



Universidad de Valladolid

E.U. de Informática (Segovia)

Ingeniería Técnica en Informática de Gestión

**Programa de Gestión
"Clínica Podológica"**

Alumno: José María Herrero Gómez

Tutora: María Luisa Martín Pérez

ÍNDICE DE CONTENIDOS

| | |
|--|-----------|
| SECCIÓN I: MEMORIA DEL PROYECTO | 7 |
| 1. IDENTIFICACIÓN DEL PROYECTO | 7 |
| 1.1. IDENTIFICACIÓN DEL PROYECTO..... | 7 |
| 1.2. ESTRUCTURA DE LA DOCUMENTACIÓN ENTREGADA..... | 7 |
| 1.3. ESTRUCTURA DEL CD | 7 |
| 2. DEFINICIÓN DEL PROBLEMA | 8 |
| 2.1. DESCRIPCIÓN DEL PROBLEMA..... | 8 |
| 2.2. MOTIVACIÓN..... | 8 |
| 2.3. OBJETIVOS | 8 |
| 3. ARQUITECTURA DEL SISTEMA..... | 9 |
| 3.1. DESCRIPCIÓN | 9 |
| 3.2. TECNOLOGÍAS..... | 10 |
| 3.3. HERRAMIENTAS | 11 |
| 4. METODOLOGÍA | 11 |
| 4.1. DESCRIPCIÓN DE LA METODOLOGÍA | 11 |
| 5. PLANIFICACIÓN Y COSTES | 12 |
| 5.1. FASES DEL PROYECTO..... | 12 |
| 5.2. ESTIMACIÓN TEMPORAL | 12 |
| 5.3. COMPARATIVA | 15 |
| 5.4. COSTES RECURSOS HUMANOS..... | 15 |
| 5.5. COSTES HARDWARE Y MATERIALES | 16 |
| 5.6. COSTES SOFTWARE | 17 |
| 5.7. COSTE DE LA APLICACIÓN | 17 |
| 6. CONSIDERACIONES ADICIONALES | 17 |
| 6.1. CONSIDERACIONES SOBRE EL ANÁLISIS | 17 |
| 6.2. CONSIDERACIONES SOBRE EL DISEÑO..... | 17 |
| 6.3. CONSIDERACIONES DE IMPLEMENTACIÓN..... | 18 |
| 7. AMPLIACIONES | 18 |
| 7.1. POSIBLES AMPLIACIONES | 18 |
| 8. CONCLUSION | 19 |
| 8.1. CONCLUSIONES | 19 |
| 8.2. BIBLIOGRAFIA | 19 |
| SECCIÓN II: MANUAL TÉCNICO..... | 23 |
| 1. ANÁLISIS..... | 23 |
| 1.1. INTRODUCCIÓN | 23 |
| 1.2. OBJETIVOS DEL SISTEMA | 23 |
| 1.3. REQUISITOS DEL SISTEMA | 25 |
| 1.3.1. REQUISITOS DE INFORMACIÓN..... | 25 |
| 1.3.2. REQUISITOS FUNCIONALES..... | 27 |
| 1.3.2.1. DIAGRAMA DE CASOS DE USO GENERAL | 27 |
| 1.3.2.2. DEFINICIÓN DE LOS ACTORES..... | 28 |
| 1.3.2.3. CASOS DE USO | 28 |

| | |
|---|------------|
| 2. DISEÑO | 45 |
| 2.1. COMPORTAMIENTO | 45 |
| 2.1.1. INTRODUCCIÓN | 45 |
| 2.1.2. DIAGRAMA DE CLASES | 47 |
| 2.1.3. DESCRIPCIÓN DE LAS CLASES | 49 |
| 2.2. MODELO DE DATOS | 63 |
| 2.2.1. INTRODUCCIÓN | 63 |
| 2.2.2. MODELO ENTIDAD-RELACIÓN | 63 |
| 2.2.3. MODELO RELACIONAL | 64 |
| 2.2.4. TRANSFORMACIÓN EN TABLAS | 65 |
| 3. PRUEBAS | 66 |
| 3.1. PRUEBAS DE INSTALACIÓN | 66 |
| 3.2. PRUEBAS DE SISTEMA | 67 |
| SECCIÓN III: MANUAL DE USUARIO | 81 |
| 1. INSTALACION DE LA APLICACIÓN | 81 |
| 1.1. REQUISITOS DE INSTALACIÓN | 81 |
| 1.2. INSTALACIÓN DE LA APLICACIÓN | 81 |
| 2. CREACIÓN BASE DE DATOS | 81 |
| 2.1. CREAR BASE DE DATOS | 81 |
| 3. USO GENERAL | 82 |
| 3.1. PANTALLA PRINCIPAL | 82 |
| 3.2. MENÚ PACIENTES | 84 |
| 3.2.1. AGREGAR PACIENTES | 85 |
| 3.2.2. BUSCAR PACIENTES | 86 |
| 3.2.3. MODIFICAR PACIENTES | 90 |
| 3.2.4. AGREGAR PACIENTES RELACIONADOS | 91 |
| 3.2.5. AGREGAR CONSULTAS | 95 |
| 3.2.6. VER CONSULTAS | 97 |
| 3.2.7. MODIFICAR CONSULTAS | 98 |
| 3.3. MENÚ CITAS | 100 |
| 3.4. MENÚ ESTADÍSTICAS | 105 |
| ANEXO: INDICE DE FIGURAS | 111 |

SECCIÓN I: MEMORIA DEL PROYECTO

1. IDENTIFICACIÓN DEL PROYECTO

1.1. IDENTIFICACIÓN DEL PROYECTO

Título: Programa de Gestión "Clínica Podológica"

Autor: José María Herrero Gómez

Tutora: María Luisa Martín Pérez

1.2. ESTRUCTURA DE LA DOCUMENTACIÓN ENTREGADA

La documentación entregada constará de 3 partes y un CD-ROM.

- **Memoria del proyecto**

En esta sección, se abordarán temas generales acerca del proyecto como las herramientas, arquitectura, metodología, motivaciones, planificaciones y presupuestos.

- **Manual Técnico**

Los puntos a tratar en esta sección serán desde una manera mucho más técnica, los requisitos, análisis, diseño, implementación y pruebas necesarias para el desarrollo de la aplicación.

- **Manual de Usuario**

El manual de usuario, hará un recorrido por toda la aplicación, explicando parte por parte cuales son las funcionalidades de la misma, así como su modo de uso.

- **CD-ROM**

El CD-ROM adjunto contiene todo lo necesario para la instalación y ejecución de la aplicación, así como toda la documentación, incluida la presente memoria.

1.3 ESTRUCTURA DEL CD

El CD-ROM esta dividido en varias carpetas, de las que se detalla su contenido a continuación.

- La carpeta **raíz** del CD-ROM contiene la memoria del proyecto y una carpeta denominada "**Software**".
- La carpeta "**Software**" a su vez contiene dos carpetas una denominada "**Fuentes**" y otra llamada "**Instalación**".
- La carpeta "**Fuentes**" contiene todos los códigos fuente del proyecto, así como las librerías necesarias para su correcto funcionamiento.
- La carpeta "**Instalación**" contiene el manual de usuario, el manual de instalación y una carpeta "**Software adicional**", que alberga todos los programas adicionales requeridos para la instalación del programa.

2. DEFINICIÓN DEL PROBLEMA

2.1. DESCRIPCIÓN DEL PROBLEMA

Los pies son la base del cuerpo humano, sobre la que soportamos nuestro peso todos los días de nuestra vida. Esto hace que sea de vital importancia cuidarlos adecuadamente. Hace unos años las patologías de los pies como callos y durezas eran algo exclusivo de personas con una avanzada edad. El auge del *running* y la moda tendiente cada vez más al calzado estrecho está provocando que cada vez más personas necesiten acudir a un podólogo para dichas dolencias y otras más específicas. Esto, y el amplio abanico de tratamientos existentes en la actualidad hace que las clínicas podológicas lleven un control detallado del historial de cada paciente para asegurar el mejor trato posible.

2.2. MOTIVACIÓN

Este proyecto surge del que es mi trabajo durante los últimos años. Mi familia tiene desde hace muchos años una Clínica Podológica en la que trabajan mi padre y ahora también mi hermana. Recientemente realizamos una reforma del local y decidimos llevar una gestión más detallada mediante una base de datos de pacientes, de la que fui el encargado. Inicialmente partimos de una base de datos en Microsoft Access que usaba otro compañero en su clínica. Sin embargo, esta solución ha resultado poco eficiente, ya que tenía demasiadas secciones que nosotros no usamos y, en cambio, echábamos en falta alguna funcionalidad que no estaba reflejada. Al final decidimos que queríamos un programa más a la medida de nuestros gustos y necesidades incluyendo, además de los datos e historial de los pacientes, una gestión de la agenda diaria de citas y una sección donde se reflejaran los cobros a los pacientes tanto los particulares, como los que acuden a consulta a través de mutuas de seguros de salud.

2.3. OBJETIVOS

El objetivo de este proyecto es realizar una aplicación CRUD (*Create, Read, Update, Delete*) que permita una gestión integral de todo lo que ocurre en una clínica de podología diariamente, que sea intuitiva, fácil de usar y ajustada a las necesidades que puedan surgir. Debe incluir un apartado de agenda donde se reflejen las citas de cada día, otro de gestión de pacientes con sus datos personales y sus antecedentes médicos, además de su historial de visitas a la clínica y, finalmente, otra sección donde se reflejen los pagos de cada visita que realicen los pacientes y poder así realizar un balance diario, mensual y de los ingresos y los pacientes atendidos a través de aseguradoras.

3. ARQUITECTURA DEL SISTEMA

3.1. DESCRIPCIÓN

La arquitectura utilizada es la llamada "*arquitectura en tres capas*", la cual define el modelo en tres niveles o capas de forma que los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Las tres capas definidas son las siguientes:

- **Capa de presentación.** Es la que se encarga de que el sistema interactúe con el usuario y viceversa, muestra el sistema al usuario, le presenta la información y obtiene la información del usuario en un mínimo de proceso. Es conocida como interfaz gráfica, y debe tener la característica de ser amigable, o sea, entendible y fácil de usar para el usuario. Esta capa se comunica únicamente con la capa intermedia o de negocio.
- **Capa de negocio.** Es donde residen las funciones que se ejecutan, se reciben las peticiones del usuario, se procesa la información y se envían las respuestas tras el proceso. Se denomina capa de negocio o capa de lógica del negocio, porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de acceso a datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.
- **Capa de acceso a datos.** Esta capa es la encargada de almacenar los datos del sistema y de los usuarios. Su función es almacenar y devolver datos a la capa de negocio, aunque para esto también es necesario en algunos casos, que tengan procedimientos y funciones dentro de la capa. En una arquitectura de tres capas, esta capa es la única que puede acceder a los mismos. Está formada por uno o varios sistemas gestores de bases de datos, localizados en un mismo servidor o en varios.

Estas capas, pueden estar localizadas todas en un mismo ordenador, como en el caso que nos ocupa, o por el contrario, cada una de las capas puede estar situada en diferentes ordenadores, para mejorar la funcionalidad de las mismas, incluso, en productos de gran complejidad, existen varios ordenadores para la capa de acceso a datos, y varios ordenadores para la capa de negocio.

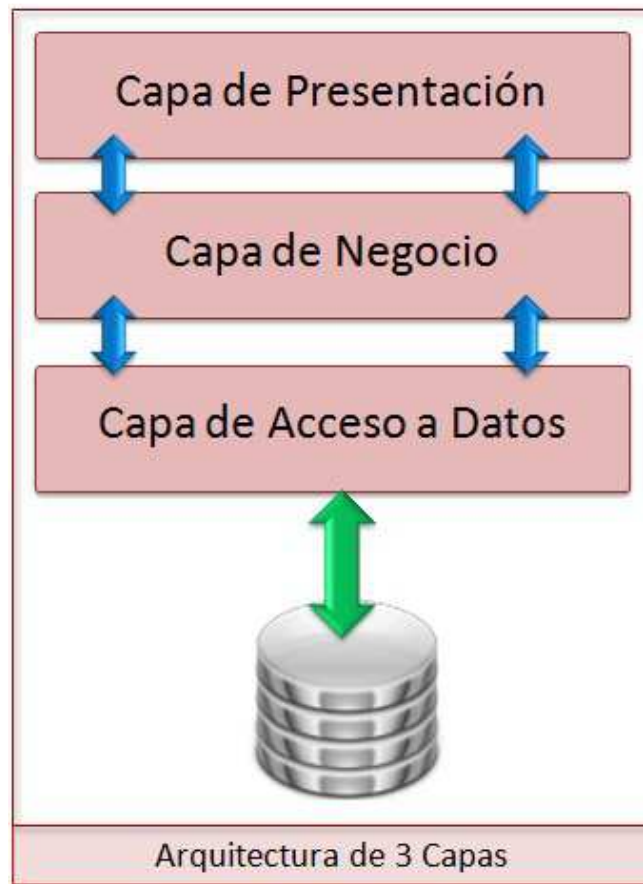


Figura 1: Arquitectura de tres capas

3.2. TECNOLOGÍAS

Las tecnologías que se han usado para el desarrollo del programa son las siguientes:

- **Java** como lenguaje de programación. Java es un lenguaje orientado a objetos, multiplataforma, lo cual nos permite que las aplicaciones desarrolladas sean portables a otros sistemas operativos. Además el ser orientado a objetos supone una gran ventaja, ya que permite al código emular con mayor exactitud los problemas del mundo real que se desean resolver. Java es un lenguaje dinámico, en el que los pequeños fragmentos de código Java se ensamblan en tiempo de ejecución en el programa y no en el momento en el que se escribe el código. Una gran ventaja es que Java dispone de una serie de funcionalidades ya implementadas y recogidas en un conjunto de paquetes que facilitan el desarrollo de las aplicaciones y que pueden ser utilizadas por cualquier desarrollador. Por último Java tiene la característica de que libera la memoria automáticamente, por lo cual los desarrolladores no se tienen que centrar en la liberación de memoria y como consecuencia mejora notablemente el rendimiento de las aplicaciones.
- **SQL** como lenguaje de base de datos. El lenguaje de consulta estructurado (*Structured Query Language*) o SQL es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla información de interés de bases de datos, así como hacer cambios en ella.

3.3. HERRAMIENTAS

- **Netbeans.** Es un entorno de desarrollo integrado (*IDE*) libre, creado principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso. NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos. La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes software llamados módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.
- **MySQL Server.** Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario desarrollado por MySQL AB. Es una base de datos muy rápida en la lectura.
- **MySQL Workbench.** es una herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, creación y mantenimiento para el sistema de base de datos MySQL.
- **StarUML.** Es un proyecto de código abierto para desarrollar UML de una manera rápida, flexible, extensible, característica y disponible de manera gratuita.
- **Microsoft Word 2003.** es una aplicación informática para procesamiento de textos. Fue creado por la empresa Microsoft, y viene integrado en el paquete ofimático denominado Microsoft Office.

4. METODOLOGÍA

4.1. DESCRIPCIÓN DE LA METODOLOGÍA

La aplicación ha sido diseñada mediante el paradigma de la programación orientada a objetos. En las fases de análisis y diseño se ha empleado el lenguaje UML (*Unified Modeling Language*) **Lenguaje Unificado de Modelado**, el cual es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad que permite de manera gráfica visualizar, especificar, construir y documentar un sistema.

El proceso de desarrollo se ha inspirado en el **Proceso Unificado de Desarrollo Software**. Es un proceso centrado en la arquitectura, la cual da una perspectiva del sistema completo y describe los elementos más importantes del sistema. También es un proceso iterativo e incremental, compuesto por cuatro fases bien definidas, análisis, diseño, implementación y pruebas. Cada una de estas fases es realizada a lo largo de una serie de iteraciones. Estas iteraciones ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo. Dicho proceso está dirigido por casos de uso, que se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones.

5. PLANIFICACIÓN Y COSTES

5.1. FASES DEL PROYECTO

El proyecto se ha dividido en cinco fases:

1. **Estudio Previo.** En esta fase se realiza un estudio del problema para valorar las necesidades reales. También se estudian aplicaciones similares para ver las funcionalidades de la competencia y mejorarlas en ciertos aspectos o crear funcionalidades nuevas para crear una ventaja competitiva.
2. **Análisis y Diseño del Sistema.** Esta fase engloba en análisis del sistema mediante los requisitos del sistema y su diseño siguiendo la metodología orientada a objetos.
3. **Implementación.** En esta fase se llevará a cabo el estudio de las tecnologías con las que se va a desarrollar a la aplicación, además de la implementación de la base de datos y el programa de gestión.
4. **Pruebas.** En esta fase se realizan las pruebas del sistema y la puesta en funcionamiento del mismo para comprobar que el programa funciona como se espera y cumple el catalogo de requisitos.
5. **Documentación.** Esta fase se encarga de plasmar de una manera clara todas las fases anteriores.

5.2. ESTIMACIÓN TEMPORAL

En las tablas que se muestran a continuación puede verse la estimación temporal inicial y el seguimiento temporal real. También se adjuntan los diagramas de Gantt asociados a cada tabla. Están divididas en las fases de diseño anteriormente citadas y puede verse el desglose de cada fase. En cada fase se indica la fecha de inicio la fecha de final y la duración total.

La jornada de trabajo prevista es de lunes a viernes de 10:00 a 14:00 y de 20:00 a 23:00 con una pausa de una hora y los fines de semana de 10:00 a 13:00 y de 17:00 a 20:00. Esto hace un total de 6 horas diarias y 42 horas semanales hasta la finalización del proyecto.

| Nombre | Duración | Inicio | Fin |
|------------------------------|----------|------------|------------|
| CLINPOD. Desarrollo completo | 107 días | 01/05/2015 | 15/08/2015 |
| ESTUDIO PREVIO | 11 días | 01/05/2015 | 11/05/2015 |
| Estudio del problema | 7 días | 01/05/2015 | 07/05/2015 |
| Aplicaciones Similares | 4 días | 08/05/2015 | 11/05/2015 |
| ANÁLISIS Y DISEÑO | 20 días | 12/05/2015 | 31/05/2015 |
| Análisis | 10 días | 12/05/2015 | 21/05/2015 |
| Diseño | 10 días | 22/05/2015 | 31/05/2015 |
| IMPLEMENTACION | 45 días | 01/06/2015 | 15/07/2015 |
| Estudio Tecnologías | 3 días | 01/06/2015 | 03/06/2015 |
| Creación Base de Datos | 2 días | 04/06/2015 | 05/06/2015 |
| Implementación | 40 días | 06/06/2015 | 15/07/2015 |
| PRUEBAS | 5 días | 16/07/2015 | 20/07/2015 |
| Batería de pruebas | 5 días | 16/07/2015 | 20/07/2015 |
| DOCUMENTACIÓN | 26 días | 21/07/2015 | 15/08/2015 |
| Memoria del proyecto | 7 días | 21/07/2015 | 27/07/2015 |
| Manual técnico | 11 días | 28/07/2015 | 07/08/2015 |
| Manual de usuario | 8 días | 08/08/2015 | 15/08/2015 |

Figura 2: Planificación temporal inicial

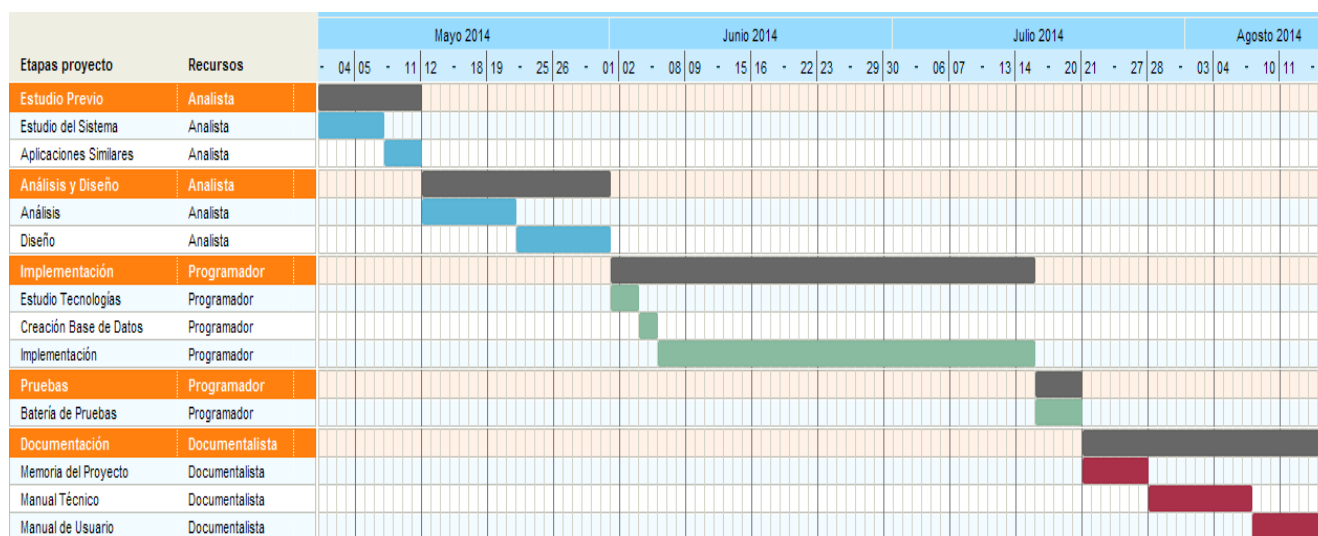


Figura 3: Planificación inicial, diagrama de Gantt

| Nombre | Duración | Inicio | Fin |
|------------------------------|----------|------------|------------|
| CLINPOD. Desarrollo completo | 107 días | 01/05/2015 | 15/08/2015 |
| ESTUDIO PREVIO | 3 días | 01/05/2015 | 03/05/2015 |
| Estudio del problema | 2 días | 01/05/2015 | 02/05/2015 |
| Aplicaciones Similares | 1 día | 03/05/2015 | 03/05/2015 |
| ANÁLISIS Y DISEÑO | 13 días | 04/05/2015 | 16/05/2015 |
| Análisis | 6 días | 04/05/2015 | 09/05/2015 |
| Diseño | 7 días | 10/05/2015 | 16/05/2015 |
| IMPLEMENTACION | 60 días | 17/05/2015 | 15/07/2015 |
| Estudio Tecnologías | 3 días | 17/05/2015 | 19/05/2015 |
| Creación Base de Datos | 2 días | 20/05/2015 | 21/05/2015 |
| Implementación | 55 días | 22/05/2015 | 15/07/2015 |
| PRUEBAS | 5 días | 16/07/2015 | 20/07/2015 |
| Batería de pruebas | 5 días | 16/07/2015 | 20/07/2015 |
| DOCUMENTACIÓN | 26 días | 21/07/2015 | 15/08/2015 |
| Memoria del proyecto | 7 días | 21/07/2015 | 27/07/2015 |
| Manual técnico | 11 días | 28/07/2015 | 07/08/2015 |
| Manual de usuario | 8 días | 08/08/2015 | 15/08/2015 |

Figura 4: Seguimiento real del proyecto

Programa de Gestión "Clínica Podológica"

José María Herrero Gómez

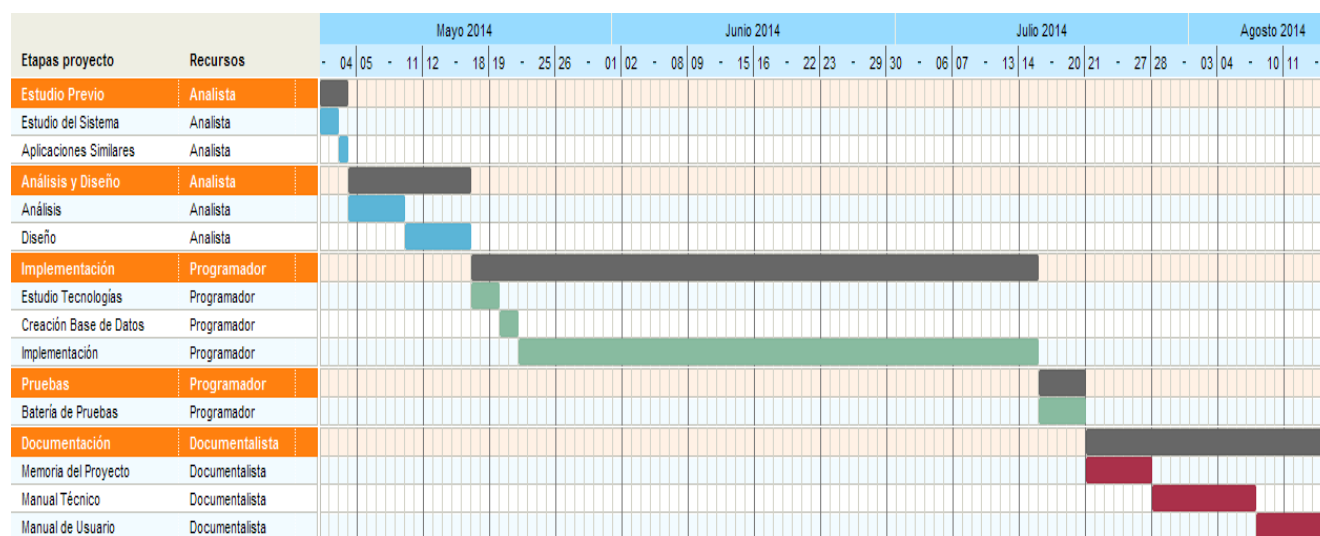


Figura 5: Seguimiento real, diagrama de Gantt

5.3. COMPARATIVA

La comparativa entre la planificación inicial y el seguimiento real del proyecto dividida por fases de desarrollo es la siguiente:

| Nombre | Estimado | Real | Desviación Días |
|-------------------|----------|-------------------|-----------------|
| Estudio Previo | 11 días | 3 días | -8 |
| Análisis y Diseño | 20 días | 13 días | -7 |
| Implementación | 45 días | 60 días (53 días) | +15 (+8) |
| Pruebas | 5 días | 5 días | 0 |
| Documentación | 27 días | 27 días | 0 |

Figura 6: Comparativa planificación inicial-seguimiento real

Como se puede ver en la tabla anterior las fases de estudio, análisis y diseño se consiguieron completar 15 días antes de lo previsto. Esto fue debido al conocimiento previo del problema y su entorno, que permitió acelerar estas fases sin suponer una pérdida en la calidad de la realización de las mismas.

En la fase de implementación se produjo un retraso importante en el que se perdió el tiempo que se había conseguido adelantar debido a una subestimación del tiempo de desarrollo necesario y a unos problemas familiares que detuvieron el proyecto durante una semana. Por esto, en realidad el tiempo de la fase de implementación solo se demoró 8 días más de lo previsto.

Finalmente, en las fases de pruebas y documentación se intentó cumplir los plazos previstos para poder completar el proyecto a tiempo y se consiguió, por lo que se terminó el proyecto en el plazo previsto al inicio.

5.4. COSTES RECURSOS HUMANOS

Para el cálculo de los costes en recursos humanos se diferenciarán 3 perfiles distintos. Estos perfiles se encargan cada uno de realizar las etapas que le son asignadas y su remuneración es acorde a la cualificación necesaria para cada puesto y a la complejidad de la tarea a realizar:

- **Analista:** Es el encargado de realizar el estudio del problema a resolver y de las fases de análisis y diseño. Se calcula una remuneración de 20€/hora.
- **Programador:** Su función es la de realizar la implementación del producto de acuerdo con los requisitos y el análisis extraído por el analista. También debe diseñar y realizar las pruebas de producto. Su remuneración es de 12€/hora.
- **Documentalista:** Se encarga de realizar toda la documentación asociada al desarrollo de la aplicación, incluyendo la memoria y los manuales técnico y de usuario. La remuneración estimada para él es de 8€/hora.

Hay que tener en cuenta que la fase de implementación sufrió un retraso de 15 días, pero en realidad solo se tienen en cuenta 8, ya que el resto no se trabajó, por lo que la semana restante no supone un incremento en el coste. Con estos perfiles los costes en recursos humanos desglosados quedarían de la siguiente forma:

| Nombre | Perfil | Duración | Duración (horas) | Coste |
|--------------------------|----------------|----------|------------------|----------------|
| ESTUDIO PREVIO | Analista | 3 días | 3x6 = 18h | 18x20 = 360€ |
| Estudio del problema | Analista | 2 días | 2x6 = 12h | 12x20 = 240€ |
| Aplicaciones Similares | Analista | 1 día | 1x6 = 6h | 6x20 = 120€ |
| ANÁLISIS Y DISEÑO | Analista | 13 días | 13x6 = 78h | 78x20 = 1560€ |
| Análisis | Analista | 6 días | 6x6 = 36h | 36x20 = 720€ |
| Diseño | Analista | 7 días | 7x6 = 42h | 42x20 = 840€ |
| IMPLEMENTACION | Programador | 53 días | 53x6 = 318h | 318x12 = 3816€ |
| Estudio Tecnologías | Programador | 3 días | 3x6 = 18h | 18x12 = 216€ |
| Creación Base de Datos | Programador | 2 días | 2x6 = 12h | 12x12 = 144€ |
| Implementación | Programador | 48 días | 48x6 = 288h | 288x12 = 3456€ |
| PRUEBAS | Programador | 5 días | 5x6 = 30h | 30x12 = 360€ |
| Batería de pruebas | Programador | 5 días | 5x6 = 30h | 30x12 = 360€ |
| DOCUMENTACIÓN | Documentalista | 27 días | 27x6 = 162h | 162x8 = 1296€ |
| Memoria del proyecto | Documentalista | 7 días | 7x6 = 42h | 42x8 = 336€ |
| Manual técnico | Documentalista | 11 días | 11x6 = 66h | 66x8 = 528€ |
| Manual de usuario | Documentalista | 8 días | 8x6 = 48h | 48x8 = 384 € |

Figura 7: Costes de recursos humanos

El coste total de los recursos humanos asociados al proyecto incluyendo las 5 fases es de 7392€. Esto supone un importante ahorro ya que si tomáramos como referencia los tiempos de la estimación inicial, el coste en este apartado sería de 8616€. El tiempo que se adelantó en las fases asociadas al analista permitió que el retraso que sufrió el programador no solo no provocara un incremento en el coste, sino que se haya mejorado el presupuesto en 1224€.

5.5. COSTES HARDWARE Y MATERIALES

El hardware informático necesario para la realización del proyecto es el ordenador personal del que ya se disponía antes del inicio del proyecto. También se ha utilizado la conexión a Internet para la actualización de los programas utilizados y la búsqueda de información y material de oficina variado. El coste que se ha aplicado al proyecto en este apartado es un 25€ del coste real. Por tanto los costes se reflejan en la siguiente tabla:

| RECURSO | COSTE | COSTE PROYECTO (25%) |
|----------------------|-------------------|----------------------|
| Ordenador personal | 600€ | 150€ |
| Teclado multifunción | 16€ | 4€ |
| Ratón ergonómico | 10€ | 2.5€ |
| Conexión Adsl | 25€x4meses = 100€ | 25€ |
| Material de oficina | 50€ | 12.5€ |
| TOTAL | 776€ | 194€ |

Figura 8: Costes hardware

5.6. COSTES SOFTWARE

El software utilizado es el descrito en el apartado de Herramientas. La mayor parte es software libre y el resto venía incluido en el precio del ordenador por lo que el coste de este apartado será 0€ como puede verse en esta tabla:

| SOFTWARE | COSTE |
|-----------------------|-------------------------------|
| Microsoft Windows XP | 0€ (incluido en el ordenador) |
| Microsoft Office 2003 | 0€ (incluido en el ordenador) |
| Netbeans IDE 8.0.1 | 0€ |
| MySQL Server | 0€ |
| MySQL Workbench | 0€ |
| StarUML | 0€ |

Figura 9: Costes software

5.7. COSTE DE LA APLICACIÓN

El coste total de la aplicación se obtiene sumando los importes de todos los apartados anteriores, recursos humanos, hardware, materiales y software.

| | |
|-----------------------------|--------------|
| Recursos humanos | 7392€ |
| Hardware y materiales | 194€ |
| Software | 0€ |
| TOTAL (IVA INCLUIDO) | 7586€ |

Figura 10: Coste total de la aplicación

6. CONSIDERACIONES ADICIONALES

6.1. CONSIDERACIONES SOBRE EL ANÁLISIS

La fase de análisis suele ser la más compleja ya que si no se realiza un estudio completo del sistema no se pueden conocer en detalle los requisitos del producto a desarrollar. En mi caso, debido a la experiencia que tenía con el programa antiguo, sabía sus puntos fuertes y débiles, conocía perfectamente el tipo de aplicación a desarrollar, los apartados que debía tener y qué funcionalidades debía aportar. Además tenía varias ideas sobre funcionalidades nuevas y posibles mejoras que seguramente no se me hubieran ocurrido si no hubiera tenido un contacto profundo con el entorno de uso. Por todo lo citado la fase de análisis resultó más fácil de lo previsto y se realizó de manera más rápida.

6.2. CONSIDERACIONES SOBRE EL DISEÑO

En el diseño he tenido en cuenta mis gustos personales a la hora de ordenar los componentes de la aplicación, ya que voy a ser el principal usuario final. En los apartados que ya existían en el producto anterior, como la introducción de datos de los pacientes he intentado que el diseño se asemeje para que la implantación no resulte traumática y no se requiera un tiempo de adaptación.

6.3. CONSIDERACIONES DE IMPLEMENTACIÓN

Para la implementación del producto se ha usado JPA (*Java Persistence API*) para realizar los accesos a la base de datos. Esto se ha realizado creando entidades que correspondían a las tablas de la base de datos. En cada entidad se definieron varias *Named Queries*, consultas JPQL (*Java Persistence Query Language*), equivalentes a las consultas SQL que se necesitaban. De esta forma no había que escribir la consulta cada vez que había que realizarla, sino que solo se hacía una llamada mediante el identificador. Para lograr una mejor separación modular para cada entidad se han creado dos módulos más, uno con el controlador de la misma y otro con los métodos de para agregar y modificar registros en la base de datos. También hay que tener en cuenta que algunas tablas tienen identificadores principales compuestos por varios campos, por lo que hay clases separadas para la entidad y para la clave principal.

La base de datos se ha realizado teniendo en cuenta que solo va a ser utilizada por un único usuario, así que al iniciar la aplicación se conecta automáticamente sin necesidad de realizar autenticación.

7. AMPLIACIONES

7.1. POSIBLES AMPLIACIONES

El siguiente proyecto está abierto y admite muchas ampliaciones en un futuro. El objetivo es terminar implementando la mayor parte de estas adiciones y, al ser un sistema que me va repercutir directamente tengo motivación para implementarlas todas. Algunas de ellas ya fueron pensadas en las fases de análisis y diseño y se descartaron para no prolongar el proyecto en el tiempo y poder priorizar las funcionalidades que se consideraban más importantes. Algunas de estas ideas son:

- Posibilidad de imprimir informes e historias de pacientes además de mostrarlas en pantalla.
- Inclusión de un navegador de Internet en la interfaz para poder facturar las consultas de las pólizas de seguros sin salir de la aplicación.
- Inclusión de un sistema de autocita por Internet. Esta funcionalidad fue descartada inicialmente debido al perfil de los pacientes que acuden a la clínica. En su mayoría no son "nativos digitales" y no han conocido otro método para concertar una cita que no sea llamar por teléfono directamente.
- Implementación de un sistema de alertas por SMS para recordar citas a los pacientes, con posibilidad de configurar alarmas para recordarles el momento adecuado para su próxima visita o revisión.
- Implantación de la receta médica digital para la prescripción de productos en lugar de usar las clásicas en papel.
- Inclusión de apartados en la aplicación para gestionar facturas, nóminas, pedidos de material, etc.
- Posibilidad de realizar backups de la base de datos automáticamente y subirlas a un servidor externo.
- Creación de una aplicación Android para poder consultar y gestionar la agenda del día, así como cualquier otro apartado de la aplicación en cualquier momento y lugar.
- Inclusión de un gestor de correo propio con agenda de contactos.

8. CONCLUSION

8.1. CONCLUSIONES

El presente proyecto me ha permitido profundizar en los conocimientos adquiridos durante la carrera y adquirir otros nuevos. El hecho de desarrollar un proyecto completo de principio a fin ejerciendo todos los roles, ha hecho que ahora dimensione mejor lo que supone crear un producto software. Me he encontrado con diferentes escollos durante el proceso, especialmente en la fase de implementación, debido principalmente a que nunca había realizado una aplicación que usase bases de datos y he requerido una fase de aprendizaje, pero en general se han resuelto todos los problemas encontrados. El resultado final es una aplicación que cumple las expectativas que me había propuesto en un principio. También me he dado cuenta de lo importante y difícil que resulta hacer una buena documentación que plasme todos los aspectos del proyecto para el posterior desarrollo del sistema. Por último, el proyecto ha servido para desarrollar y valorar más aún otras cualidades no relacionadas con la informática sino con todos los aspectos de la vida, como dedicación, constancia, responsabilidad, organización, innovación o afán de superación.

8.2. BIBLIOGRAFIA

- ELMASRI, NAVATHE. Fundamentos de Sistemas de Bases de Datos. Editorial Addison-Wesley 2007
- SOMMERVILLE, I. Ingeniería del Software. 6ed. Addison Wesley. 2002
- SCHMULLER, J. Aprendiendo UML en 24 horas. Editorial Prentice Hall. 2001
- <http://www.mysql.com>
- <http://www.mysqla.com.ar>
- <http://www.mysqltutorial.org>
- <http://www.w3schools.com>
- <http://www.stackoverflow.com>
- <http://www.lawebdelprogramador.com>
- <http://www.wikipedia.org>
- <http://www.google.com>
- <http://uvadoc.uva.es>
- <http://www.netbeans.com>
- <http://www.tomsplanner.es>
- <http://toedter.com>
- <http://staruml.sourceforge.net>
- <http://www.uml.org>
- Apuntes de Ingeniería del Software I y II
- Apuntes de Bases de Datos

SECCIÓN II: MANUAL TÉCNICO

1. ANÁLISIS

1.1. INTRODUCCIÓN

El objetivo de este proyecto es el desarrollo de una aplicación que permita la gestión diaria de una consulta de podología, incluyendo tareas como la gestión de los datos e historial de los pacientes, la gestión de las citas y la realización de informes de actividad.

1.2. OBJETIVOS DEL SISTEMA

A continuación se detallarán los objetivos principales que debe cumplir el sistema a desarrollar.

| | |
|--------------------|--|
| OBJ-01 | <i>Gestión de ficha con los datos personales del paciente</i> |
| Descripción | El sistema debe permitir la introducción, visualización y modificación de los datos personales del paciente, así como su consolidación en la base de datos. Los datos personales incluyen el nombre, dirección, teléfono, etc. del paciente, así como su seguro médico, de tener uno, sus antecedentes médicos, alergias y otros datos que pudieran ser de interés a la hora de atenderle. |
| Importancia | Objetivo principal del sistema |
| Estabilidad | Alta |
| Comentarios | Ninguno |

Figura 11: Objetivo OBJ-01

| | |
|--------------------|---|
| OBJ-02 | <i>Gestión del historial clínico del paciente</i> |
| Descripción | El sistema debe permitir la introducción, visualización y modificación del historial del paciente en la consulta, el listado de visitas con fecha y la prestación realizada en cada caso. También se debe incluir el importe y la fecha de abono del mismo. |
| Importancia | Objetivo principal del sistema |
| Estabilidad | Alta |
| Comentarios | Ninguno |

Figura 12: Objetivo OBJ-02

| | |
|--------------------|---|
| OBJ-03 | <i>Interfaz amigable</i> |
| Descripción | El sistema debe presentar una interfaz clara y fácil de usar, con todos los elementos necesarios para la gestión accesibles y presentados de manera sencilla para que la experiencia de usuario sea óptima. |
| Importancia | Objetivo principal del sistema |
| Estabilidad | Alta |
| Comentarios | Ninguno |

Figura 13: Objetivo OBJ-03

| | |
|--------------------|--|
| OBJ-04 | <i>Gestión de citas de consulta</i> |
| Descripción | El sistema debe presentar una pequeña agenda de citas ordenadas por día y permitir su introducción y modificación. La agenda debe poder almacenar citas de pacientes que ya tengan historial en la clínica o pacientes nuevos. |
| Importancia | Objetivo secundario del sistema |
| Estabilidad | Media |
| Comentarios | Ninguno |

Figura 14: Objetivo OBJ-04

| | |
|--------------------|--|
| OBJ-05 | <i>Presentación de balances</i> |
| Descripción | El sistema debe permitir recuperar los datos de todas las prestaciones realizadas durante un periodo de tiempo, día, mes, año o todo, y presentarlos para poder llevar un control económico de la empresa. |
| Importancia | Objetivo secundario del sistema |
| Estabilidad | Media |
| Comentarios | Ninguno |

Figura 15: Objetivo OBJ-05

1.3. REQUISITOS DEL SISTEMA

1.3.1. REQUISITOS DE INFORMACIÓN

En este apartado se detallan los requisitos de información identificados para el correcto funcionamiento de la aplicación.

| | |
|-----------------------------|---|
| IRQ-01 | <i>Información y datos personales del paciente</i> |
| Objetivos asociados | OBJ-01 Gestión de ficha con los datos personales del paciente OBJ-03 Interfaz amigable |
| Requisitos asociados | UC-01 Introducir datos paciente |
| Descripción | El sistema debe almacenar los datos personales del paciente en la base de datos para su posterior uso. |
| Datos específicos | <ul style="list-style-type: none"> • Nombre • Apellidos • Dirección • Código Postal • Localidad • Provincia • Teléfono • Fecha de Nacimiento • Mutua aseguradora • Antecedentes médicos • Alergias • Anticoagulantes • Diabético |
| Estabilidad | Alta |
| Comentarios | Ninguno |

Figura 16: Requisito IRQ-01

| | |
|-----------------------------|--|
| IRQ-02 | <i>Información de visitas clínicas del paciente</i> |
| Objetivos asociados | OBJ-02 Gestión del historial clínico del paciente OBJ-03 Interfaz amigable |
| Requisitos asociados | UC-02 Introducir visitas paciente |
| Descripción | El sistema debe almacenar los datos de las visitas del paciente a la clínica en la base de datos para su posterior uso. |
| Datos específicos | <ul style="list-style-type: none"> • Fecha Realización • Acto Médico • Prestación Realizada • Importe • Fecha de pago |
| Estabilidad | Alta |
| Comentarios | Ninguno |

Figura 17: Requisito IRQ-02

| | |
|-----------------------------|---|
| IRQ-03 | <i>Información de citas de la clínica</i> |
| Objetivos asociados | OBJ-03 Interfaz amigable OBJ-04 Gestión de citas de consulta |
| Requisitos asociados | UC-03 Introducir citas |
| Descripción | El sistema debe almacenar los datos de la agenda de citas de la consulta para su uso. |
| Datos específicos | <ul style="list-style-type: none"> • Fecha • Nombre • Duración |
| Estabilidad | Media |
| Comentarios | Ninguno |

Figura 18: Requisito IRQ-03

| | |
|-----------------------------|--|
| IRQ-04 | <i>Información de balances de la clínica</i> |
| Objetivos asociados | OBJ-03 Interfaz amigable OBJ-05 Presentación de balances |
| Requisitos asociados | UC-04 Mostrar balances |
| Descripción | El sistema debe ofrecer la posibilidad al usuario de poder ver balances, diarios, mensuales, anuales o totales de las atenciones realizadas en la clínica. Los datos deben poder ser agrupados por actos médicos o mostrarse al completo. Además deben poder visualizarse las consultas que están pendientes de cobro. |
| Datos específicos | <ul style="list-style-type: none"> • Nombre • Apellidos • Acto Médico • Importe • Acto Médico • Cantidad • Recaudación |
| Estabilidad | Media |
| Comentarios | Ninguno |

Figura 19: Requisito IRQ-04

1.3.2. REQUISITOS FUNCIONALES

A continuación se especificarán los requisitos funcionales sobre lo que debería hacer el sistema, obtenidos durante el proceso de estudio del sistema.

1.3.2.1. DIAGRAMA DE CASOS DE USO GENERAL

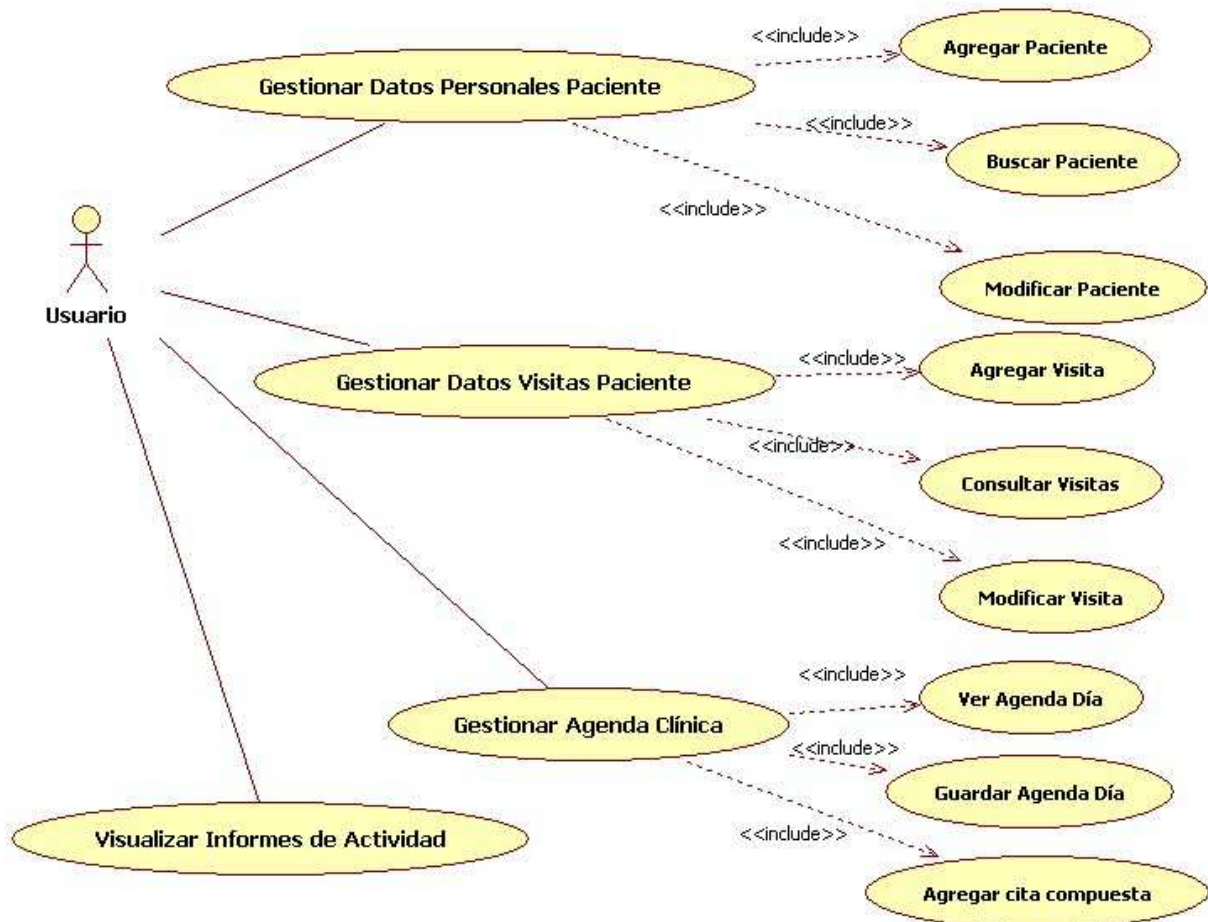


Figura 20: Diagrama de casos de uso

1.3.2.2. DEFINICIÓN DE LOS ACTORES

| | |
|--------------------|---|
| ACT-01 | <i>Usuario Principal</i> |
| Descripción | Actor que representa al usuario único de la aplicación, con capacidad para realizar todas las acciones disponibles en la misma. |
| Comentarios | Ninguno |

Figura 21: Definición de actor ACT-01

1.3.2.3. CASOS DE USO

| | | |
|-----------------------------|---|---|
| UC-01 | <i>Gestionar Datos Personales Paciente</i> | |
| Objetivos asociados | OBJ-01 Gestión de ficha con los datos personales del paciente OBJ-03 Interfaz amigable | |
| Requisitos asociados | IRQ-01 Información y datos personales del paciente | |
| Descripción | El sistema guardará los datos personales del paciente introducidos a través del formulario en la base de datos. | |
| Precondición | Seleccionar en el menú principal la opción de Gestionar Pacientes. | |
| Secuencia normal | <i>P1</i> | El usuario introduce los datos personales obtenidos del paciente en el formulario y presiona el botón adecuado. |
| | <i>P2</i> | El sistema comprueba que los datos sean al menos los mínimos requeridos para su guardado y que estén en el formato correcto. |
| | <i>P3</i> | Se le muestra al usuario una confirmación de que la acción se ha realizado de forma satisfactoria y le comunica el identificador interno asignado al paciente en el caso de que la acción sea el alta de un paciente nuevo. |
| Poscondición | Los datos personales del paciente han sido guardados en la base de datos. | |
| Excepciones | En el caso de que la consolidación de la información a la base de datos no se realice correctamente se muestra un dialogo al usuario. | |
| Estabilidad | Alta | |
| Comentarios | Ninguno | |

Figura 22: Caso de uso UC-01

| | | |
|-----------------------------|---|---|
| SUBCASO DE USO-01 | <i>Agregar Paciente nuevo</i> | |
| Objetivos asociados | OBJ-01 Gestión de ficha con los datos personales del paciente OBJ-03 Interfaz amigable | |
| Requisitos asociados | IRQ-01 Información y datos personales del paciente | |
| Descripción | El sistema guardará los datos personales del paciente nuevo introducidos a través del formulario en la base de datos. | |
| Precondición | Seleccionar en el menú principal la opción de Gestionar Pacientes y no haber abierto antes la ficha de otro paciente o haber limpiado el formulario tras haberlo hecho. | |
| Secuencia normal | <i>P1</i> | El usuario introduce los datos personales obtenidos del paciente en el formulario y presiona el botón "Añadir Paciente". |
| | <i>P2</i> | El sistema comprueba que los datos sean al menos los mínimos requeridos para su guardado y que estén en el formato correcto. |
| | <i>P3</i> | Se le muestra al usuario una confirmación de que la acción se ha realizado de forma satisfactoria y se le comunica el identificador interno asignado al paciente. |
| Poscondición | Los datos personales del paciente han sido guardados en la base de datos. | |
| Excepciones | En el caso de que la consolidación de la información a la base de datos no se realice correctamente se muestra un dialogo al usuario. | |
| Estabilidad | Alta | |
| Comentarios | Ninguno | |

Figura 23: Caso de uso UC-01, subcaso 01

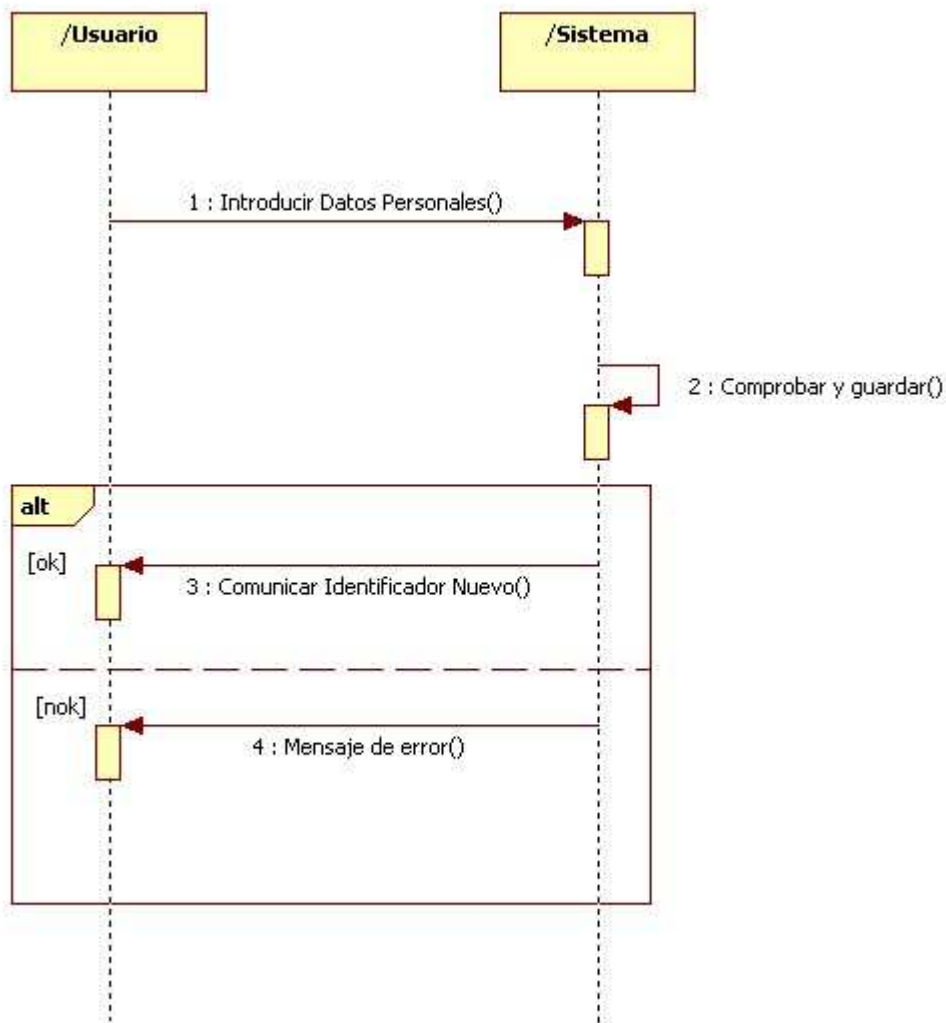


Figura 24: Caso de uso UC-01, subcaso 01. Diagrama de interacción

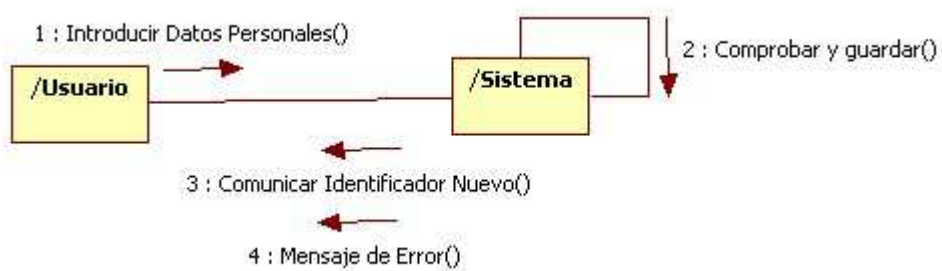


Figura 25: Caso de uso UC-01, subcaso 01. Diagrama de colaboración

| | | |
|-----------------------------|---|--|
| SUBCASO DE USO-02 | <i>Buscar Paciente</i> | |
| Objetivos asociados | OBJ-01 Gestión de ficha con los datos personales del paciente OBJ-02 Gestión del historial clínico del paciente OBJ-03 Interfaz amigable | |
| Requisitos asociados | IRQ-01 Información y datos personales del paciente IRQ-02 Información de visitas clínicas del paciente | |
| Descripción | El sistema mostrará los datos personales almacenados en la base de datos del paciente solicitado por el usuario a través de los apellidos o su identificador. | |
| Precondición | Seleccionar en el menú principal la opción de Gestionar Pacientes. | |
| Secuencia normal | <i>P1</i> | El usuario introduce en la barra de búsqueda superior los apellidos o en la barra de búsqueda inferior el identificador del paciente que desee buscar. |
| | <i>P2</i> | El sistema busca los datos en la base de datos. |
| | <i>P3</i> | Se le muestra al usuario la ficha completa del paciente obtenido. |
| Poscondición | La ficha del paciente se muestra en el formulario. | |
| Excepciones | En el caso de que el usuario introduzca un identificador que no existe en la base de datos el sistema muestra un error. | |
| Estabilidad | Alta | |
| Comentarios | Ninguno | |

Figura 26: Caso de uso UC-01, subcaso 02

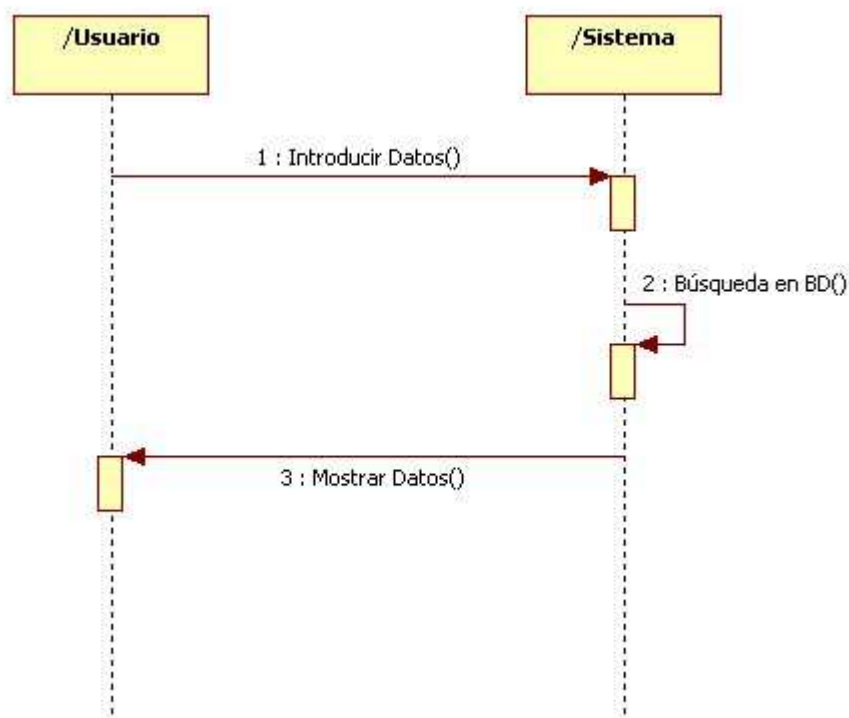


Figura 27: Caso de uso UC-01, subcaso 02. Diagrama de interacción

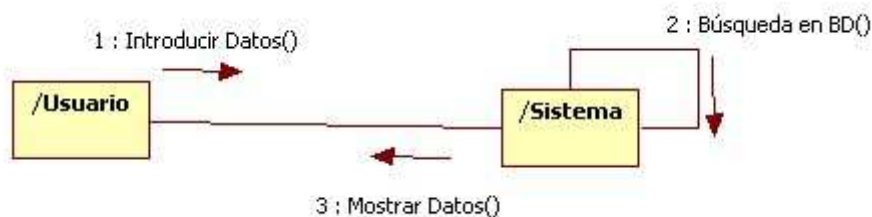


Figura 28: Caso de uso UC-01, subcaso 02. Diagrama de colaboración

| | | |
|-----------------------------|---|--|
| SUBCASO DE USO-03 | <i>Modificar Paciente</i> | |
| Objetivos asociados | OBJ-01 Gestión de ficha con los datos personales del paciente OBJ-03 Interfaz amigable | |
| Requisitos asociados | IRQ-01 Información y datos personales del paciente | |
| Descripción | El sistema guardará los nuevos datos personales del paciente introducidos a través del formulario en la base de datos. | |
| Precondición | Seleccionar en el menú principal la opción de Gestionar Pacientes y haber abierto antes la ficha de un paciente. | |
| Secuencia normal | <i>P1</i> | El usuario introduce los datos personales nuevos del paciente en cuestión en el formulario y presiona el botón "Modificar Paciente". |
| | <i>P2</i> | El sistema comprueba que los datos sean al menos los mínimos requeridos para su guardado y que estén en el formato correcto. |
| | <i>P3</i> | Se le muestra al usuario una confirmación de que la acción se ha realizado de forma satisfactoria. |
| Poscondición | Los nuevos datos personales del paciente han sido guardados en la base de datos. | |
| Excepciones | En el caso de que la consolidación de la información a la base de datos no se realice correctamente se muestra un dialogo al usuario. | |
| Estabilidad | Alta | |
| Comentarios | Ninguno | |

Figura 29: Caso de uso UC-01, subcaso 03

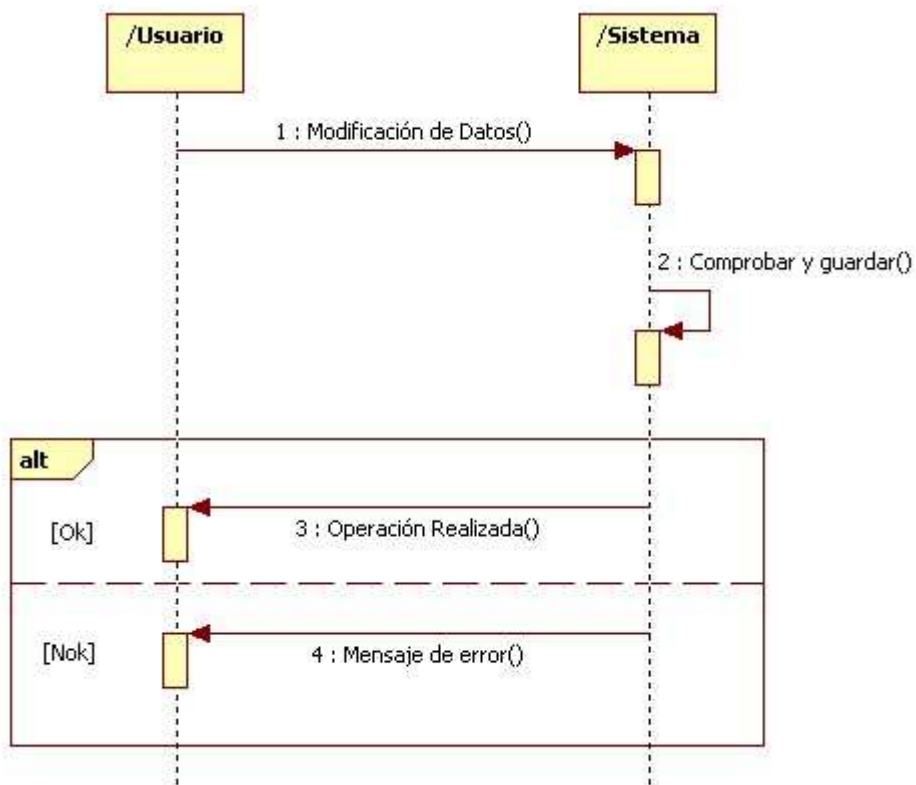


Figura 30: Caso de uso UC-01, subcaso 03. Diagrama de interacción

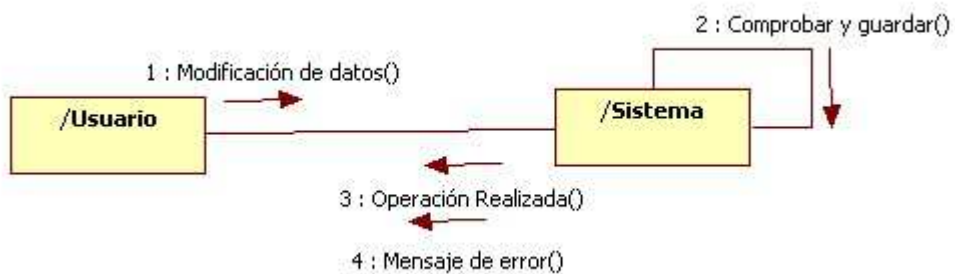


Figura 31: Caso de uso UC-01, subcaso 03. Diagrama de colaboración

| | | |
|-----------------------------|--|--|
| UC-02 | <i>Gestionar Datos Visitas Paciente</i> | |
| Objetivos asociados | OBJ-02 Gestión del historial clínico del paciente OBJ-03 Interfaz amigable | |
| Requisitos asociados | IRQ-02 Información de visitas clínicas del paciente | |
| Descripción | El sistema guardará los datos correspondientes a las visitas a la clínica del paciente introducidos a través del formulario en la base de datos. | |
| Precondición | Seleccionar en el menú principal la opción de Gestionar Pacientes y abrir la ficha de algún paciente. | |
| Secuencia normal | <i>P1</i> | El usuario introduce los datos de la visita realizada por el paciente en el formulario y presiona el botón adecuado. |
| | <i>P2</i> | El sistema comprueba que los datos sean al menos los mínimos requeridos para su guardado y que estén en el formato correcto. |
| | <i>P3</i> | Se le muestra al usuario una confirmación de que la acción se ha realizado de forma satisfactoria. |
| Poscondición | Los datos de la visita médica del paciente han sido guardados dentro de su ficha en la base de datos. | |
| Excepciones | En el caso de que la consolidación de la información a la base de datos no se realice correctamente se muestra un dialogo al usuario. | |
| Estabilidad | Alta | |
| Comentarios | Ninguno | |

Figura 32: Caso de uso UC-02

| | | |
|-----------------------------|---|--|
| SUBCASO DE USO-01 | <i>Agregar Visita</i> | |
| Objetivos asociados | OBJ-02 Gestión del historial clínico del paciente OBJ-03 Interfaz amigable | |
| Requisitos asociados | IRQ-02 Información de visitas clínicas del paciente | |
| Descripción | El sistema guardará los datos correspondientes a la última visita a la clínica realizada por el paciente en la base de datos. | |
| Precondición | Seleccionar en el menú principal la opción de Gestionar Pacientes, abrir la ficha de algún paciente y presionar el botón "Añadir Consulta". | |
| Secuencia normal | <i>P1</i> | El usuario introduce los datos de la visita a la clínica realizada por el paciente en el formulario y presiona el botón "Añadir Consulta". |
| | <i>P2</i> | El sistema comprueba que los datos sean al menos los mínimos requeridos para su guardado y que estén en el formato correcto. |
| | <i>P3</i> | Se le muestra al usuario una confirmación de que la acción se ha realizado de forma satisfactoria. |
| Poscondición | Los datos de la visita médica del paciente han sido guardados dentro de su ficha en la base de datos. | |
| Excepciones | En el caso de que la consolidación de la información a la base de datos no se realice correctamente se muestra un dialogo al usuario. | |
| Estabilidad | Alta | |
| Comentarios | Ninguno | |

Figura 33: Caso de uso UC-02, subcaso 01

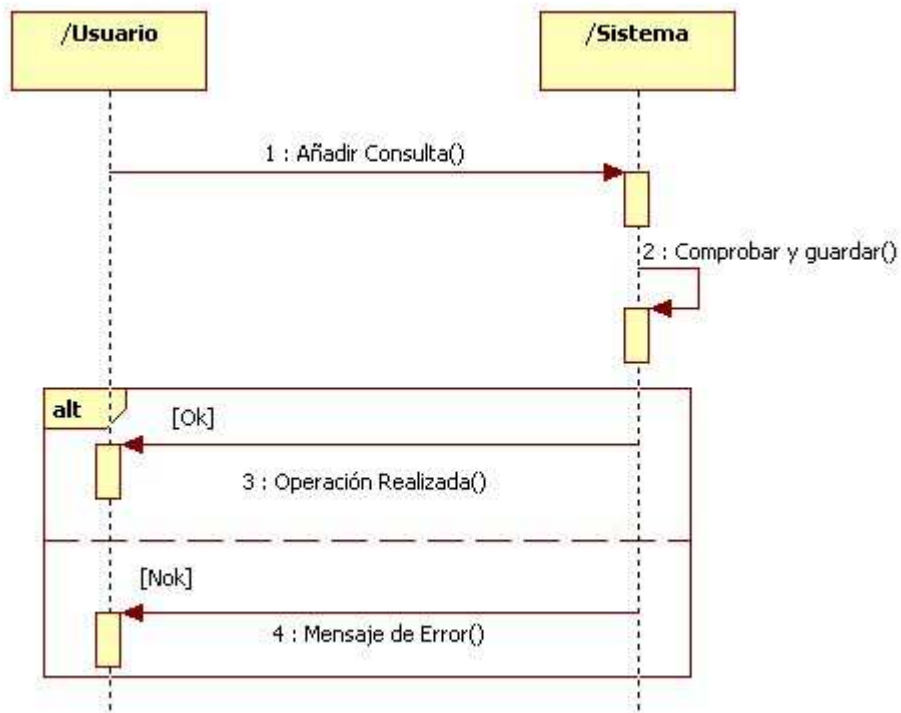


Figura 34: Caso de uso UC-02, subcaso 01. Diagrama de interacción

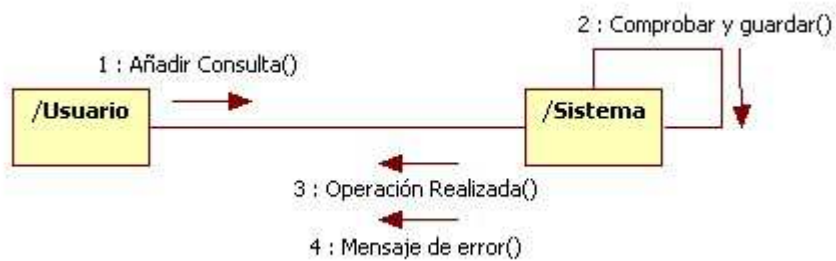


Figura 35: Caso de uso UC-02, subcaso 01. Diagrama de colaboración

| | | |
|-----------------------------|--|---|
| SUBCASO DE USO-02 | <i>Consultar Visitas</i> | |
| Objetivos asociados | OBJ-02 Gestión del historial clínico del paciente OBJ-03 Interfaz amigable | |
| Requisitos asociados | IRQ-02 Información de visitas clínicas del paciente | |
| Descripción | El sistema mostrará los datos correspondientes a la visita a la clínica realizada por el paciente solicitada por el usuario que se encuentran en la base de datos. | |
| Precondición | Seleccionar en el menú principal la opción de Gestionar Pacientes, abrir la ficha de algún paciente y hacer doble clic sobre alguna consulta de su historial. | |
| Secuencia normal | P1 | El usuario selecciona la consulta de su interés de la ficha del usuario y hace doble clic sobre ella. |
| | P2 | El sistema busca los datos de la consulta en cuestión en la base de datos. |
| | P3 | Se le muestran al usuario los datos de la consulta solicitada. |
| Poscondición | Los datos de la visita médica del paciente se muestran en el formulario. | |
| Excepciones | Ninguna. | |
| Estabilidad | Alta | |
| Comentarios | Ninguno | |

Figura 36: Caso de uso UC-02, subcaso 02

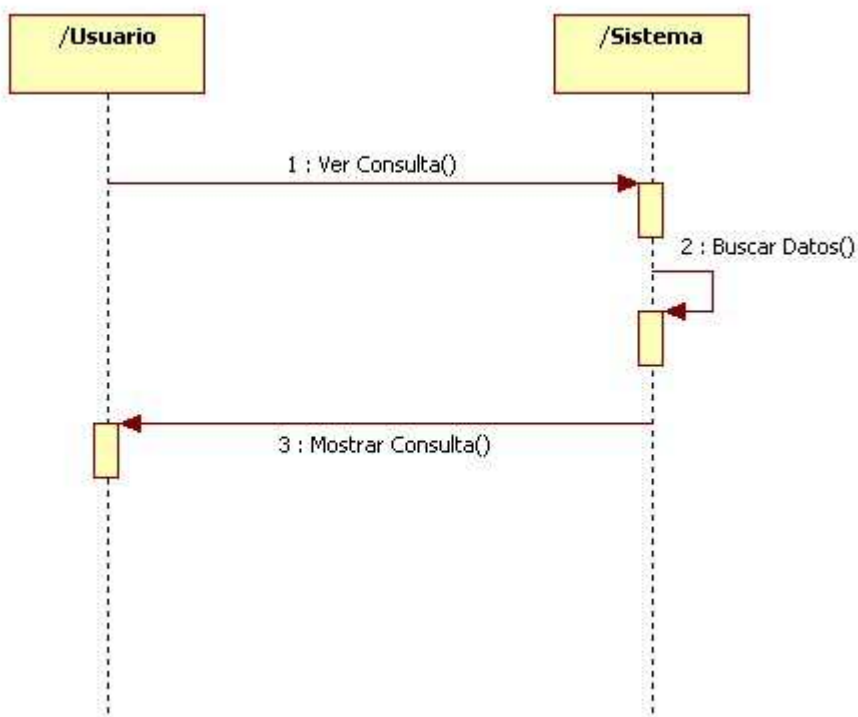


Figura 37: Caso de uso UC-02, subcaso 02. Diagrama de interacción

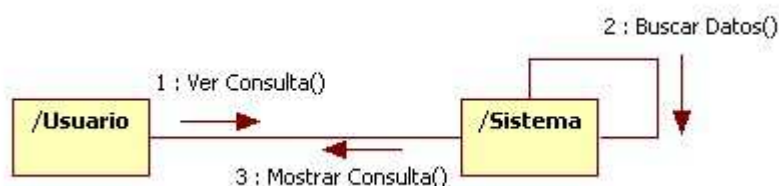


Figura 38: Caso de uso UC-02, subcaso 02. Diagrama de colaboración

| | | |
|-----------------------------|---|--|
| SUBCASO DE USO-03 | <i>Modificar Visita</i> | |
| Objetivos asociados | OBJ-02 Gestión del historial clínico del paciente OBJ-03 Interfaz amigable | |
| Requisitos asociados | IRQ-02 Información de visitas clínicas del paciente | |
| Descripción | El sistema guardará los nuevos datos correspondientes a la visita a la clínica seleccionada por el usuario en la base de datos. | |
| Precondición | Seleccionar en el menú principal la opción de Gestionar Pacientes, abrir la ficha de algún paciente y hacer doble clic sobre alguna consulta de su historial. | |
| Secuencia normal | <i>P1</i> | El usuario modifica los datos de la visita a la clínica realizada por el paciente en el formulario y presiona el botón "Modificar Consulta". |
| | <i>P2</i> | El sistema comprueba que los datos sean al menos los mínimos requeridos para su guardado y que estén en el formato correcto. |
| | <i>P3</i> | Se le muestra al usuario una confirmación de que la acción se ha realizado de forma satisfactoria. |
| Poscondición | Los nuevos datos de la visita médica del paciente han sido guardados dentro de su ficha en la base de datos. | |
| Excepciones | En el caso de que la consolidación de la información a la base de datos no se realice correctamente se muestra un dialogo al usuario. | |
| Estabilidad | Alta | |
| Comentarios | Ninguno | |

Figura 39: Caso de uso UC-02, subcaso 03

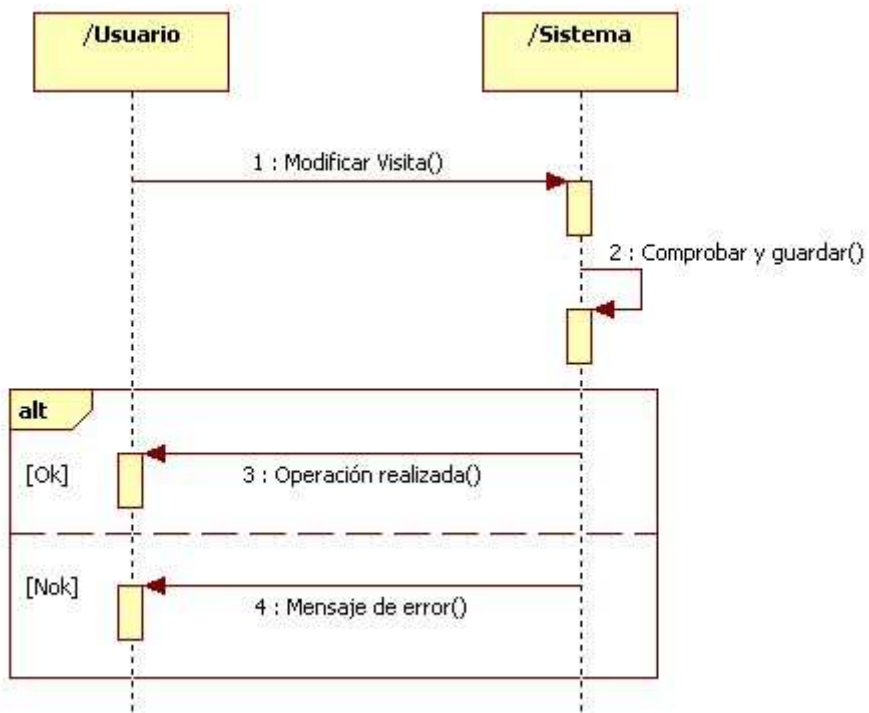


Figura 40: Caso de uso UC-02, subcaso 03. Diagrama de interacción



Figura 41: Caso de uso UC-02, subcaso 03. Diagrama de colaboración

| | | |
|-----------------------------|--|--|
| UC-03 | <i>Gestionar Agenda Clínica</i> | |
| Objetivos asociados | OBJ-03 Interfaz amigable OBJ-04 Gestión de citas de consulta | |
| Requisitos asociados | IRQ-03 Información de citas de la clínica | |
| Descripción | El sistema mostrará y guardará los datos correspondientes a la agenda diaria de la clínica introducidos a través del formulario en la base de datos. | |
| Precondición | Seleccionar en el menú principal la opción de Gestionar Citas. | |
| Secuencia normal | <i>P1</i> | El usuario introduce los datos de la agenda correspondientes al día en cuestión a través del formulario. |
| | <i>P2</i> | El sistema guarda los datos en la base de datos. |
| | <i>P3</i> | Se le muestra al usuario una confirmación de que la acción se ha realizado de forma satisfactoria. |
| Poscondición | Los datos de la agenda del día han sido guardados en la base de datos. | |
| Excepciones | En el caso de que la consolidación de la información a la base de datos no se realice correctamente se muestra un dialogo al usuario. | |
| Estabilidad | Alta | |
| Comentarios | Ninguno | |

Figura 42: Caso de uso UC-03

| | | |
|-----------------------------|---|--|
| SUBCASO DE USO-01 | <i>Ver Agenda Día</i> | |
| Objetivos asociados | OBJ-03 Interfaz amigable OBJ-04 Gestión de citas de consulta | |
| Requisitos asociados | IRQ-03 Información de citas de la clínica | |
| Descripción | El sistema mostrará los datos correspondientes a la agenda diaria de la clínica del día seleccionado almacenados en la base de datos. | |
| Precondición | Seleccionar en el menú principal la opción de Gestionar Citas y elegir un día en el calendario. Por defecto se muestra el día actual. | |
| Secuencia normal | <i>P1</i> | El usuario selecciona el día que quiere ver en el calendario. |
| | <i>P2</i> | El sistema busca la agenda del día seleccionado en la base de datos. |
| | <i>P3</i> | Se le muestra en el formulario la relación de citas del día en cuestión. |
| Poscondición | Los datos de la agenda se muestran en el formulario. | |
| Excepciones | Ninguna. | |
| Estabilidad | Alta | |
| Comentarios | Ninguno | |

Figura 43: Caso de uso UC-03, subcaso 01

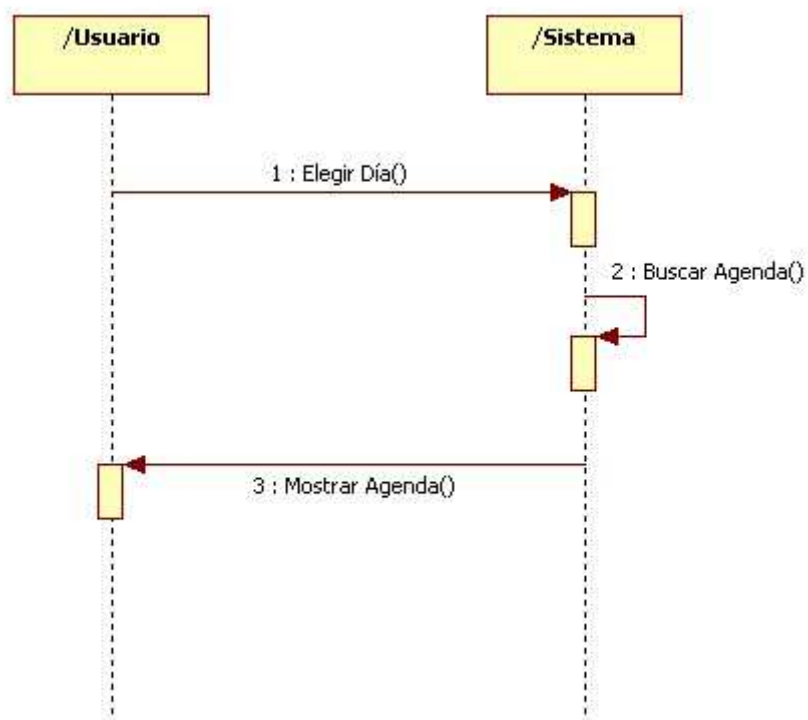


Figura 44: Caso de uso UC-03, subcaso 01. Diagrama de interacción

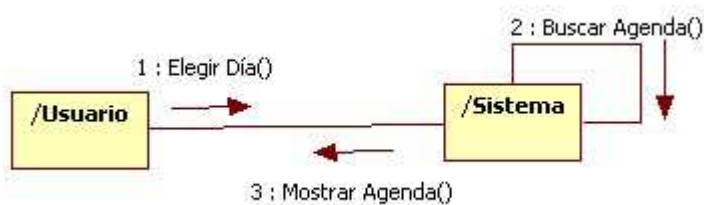


Figura 45: Caso de uso UC-03, subcaso 01. Diagrama de colaboración

| | | |
|-----------------------------|---|--|
| SUBCASO DE USO-02 | <i>Guardar Agenda Día</i> | |
| Objetivos asociados | OBJ-03 Interfaz amigable OBJ-04 Gestión de citas de consulta | |
| Requisitos asociados | IRQ-03 Información de citas de la clínica | |
| Descripción | El sistema guardará los datos correspondientes a la agenda diaria de la clínica introducidos por el usuario en la base de datos. | |
| Precondición | Seleccionar en el menú principal la opción de Gestionar Citas, elegir un día en el calendario y presionar el botón "Guardar Cambios". Por defecto se muestra el día actual. | |
| Secuencia normal | <i>P1</i> | El usuario introduce en el formulario los datos de la agenda, seleccionando los datos de pacientes de la lista o introduciendo otros nuevos. |
| | <i>P2</i> | El sistema guarda la agenda del día seleccionado en la base de datos. |
| | <i>P3</i> | Se le muestra al usuario una confirmación de que la acción se ha realizado de forma satisfactoria. |
| Poscondición | Los datos de la agenda del día se han guardado y se muestran en el formulario. | |
| Excepciones | Ninguna | |
| Estabilidad | Alta | |
| Comentarios | Ninguno | |

Figura 46: Caso de uso UC-03, subcaso 02

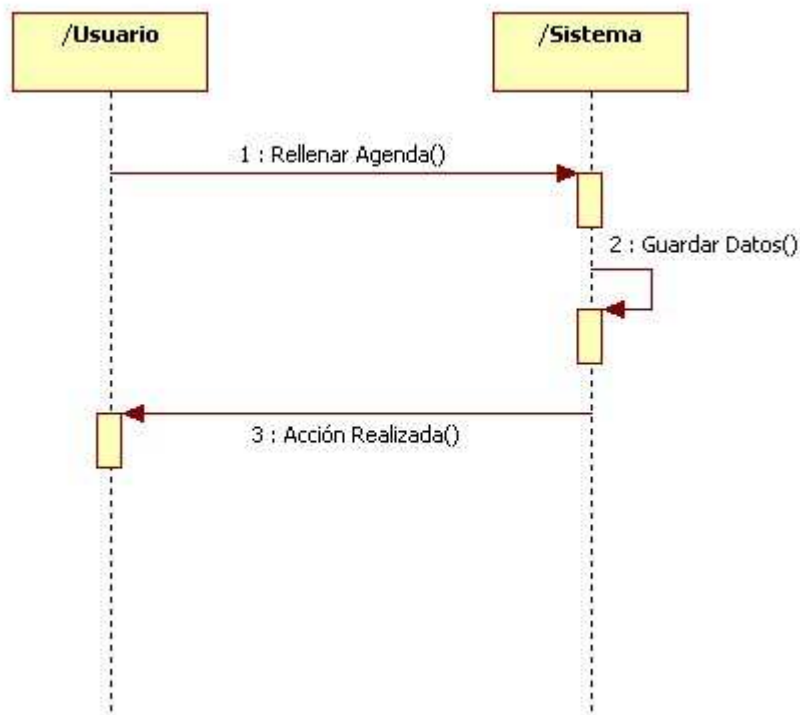


Figura 47: Caso de uso UC-03, subcaso 02. Diagrama de interacción



Figura 48: Caso de uso UC-03, subcaso 02. Diagrama de colaboración

| | | |
|-----------------------------|--|--|
| SUBCASO DE USO-03 | <i>Agregar Cita Compuesta</i> | |
| Objetivos asociados | OBJ-03 Interfaz amigable OBJ-04 Gestión de citas de consulta | |
| Requisitos asociados | IRQ-03 Información de citas de la clínica | |
| Descripción | El sistema guardará los nuevos datos correspondientes a la cita introducida por el usuario en la base de datos | |
| Precondición | Seleccionar en el menú principal la opción de Gestionar Citas y presionar el botón "Nueva Cita..." | |
| Secuencia normal | <i>P1</i> | El usuario introduce los datos de la cita en el formulario y presiona el botón "Añadir Cita". |
| | <i>P2</i> | El sistema comprueba que los datos sean al menos los mínimos requeridos para su guardado y que estén en el formato correcto. |
| | <i>P3</i> | Se le muestra al usuario una confirmación de que la acción se ha realizado de forma satisfactoria. |
| Poscondición | Los datos de la cita han sido guardados dentro de la agenda del día correspondiente en la base de datos. | |
| Excepciones | En el caso de que la consolidación de la información a la base de datos no se realice correctamente se muestra un dialogo al usuario. | |
| Estabilidad | Alta | |
| Comentarios | Por defecto las citas que se agregan a través del formulario principal son en un horario determinado (17:00-20:00) y tienen una duración de 15 minutos, pero a veces puede haber citas fuera de ese horario y que tengan otra duración por ser una actividad específica (operaciones, estudios biomecánicos, etc.). Este caso de uso permite guardar una cita de ese tipo. | |

Figura 49: Caso de uso UC-03, subcaso 03

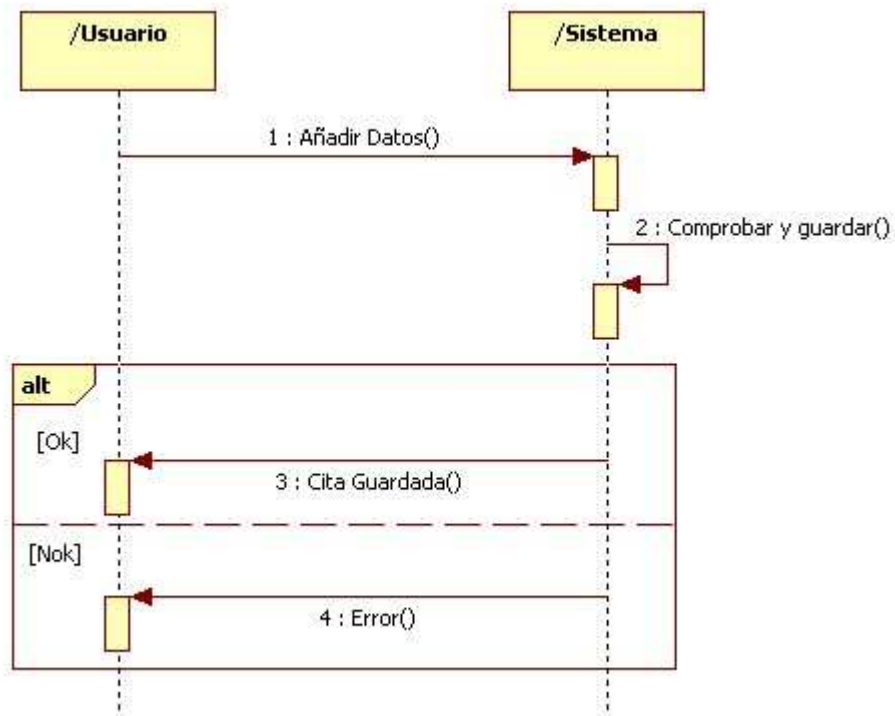


Figura 50: Caso de uso UC-03, subcaso 03. Diagrama de interacción

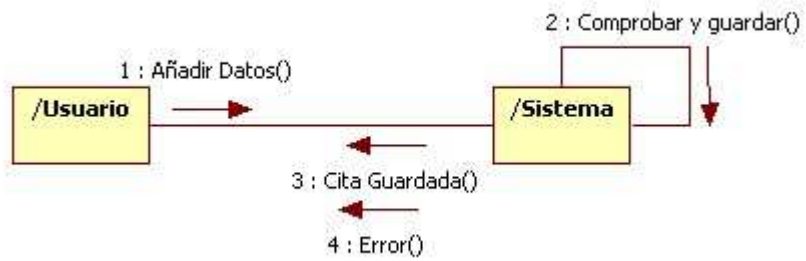


Figura 51: Caso de uso UC-03, subcaso 03. Diagrama de colaboración

| | | |
|-----------------------------|---|--|
| UC-04 | <i>Visualizar Informes de Actividad</i> | |
| Objetivos asociados | OBJ-03 Interfaz amigable OBJ-04 Presentación de balances | |
| Requisitos asociados | IRQ-04 Información de balances de la clínica | |
| Descripción | El sistema mostrará al usuario diferentes balances de actividad de la clínica según las opciones seleccionadas. | |
| Precondición | Seleccionar en el menú principal la opción de Estadísticas. | |
| Secuencia normal | <i>P1</i> | El usuario selecciona en el formulario el tipo de balance deseado y presiona el botón "Mostrar datos". |
| | <i>P2</i> | El sistema realiza una consulta en la base de datos según las opciones que el usuario ha elegido. |
| | <i>P3</i> | Se muestra en el formulario el informe solicitado. |
| Poscondición | Los datos del informe aparecen en el formulario. | |
| Excepciones | Ninguna. | |
| Estabilidad | Alta | |
| Comentarios | Ninguno | |

Figura 52: Caso de uso UC-04

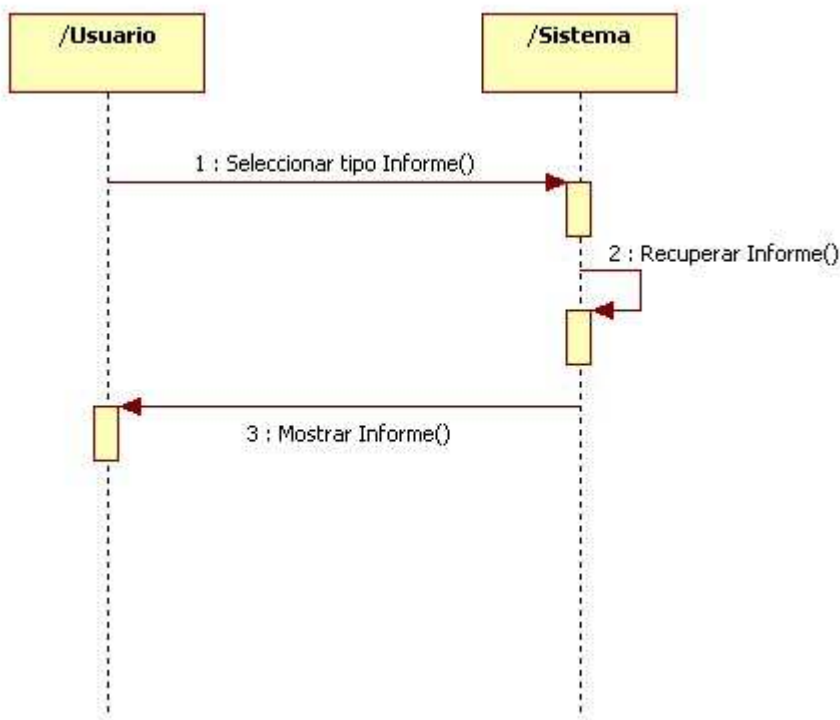


Figura 53: Caso de uso UC-04. Diagrama de interacción

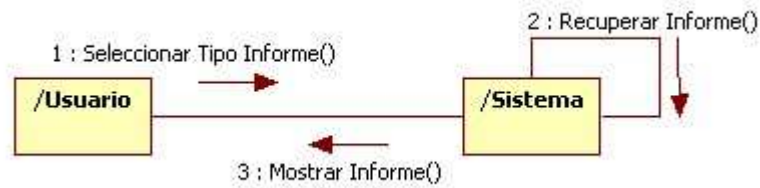


Figura 54: Caso de uso UC-04. Diagrama de colaboración

2. DISEÑO

2.1. COMPORTAMIENTO

2.1.1. INTRODUCCIÓN

En la fase de análisis se definieron los requisitos del sistema, es decir, lo que el sistema debe hacer. En la fase que abordaremos a continuación se procede a su interpretación y realización física, esto es, cómo funciona el sistema. Para ello se realizó un diagrama de clases que describe las entidades lógicas que componen nuestro programa y otros diagramas que nos ayudan a entender el funcionamiento de la aplicación.



2.1.3. DESCRIPCIÓN DE LAS CLASES

A continuación se detallarán cada una de las clases del diagrama anterior y una pequeña descripción de cada una.

ClinPodApp: Es la clase principal de la aplicación que se encarga de lanzar la misma y mostrar la pantalla principal.

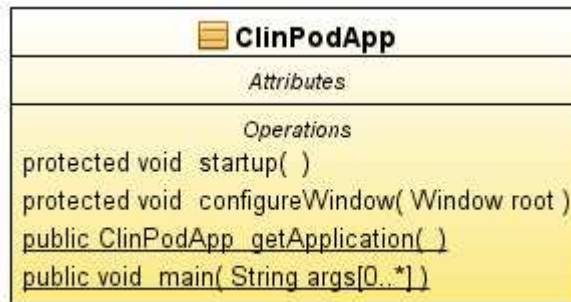


Figura 56: Clase ClinPodApp

ClinPodView: Es la clase que controla la ventana de inicio de la aplicación y los diferentes botones y menús de la misma.

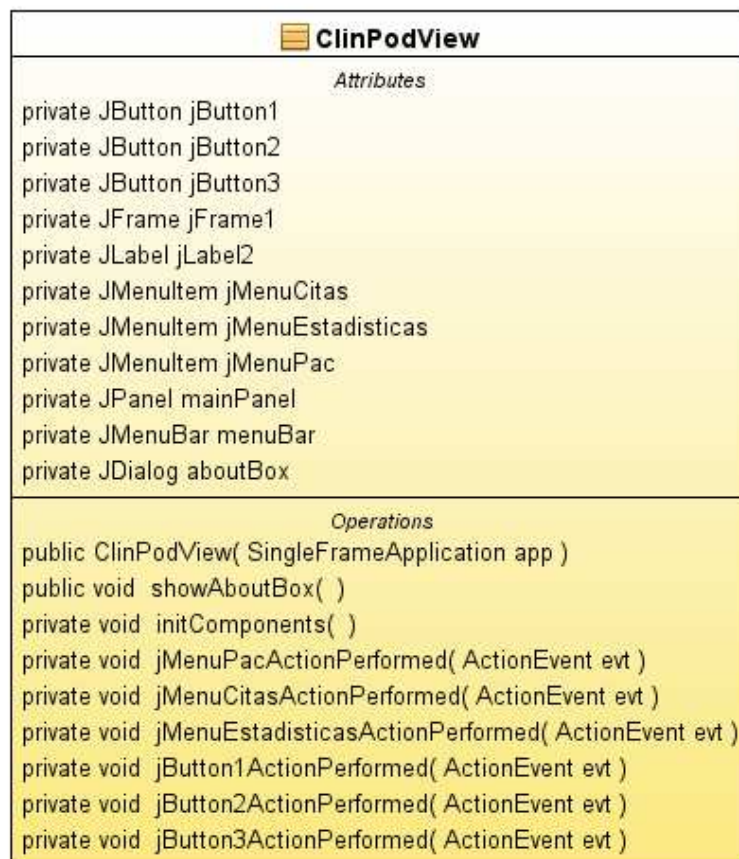


Figura 57: Clase ClinPodView

ClinPodAboutBox: Contiene el dialogo con el "Acerca de..." de la aplicación.



Figura 58: Clase ClinPodAboutBox

FormRelaciones: Clase que muestra el formulario de creación de relaciones entre pacientes para su grabado en la base de datos.

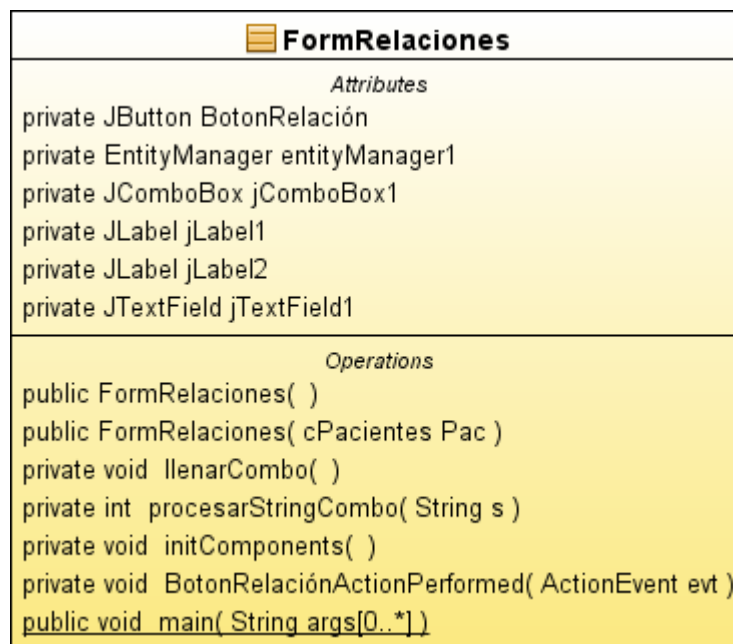


Figura 59: Clase FormRelaciones

FormPacientes: Contiene el formulario de introducción y visualización de los datos de los pacientes.

```

FormPacientes
    Attributes
private DefaultListModel ModeloLista
private DefaultListModel ModeloLista1
private JButton BotonConsultas
private JButton BotonRelaciones
private JComboBox ComboBusqueda
private JComboBox ComboSexo
private JLabel LabelHistoria
private JList ListaActos
private JList ListaRelacionados
private JTextField TextoAlergias
private JTextArea TextoAntecedentes
private JTextField TextoApellidos
private JTextField TextoCP
private JTextField TextoDiabetico
private JTextField TextoDireccion
private JTextField TextoLocalidad
private JTextField TextoMovil
private JTextField TextoMutua
private JTextField TextoNombre
private JTextField TextoParticular
private JTextField TextoProvincia
private JTextField TextoRegistro
private JTextField TextoSinrom
private JTextField TextoTrabajo
private JButton botonAgregar
private JButton botonAnterior
private JButton botonLimpiar
private JButton botonModificar
private JButton botonPrimero
private JButton botonSiguiente
private JButton botonUltimo
private EntityManager entityManager1
private JDateChooser jFechaNac
private JLabel jLabel1
private JLabel jLabel10
private JLabel jLabel11
private JLabel jLabel12
private JLabel jLabel13
private JLabel jLabel14
private JLabel jLabel15
private JLabel jLabel16
private JLabel jLabel17
private JLabel jLabel18
private JLabel jLabel19
private JLabel jLabel2
private JLabel jLabel20
private JLabel jLabel21
private JLabel jLabel3
private JLabel jLabel4
private JLabel jLabel5
private JLabel jLabel6
private JLabel jLabel7
private JLabel jLabel8
private JLabel jLabel9
private JScrollPane jScrollPane1
private JScrollPane jScrollPane2
private JScrollPane jScrollPane3

    Operations
public FormPacientes( )
private void initComponents( )
private void botonAgregarActionPerformed( ActionEvent evt )
private void botonModificarActionPerformed( ActionEvent evt )
private void botonAgregarFocusLost( FocusEvent evt )
private void botonModificarFocusLost( FocusEvent evt )
private void TextoRegistroActionPerformed( ActionEvent evt )
private void botonPrimeroActionPerformed( ActionEvent evt )
private void botonUltimoActionPerformed( ActionEvent evt )
private void botonAnteriorActionPerformed( ActionEvent evt )
private void botonSiguienteActionPerformed( ActionEvent evt )
private void botonLimpiarActionPerformed( ActionEvent evt )
private void BotonRelacionesActionPerformed( ActionEvent evt )
private void ListaRelacionadosMouseClicked( MouseEvent evt )
private void BotonConsultasActionPerformed( ActionEvent evt )
private void ListaActosMouseClicked( MouseEvent evt )
private Date ParsearFecha( )
private String FechaToString( Date d )
private cPacientes ObtenerDatos( )
private void llenarForm( cPacientes Pac )
private void limpiarForm( )
private void llenarCombo( )
private int procesarStringCombo( String s )
private void iniciarListaRelaciones( )
private void llenarListaRelaciones( )
private void iniciarListaConsultas( )
private void llenarListaConsultas( )
public void main( String args[0..*] )
    
```

Figura 60: Clase FormPacientes

FormConsultas: Subformulario de FormPacientes que permite introducir y modificar visitas médicas en la ficha de un paciente.


|  FormConsultas |
|---|
| <i>Attributes</i> |
| private int id private JButton BotonAñad private JButton BotonMod private JComboBox ComboActo private EntityManager entityManager1 private JDateChooser jFechaActo private JDateChooser jFechaPago private JTextField jlImporte private JLabel jLabel1 private JLabel jLabel2 private JLabel jLabel3 private JLabel jLabel4 private JLabel jLabel5 private JLabel jLabel6 private JTextArea jPrestación private JScrollPane jScrollPane1 |
| <i>Operations</i> |
| public FormConsultas() public FormConsultas(String S, int I) public FormConsultas(String S, int I, cHistorial H) private void initComponents() private void BotonAñadActionPerformed((ActionEvent evt) private void BotonModActionPerformed((ActionEvent evt) <u>public void main(String args[0..*])</u> |

Figura 61: Clase FormConsultas

FormAgenda: Contiene el formulario con la agenda de citas medicas de la aplicación.


|  FormAgenda |
|---|
| <i>Attributes</i> |
| private WindowListener wl private JButton BotonAct private JButton BotonAnt private JButton BotonGuardar private JButton BotonNuevo private JButton BotonSig private JComboBox ComboTabla private JTable TablaCitas private EntityManager entityManager1 private JDateChooser jFechaAgenda private JLabel jLabel1 private JScrollPane jScrollPane1 |
| <i>Operations</i> |
| public FormAgenda() private void initComponents() private void BotonAntActionPerformed(ActionEvent evt) private void BotonSigActionPerformed(ActionEvent evt) private void BotonGuardarActionPerformed(ActionEvent evt) private void BotonNuevoActionPerformed(ActionEvent evt) private void BotonGuardarFocusLost(FocusEvent evt) private void BotonActActionPerformed(ActionEvent evt) private void vaciarTabla() private int procesarString(String s) private int procesarStringMin(String s) private void ordenarTabla() private void llenarTabla() private void llenarCombo() <u>public void main(String args[0..*])</u> public void WindowClosing(ActionEvent evt) |

Figura 62: Clase FormAgenda

FormCitaNueva: Subformulario de FormConsultas que permite crear una cita en una fecha y horarios determinados y con una duración a elegir para su guardado en la agenda de la aplicación.

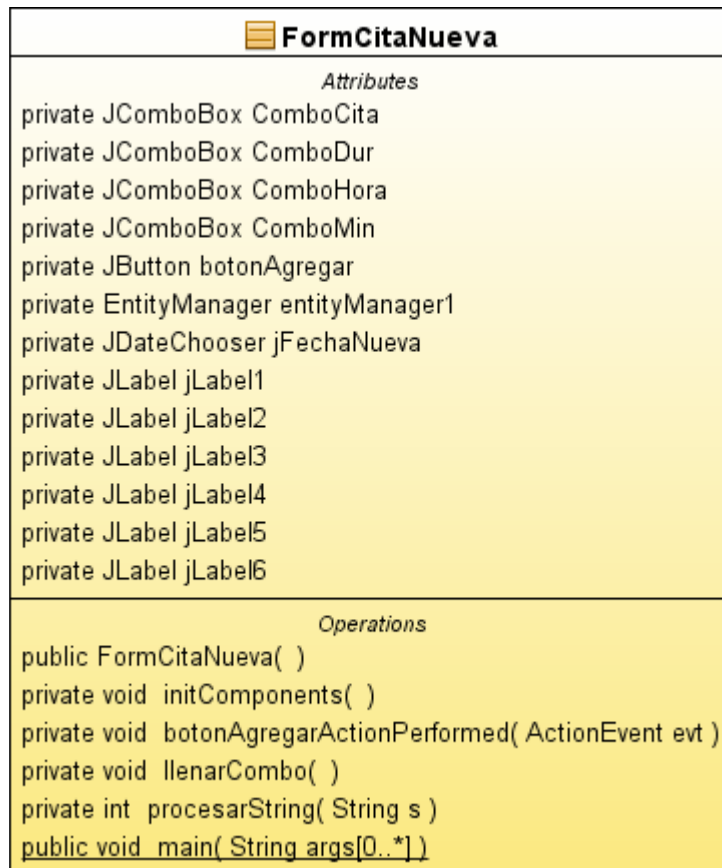


Figura 63: Clase FormCitaNueva

FormEstadisticas: Clase que contiene el formulario de visualización de informes por día, mes, año o totales


|  FormEstadisticas |
|--|
| <i>Attributes</i> |
| <pre> private JTable TablaRes private JYearChooser añoAño private JButton botonMostrar private EntityManager entityManager1 private JDateChooser fechaDia private ButtonGroup grupoFechas private ButtonGroup grupoTipo private JRadioButton jActos private JRadioButton jAño private JRadioButton jDia private JLabel jlImporte private JLabel jLabel1 private JLabel jLabel2 private JLabel jLabel3 private JLabel jLabel4 private JLabel jLabel5 private JRadioButton jMes private JRadioButton jPaciente private JRadioButton jPendiente private JScrollPane jScrollPane1 private JRadioButton jTodo private JLabel jTotal private JYearChooser mesAño private JMonthChooser mesMes </pre> |
| <i>Operations</i> |
| <pre> public FormEstadisticas() private void initComponents() private void botonMostrarActionPerformed(ActionEvent evt) public void CrearModelo1() public void CrearModelo2() public void CrearModelo3() public void DiaActos() public void DiaPacientes() private void DiaPendientes() private void MesActos() private void MesPacientes() private void MesPendientes() private void AñoActos() private void AñoPacientes() private void AñoPendientes() private void TodoActos() private void TodoPacientes() private void TodoPendientes() private String FechaToString(Date d) public void main(String args[0..*]) </pre> |

Figura 64: Clase FormEstadisticas

cPacientes: Esta clase representa la entidad "Paciente" dentro del modelo de nuestra aplicación.

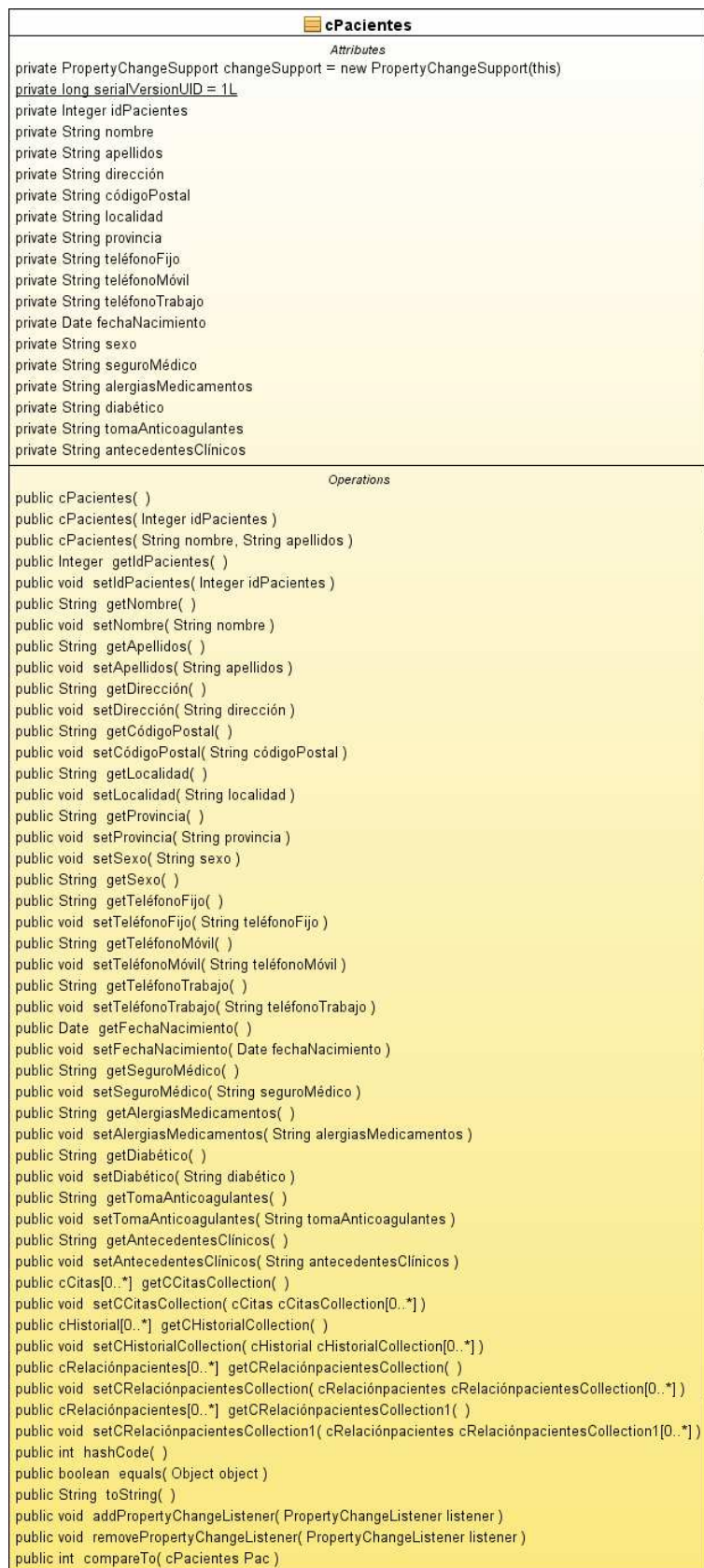


Figura 65: Clase cPacientes

cCitas: Clase que representa la entidad "Cita" en el modelo de la aplicación.



Figura 66: Clase cCitas

cCitasPK: Esta clase representa la clave primaria compuesta de la entidad "Cita".

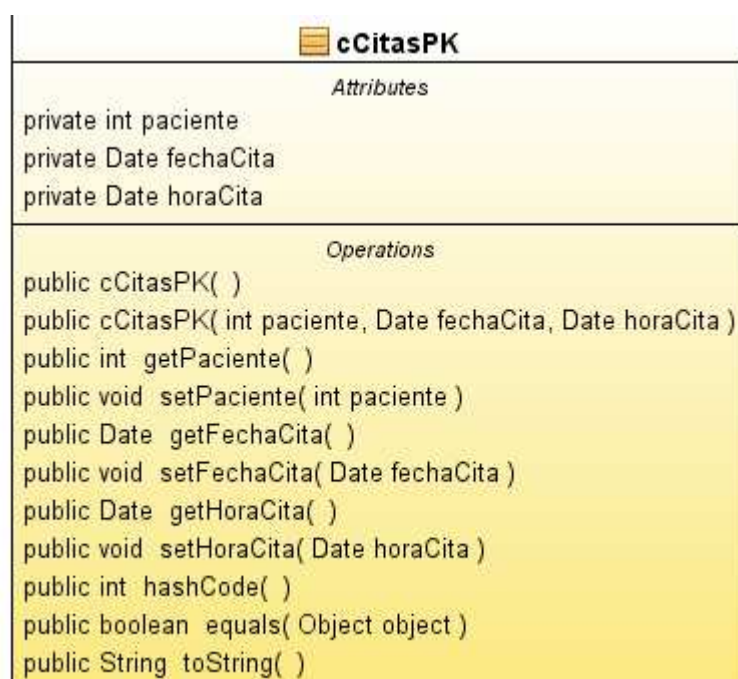


Figura 67: Clase cCitasPK

cHistorial: Clase que representa la entidad "Historial" de nuestra aplicación.

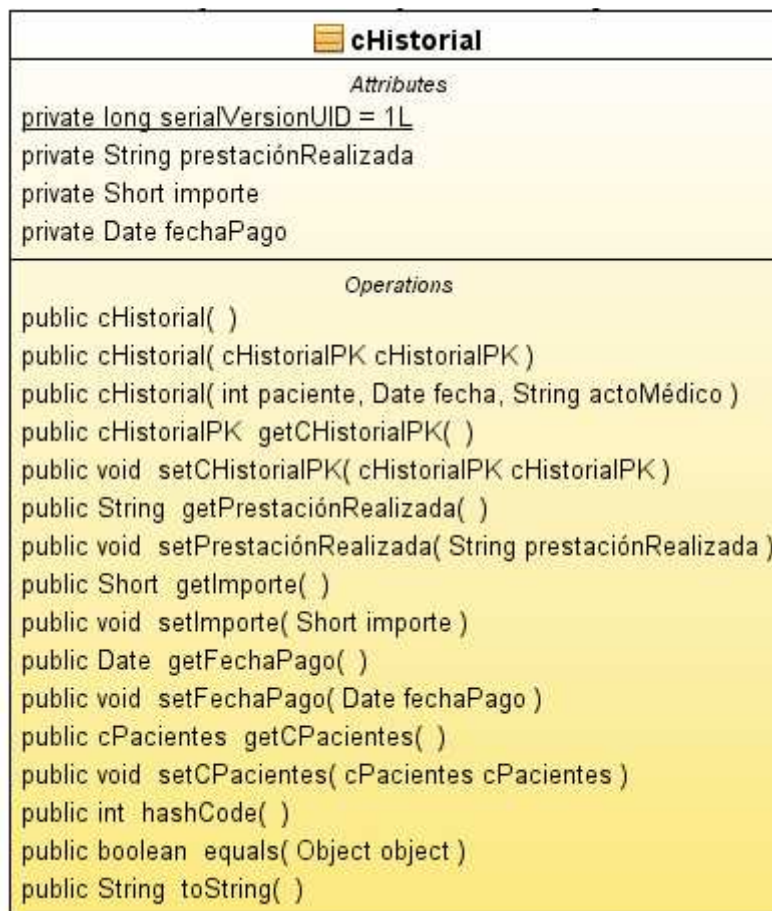


Figura 68: Clase cHistorial

cHistorialPK: Esta clase representa la clave primaria compuesta de la entidad Historial de la aplicación.

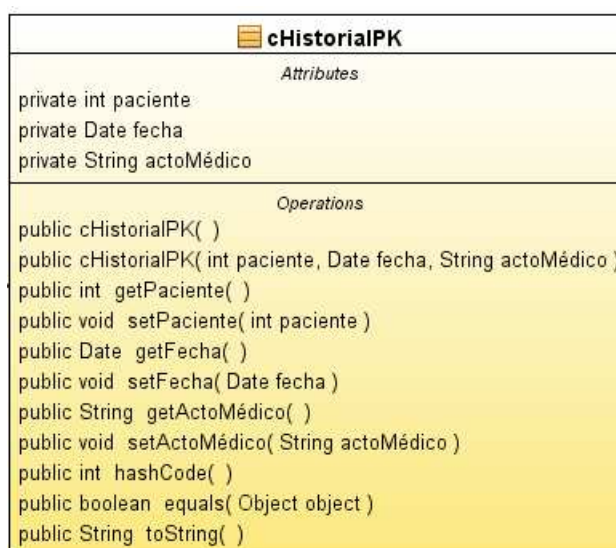


Figura 69: Clase cHistorialPK

cRelaciónPacientes: Esta clase representa la relación existente entre dos entidades de tipo "Pacientes".


|  cRelaciónpacientes |
|--|
| <i>Attributes</i> |
| <pre>private long serialVersionUID = 1L private String parentesco</pre> |
| <i>Operations</i> |
| <pre>public cRelaciónpacientes() public cRelaciónpacientes(cRelaciónpacientesPK cRelaciónpacientesPK) public cRelaciónpacientes(int paciente1 , int paciente2) public cRelaciónpacientesPK getCRelaciónpacientesPK() public void setCRelaciónpacientesPK(cRelaciónpacientesPK cRelaciónpacientesPK) public String getParentesco() public void setParentesco(String parentesco) public cPacientes getCPacientes() public void setCPacientes(cPacientes cPacientes) public cPacientes getCPacientes1() public void setCPacientes1(cPacientes cPacientes1) public int hashCode() public boolean equals(Object object) public String toString()</pre> |

Figura 70: Clase cRelaciónPacientes

cRelaciónPacientesPK: Clase que representa la clave primaria compuesta de la entidad "Relación".


|  cRelaciónpacientesPK |
|--|
| <i>Attributes</i> |
| <pre>private int paciente1 private int paciente2</pre> |
| <i>Operations</i> |
| <pre>public cRelaciónpacientesPK() public cRelaciónpacientesPK(int paciente1 , int paciente2) public int getPaciente1() public void setPaciente1(int paciente1) public int getPaciente2() public void setPaciente2(int paciente2) public int hashCode() public boolean equals(Object object) public String toString()</pre> |

Figura 71: Clase cRelaciónPacientesPK

cCitasJpaController: Clase que contiene el controlador para el manejo de la clase cCitas con la base de datos.

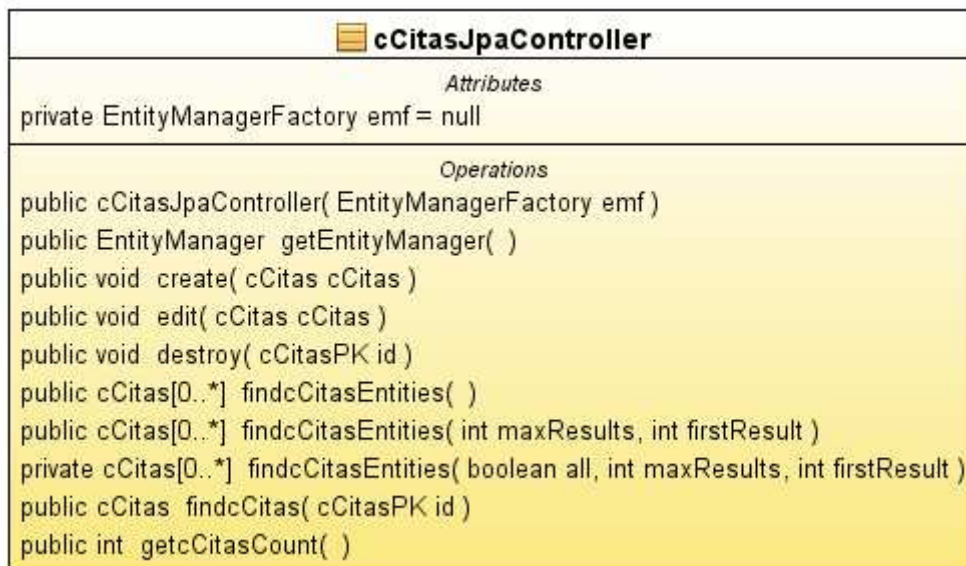


Figura 72: Clase cCitasJpaController

cHistorialJpaController: Clase que contiene el controlador para el manejo de la clase cHistorial con la base de datos.

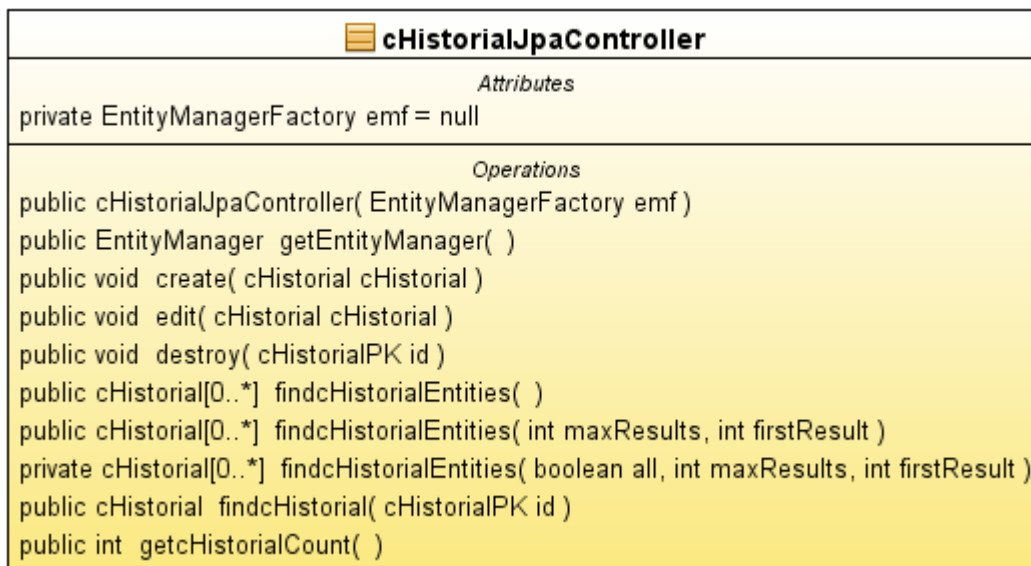


Figura 73: Clase cHistorialJpaController

cPacientesJpaController: Clase que contiene el controlador para el manejo de la clase cPacientes con la base de datos.

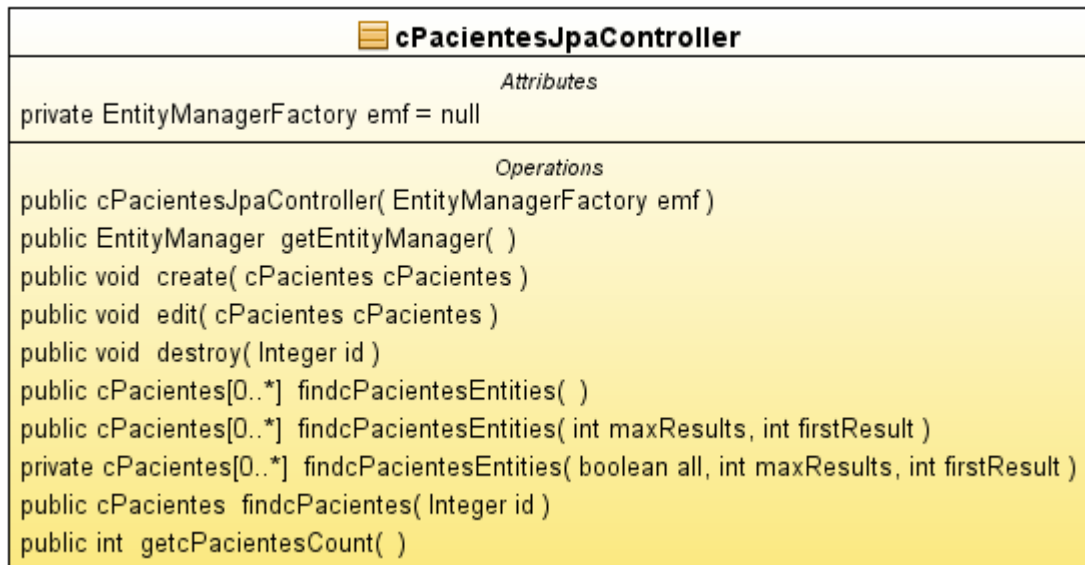


Figura 74: Clase cPacientesJpaController

cRelaciónpacientesJpaController: Clase que contiene el controlador para el manejo de la clase cRelaciónPacientes con la base de datos.

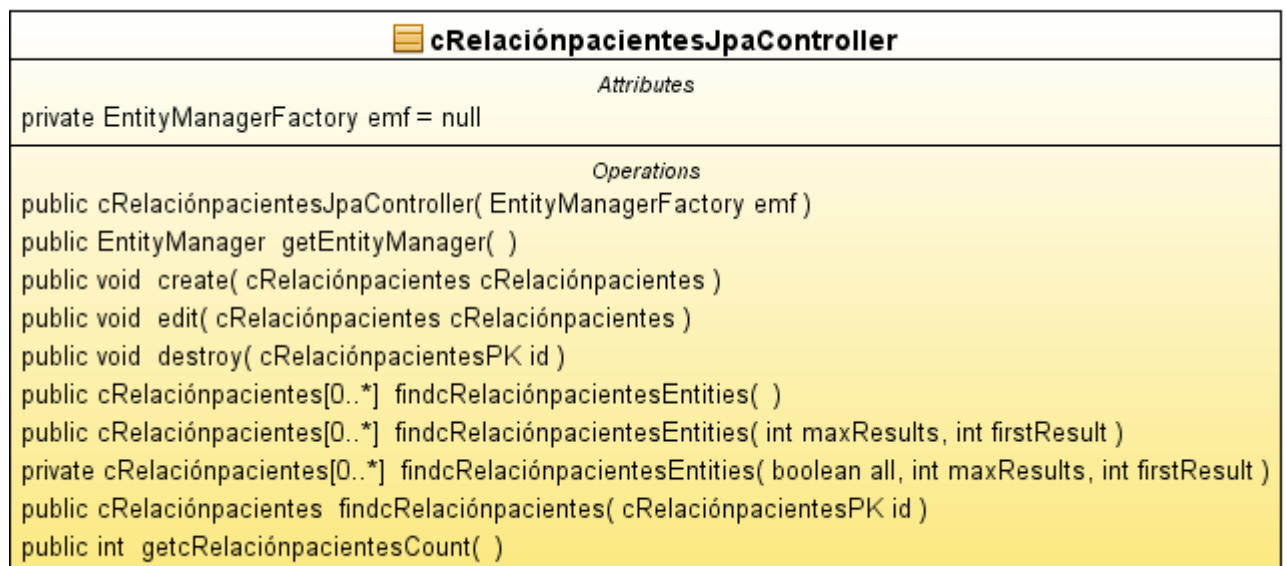


Figura 75: Clase cRelaciónpacientesJpaController

2.1.4. DIAGRAMAS DE ESTADOS

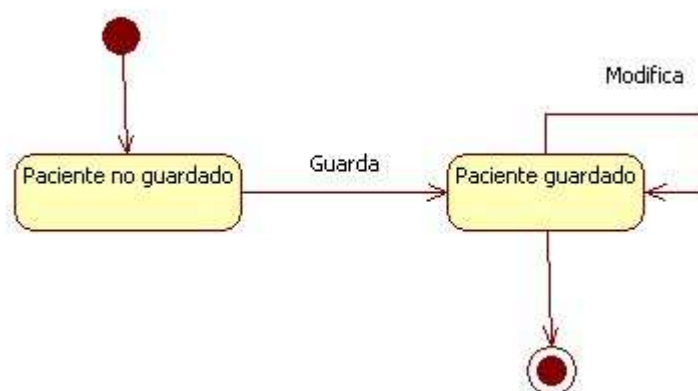


Figura 76: Diagrama de estados paciente



Figura 77: Diagrama de estados cita

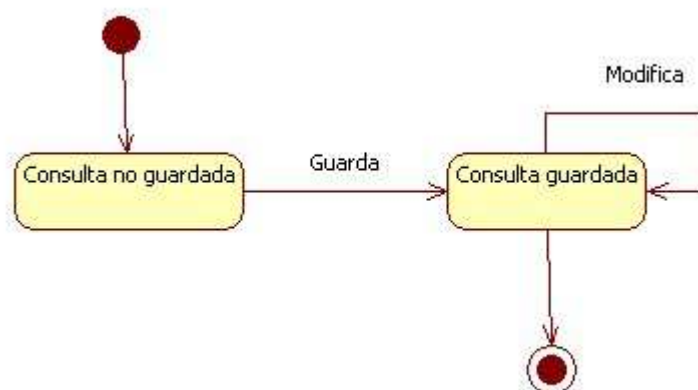


Figura 78: Diagrama de estados consulta

2.2. MODELO DE DATOS

2.2.1. INTRODUCCIÓN

A continuación describiremos el modelo de datos utilizado para la creación de la base de datos de nuestra aplicación. El modelo de datos posteriormente se transformó en clases para el desarrollo de la aplicación como hemos podido ver en los apartados anteriores. Veremos el modelo de datos según el modelo entidad-relación, después lo transformaremos al modelo relacional y por último describiremos las tablas que se crearon finalmente y los campos que contiene cada una de ellas.

2.2.2. MODELO ENTIDAD-RELACIÓN

El modelo entidad-relación es un tipo de diagrama sencillo que describe la realidad de un sistema de información mediante entidades, que serían los sujetos relevantes de dicho sistema y relaciones que expresan las interacciones que puede haber entre las distintas entidades. No se han incluido los campos que contendría cada entidad para una mejor comprensión del diagrama. Dichos campos se detallarán en las secciones siguientes.

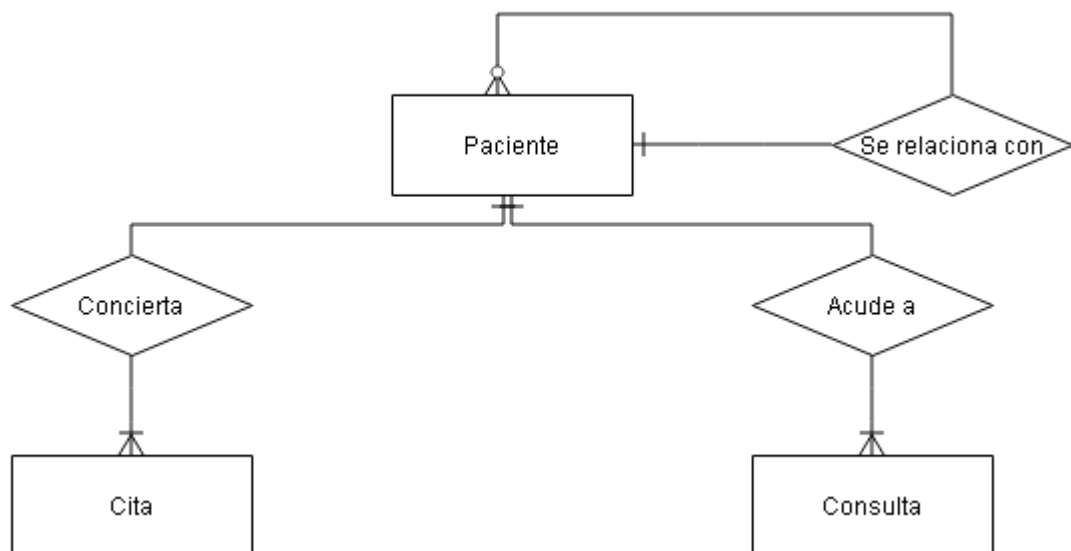


Figura 79: Diagrama Entidad-Relación

2.2.3. MODELO RELACIONAL

Este modelo, al contrario que el anterior, almacena todos los datos mediante relaciones, entendiendo la base de datos como una colección de relaciones. De manera simple, una relación representa una tabla que no es más que un conjunto de filas, cada fila es un conjunto de campos y cada campo representa un valor que interpretado describe el mundo real. Cada fila también se puede denominar tupla o registro y a cada columna también se le puede llamar campo o atributo.

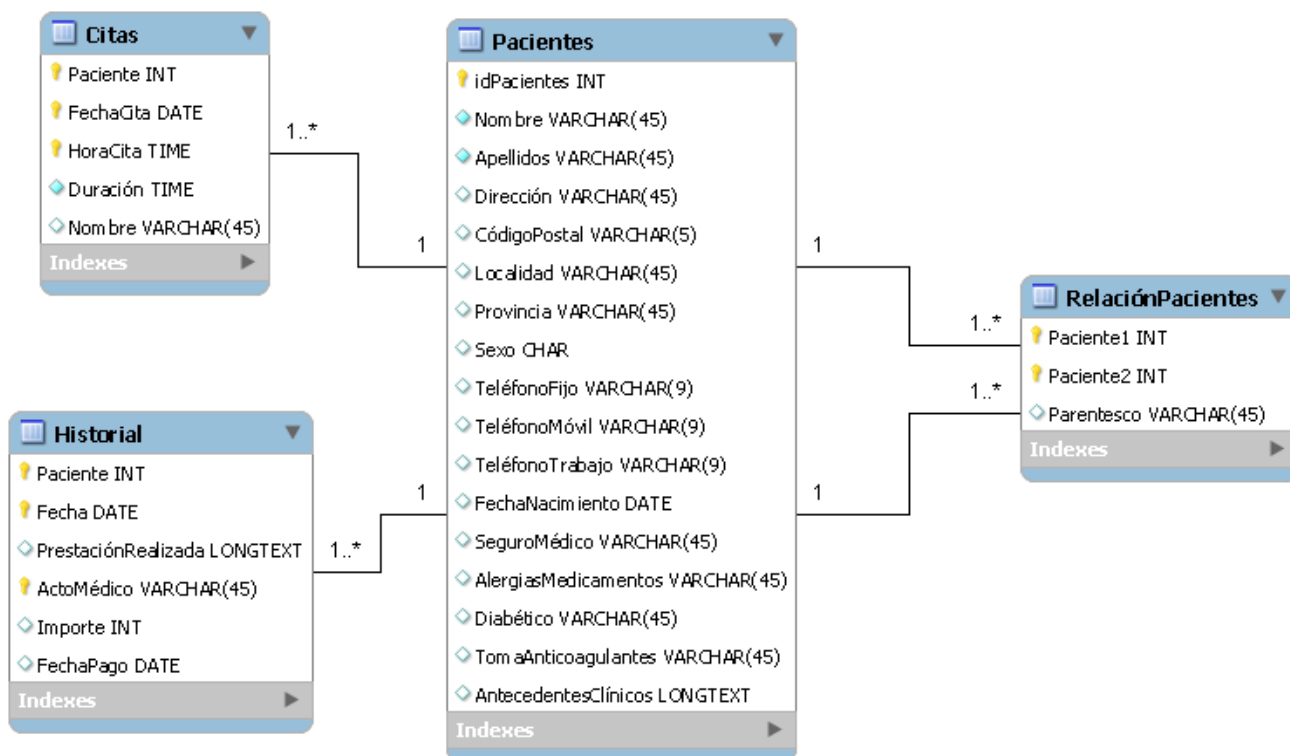


Figura 80: Diagrama Relacional

2.2.4. TRANSFORMACIÓN EN TABLAS

Tabla Pacientes: La tabla pacientes almacena toda la información relativa al paciente, que es fija y no cambia con el tiempo, como los datos personales, de contacto y su historial médico anterior.

| Campo | Tipo | Nulo | Extra | Comentario |
|----------------------|-------------|------|------------------|---|
| <u>idPacientes</u> | INT | No | Auto-incremental | Identificador interno que el paciente tiene dentro de la base de datos |
| Nombre | VARCHAR(45) | No | | |
| Apellidos | VARCHAR(45) | No | | |
| Dirección | VARCHAR(45) | Sí | | |
| CódigoPostal | VARCHAR(5) | Sí | | |
| Localidad | VARCHAR(45) | Sí | | |
| Provincia | VARCHAR(45) | Sí | | |
| Sexo | CHAR | Sí | | Solo se almacena H para Hombres y M para Mujeres |
| TeléfonoFijo | VARCHAR(9) | Sí | | |
| TeléfonoMovil | VARCHAR(9) | Sí | | |
| TeléfonoTrabajo | VARCHAR(9) | Sí | | |
| FechaNacimiento | DATE | Sí | | |
| SeguroMédico | VARCHAR(45) | Sí | | |
| AlergiasMedicamentos | VARCHAR(45) | Sí | | |
| Diabético | VARCHAR(45) | Sí | | Además de si es diabético se anota el tipo de tratamiento (insulina, pastillas, etc.) |
| TomaAnticoagulantes | VARCHAR(45) | Sí | | Anotar el medicamento que toma |
| AntecedentesClínicos | LONGTEXT | Sí | | |

Figura 81: Tabla Pacientes

Tabla Historial: La tabla historial contiene toda la información acerca de las visitas de cada paciente a la clínica, como la fecha y lo que se le realizó, además de la información sobre el pago de dicha consulta.

| Campo | Tipo | Nulo | Extra | Comentario |
|---------------------|-------------|------|-------|---|
| <u>Pacientes</u> | INT | No | | Identificador interno del paciente al que se refiere el registro. |
| <u>Fecha</u> | DATE | No | | |
| <u>ActoMédico</u> | VARCHAR(45) | No | | Genérico (Quiropodia, Uña incarnada, Papiloma) |
| PrestaciónRealizada | LONGTEXT | Sí | | Detalles de la consulta |
| Importe | INT | Sí | | Solo importes enteros |
| FechaPago | DATE | Sí | | |

Figura 82: Tabla Historial

Tabla RelaciónPacientes: La función principal de esta tabla es agrupar pacientes que suelen acudir a consulta juntos y agilizar así el proceso de búsqueda de sus fichas cada vez que nos visiten.

| Campo | Tipo | Nulo | Extra | Comentario |
|------------------|-------------|------|-------|---|
| <u>Paciente1</u> | INT | No | | Identificador interno del paciente |
| <u>Paciente2</u> | INT | No | | Identificador interno del paciente |
| Parentesco | VARCHAR(45) | Sí | | Relación existente entre ambos pacientes. |

Figura 83: Tabla Historial

Tabla Citas: Esta tabla es la encargada de almacenar toda la agenda de citas que usa nuestra aplicación. Cada cita incluye el nombre e identificador del paciente, el día y hora y la duración estimada de la consulta.

| Campo | Tipo | Nulo | Extra | Comentario |
|------------------|-------------|------|-------|---|
| <u>Paciente</u> | INT | No | | Identificador interno del paciente al que se refiere el registro. |
| <u>FechaCita</u> | DATE | No | | |
| <u>HoraCita</u> | TIME | No | | |
| Duración | TIME | No | | |
| Nombre | VARCHAR(45) | Sí | | |

Figura 84: Tabla Citas

3. PRUEBAS

3.1. PRUEBAS DE INSTALACIÓN

La aplicación ha sido probada con éxito en tres ordenadores distintos. En todos los casos la instalación se realizó correctamente y la aplicación funciona sin apreciarse diferencias ni problemas en el uso. A continuación detallamos las características de dichos ordenadores.

- **Ordenador 1:** Intel Core 2 Duo 2.33GHz, 2GB RAM. Sistema operativo Windows XP.
- **Ordenador 2:** Intel 4 2.66GHz, 1.50GB RAM. Sistema operativo Windows XP.
- **Ordenador 3:** Intel Core i7 2.50GHz, 8GB RAM. Sistema operativo Windows 10.

3.2 PRUEBAS DE SISTEMA

Las pruebas de sistema son importantes para verificar y comprobar la calidad de un producto software. Permiten posibles fallos de implementación calidad o usabilidad. Existen dos enfoques principales a la hora de realizar las pruebas.

- Pruebas de caja blanca: se basan en comprobar la estructura interna del programa, recorriendo los diferentes caminos de ejecución del mismo. Se requiere un amplio conocimiento de la aplicación a probar para poder realizarlas bien.
- Pruebas de caja negra: se basan en comprobar únicamente que el programa hace lo que tiene que hacer, analizando las entradas que se realizan a la aplicación y las salidas con las que esta responde. Es el enfoque más utilizado al ser el más cercano a la experiencia del usuario.

En este caso nos interesa solamente probar la robustez del sistema por lo que se han elegido las pruebas de caja negra.

| | |
|------------------------|---|
| PRUEBA DE USO 1 | |
| Objetivo | Comprobar que la aplicación se inicia correctamente. |
| Descripción | Al ejecutar la aplicación debe mostrarse la pantalla inicial de la aplicación con todos sus componentes. |
| Entrada | Abrimos la aplicación desde nuestro ordenador. |
| Salida | Aparece la pantalla de inicio de nuestro programa y todos los botones y etiquetas se muestran satisfactoriamente. |
| Resultado | Se obtiene el resultado esperado. |

Figura 85: Prueba de Uso 1

| | |
|------------------------|--|
| PRUEBA DE USO 2 | |
| Objetivo | Comprobar que el formulario de gestión de pacientes se muestra al pulsar el botón correspondiente. |
| Descripción | Al seleccionar el menú Gestión de Pacientes debe abrirse el formulario correspondiente. |
| Entrada | Presionamos el botón Gestión de Pacientes. |
| Salida | Aparece el formulario que permite ver, agregar y modificar pacientes. |
| Resultado | Se obtiene el resultado esperado. |

Figura 86: Prueba de Uso 2

| | |
|------------------------|---|
| PRUEBA DE USO 3 | |
| Objetivo | Comprobar la creación de un paciente nuevo. |
| Descripción | Al presionar el botón Agregar Paciente el programa debe guardar los datos en la base de datos y devolver un mensaje de confirmación con el identificador de la historia del Paciente. |
| Entrada | Rellenamos el formulario y pulsamos el botón. |
| Salida | Se muestra el mensaje de confirmación y el identificador de usuario. Además al acceder a la base de datos se observa que todos los datos han sido guardados en la misma. |
| Resultado | Se obtiene el resultado esperado. |

Figura 87: Prueba de Uso 3

| | |
|------------------------|---|
| PRUEBA DE USO 4 | |
| Objetivo | Comprobar la modificación de un paciente. |
| Descripción | Al pulsar el botón de Modificar Paciente la aplicación debe comprobar que el identificador existe y guardar los datos. Después mostrará un mensaje de confirmación. |
| Entrada 1 | Abrimos la ficha de un paciente y la modificamos, después apretamos el botón modificar paciente. |
| Salida 1 | Se muestra el mensaje de confirmación. Además al acceder a la base de datos se observa que todos los datos han sido guardados en la misma. |
| Entrada 2 | Desde el formulario vacío se presiona el botón de Modificar Paciente. |
| Salida 2 | Al no detectar ningún identificador de paciente, el programa nos devuelve un mensaje de error. |
| Resultado | Se obtiene el resultado esperado. |

Figura 88: Prueba de Uso 4

| | |
|------------------------|---|
| PRUEBA DE USO 5 | |
| Objetivo | Comprobar la búsqueda por id de un paciente. |
| Descripción | Al introducir en la barra de búsqueda por id del formulario de pacientes un id, la aplicación abrirá la ficha del paciente. En caso de que el id no exista o se deje en blanco se mostrará un error |
| Entrada 1 | Introducimos un id de Paciente existente en la base de datos. |
| Salida 1 | Se muestra la historia y todos los datos personales del paciente con dicho id. |
| Entrada 2 | Introducimos un id no existente en la base de datos. |
| Salida 2 | El sistema nos devuelve un mensaje de error. |
| Entrada 3 | Dejamos en blanco la casilla de búsqueda. |
| Salida 3 | El sistema nos devuelve un mensaje de error. |
| Resultado | Se obtiene el resultado esperado. |

Figura 89: Prueba de Uso 5

| | |
|------------------------|--|
| PRUEBA DE USO 6 | |
| Objetivo | Comprobar la búsqueda por apellidos de un paciente. |
| Descripción | Al introducir en la barra de búsqueda por apellidos del formulario de pacientes los apellidos de un paciente, la aplicación nos irá sugiriendo el resto del nombre y mostrará la ficha del paciente. |
| Entrada | Introducimos varias letras en el cajón de búsqueda por apellidos |
| Salida | El programa nos sugiere el resto del nombre y nos muestra la ficha de dicho paciente. |
| Resultado | Se obtiene el resultado esperado. |

Figura 90: Prueba de Uso 6

| | |
|------------------------|---|
| PRUEBA DE USO 7 | |
| Objetivo | Comprobar que el formulario de gestión de relaciones entre pacientes se muestra al pulsar el botón correspondiente para el paciente actual. |
| Descripción | Al seleccionar el menú Añadir Relaciones... debe abrirse el formulario correspondiente. |
| Entrada 1 | Presionamos el botón Añadir Relaciones... desde la ficha de un paciente. |
| Salida 1 | Aparece el formulario que permite agregar relaciones entre pacientes. |
| Entrada 2 | Presionamos el botón Relaciones sin abrir la ficha de un paciente. |
| Salida 2 | No se muestra el formulario. |
| Resultado | Se obtiene el resultado esperado. |

Figura 91: Prueba de Uso 7

| | |
|------------------------|---|
| PRUEBA DE USO 8 | |
| Objetivo | Comprobar la creación de una nueva relación entre pacientes. |
| Descripción | Al abrir el menú de relaciones y seleccionar un id de otro paciente se añadirá ese paciente como paciente relacionado del actual, junto con la descripción de la relación que se haya dado. |
| Entrada 1 | Abrimos el formulario de relaciones desde la ficha de un paciente, seleccionamos el id de un paciente y rellenamos el parentesco. Después pulsamos el botón de agregar relación |
| Salida 1 | Se muestra el mensaje de confirmación y la nueva relación aparece en la ficha del paciente. Además al acceder a la base de datos se observa que los datos han sido guardados en la misma. |
| Entrada 2 | Seleccionamos el id del mismo paciente desde el que se abrió el formulario. |
| Salida 2 | Se muestra un mensaje de error. |
| Resultado | Se obtiene el resultado esperado. |

Figura 92: Prueba de Uso 8

| | |
|------------------------|--|
| PRUEBA DE USO 9 | |
| Objetivo | Comprobar que al seleccionar un paciente relacionado con el actual se abre su ficha de historial médico. |
| Descripción | Al seleccionar en la ficha de un paciente un paciente relacionado y hacer doble clic debe abrirse la historia del otro paciente. |
| Entrada | Presionamos una relación de la ficha de un paciente. |
| Salida | Aparecen los datos personales y el historial del paciente seleccionado. |
| Resultado | Se obtiene el resultado esperado. |

Figura 93: Prueba de Uso 9

| | |
|-------------------------|--|
| PRUEBA DE USO 10 | |
| Objetivo | Comprobar que el formulario de gestión de consultas se abre en modo de adición para el paciente actual al pulsar el botón correspondiente. |
| Descripción | Al seleccionar el menú Añadir Consulta debe abrirse el formulario correspondiente para agregar una consulta. |
| Entrada 1 | Presionamos el botón Añadir Consulta tras abrir la ficha de un paciente. |
| Salida 1 | Aparece el formulario que permite agregar consultas al paciente. |
| Entrada 2 | Presionamos el botón Añadir Consultas sin abrir la ficha de un paciente. |
| Salida 2 | No se muestra el formulario. |
| Resultado | Se obtiene el resultado esperado. |

Figura 94: Prueba de Uso 10

| | |
|-------------------------|---|
| PRUEBA DE USO 11 | |
| Objetivo | Comprobar la creación de una consulta en la ficha de un paciente. |
| Descripción | Al rellenar los datos del formulario y pulsar el botón Añadir Consulta se debe guardar esa consulta en la ficha del usuario y mostrar un mensaje de confirmación. |
| Entrada 1 | Rellenamos el formulario y apretamos el botón. |
| Salida 1 | Se muestra el mensaje que nos informa de que los datos se han guardado. Además al acceder a la base de datos se observa que todos los datos han sido guardados en la misma. |
| Entrada 2 | Pulsamos el botón sin rellenar alguno de los campos del formulario. |
| Salida 2 | Se muestra un mensaje de error. |
| Resultado | Se obtiene el resultado esperado. |

Figura 95: Prueba de Uso 11

| | |
|-------------------------|---|
| PRUEBA DE USO 12 | |
| Objetivo | Comprobar que el formulario de gestión de consultas se abre en modo de modificación para el paciente actual al pulsar sobre una consulta dentro de la ficha del paciente. |
| Descripción | Al hacer doble clic sobre una consulta debe abrirse el formulario correspondiente. |
| Entrada | Presionamos sobre una de las consultas de la ficha de un paciente. |
| Salida | Aparece el formulario que permite modificar dicha consultas en la ficha de ese paciente. |
| Resultado | Se obtiene el resultado esperado. |

Figura 96: Prueba de Uso 12

| | |
|-------------------------|---|
| PRUEBA DE USO 13 | |
| Objetivo | Comprobar la modificación de una consulta en la ficha de un paciente. |
| Descripción | Al rellenar los datos del formulario y pulsar el botón Modificar Consulta se debe guardar esa consulta en la ficha del usuario y mostrar un mensaje de confirmación. |
| Entrada 1 | Rellenamos el formulario y apretamos el botón. |
| Salida 1 | Se muestra el mensaje que nos informa de que los datos se han guardado. Además al acceder a la base de datos se observa que todos los datos han sido guardados en la misma. |
| Entrada 2 | Pulsamos el botón sin rellenar alguno de los campos del formulario. |
| Salida 2 | Se muestra un mensaje de error. |
| Resultado | Se obtiene el resultado esperado. |

Figura 97: Prueba de Uso 13

| | |
|-------------------------|---|
| PRUEBA DE USO 14 | |
| Objetivo | Comprobar que el botón de Primero del formulario de pacientes funciona correctamente. |
| Descripción | Al pulsar el botón se debe mostrar el primer registro. |
| Entrada | Presionamos el botón. |
| Salida | Se abre el primer registro de la base de datos de Pacientes. |
| Resultado | Se obtiene el resultado esperado. |

Figura 98: Prueba de Uso 14

| | |
|-------------------------|--|
| PRUEBA DE USO 15 | |
| Objetivo | Comprobar que el botón de Último del formulario de pacientes funciona correctamente. |
| Descripción | Al pulsar el botón se debe mostrar el último registro. |
| Entrada | Presionamos el botón. |
| Salida | Se abre el último registro de la base de datos de Pacientes. |
| Resultado | Se obtiene el resultado esperado. |

Figura 99: Prueba de Uso 15

| | |
|-------------------------|--|
| PRUEBA DE USO 16 | |
| Objetivo | Comprobar que el botón de Anterior del formulario de pacientes funciona correctamente. |
| Descripción | Al pulsar el botón se debe mostrar el registro anterior al actual. |
| Entrada 1 | Presionamos el botón. |
| Salida 1 | Se abre el registro anterior de la base de datos de Pacientes. |
| Entrada 2 | Presionamos el botón en el primer registro de la base de datos. |
| Salida 2 | Se muestra un error. |
| Resultado | Se obtiene el resultado esperado. |

Figura 100: Prueba de Uso 16

| | |
|-------------------------|---|
| PRUEBA DE USO 17 | |
| Objetivo | Comprobar que el botón de Siguiente del formulario de pacientes funciona correctamente. |
| Descripción | Al pulsar el botón se debe mostrar el registro siguiente al actual |
| Entrada 1 | Presionamos el botón. |
| Salida 1 | Se abre el registro siguiente de la base de datos de Pacientes. |
| Entrada 2 | Presionamos el botón en el último registro de la base de datos. |
| Salida 2 | Se muestra un error. |
| Resultado | Se obtiene el resultado esperado. |

Figura 101: Prueba de Uso 17

| | |
|-------------------------|--|
| PRUEBA DE USO 18 | |
| Objetivo | Comprobar que el formulario de gestión de citas se muestra al pulsar el botón correspondiente. |
| Descripción | Al seleccionar el menú Gestión de Citas debe abrirse el formulario correspondiente. |
| Entrada | Presionamos el botón Gestión de Citas. |
| Salida | Aparece el formulario que permite ver, agregar y modificar citas. |
| Resultado | Se obtiene el resultado esperado. |

Figura 102: Prueba de Uso 18

| | |
|-------------------------|---|
| PRUEBA DE USO 19 | |
| Objetivo | Comprobar el correcto guardado de las citas. |
| Descripción | Al pulsar el botón Guardar Cambios se deben guardar las citas introducidas en el formulario para el día seleccionado y mostrar un mensaje de confirmación. |
| Entrada | Rellenamos el formulario y apretamos el botón. |
| Salida | Se muestra el mensaje que nos informa de que los datos se han guardado. Además al acceder a la base de datos se observa que todos los datos han sido guardados en la misma. |
| Resultado | Se obtiene el resultado esperado. |

Figura 103: Prueba de Uso 19

| | |
|-------------------------|--|
| PRUEBA DE USO 20 | |
| Objetivo | Comprobar que el botón de selección de día del formulario de citas funciona correctamente. |
| Descripción | Al seleccionar una fecha y pulsar el botón Actualizar se debe mostrar la agenda de citas para esa fecha. |
| Entrada | Seleccionamos una fecha mediante el calendario y presionamos el botón correspondiente. |
| Salida | Se recupera la agenda de citas del día elegido en la base de datos de Citas. |
| Resultado | Se obtiene el resultado esperado. |

Figura 104: Prueba de Uso 20

| | |
|-------------------------|---|
| PRUEBA DE USO 21 | |
| Objetivo | Comprobar que el botón de Día Anterior del formulario de citas funciona correctamente. |
| Descripción | Al pulsar el botón se debe mostrar la agenda de citas para la fecha anterior a la actual. |
| Entrada | Presionamos el botón. |
| Salida | Se recupera la agenda de citas del día anterior al actual de la base de datos de Citas. |
| Resultado | Se obtiene el resultado esperado. |

Figura 105: Prueba de Uso 21

| | |
|-------------------------|--|
| PRUEBA DE USO 22 | |
| Objetivo | Comprobar que el botón de Día Siguiente del formulario de citas funciona correctamente. |
| Descripción | Al pulsar el botón se debe mostrar la agenda de citas para la fecha siguiente a la actual. |
| Entrada | Presionamos el botón. |
| Salida | Se recupera la agenda de citas del día siguiente al actual de la base de datos de Citas. |
| Resultado | Se obtiene el resultado esperado. |

Figura 106: Prueba de Uso 22

| | |
|-------------------------|---|
| PRUEBA DE USO 23 | |
| Objetivo | Comprobar que el formulario para añadir una cita nueva se muestra al pulsar el botón correspondiente. |
| Descripción | Al seleccionar el menú Nueva Cita... debe abrirse el formulario correspondiente. |
| Entrada | Presionamos el botón Nueva Cita... |
| Salida | Aparece el formulario que permite agregar una cita nueva en la agenda seleccionando todos los detalles de la misma. |
| Resultado | Se obtiene el resultado esperado. |

Figura 107: Prueba de Uso 23

| | |
|-------------------------|---|
| PRUEBA DE USO 24 | |
| Objetivo | Comprobar la creación de una cita mediante el formulario de Cita Nueva. |
| Descripción | Al rellenar los datos del formulario y pulsar el botón Añadir Cita se debe guardar esa cita en la base de datos y mostrar un mensaje de confirmación. |
| Entrada 1 | Rellenamos el formulario y apretamos el botón. |
| Salida 1 | Se muestra el mensaje que nos informa de que los datos se han guardado. Además al acceder a la base de datos se observa que todos los datos han sido guardados en la misma. |
| Entrada 2 | Pulsamos el botón sin rellenar alguno de los campos del formulario. |
| Salida 2 | Se muestra un mensaje de error. |
| Resultado | Se obtiene el resultado esperado. |

Figura 108: Prueba de Uso 24

| | |
|-------------------------|--|
| PRUEBA DE USO 25 | |
| Objetivo | Comprobar que el formulario de estadísticas se muestra al pulsar el botón correspondiente. |
| Descripción | Al seleccionar el menú Estadísticas debe abrirse el formulario correspondiente. |
| Entrada | Presionamos el botón Estadísticas. |
| Salida | Aparece el formulario que permite generar y ver diferentes informes de actividad. |
| Resultado | Se obtiene el resultado esperado. |

Figura 109: Prueba de Uso 25

| | |
|-------------------------|---|
| PRUEBA DE USO 26 | |
| Objetivo | Comprobar que se muestra el informe por día y agrupado por actos en el formulario de estadísticas. |
| Descripción | Al seleccionar las opciones correspondientes debe mostrarse en el formulario el informe correspondiente y las sumas de los totales correspondiente. |
| Entrada | Seleccionamos las opciones Día y Agrupar por actos y pulsamos el botón Mostrar Datos |
| Salida | Aparece en el formulario el resumen de la actividad del día agrupado por actos y los totales se suman correctamente. |
| Resultado | Se obtiene el resultado esperado. |

Figura 110: Prueba de Uso 26

| | |
|-------------------------|---|
| PRUEBA DE USO 27 | |
| Objetivo | Comprobar que se muestra el informe por día y agrupado por paciente en el formulario de estadísticas. |
| Descripción | Al seleccionar las opciones correspondientes debe mostrarse en el formulario el informe correspondiente y las sumas de los totales correspondiente. |
| Entrada | Seleccionamos las opciones Día y Agrupar por paciente y pulsamos el botón Mostrar Datos |
| Salida | Aparece en el formulario el resumen de la actividad del día agrupado por pacientes y los totales se suman correctamente. |
| Resultado | Se obtiene el resultado esperado. |

Figura 111: Prueba de Uso 27

| | |
|-------------------------|---|
| PRUEBA DE USO 28 | |
| Objetivo | Comprobar que se muestra el informe por día y sólo consultas pendientes en el formulario de estadísticas. |
| Descripción | Al seleccionar las opciones correspondientes debe mostrarse en el formulario el informe correspondiente y las sumas de los totales correspondiente. |
| Entrada | Seleccionamos las opciones Día y Consultas Pendientes y pulsamos el botón Mostrar Datos |
| Salida | Aparece en el formulario el resumen de la actividad del día mostrando las consultas pendientes y los totales se suman correctamente. |
| Resultado | Se obtiene el resultado esperado. |

Figura 112: Prueba de Uso 28

| | |
|-------------------------|---|
| PRUEBA DE USO 29 | |
| Objetivo | Comprobar que se muestra el informe por mes y agrupado por actos en el formulario de estadísticas. |
| Descripción | Al seleccionar las opciones correspondientes debe mostrarse en el formulario el informe correspondiente y las sumas de los totales correspondiente. |
| Entrada | Seleccionamos las opciones Mes y Agrupar por actos y pulsamos el botón Mostrar Datos |
| Salida | Aparece en el formulario el resumen de la actividad del mes agrupado por actos y los totales se suman correctamente. |
| Resultado | Se obtiene el resultado esperado. |

Figura 113: Prueba de Uso 29

| | |
|-------------------------|---|
| PRUEBA DE USO 30 | |
| Objetivo | Comprobar que se muestra el informe por mes y agrupado por paciente en el formulario de estadísticas. |
| Descripción | Al seleccionar las opciones correspondientes debe mostrarse en el formulario el informe correspondiente y las sumas de los totales correspondiente. |
| Entrada | Seleccionamos las opciones Mes y Agrupar por paciente y pulsamos el botón Mostrar Datos |
| Salida | Aparece en el formulario el resumen de la actividad del mes agrupado por pacientes y los totales se suman correctamente. |
| Resultado | Se obtiene el resultado esperado. |

Figura 114: Prueba de Uso 30

| | |
|-------------------------|---|
| PRUEBA DE USO 31 | |
| Objetivo | Comprobar que se muestra el informe por mes y sólo consultas pendientes en el formulario de estadísticas. |
| Descripción | Al seleccionar las opciones correspondientes debe mostrarse en el formulario el informe correspondiente y las sumas de los totales correspondiente. |
| Entrada | Seleccionamos las opciones Mes y Consultas Pendientes y pulsamos el botón Mostrar Datos |
| Salida | Aparece en el formulario el resumen de la actividad del mes mostrando las consultas pendientes y los totales se suman correctamente. |
| Resultado | Se obtiene el resultado esperado. |

Figura 115: Prueba de Uso 31

| | |
|-------------------------|---|
| PRUEBA DE USO 32 | |
| Objetivo | Comprobar que se muestra el informe por año y agrupado por actos en el formulario de estadísticas. |
| Descripción | Al seleccionar las opciones correspondientes debe mostrarse en el formulario el informe correspondiente y las sumas de los totales correspondiente. |
| Entrada | Seleccionamos las opciones Año y Agrupar por actos y pulsamos el botón Mostrar Datos |
| Salida | Aparece en el formulario el resumen de la actividad del año agrupado por actos y los totales se suman correctamente. |
| Resultado | Se obtiene el resultado esperado. |

Figura 116: Prueba de Uso 32

| | |
|-------------------------|---|
| PRUEBA DE USO 33 | |
| Objetivo | Comprobar que se muestra el informe por año y agrupado por paciente en el formulario de estadísticas. |
| Descripción | Al seleccionar las opciones correspondientes debe mostrarse en el formulario el informe correspondiente y las sumas de los totales correspondiente. |
| Entrada | Seleccionamos las opciones Año y Agrupar por paciente y pulsamos el botón Mostrar Datos |
| Salida | Aparece en el formulario el resumen de la actividad del año agrupado por pacientes y los totales se suman correctamente. |
| Resultado | Se obtiene el resultado esperado. |

Figura 117: Prueba de Uso 33

| | |
|-------------------------|---|
| PRUEBA DE USO 34 | |
| Objetivo | Comprobar que se muestra el informe por año y sólo consultas pendientes en el formulario de estadísticas. |
| Descripción | Al seleccionar las opciones correspondientes debe mostrarse en el formulario el informe correspondiente y las sumas de los totales correspondiente. |
| Entrada | Seleccionamos las opciones Año y Consultas Pendientes y pulsamos el botón Mostrar Datos |
| Salida | Aparece en el formulario el resumen de la actividad del año mostrando las consultas pendientes y los totales se suman correctamente. |
| Resultado | Se obtiene el resultado esperado. |

Figura 118: Prueba de Uso 34

| | |
|-------------------------|---|
| PRUEBA DE USO 35 | |
| Objetivo | Comprobar que se muestra el informe total y agrupado por actos en el formulario de estadísticas. |
| Descripción | Al seleccionar las opciones correspondientes debe mostrarse en el formulario el informe correspondiente y las sumas de los totales correspondiente. |
| Entrada | Seleccionamos las opciones Todo y Agrupar por actos y pulsamos el botón Mostrar Datos |
| Salida | Aparece en el formulario el resumen de la actividad total agrupado por actos y los totales se suman correctamente. |
| Resultado | Se obtiene el resultado esperado. |

Figura 119: Prueba de Uso 35

| | |
|-------------------------|---|
| PRUEBA DE USO 36 | |
| Objetivo | Comprobar que se muestra el informe total y agrupado por paciente en el formulario de estadísticas. |
| Descripción | Al seleccionar las opciones correspondientes debe mostrarse en el formulario el informe correspondiente y las sumas de los totales correspondiente. |
| Entrada | Seleccionamos las opciones Todo y Agrupar por paciente y pulsamos el botón Mostrar Datos |
| Salida | Aparece en el formulario el resumen de la actividad total agrupado por pacientes y los totales se suman correctamente. |
| Resultado | Se obtiene el resultado esperado. |

Figura 120: Prueba de Uso 36

| | |
|-------------------------|---|
| PRUEBA DE USO 37 | |
| Objetivo | Comprobar que se muestra el informe total y sólo consultas pendientes en el formulario de estadísticas. |
| Descripción | Al seleccionar las opciones correspondientes debe mostrarse en el formulario el informe correspondiente y las sumas de los totales correspondiente. |
| Entrada | Seleccionamos las opciones Todo y Consultas Pendientes y pulsamos el botón Mostrar Datos |
| Salida | Aparece en el formulario el resumen de la actividad total mostrando las consultas pendientes y los totales se suman correctamente. |
| Resultado | Se obtiene el resultado esperado. |

Figura 121: Prueba de Uso 37

SECCIÓN III: MANUAL DE USUARIO

1. INSTALACION DE LA APLICACIÓN

1.1. REQUISITOS DE INSTALACIÓN

- Máquina virtual Java
- Servidor MySQL.

1.2. INSTALACIÓN DE LA APLICACIÓN

Para realizar la instalación de la aplicación tan solo hay que descomprimir el archivo "ClinPod.zip" que se encuentra en el CD en la ubicación de nuestra preferencia. Una vez descomprimido ya se puede ejecutar, pero solo podremos ver la pantalla de inicio. Es necesario que creamos la base de datos para poder explorar la aplicación al máximo de sus posibilidades.

2. CREACIÓN BASE DE DATOS

2.1. CREAR BASE DE DATOS

Para realizar la creación de la base de datos se debe ejecutar el script SQL adjunto desde la consola de MySQL utilizando un usuario con permisos de creación. Para ello abra la consola del sistema y posicione en la carpeta donde se encuentra instalado el servidor MySQL (habitualmente será algo del tipo *C:\Program Files\MySQL\MySQL Server 5.7\bin*). Después ejecute la siguiente instrucción:

```
mysql -u UsuarioMySQL -p< C:\SQLScript.sql
```

Donde habrá que sustituir *UsuarioMySQL* por nuestro nombre de usuario y *C:\SQLScript.sql* por la ruta completa del script de creación en nuestro disco duro. Nos pedirá nuestra contraseña y ejecutará el script.

Una vez realizados estos pasos la base de datos estará creada y lista para ejecutar la aplicación.

La aplicación está configurada por defecto para funcionar con los siguientes datos de conexión de MySQL:

Servidor: localhost
Puerto: 3306
Nombre de la base de datos: bdclin
Usuario: usuario
Contraseña: contrasena

Estos datos se almacenan en el archivo *conexion.conf* que debe estar en el mismo directorio que el programa. Se puede modificar con cualquier editor de texto, pero es importante conservar el mismo formato y solo editar la parte derecha de cada expresión (a la derecha del símbolo "=") con lo que corresponda.

3. USO GENERAL

3.1. PANTALLA PRINCIPAL

Al abrir la aplicación nos encontramos con la siguiente pantalla:



Figura 122: Pantalla inicial de la aplicación

Esta es la pantalla principal de la aplicación desde la que podremos acceder a todas las funcionalidades que nos ofrece la aplicación. Como podemos apreciar la aplicación está dividida en tres secciones. Menú Pacientes, donde podremos gestionar toda la información de los pacientes que visiten la clínica como datos personales, historial de consultas, etc. Desde el menú Citas se gestiona toda la agenda diaria de la consulta de podología. Por último, en Estadísticas tendremos la posibilidad de visualizar diferentes informes sobre la actividad de la clínica. Esta ventana permanecerá abierta en todo momento mientras estemos usando la aplicación para poder acceder a los tres apartados en el momento que lo necesitemos. Las distintas opciones también son accesibles desde la barra de menús, como puede verse en la siguiente imagen:



Figura 123: Menú Archivo



Figura 124: Menú Ayuda

Si elegimos la opción Acerca de... se nos mostrará la información sobre la aplicación, como el creador o la página web de la empresa.



Figura 125: Dialogo Acerca de...

Cabe destacar que los formularios generados por el menú de la pantalla principal son independientes, lo que hace posible que puedan ser visualizados simultáneamente, para poder realizar varias gestiones en diferentes apartados simultáneamente.

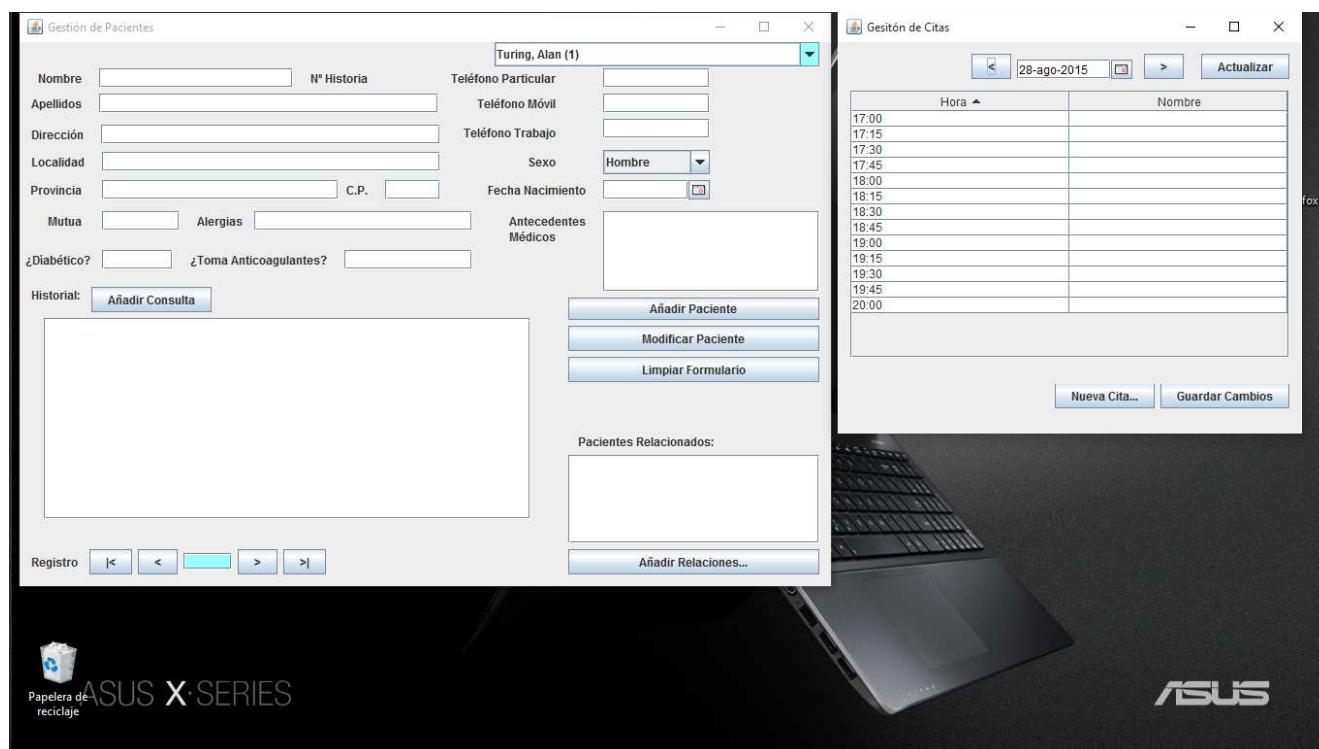


Figura 126: Vista de dos ventanas de la aplicación funcionando simultáneamente

3.2. MENÚ PACIENTES

Al acceder al menú de pacientes nos encontraremos con un formulario de introducción de datos en blanco. A lo largo de las diferentes secciones iremos desgranando todas las opciones que este nos ofrece.

3.2.1. AGREGAR PACIENTES

The screenshot displays a web-based form titled "Gestión de Pacientes". The form is organized into several sections:

- Personal Information:** Includes fields for "Nombre", "Apellidos", "Dirección", "Localidad", "Provincia", "C.P.", "Sexo" (set to "Hombre"), and "Fecha Nacimiento".
- Contact Information:** Includes fields for "Teléfono Particular", "Teléfono Móvil", and "Teléfono Trabajo".
- Medical History:** Includes fields for "Mutua", "Alergias", "Antecedentes Médicos", "¿Diabético?", and "¿Toma Anticoagulantes?".
- Actions:** A "Historial:" section contains an "Añadir Consulta" button. On the right side, there are three stacked buttons: "Añadir Paciente", "Modificar Paciente", and "Limpiar Formulario".
- Related Patients:** A section labeled "Pacientes Relacionados:" contains an empty box and an "Añadir Relaciones..." button.
- Navigation:** At the bottom left, there is a "Registro" label and a set of navigation buttons: "|<", "<", a highlighted cyan button, ">", and ">|".

Figura 127: Gestión de Pacientes

Para agregar un paciente nuevo tan solo hay que rellenar los datos que hemos obtenido del mismo en los apartados correspondientes del formulario. Cuando lo tengamos todo presionamos el botón agregar paciente. Si todo ha funcionado correctamente, el sistema nos informará de que el paciente ha sido agregado y del id que se le ha asignado en el sistema. En cualquier momento que lo necesitemos podemos limpiar el formulario para empezar de nuevo usando el botón correspondiente.

The screenshot shows a software window titled "Gestión de Pacientes". At the top right, there are standard window controls (minimize, maximize, close) and a search bar. The main area contains a form with the following fields and controls:

- Form Fields:** Nombre, Apellidos, Dirección, Localidad, Provincia, C.P., Mutua, Alergias, ¿Diabético?, ¿Toma Anticoagulantes?, N° Historia, Teléfono Particular, Teléfono Móvil, Teléfono Trabajo, Sexo (dropdown menu set to "Hombre"), Fecha Nacimiento (with a calendar icon), and Antecedentes Médicos (a large text area).
- Buttons:** "Añadir Consulta" (next to the "Historial:" label), "Añadir Paciente", "Modificar Paciente", "Limpiar Formulario", and "Añadir Relaciones...".
- Message:** "Paciente agregado correctamente, id=5" is displayed in the center of the form.
- Navigation:** A "Registro" section at the bottom left contains navigation buttons: "|<", "<", a highlighted cyan button, ">", and ">|".

Figura 128: Confirmación de paciente agregado

3.2.2. BUSCAR PACIENTES

Una vez que tengamos varios pacientes grabados en nuestra base de datos, nos será muy útil poder buscarlos por diferentes criterios. El primero de ellos es a través de su id, introduciendo el número de historia en el cajón de la parte inferior y pulsando *enter* o moviéndonos por los registros con los botones de avance, retroceso, primero y último.

Gestión de Pacientes

Nombre: John N° Historia: 3

Apellidos: Von Neumann

Dirección: []

Localidad: []

Provincia: [] C.P.: []

Mutua: [] Alergias: []

¿Diabético?: [] ¿Toma Anticoagulantes?: []

Teléfono Particular: []

Teléfono Móvil: []

Teléfono Trabajo: []

Sexo: Hombre

Fecha Nacimiento: []

Antecedentes Médicos: []

Historial: Añadir Consulta

Registro: |< < 3 > >|

Añadir Paciente

Modificar Paciente

Limpiar Formulario

Pacientes Relacionados: []

Añadir Relaciones...

Figura 129: Búsqueda por id de Paciente

La segunda opción de la que disponemos es buscar el paciente introduciendo sus apellidos en el cajón de búsqueda superior derecho. Vemos que ambos elementos de búsqueda están resaltados en color azul para distinguirlos del resto de campos del formulario.

The screenshot shows a software window titled "Gestión de Pacientes" with a search dropdown menu open. The search results are:

- Cray, Seymour (2)
- Tanenbaum, Andrew (4)
- Turing, Alan (1)
- Von Neumann, John (3)

The form includes the following fields and controls:

- Nombre:
- Nº Historia:
- Apellidos:
- Dirección:
- Localidad:
- Provincia: C.P.:
- Mutua: Alergias:
- ¿Diabético?: ¿Toma Anticoagulantes?:
- Sexo:
- Fecha Nacimiento:
- Antecedentes Médicos:
- Historial:
- Registro:
- Buttons:

Figura 130: Búsqueda por apellidos del paciente

La búsqueda por apellidos puede usarse seleccionando un elemento de la lista con el ratón o escribiendo directamente para movernos rápidamente a la posición deseada.

The screenshot shows a software window titled "Gestión de Pacientes". The main form contains several input fields: "Nombre" (Andrew), "Nº Historia" (4), "Apellidos" (Tanenbaum), "Dirección", "Localidad", "Provincia", "C.P.", "Mutua", "Alergias", "¿Diabético?", and "¿Toma Anticoagulantes?". A dropdown menu is open over the "Apellidos" field, displaying a list of suggestions: "Tanenbaum, Andrew (4)", "Cray, Seymour (2)", "Lovelace, Ada (5)", "Tanenbaum, Andrew (4)", "Turing, Alan (1)", and "Von Neumann, John (3)". Below the form are buttons for "Añadir Consulta", "Añadir Paciente", "Modificar Paciente", "Limpiar Formulario", and "Añadir Relaciones...". A "Historial" section shows a record: "25/08/2015 Papiloma Mapre". At the bottom, there is a "Registro" section with navigation buttons and a page number "4".

Figura 131: Función de autocompletado

Como podemos apreciar la búsqueda por apellidos dispone de autocompletado, de forma que se nos va sugiriendo el resto del nombre a medida que vamos escribiendo.

3.2.3. MODIFICAR PACIENTES

A veces nos resulta útil modificar los datos personales de un paciente, bien porque haya cambiado de dirección, para añadir otro número de teléfono o para añadir algún dato médico relevante nuevo. Para hacer estos cambios primero hay que ubicarse en la ficha del paciente que queramos cambiar. Después trabajamos con los datos y cuando hayamos terminado pulsamos Modificar Paciente. Si la operación se completa con éxito obtendremos un mensaje de confirmación

The screenshot shows a web-based form titled "Gestión de Pacientes". The form is organized into several sections:

- Personal Data:** Fields for "Nombre", "Apellidos", "Dirección", "Localidad", "Provincia", "C.P.", "Mutua", and "Alergias".
- Contact Information:** Fields for "Teléfono Particular", "Teléfono Móvil", and "Teléfono Trabajo".
- Medical Information:** Fields for "Sexo" (set to "Hombre"), "Fecha Nacimiento", and "Antecedentes Médicos".
- Health Status:** Checkboxes for "¿Diabético?" and "¿Toma Anticoagulantes?".
- History:** A section with a "Historial:" label and an "Añadir Consulta" button.
- Confirmation:** A message "Paciente modificado correctamente" is displayed in the center.
- Navigation:** A "Registro" section at the bottom left with navigation buttons (|<, <, >, >|).
- Buttons:** A vertical stack of buttons on the right: "Añadir Paciente", "Modificar Paciente", "Limpiar Formulario", and "Añadir Relaciones...".
- Related Patients:** A section titled "Pacientes Relacionados:" with an empty box below it.

Figura 132: Confirmación de modificación de paciente

3.2.4. AGREGAR PACIENTES RELACIONADOS

En las consultas médicas es muy habitual que haya pacientes que acudan habitualmente juntos a la clínica: matrimonios, padres e hijos, monjas de la misma congregación... Por esto es de vital importancia contar con un mecanismo que permita encontrar rápidamente las historias sin necesidad de tener que buscar cada paciente uno a uno. Para añadir una relación entre pacientes debemos situarnos en la historia de uno de los pacientes y hacer clic en el botón Añadir Relaciones.

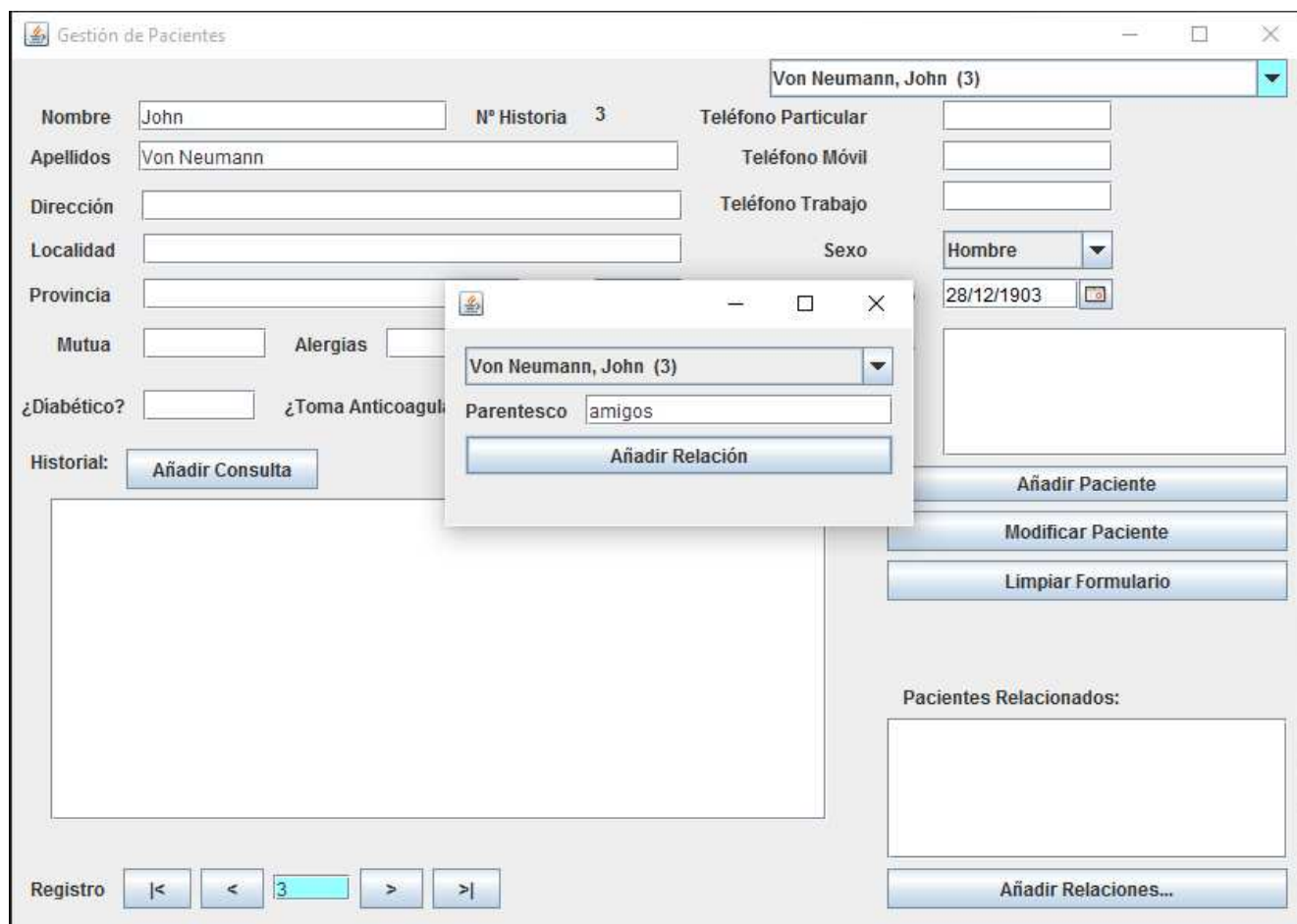


Figura 133: Agregando pacientes relacionados

Nos aparecerá un pequeño formulario en el que podremos elegir del desplegable a otro paciente para relacionarlo con el actual. Obviamente no podemos relacionar el paciente consigo mismo, por lo que si lo intentamos nos saltará un error.

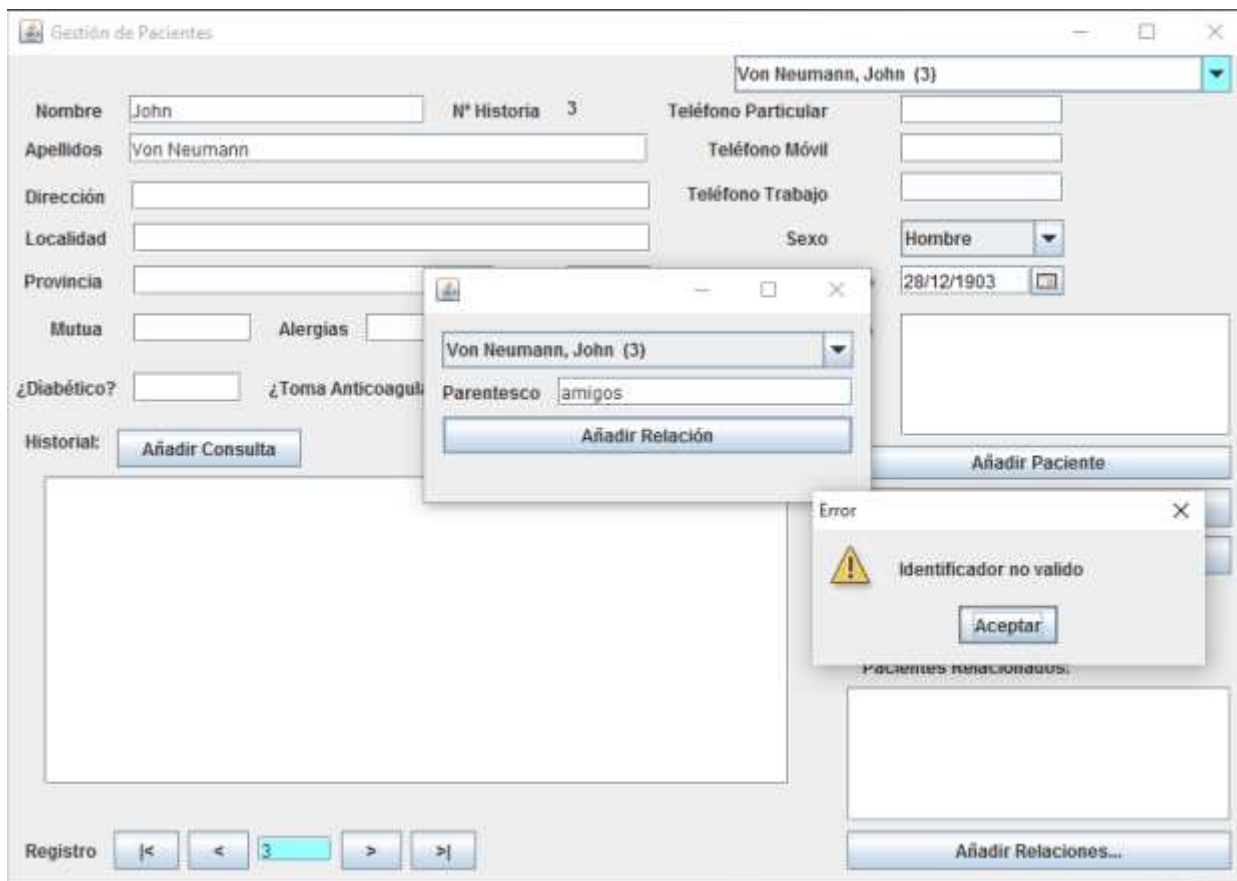


Figura 134: Error al intentar relacionar un paciente consigo mismo

Si por el contrario la relación es correcta recibiremos un mensaje de que la relación entre los pacientes se ha creado, por lo que a partir de ahora en la ficha de ambos habrá una referencia al otro y si hacemos doble clic sobre ella saltaremos directamente a la ficha correspondiente.

The screenshot displays a software window titled "Gestión de Pacientes". The main form contains the following fields and controls:

- Nombre:** John
- Apellidos:** Von Neumann
- Nº Historia:** 3
- Sexo:** Hombre
- Fecha de nacimiento:** 28/12/1903
- Parentesco:** amigos
- Buttons:** Añadir Consulta, Añadir Paciente, Modificar Paciente, Limpiar Formulario, Añadir Relaciones...

A modal dialog box is open in the center, titled "Tanenbaum, Andrew (4)". It contains the following elements:

- Parentesco:** amigos
- Button:** Añadir Relación
- Message:** Se añadió la relación al paciente

At the bottom of the main window, there is a "Registro" section with navigation buttons: |<, <, 3, >, >|.

Figura 135: Relación agregada correctamente

Gestión de Pacientes

Tanenbaum, Andrew (4)

Nombre: Andrew N° Historia: 4 Teléfono Particular:

Apellidos: Tanenbaum Teléfono Móvil:

Dirección: Teléfono Trabajo:

Localidad: Sexo: Hombre

Provincia: C.P.: Fecha Nacimiento:

Mutua: Alergias: Antecedentes Médicos:

¿Diabético?: ¿Toma Anticoagulantes?:

Historial:

Pacientes Relacionados:
3, John Von Neumann, amigos

Registro: |< < 4 > >|

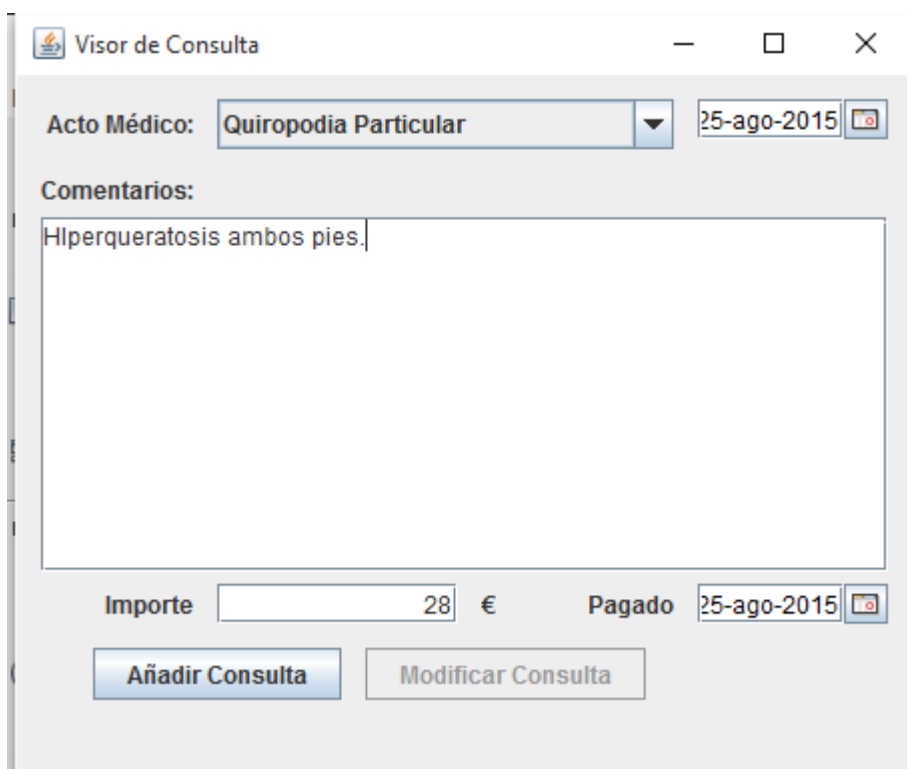
Figura 136: Vista de un paciente con una relación

3.2.5. AGREGAR CONSULTAS

La sección de consultas es la más importante de la aplicación, ya que es la que nos permite registrar cada visita de cada paciente para en un futuro poder consultar lo que se le realizó, cuanto tiempo ha pasado desde la consulta anterior y, en definitiva llevar un seguimiento de la actividad del paciente en la clínica. Para acceder al menú que nos permite gestionar todas estas funcionalidades apretamos el botón Añadir Consulta.

Figura 137: Visor de consulta

Aquí podremos añadir todos los datos relativos a la consulta como el acto médico (breve descripción genérica de lo que se ha realizado), la fecha, comentarios en detalle sobre la consulta, el importe o la fecha de pago. Lo rellenamos con los datos propios de la consulta y pulsamos sobre Añadir consulta.



Visor de Consulta

Acto Médico: Quiropodia Particular 25-ago-2015

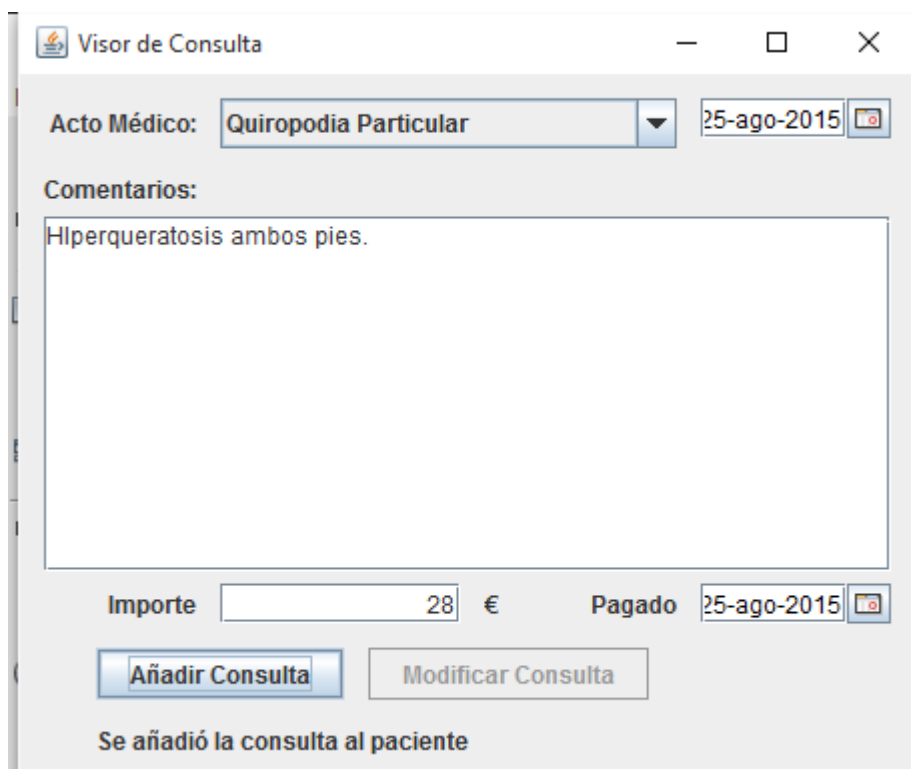
Comentarios:
Hiperqueratosis ambos pies.

Importe 28 € Pagado 25-ago-2015

Añadir Consulta Modificar Consulta

Figura 138: Rellenando los datos de una consulta

Cuando la consulta se haya grabado, veremos un mensaje avisándonos de ello y en adelante la consulta se mostrará en la ficha del paciente.



Visor de Consulta

Acto Médico: Quiropodia Particular 25-ago-2015

Comentarios:
Hiperqueratosis ambos pies.

Importe 28 € Pagado 25-ago-2015

Añadir Consulta Modificar Consulta

Se añadió la consulta al paciente

Figura 139: Consulta guardada

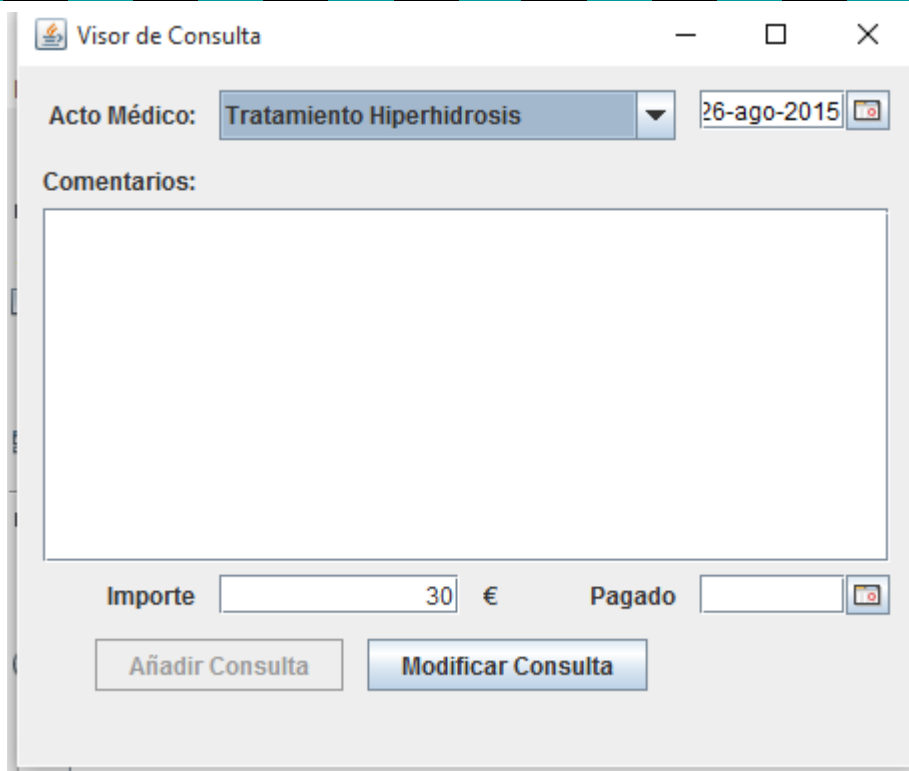
3.2.6. VER CONSULTAS

Las consultas se muestran en el apartado correspondiente dentro de la historia del paciente de forma separada y ordenadas de forma que se visualicen en la parte superior las más recientes. Si en algún momento necesitamos consultar el detalle sobre alguna consulta en particular solo tendremos que hacer doble clic sobre la consulta de la que queramos saber más y se nos volverá a abrir el mismo formulario con los datos que habíamos introducido ya guardados.

The screenshot shows a software window titled "Gestión de Pacientes". At the top right, there is a search bar containing "Turing, Alan (1)". Below this, the patient's details are organized into several sections:

- Personal Data:** Nombre: Alan, Apellidos: Turing, N° Historia: 1.
- Contact Information:** Teléfono Particular, Teléfono Móvil, and Teléfono Trabajo (all empty).
- Demographics:** Dirección, Localidad, Provincia, C.P., Sexo: Hombre, Fecha Nacimiento.
- Medical History:** Mutua, Alergias, ¿Diabético?, and ¿Toma Anticoagulantes? (all empty).
- Antecedentes Médicos:** A large empty text area.
- Historial:** A section with an "Añadir Consulta" button and a list of consultations:
 - 26/08/2015 Tratamiento Hiperhidrosis
 - 25/08/2015 Quiropodia Particular
- Navigation:** A "Registro" section with navigation buttons (|<, <, 1, >, >|).
- Actions:** A vertical stack of buttons: "Añadir Paciente", "Modificar Paciente", "Limpiar Formulario", "Pacientes Relacionados:" (with an empty box below), and "Añadir Relaciones..."

Figura 140: Vista de consultas en la ficha del paciente



Visor de Consulta

Acto Médico: **Tratamiento Hiperhidrosis** 26-ago-2015

Comentarios:

Importe 30 € Pagado

Añadir Consulta Modificar Consulta

Figura 141: Modo de modificación de consulta

3.2.7. MODIFICAR CONSULTAS

Una vez que hemos abierto una consulta en concreto podemos apreciar que el formulario que aparece ahora es algo distinto, ahora tenemos los datos que habíamos introducido en pantalla y el botón modificar aparece habilitado. Para realizar algún cambio sobre esa consulta solo hay que rectificar los datos que queramos y pulsar el botón de Modificar consulta. Una vez más, recibiremos una confirmación en el caso de que la operación se realice.

Visor de Consulta

Acto Médico: 26-ago-2015

Comentarios:

Importe € Pagado 31-ago-2015

Se modificó la consulta al paciente

Figura 142: Consulta modificada

3.3. MENÚ CITAS

El apartado de Gestión de Citas nos permite tener una visión general de la agenda de citas de la clínica para cada día. Al abrirlo se nos mostrará una tabla con el horario habitual de la consulta (17:00 a 20:00) en filas separadas por intervalos de 15 minutos. Podemos elegir el día que necesitamos desplazándonos con las flechas, escribiendo el día, o eligiéndolo con el calendario.

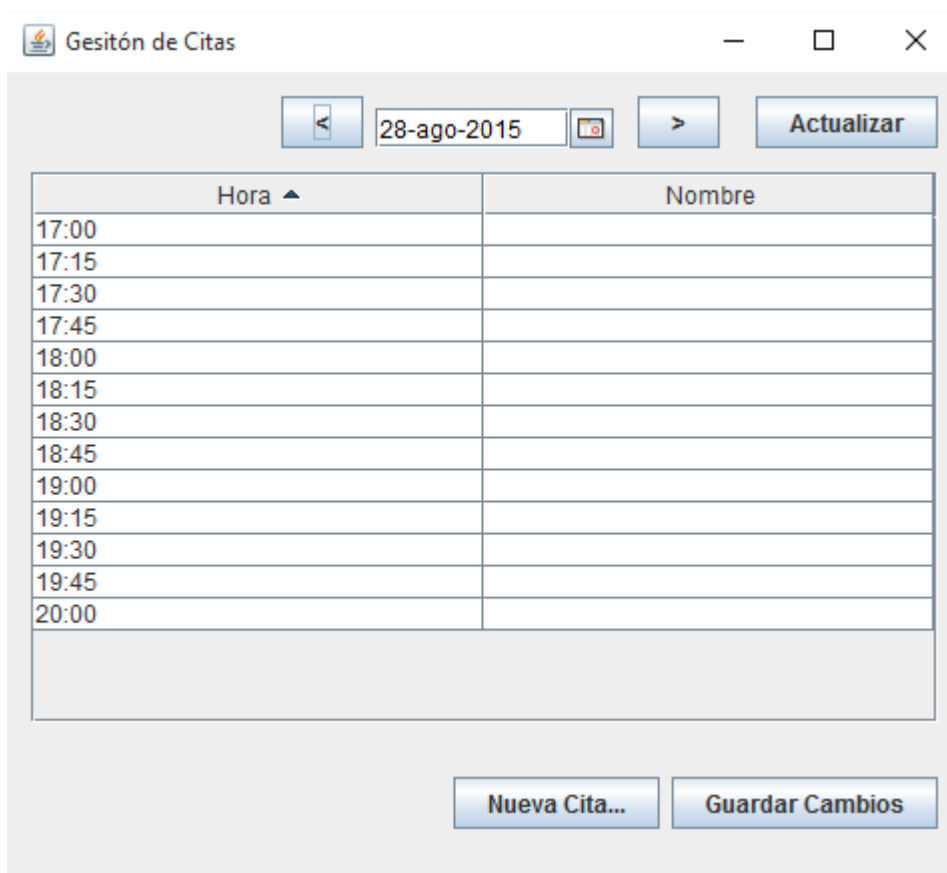


Figura 143: Gestión de citas

Las celdas de la izquierda son fijas y no se pueden modificar, mientras que las de la derecha son en las que tendremos que introducir el nombre del paciente que ha cogido ese horario de cita. El nombre puede escribirse o elegirse del desplegable de los pacientes que ya tenemos almacenados en nuestra base de datos.

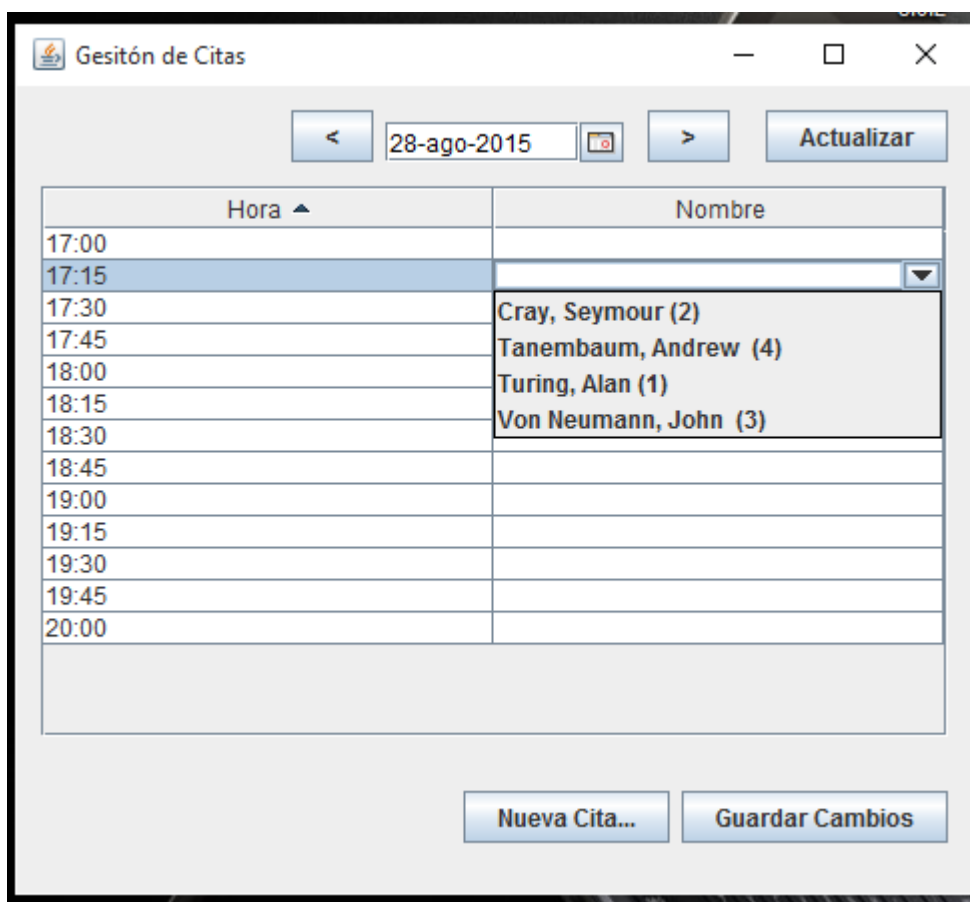


Figura 144: Desplegable para elegir un paciente de la base de datos

Podemos añadir todas las citas que necesitemos y cuando lo tengamos completo deberemos apretar el botón Guardar Cambios para que los datos se consoliden en la base de datos. Como siempre, la aplicación nos informará del guardado de los datos.

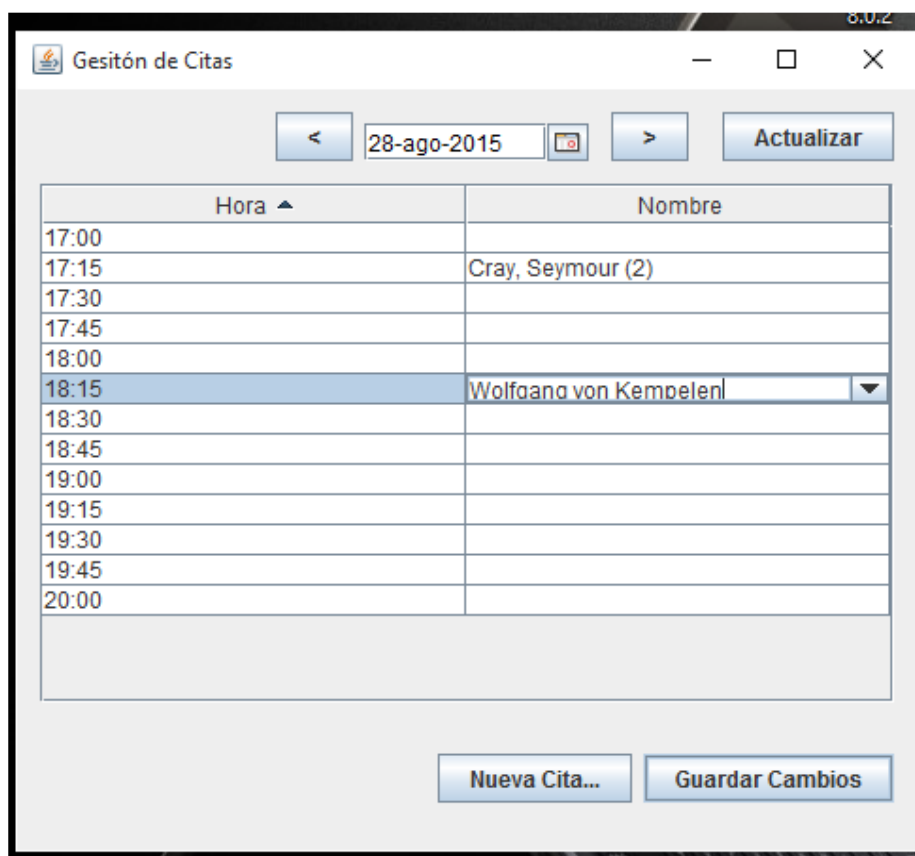


Figura 145: Agregando un paciente manualmente

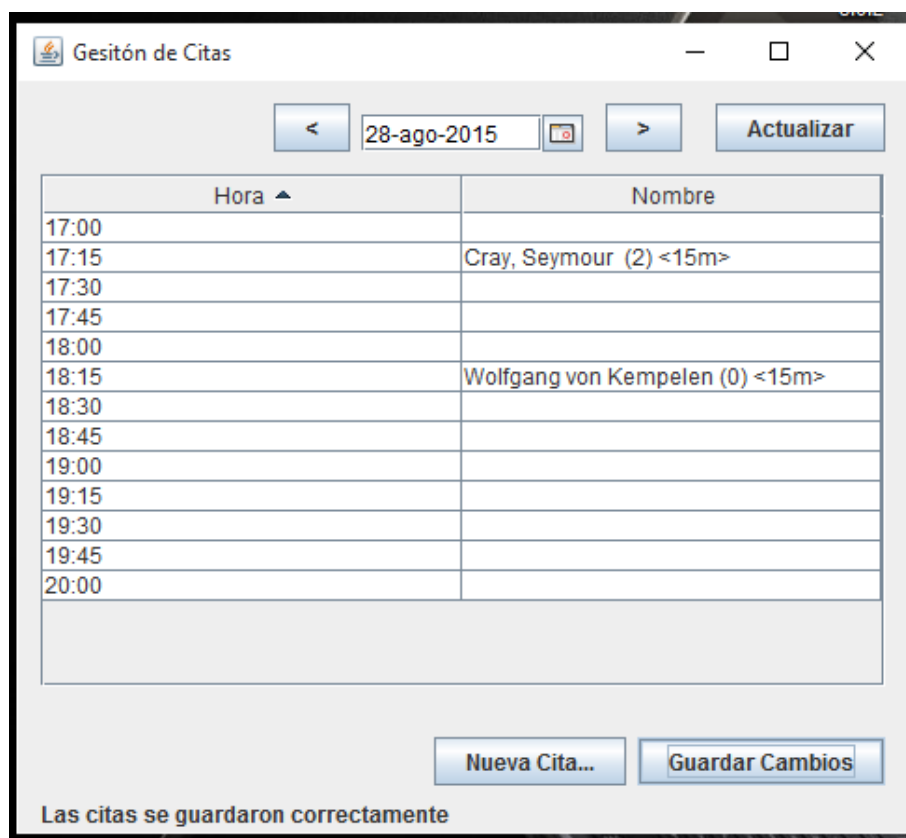
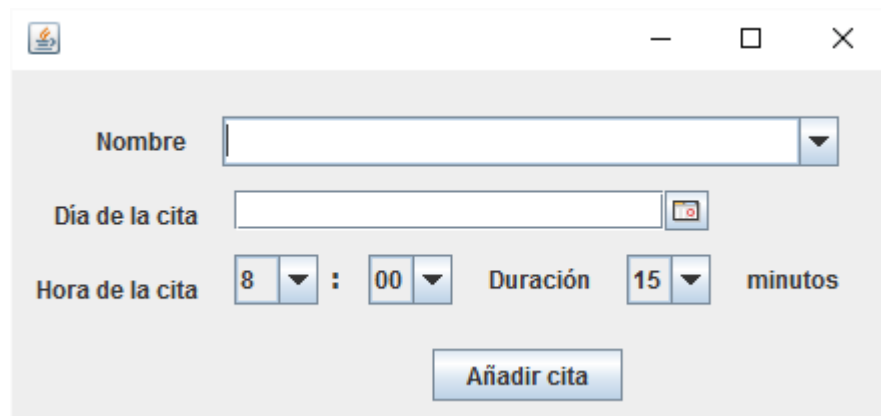


Figura 146: Citas guardadas

Podemos ver que una vez guardados los datos en la celda correspondiente nos aparece el nombre y apellidos el identificador en el caso de pacientes que ya tengan historia en la clínica (para los pacientes no registrados en la base de datos el sistema mostrará el identificador genérico "0") y la duración de la consulta, que por defecto es 15 minutos.

Puede ocurrir que a veces se concierte una cita fuera del horario habitual o se necesite reservar un mayor tiempo de cita para actividades de mayor complejidad (estudios de la pisada, cirugías...). En ese caso no podemos añadir la cita desde la tabla, sino que tendremos que abrir el menú Nueva Cita...

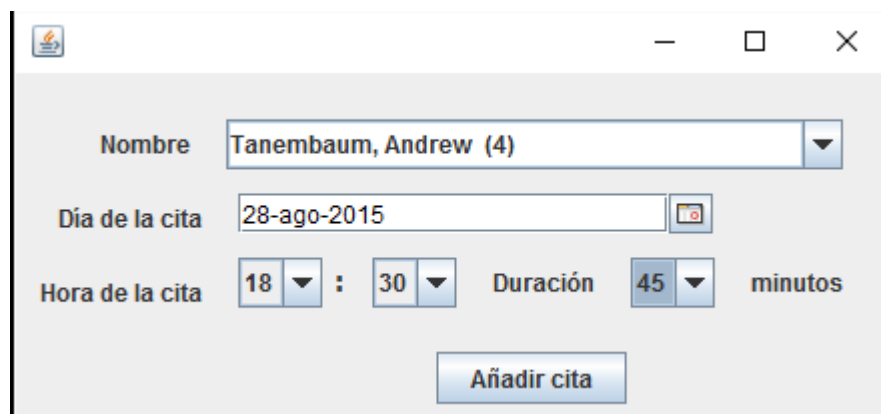


The screenshot shows a software window titled "Añadir cita" (Add appointment). It contains the following fields and controls:

- Nombre:** A text input field that is currently empty.
- Día de la cita:** A date selection field showing a calendar icon.
- Hora de la cita:** Two dropdown menus for hours and minutes, currently set to "8" and "00".
- Duración:** A dropdown menu for duration, currently set to "15".
- minutes:** A text label indicating the unit of the duration.
- Añadir cita:** A button at the bottom to submit the form.

Figura 147: Agregar cita compuesta

Aquí podremos fijar el nombre (de nuevo podremos hacerlo por el desplegable o escribir el nombre), el día de la cita, la hora o la duración entre 15 minutos y una hora.



The screenshot shows the same "Añadir cita" window, but with the following data entered:

- Nombre:** A dropdown menu showing "Tanenbaum, Andrew (4)".
- Día de la cita:** A date selection field showing "28-ago-2015".
- Hora de la cita:** Two dropdown menus for hours and minutes, currently set to "18" and "30".
- Duración:** A dropdown menu for duration, currently set to "45".
- minutes:** A text label indicating the unit of the duration.
- Añadir cita:** A button at the bottom to submit the form.

Figura 148: Completando los datos de la cita

Cuando lo tengamos relleno apretamos el botón para añadir la cita y esperamos el mensaje que nos confirma que la operación se ha realizado.

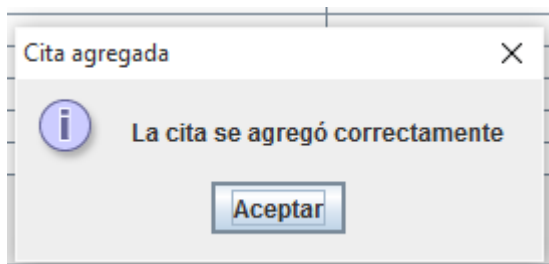


Figura 149: Cita agregada

Podemos ver que las citas se muestran todas con el mismo formato en el formulario de citas.

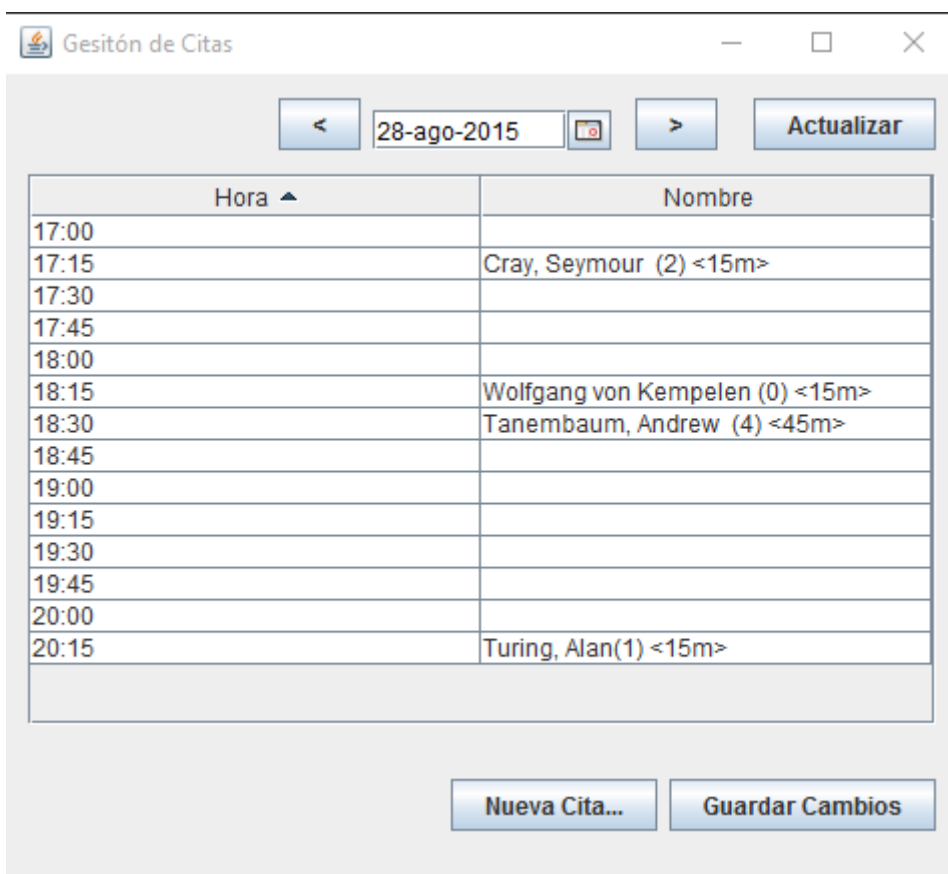


Figura 150: Formulario de citas con varios tipos de citas

3.4. MENÚ ESTADÍSTICAS

La última opción seleccionable desde el menú principal es la sección de estadísticas aquí podremos visualizar hasta 12 tipos de informes distintos según las opciones que escojamos. La aplicación nos da la opción de ver informes diarios, mensuales, anuales y totales y la capacidad de mostrar los datos de tres maneras distintas. La primera de ellas es agrupando por el acto médico para poder ver el balance global de la consulta. La siguiente opción es ver todas las consultas con los nombres de los pacientes y las fechas de realización. Por último podremos llevar un registro de las consultas que se han realizado y están pendientes de cobro. Solo tenemos que seleccionar las opciones que más nos interesen y pulsar el botón para que se muestre el informe.

Estadísticas

Periodo de Consulta: Día 25-ago-2015 Mes agosto 2015 Año 2015 Todo

Tipo de Consulta: Agrupar por actos Agrupar por paciente Consultas Pendientes

Mostrar Datos

| Acto Médico | Cantidad | Recaudación |
|---------------------------|----------|-------------|
| Papiloma Mapre | 1 | 0 |
| Quiropodia Particular | 2 | 56 |
| Tratamiento Hiperhidrosis | 1 | 30 |

Total actos: 4 Total importe 86 €

Figura 151: Informe por día y actos

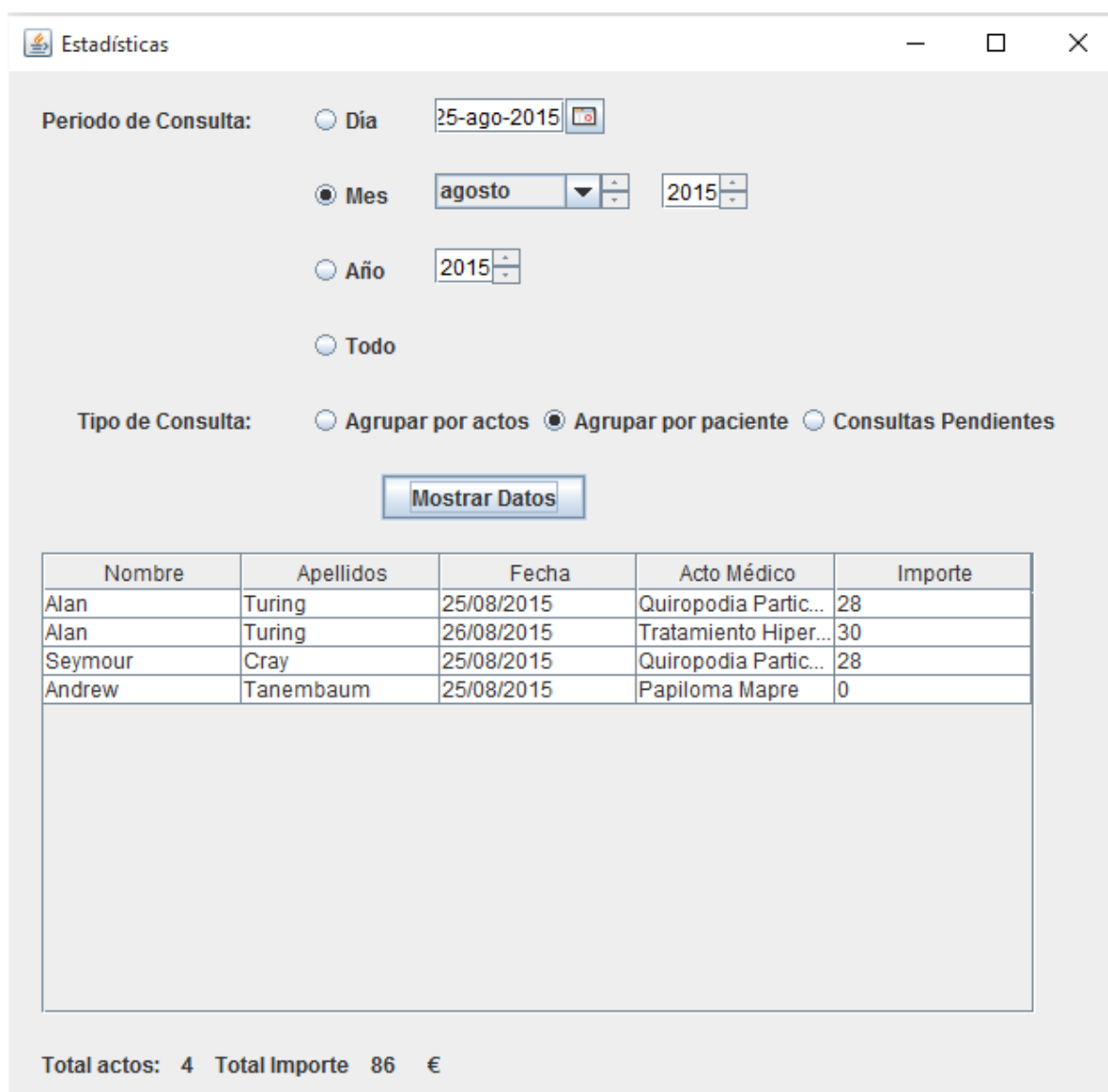


Figura 152: Informe por mes y pacientes

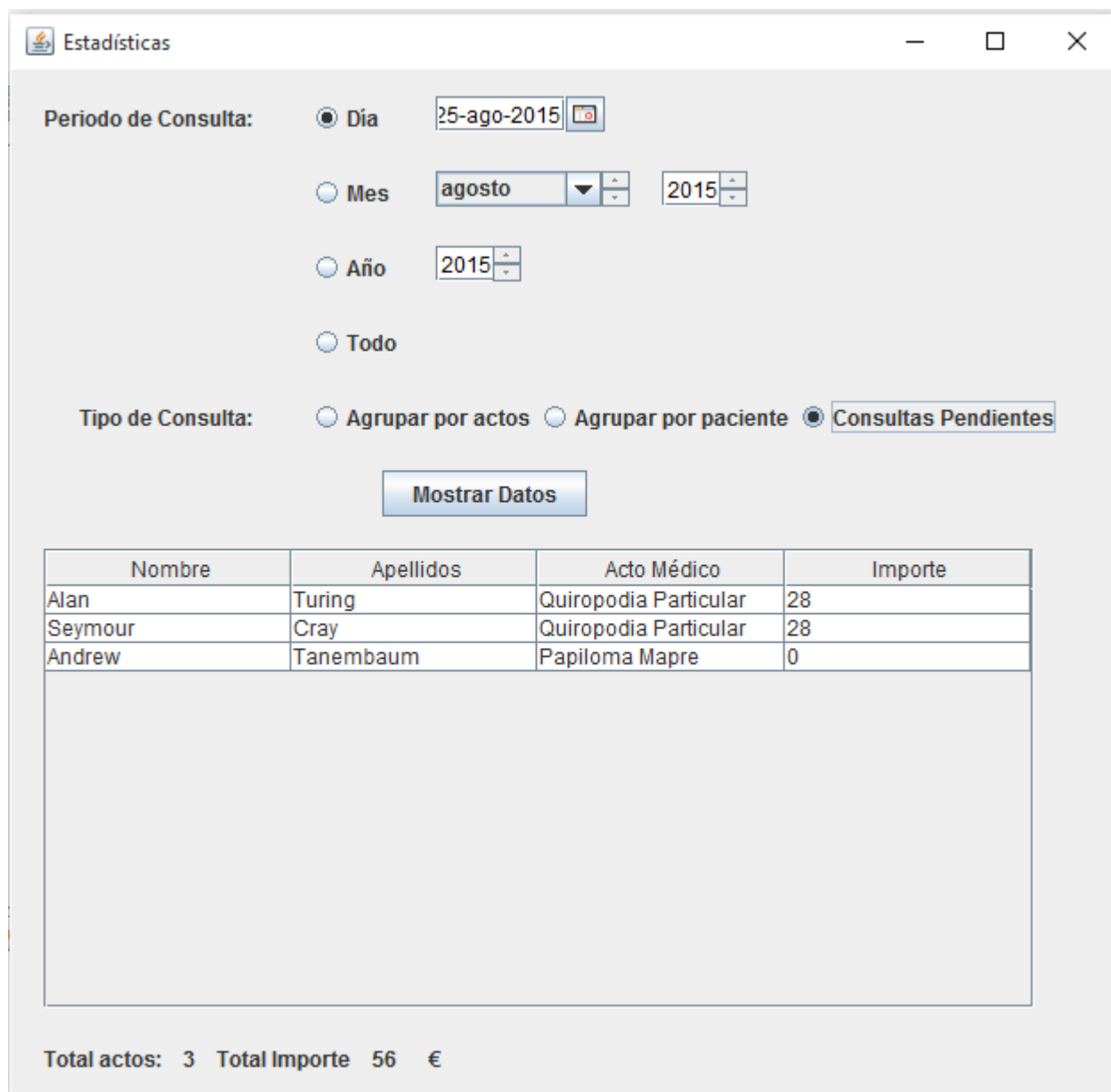


Figura 153: Informe por día y consultas pendientes

ANEXO: INDICE DE FIGURAS

| | |
|---|----|
| Figura 1: Arquitectura de tres capas | 10 |
| Figura 2: Planificación temporal inicial | 12 |
| Figura 3: Planificación inicial, diagrama de Gantt | 13 |
| Figura 4: Seguimiento real del proyecto..... | 13 |
| Figura 5: Seguimiento real, diagrama de Gantt..... | 14 |
| Figura 6: Comparativa planificación inicial-seguimiento real | 15 |
| Figura 7: Costes de recursos humanos..... | 16 |
| Figura 8: Costes hardware | 16 |
| Figura 9: Costes software | 17 |
| Figura 10: Coste total de la aplicación | 17 |
| Figura 11: Objetivo OBJ-01 | 23 |
| Figura 12: Objetivo OBJ-02 | 23 |
| Figura 13: Objetivo OBJ-03 | 23 |
| Figura 14: Objetivo OBJ-04 | 24 |
| Figura 15: Objetivo OBJ-05 | 24 |
| Figura 16: Requisito IRQ-01 | 25 |
| Figura 17: Requisito IRQ-02 | 25 |
| Figura 18: Requisito IRQ-03 | 26 |
| Figura 19: Requisito IRQ-04 | 26 |
| Figura 21: Definición de actor ACT-01..... | 28 |
| Figura 22: Caso de uso UC-01..... | 28 |
| Figura 23: Caso de uso UC-01, subcaso 01 | 29 |
| Figura 24: Caso de uso UC-01, subcaso 01. Diagrama de interacción..... | 30 |
| Figura 25: Caso de uso UC-01, subcaso 01. Diagrama de colaboración..... | 30 |
| Figura 26: Caso de uso UC-01, subcaso 02..... | 31 |
| Figura 27: Caso de uso UC-01, subcaso 02. Diagrama de interacción..... | 31 |
| Figura 28: Caso de uso UC-01, subcaso 02. Diagrama de colaboración..... | 32 |
| Figura 29: Caso de uso UC-01, subcaso 03 | 32 |
| Figura 30: Caso de uso UC-01, subcaso 03. Diagrama de interacción..... | 33 |
| Figura 31: Caso de uso UC-01, subcaso 03. Diagrama de colaboración..... | 33 |
| Figura 32: Caso de uso UC-02..... | 34 |
| Figura 33: Caso de uso UC-02, subcaso 01 | 34 |
| Figura 34: Caso de uso UC-02, subcaso 01. Diagrama de interacción..... | 35 |
| Figura 35: Caso de uso UC-02, subcaso 01. Diagrama de colaboración..... | 35 |
| Figura 36: Caso de uso UC-02, subcaso 02 | 36 |
| Figura 37: Caso de uso UC-02, subcaso 02. Diagrama de interacción..... | 36 |
| Figura 38: Caso de uso UC-02, subcaso 02. Diagrama de colaboración..... | 37 |
| Figura 39: Caso de uso UC-02, subcaso 03 | 37 |
| Figura 40: Caso de uso UC-02, subcaso 03. Diagrama de interacción..... | 38 |
| Figura 41: Caso de uso UC-02, subcaso 03. Diagrama de colaboración..... | 38 |
| Figura 42: Caso de uso UC-03..... | 39 |
| Figura 43: Caso de uso UC-03, subcaso 01 | 39 |
| Figura 44: Caso de uso UC-03, subcaso 01. Diagrama de interacción..... | 40 |
| Figura 45: Caso de uso UC-03, subcaso 01. Diagrama de colaboración..... | 40 |
| Figura 46: Caso de uso UC-03, subcaso 02 | 41 |
| Figura 47: Caso de uso UC-03, subcaso 02. Diagrama de interacción..... | 41 |
| Figura 48: Caso de uso UC-03, subcaso 02. Diagrama de colaboración..... | 42 |

| | |
|--|----|
| Figura 49: Caso de uso UC-03, subcaso 03 | 42 |
| Figura 50: Caso de uso UC-03, subcaso 03. Diagrama de interacción | 43 |
| Figura 51: Caso de uso UC-03, subcaso 03. Diagrama de colaboración | 43 |
| Figura 52: Caso de uso UC-04 | 44 |
| Figura 53: Caso de uso UC-04. Diagrama de interacción | 44 |
| Figura 54: Caso de uso UC-04. Diagrama de colaboración | 45 |
| Figura 55: Diagrama de clases | 47 |
| Figura 56: Clase ClinPodApp | 49 |
| Figura 57: Clase ClinPodView | 49 |
| Figura 58: Clase ClinPodAboutBox | 50 |
| Figura 59: Clase FormRelaciones | 50 |
| Figura 60: Clase FormPacientes | 51 |
| Figura 61: Clase FormConsultas | 52 |
| Figura 62: Clase FormAgenda | 53 |
| Figura 63: Clase FormCitaNueva | 54 |
| Figura 64: Clase FormEstadisticas | 55 |
| Figura 65: Clase cPacientes | 56 |
| Figura 66: Clase cCitas | 57 |
| Figura 67: Clase cCitasPK | 57 |
| Figura 68: Clase cHistorial | 58 |
| Figura 69: Clase cHistorialPK | 58 |
| Figura 70: Clase cRelaciónPacientes | 59 |
| Figura 71: Clase cRelaciónPacientesPK | 59 |
| Figura 72: Clase cCitasJpaController | 60 |
| Figura 73: Clase cHistorialJpaController | 60 |
| Figura 74: Clase cPacientesJpaController | 61 |
| Figura 75: Clase cRelaciónPacientesJpaController | 61 |
| Figura 76: Diagrama de estados paciente | 62 |
| Figura 77: Diagrama de estados cita | 62 |
| Figura 78: Diagrama de estados consulta | 62 |
| Figura 79: Diagrama Entidad-Relación | 63 |
| Figura 80: Diagrama Relacional | 64 |
| Figura 81: Tabla Pacientes | 65 |
| Figura 82: Tabla Historial | 65 |
| Figura 83: Tabla Historial | 66 |
| Figura 84: Tabla Citas | 66 |
| Figura 85: Prueba de Uso 1 | 67 |
| Figura 86: Prueba de Uso 2 | 67 |
| Figura 87: Prueba de Uso 3 | 67 |
| Figura 88: Prueba de Uso 4 | 68 |
| Figura 89: Prueba de Uso 5 | 68 |
| Figura 90: Prueba de Uso 6 | 68 |
| Figura 91: Prueba de Uso 7 | 69 |
| Figura 92: Prueba de Uso 8 | 69 |
| Figura 93: Prueba de Uso 9 | 69 |
| Figura 94: Prueba de Uso 10 | 70 |
| Figura 95: Prueba de Uso 11 | 70 |
| Figura 96: Prueba de Uso 12 | 70 |
| Figura 97: Prueba de Uso 13 | 71 |

| | |
|---|-----|
| Figura 98: Prueba de Uso 14 | 71 |
| Figura 99: Prueba de Uso 15 | 71 |
| Figura 100: Prueba de Uso 16 | 71 |
| Figura 101: Prueba de Uso 17 | 72 |
| Figura 102: Prueba de Uso 18 | 72 |
| Figura 103: Prueba de Uso 19 | 72 |
| Figura 104: Prueba de Uso 20 | 72 |
| Figura 105: Prueba de Uso 21 | 73 |
| Figura 106: Prueba de Uso 22 | 73 |
| Figura 107: Prueba de Uso 23 | 73 |
| Figura 108: Prueba de Uso 24 | 73 |
| Figura 109: Prueba de Uso 25 | 74 |
| Figura 110: Prueba de Uso 26 | 74 |
| Figura 111: Prueba de Uso 27 | 74 |
| Figura 112: Prueba de Uso 28 | 75 |
| Figura 113: Prueba de Uso 29 | 75 |
| Figura 114: Prueba de Uso 30 | 75 |
| Figura 115: Prueba de Uso 31 | 76 |
| Figura 116: Prueba de Uso 32 | 76 |
| Figura 117: Prueba de Uso 33 | 76 |
| Figura 118: Prueba de Uso 34 | 77 |
| Figura 119: Prueba de Uso 35 | 77 |
| Figura 120: Prueba de Uso 36 | 77 |
| Figura 121: Prueba de Uso 37 | 78 |
| Figura 122: Pantalla inicial de la aplicación..... | 82 |
| Figura 123: Menú Archivo | 83 |
| Figura 124: Menú Ayuda..... | 83 |
| Figura 125: Dialogo Acerca de..... | 84 |
| Figura 126: Vista de dos ventanas de la aplicación funcionando simultáneamente..... | 84 |
| Figura 127: Gestión de Pacientes | 85 |
| Figura 128: Confirmación de paciente agregado | 86 |
| Figura 129: Búsqueda por id de Paciente | 87 |
| Figura 130: Búsqueda por apellidos del paciente | 88 |
| Figura 131: Función de autocompletado | 89 |
| Figura 132: Confirmación de modificación de paciente..... | 90 |
| Figura 133: Agregando pacientes relacionados | 91 |
| Figura 134: Error al intentar relacionar un paciente consigo mismo..... | 92 |
| Figura 135: Relación agregada correctamente | 93 |
| Figura 136: Vista de un paciente con una relación..... | 94 |
| Figura 137: Visor de consulta..... | 95 |
| Figura 138: Rellenando los datos de una consulta..... | 96 |
| Figura 139: Consulta guardada..... | 96 |
| Figura 140: Vista de consultas en la ficha del paciente..... | 97 |
| Figura 141: Modo de modificación de consulta | 98 |
| Figura 142: Consulta modificada..... | 99 |
| Figura 143: Gestión de citas | 100 |
| Figura 144: Desplegable para elegir un paciente de la base de datos..... | 101 |
| Figura 145: Agregando un paciente manualmente | 102 |
| Figura 146: Citas guardadas | 102 |

| | |
|--|-----|
| Figura 147: Agregar cita compuesta | 103 |
| Figura 148: Completando los datos de la cita | 103 |
| Figura 149: Cita agregada | 104 |
| Figura 150: Formulario de citas con varios tipos de citas..... | 104 |
| Figura 151: Informe por día y actos..... | 105 |
| Figura 152: Informe por mes y pacientes..... | 106 |
| Figura 153: Informe por día y consultas pendientes | 107 |