/writer for viewing and permanent CD-R archiving of cardiovascular X-ray angiography images', *Computers in Cardiology*, pp. 649-652.

Holle, R & Zahlmann, G 1999, 'Evaluation of Telemedical Services'*, IEEE Transactions on Information Technology in Biomedicine,* vol. 3, n.º 2, pp. 84-91.

Horsch, A & Balbach, T 1999, 'Telemedical Information Systems', *IEEE Transactions on Information Technology in Biomedicine,* vol. 3, n.º 3, pp. 166-175.

Huang, E.W. & Liou, D. (2007). Performance Analysis of a Medical Record Exchanges Model. IEEE Transactions on Information Technology in Biomedicine, vol. 11, n. 2, 153-160.

IDG Communications. La historia clínica electrónica estará funcionando en dos años en toda España. Disponible en: http://www.idg.es/computerworld/articulo.asp?id=194483. Última visita el 17 de Agosto de 2009.

IEEE 2008. [Online]. [Accediendo Agosto 2008]. Disponible desde WWW:

http://ieeexplore.ieee.org/Xplore/conhome.jsp

Inamura, K, Kondoh, H & Takeda, H 1996, 'Development and operation of PACS/teleradiology in Japan', *IEEE Communications Magazine*, vol. 3, n.º 7, pp. 46-51.

Institute of Medicine (IOM) *of the National Academies* 2008. *Página Web del IOM of the National Academies, con informacion sobre Sistemas de HCE.* [Online]. [Accediendo Agosto 2008]. Disponible desde WWW: http://iom.edu/report.asp?id=14391

Integrating the Healthcare Enterprise (IHE) 2008. [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: http://www.ihe.net

International Organization for Standardization (ISO) 2008. [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: http://www.iso.org

ISI *Web of Knowledge* 2008. [Online]. [Accediendo Agosto 2008]. Disponible desde WWW: http://isiknowledge.com

Iyer, V, Hanes, E, Wright, B & Pfaeffle, T 2007, *Oracle Database JDBC Developer's Guide and Reference, 10g Release 2 (10.2)*, guía de JDBC de Oracle 10g.

James, A, Wilcox, Y & Naguib, RNG 2001, 'A Telematic System for Oncology Based on Electronic Health and Patient Records', *IEEE Transactions on Information Technology in Biomedicine,* vol. 2, n.º 1, pp. 16-17.

Junta de Andalucía 2008. [Online]. [Accediendo Septiembre 2008]. Disponible desde WWW: http://www.juntadeandalucia.es/salud/principal/noticias.asp?codcontenido=675

Karagiannis, GE, Stamatopoulos, VG, Rigby, M, Kotis, T, Negroni, E, Munoz, A & Mathes, I 2007, 'Web-based personal health records: the personal electronic health record (pEHR) multicentred trial'. *Journal of Telemedicine and Telecare,* vol. 13, suppl. 1, vol. 3, pp. 32-34.

Kawasaki, S, Ito, S, Satoh, S, Mori, Y, Saito, T, Fukushima, H, Kato, S & Sekihara, H 2003, 'Use of Telemedicine in Periodic Screening of Diabetic Retinopathy', *Telemedicine Journal and e-Health*, vol. 9, n.º 3, pp. 235-239.

Kim, EH 2006, 'Web-based Personal-Centered Electronic Health Record for Elderly Population', Proceedings *of the 1st Distributed Diagnosis and Home Healthcare (D2H2) Conference,* pp. 144-147.

Kim, HS, Tran, T & Cho, H 2006, 'A Clinical Document Architecture (CDA) to Generate Clinical Documents within a Hospital Information System for E-Healthcare Services', *The Sixth IEEE International Conference on Computer and Information Technology*, pp. 254-254.

Klein, R 2003, 'Screening interval for retinopathy in type 2 diabetes', *Lancet*, vol. 361, pp 190–191.

Komiya, R 2005, 'A proposal for telemedicine reference model for future standardization', *Proceedings of 7th International Workshop on Enterprise networking and Computing in Healthcare Industry*, pp. 224-228.

Korth, HF, Silberschatz, A & Sudarshan, S 2002, *Fundamentos de Bases de Datos*, Mc-Graw Hill, 4ª edición.

Kroenke, D 2000, *Database Processing Fundamentals, Design and Implementation.* Prentice Hall International, New Jersey.

Kuzmak, PM & Dayhoff, RE 2000, 'The use of digital imaging and communications in medicine (DICOM) in the integration of imaging into the electronic patient record at the Department of Veterans Affairs', *Digital Imaging*, vol. 13, n.º 2, pp. 133-137.

Kuzmak, PM & Dayhoff, RE 2003, 'Experience with DICOM for the clinical specialities in the healthcare enterprise', *Proceedings of the SPIE*, vol. 5033, pp. 18-29.

Kwak YS, 2005, 'International standards for building Electronic Health Record (EHR)', *Proceedings of 7th International Workshop on Enterprise networking and Computing in Healthcare Industry, Healthcom*, pp. 18-23.

Ledesma, MC & Garavís, JL 2004, *Aplicaciones de las Tecnologías de Información y Comunicación al ámbito de la Salud en Tierra de Peñaranda*.

Liou DM, Huang, EW, Chen, TT & Hsiao, SH 2000, 'Design and implementation of a Web-based HL7 validation system', *IEEE EMBS International Conference on Information Technology Applications in Biomedicine*, pp. 347-352.

Loney, K 2004, *Oracle Database 10g: The complete referente (Series: Osborne Oracle Press S.),* Editorial McGraw-Hill.

Lorenzo, JA 2004, *Manual de iniciación a Java*, Grupo Universitario de Informática, Universidad de Valladolid, España.

Los, R, van Ginneken, AM, de Wilde, M & van der Lei, J 2004, 'OpenSDE: Row Modeling Applied to Generic Structured Data Entry', *Journal of the American Medical Informatics Association,* vol. 11, pp. 162-165.

Magdalena, JR, Soria, E, Serrano, A, Calpe, J, Guerrero, J & Martínez, M 1999, 'NEMESIS: a new telemedicine approach to cardiologic *software*, *Computers in Cardiology*, pp. 37-40.

Marcheschi, P, Mazzarisi, A, Dalmiani, S & Benassi, A 2004, 'HL7 clinical document architecture to share cardiological images and structured data in next generation', *Computers in Cardiology*, pp. 617-620.

Marcheschi, P, Positano, V, Ferdeghini, EM, Mazzarisi, A & Benassi, A 2003, 'An open source based application for integration and sharing of multi-modal cardiac image data in a heterogeneous environment', *Computers in Cardiology*, pp. 367-370.

Margolis, A 2002, *Confidencialidad de la Información Clínica en Formato Electrónico. Algunas Reflexiones*. Disponible desde WWW: http://www.suis.org.uy/infosuis/temas/temas_05-1.shtml

Medline 2008. [Online]. [Accediendo Agosto 2008]. Disponible desde WWW: http://isiknowledge.com

Meier, W. (2003). eXist: An Open Source Native XML Database, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2593(1), 169-183.

Merelo, JJ 2004, *Programando con Java*, Manual de programación de Java.

Monteagudo, JL 2001, *El Marco de Desarrollo de la e-Salud en España,* Instituto de Salud Carlos III, Madrid, España.

Monteagudo, JL 2003, *La estandarización en Tecnologías de la Información y las Comunicaciones para Sanidad, Instituto de Salud Carlos III*, Madrid, España.

MySQL 2008. Página Web oficial de la base de datos MySQL 2008. [Online]. [Accediendo Junio 2008]. Disponible desde WWW: http://www.mysql.com

National Electrical Manufacturers Association (NEMA) 2008, *Digital Imaging and Communications in Medicine (DICOM). Part 1: Introduction and Overview.* [Online]. [Accediendo Mayo 2008]. Disponible desde WWW:

ftp://medical.nema.org/medical/dicom/2008/08_01pu.doc

National Electrical Manufacturers Association (NEMA) 2008, *Digital Imaging and Communications in Medicine (DICOM). Part 2: Conformance.* [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: ftp://medical.nema.org/medical/dicom/2008/08_02pu.doc

National Electrical Manufacturers Association (NEMA) 2008, *Digital Imaging and Communications in Medicine (DICOM). Part 3: Information Object Definitions.* [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: ftp://medical.nema.org/medical/dicom/2008/08_03pu.doc

National Electrical Manufacturers Association (NEMA) 2008, *Digital Imaging and Communications in Medicine (DICOM). Part 4*: *Service Class Specifications.* [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: ftp://medical.nema.org/medical/dicom/2008/08_04pu.doc

National Electrical Manufacturers Association (NEMA) 2008, *Digital Imaging and Communications in Medicine (DICOM). Part 5: Data Structures and Encoding*. [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: ftp://medical.nema.org/medical/dicom/2008/08_05pu.doc

National Electrical Manufacturers Association (NEMA) 2008, *Digital Imaging and Communications in Medicine (DICOM). Part 6: Data Dictionary.* [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: ftp://medical.nema.org/medical/dicom/2008/08_06pu.doc

National Electrical Manufacturers Association (NEMA) 2008, *Digital Imaging and Communications in Medicine (DICOM). Part 7: Message Exchange.* [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: ftp://medical.nema.org/medical/dicom/2008/08_07pu.doc

National Electrical Manufacturers Association (NEMA) 2008, *Digital Imaging and Communications in Medicine (DICOM). Part 8: Network Communication Support for Message Exchange.* [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: ftp://medical.nema.org/medical/dicom/2008/08_08pu.doc

National Electrical Manufacturers Association (NEMA) 2008, *Digital Imaging and Communications in Medicine (DICOM). Part 10: Media Storage and File Format for Media Interchange.* [Online]. [Accediendo Mayo 2008]. Disponible desde WWW:

ftp://medical.nema.org/medical/dicom/2008/08_10pu.doc

National Electrical Manufacturers Association (NEMA) 2008, *Digital Imaging and Communications in Medicine (DICOM). Part 11: Media Storage Application Profiles.* [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: ftp://medical.nema.org/medical/dicom/2008/08_11pu.doc

National Electrical Manufacturers Association (NEMA) 2008, *Digital Imaging and Communications in Medicine (DICOM). Part 12: Media Formats and Physical Media for Media Interchange.* [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: ftp://medical.nema.org/medical/dicom/2008/08_12pu.doc

National Electrical Manufacturers Association (NEMA) 2008, *Digital Imaging and Communications in Medicine (DICOM). Part 14: Grayscale Standard Display Function.* [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: ftp://medical.nema.org/medical/dicom/2008/08_14pu.doc

National Electrical Manufacturers Association (NEMA) 2008, *Digital Imaging and Communications in Medicine (DICOM). Part 15: Security and System Management Profiles.* [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: ftp://medical.nema.org/medical/dicom/2008/08_15pu.doc

National Electrical Manufacturers Association (NEMA) 2008, *Digital Imaging and Communications in Medicine (DICOM). Part 16: Content Mapping Resource.* [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: ftp://medical.nema.org/medical/dicom/2008/08_16pu.doc

National Electrical Manufacturers Association (NEMA) 2008, *Digital Imaging and Communications in Medicine (DICOM). Part 17: Explanatory Information.* [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: ftp://medical.nema.org/medical/dicom/2008/08_17pu.doc

National Electrical Manufacturers Association (NEMA) 2008, *Digital Imaging and Communications in Medicine (DICOM). Part 18: Web Access to DICOM Persistent Objects (WADO).* [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: ftp://medical.nema.org/medical/dicom/2008/08_18pu.doc

Noh, S.Y., Gadia, S.K. & Ma, S. (2008). An XML-based methodology for parametric temporal database model implementation. The Journal of Systems and *Software*, vol. 81, 929–

948.

Noseworthy, T 2004, ‘ Advancing electronic health records in Canada: why, how and key learnings of potential value to China’, *IDEAS Workshop on Medical Information Systems: The Digital Hospital*, pp. 129-132.

Ocularis 2008. *El proyecto divulgativo sobre la visión. Retinopatía Diabética: Conceptos.* [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: http://ocularis.es/blog/?p=179

OpenEHR 2008. [Online]. [Accediendo Mayo 2008]. Disponible desde WWW: http://www.openehr.org/home.html

Oracle 2008. Página Web oficial de la base de datos Oracle. [Online]. [Accediendo Junio 2008]. Disponible desde WWW: http://www.oracle.com

Otech 2008. *DICOM básico en español.* [Online]. [Accediendo Junio 2008]. Disponible desde WWW: http://www.otechimg.com/pdf/dicombasicsspanish_ch1.pdf

Paul, L, Pearlson, KE & McDaniel, RR 1999, ‘Assessing technological barriers to telemedicine: technology-management implications’, *IEEE Transactions on Engineering Management*, vol. 46, n.º 3, pp. 279–288.

Paul, P, Raman, R, Rani, P, Deshmukh, H & Sharma, T 2006, ‘Patient Satisfaction Levels During Teleophthalmology Consultation in Rural South India’, *Telemedicine and e-Health Journal*, vol. 12, n.º 5, pp. 571-578.

Pérez, C 2004, *Oracle 10g: Administración y Análisis de Bases de Datos,* Editorial Ra-Ma.

Perís, R 1974, *Integración y recuperación de la información*, Medicina Española, Archivo Central de historias clínicas, vol. 72, pp. 35–43.

Pruna, SS, Dixon, R & Harris, ND 1998, ‘BlackSea TeleDiab: Diabetes computer system with communication technology for black sea region’, *IEEE Transactions on Information Technology in Biomedicine* , vol. 2, n.º 3, pp. 193-196.

Schadow, G 2000, HL7 Version 3 Data Types Ballot, Draft 1, Revision 1.2. Disponible en: http://aurora.rg.iupui.edu/v3dt/V3DT-ballot-1-print.pdf

Segura, J, Roldán, C, Galera, J & Naval, J 2008, ‘What Do Spanish Physicians Believe and Expect about Telemedicine?. Results of a Delphi-Based Survey’, *Telemedicine and e-*

Health Journal, pp. 42–48.

SEIS, Sociedad Española de Informática de la Salud 2003, *De la historia clínica a la historia de la salud electrónica,* Pamplona, España.

Serrano MA *et ál.* 2007. Teleoftalmología, un arma eficaz. *Archivos de la Sociedad Canaria de Oftalmología*, ISSN 0211-2698, 18, 92-96.

SESCAM. Servicio de Salud de Castilla La mancha. Disponible en: http://sescam.jccm.es/web1/home.do?main=/ciudadanos/avancesMedTecn/ykonos.html. Última visita el 17 de Agosto de 2009.

Siika, AM *et ál.* 2005, 'An electronic medical record system for ambulatory care of HIV-infected patients in Kenya', *International Journal of Medical Informatics*, vol. 74, nº 5, pp. 345-355.

Silberschatz, A, Henry, K & Susarshan, S 2006, *Fundamentos de Bases de Datos,* Quinta Edición, Editorial McGraw-Hill.

Sistema Nacional de Salud 2009. Plan de Calidad para el Sistema Nacional de Salud. Disponible en: http://www.msc.es/organizacion/sns/planCalidadSNS/tic028.htm. Última visita el 17 de Agosto de 2009.

Spil, TAM & Katsma, CP 2007, 'Balancing Supply and Demand of an Electronic Health Record in the Netherlands', *40th Annual Hawaii International Conference on System Sciences,* pp. 134.

Staken, K 2001, *Introduction to Native XML Databases.* [Online]. [Accediendo Diciembre 2008]. Disponible desde WWW: http://www.xml.com

Taddei, A, Carpeggiani, C, Emdin, M., Balocchi, R, Dalmiani, S, Cecchetti, G, Macerata, A, Pierotti, D & Marchesi, C 1997, 'Development of an electronic medical record for patient care incardiology'. *Computers in Cardiology,* vol. 7, nº 10, pp. 641-644.

Tang, R, Morales, M, Giselle, R & Schiffman, J 2005, 'Telemedicine for eye care', *Journal of Telemedicine and Telecare*, vol. 11, n.º 8, pp. 391-396.

Taylor, D, Naguib, RNB, Amin, S, James, A & Boulton, S 2002, 'Combining Dental Epidemiological Relational Databases', *Proccedings of the Second Joint EMBS/BMES Conference,* Houston, TX, USA, pp. 2209-2210.

Tennant, M, Rudnisky, C, Hinz, B, MacDonald, I & Greve, M 2000, 'Tele-Ophthalmology via Stereoscopic Digital Imaging: A Pilot Project', *Telemedicine and e-Health Journal*, vol. 2, n.º 4, pp. 583-587.

The XML:DB Project. "XML:DB Database API Working Draft" 2008. [Online]. [Accediendo Junio 2008]. Disponible desde WWW: http://www.xmldb.org/xapi

Thiyagarajan, C & Clarke, M 2006, 'A Systematic Review of Technical Evaluation in Telemedicine Systems', *28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6320-6323.

Treins, M, Cure, O & Salzano, G 2006, 'On the Interest of Using HL7 CDA Release 2 for the Exchange of Annotated Medical Documents', *19th IEEE International Symposium on Computer-Based Medical Systems*, pp. 524-532.

Trivedi, K. (2001). Probability and statistics with reliability. Queuing and Computer Science Application, 2nd ed. New York: Wisley.

Ulloa, C. (2006). Metodología para evaluar el rendimiento de sistemas de almacenamiento y recuperación de documentos XML en bases de datos nativas para XML y no nativas habilitadas para XML. Síntesis Tecnológica, vol. 3, n. 1, 1-13.

Uslu, AM & Stausberg, J 2008, 'Value of the electronic patient record: An analysis of the literature', *Journal of Biomedical Informatics,* vol. 41, nº 4, pp. 675-682.

Van der Meijden, MJ, Tange, HJ, Boiten, J, Troost, J & Hasman A 2000, 'An experimental electronic patient record for stroke patients. Part 1: Situation analysis'. International Journal of Medical Informatics, vol. 58, nº 59, pp. 111–125.

Vazquez, EN, Komar, D, Martinez Sagasta, C, de Oliveira Ribas, LF & Lyra MR, *Implementación de una Historia Clínica Electrónica en un plan de salud a nivel nacional en Brasil.* [Online]. [Accediendo Octubre 2008]. Disponible desde WWW: http://www.telemedicina.buap.mx/PaginaNueva/Archivos/HCE_3.pdf

Viet, N 2006, *XML Native Database Systems.* Czech Technical University, Prague.

Vijan, S, Hofer, TP & Hayward RA 2000, 'Cost-utility analysis of screening intervals for diabetic retinopathy in patients with type 2 diabetes mellitus', *JAMA*, vol. 283, pp. 889–896.

Web Development Helper 2008, Página *Web* de la aplicación Web Development Helper. [Online]. [Accediendo Junio 2008]. Disponible desde WWW: http://projects.nikhilk.net/WebDevHelper

Wei, J, Valentino, J, Bell, D & Baker, R 2006, 'A Web-based Telemedicine System for Diabetic Retinopathy Screening Using Digital Fundus Photography', *Telemedicine and e-Health Journal*, vol. 12, n.º 1, pp. 50-57.

Wen, HC, Ho, YS, Jian, WS, Li, HC & Hsu, YHE (2007), 'Scientific production of electronic health record research, 1991-2005', *Computer Methods and Programs in Biomedicine,* vol. 86, pp. 191-196.

Wild, S, Roglic, G, Green, A, Sicree, R & King, H 2004, 'Global Prevalence of Diabetes. Estimates for the year 2000 and projections for 2030', *Diabetes Care*, vol. 27, n.º 5, pp. 1047–1053.

WITSA Global ICT Excellence Awards. Disponible en: http://www.witsa.org/ICT_Awards/. Última visita el 17 de Agosto de 2009.

Xindice 2008. *Página Web oficial de Apache Xindice* 2008. [Online]. [Accediendo Junio 2008]. Disponible desde WWW: http://xml.apache.org/xindice

Yogesan, K, Kumar, S, Goldschmidt, L & Cuadros, J 2006, *Teleophthalmology*, Springer.

Yu-Hua, A, Shao-Yuan, F & Ji-Feng, S 2000, 'A survey of telemedicine', *Beijing Biomedical Engineering*, vol. 20, pp. 249–252.

Zach, S 1996, 'Telemedicine overview and Abstract', *Nineteenth Convention of Electrical and Electronics Engineers*, Israel, pp. 409–412.

Zhang, E 2003, *Use of Native XML Database In TAPoR Project,* Alberta, Canadá.