

UNIVERSIDAD DE



VALLADOLID

E.T.S.I. TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE ESPECÍFICAS DE
TELECOMUNICACIÓN, MENCIÓN EN SISTEMAS DE
TELECOMUNICACIÓN

Aplicación Android para pacientes de fisioterapia

Autor:

Javier Mallo Herráez

Tutora:

Dña. Míriam Antón Rodríguez

Valladolid, 13 de Abril de 2015

TÍTULO: **Aplicación Android para pacientes de
fisioterapia**

AUTOR: **Javier Mallo Herráez**

TUTORA: **Dña. Míriam Antón Rodríguez**

DEPARTAMENTO: **Teoría de la Señal y Comunicaciones e
Ingeniería Telemática**

TRIBUNAL

PRESIDENTA: **Dña. Míriam Antón Rodríguez**

VOCAL: **D. Mario Martínez Zarzuela**

SECRETARIO **D. David González Ortega**

SUPLENTE **Dña. M^a Ángeles Pérez Juárez**

SUPLENTE **D. Francisco Javier Díaz Pernas**

FECHA: **13 de Abril de 2015**

CALIFICACIÓN:

RESUMEN DEL PROYECTO

Las aplicaciones móviles de apoyo a la salud se están convirtiendo en herramientas necesarias en el día a día, tanto las destinadas a los profesionales sanitarios como las dirigidas solamente a los pacientes.

Sin embargo, no todas las áreas de la medicina están suficientemente apoyadas por este tipo de aplicaciones. En concreto, la fisioterapia es uno de las áreas de la salud que necesita un mayor esfuerzo en este sentido.

En éste Trabajo Fin de Grado se propone el desarrollo de una aplicación móvil Android que proporcione a los pacientes de fisioterapia una herramienta de consulta de información específica y de solicitud de cuidados en clínica. Con esta aplicación para smartphones se pretende que el paciente pueda obtener información sobre las clínicas asociadas a la aplicación (como llegar, teléfono, mail...), así como poder realizar una reserva en dichas clínicas para poder tener una cita con el fisioterapeuta. Además, el paciente podrá interactuar con su fisioterapeuta asignado pudiendo contactar con él y recibir consejos y ejercicios adecuados para su problema.

ABSTRACT

The mobile phone applications supporting the health are becoming essential tools in our daily life, either the ones meant for professionals as the one meant just for patients.

However, not all areas of medicine are sufficiently supported by these types of applications. In fact, physiotherapy is one of the areas of health that requires a bigger effort in this regard.

In this Final Bachelor Degree Project we propose the development of an Android mobile application that provides to the physiotherapy patients a consultation tool of specific information and application in clinical care is proposed. With this smartphone application is pretended that the patient could obtain information about the clinics associated with the application (how to get, telephone, mail...), as well as to make a reservation at these clinics in order to have an appointment with the therapist. In addition, the patient may interact with the assigned physiotherapist having contact to get advice and appropriate exercises for his problem.

PALABRAS CLAVE Y KEYWORDS

Eclipse, Android, layout, XML, java, activity, physiotherapy, base de datos.

AGRADECIMIENTOS

Agradecer a mis padres Agustín y María José, a mi abuela Justa y a mi hermana Noelia por haber estado a mi lado durante todos estos años apoyándome tanto en los exámenes como en este proyecto.

Agradecer el apoyo durante toda la carrera a mis compañeros de clase y de proyecto, "Los Argucieros", Javier, Jairo, Rubén, Mario, David, Pope, Elena e Isma porque sin ellos se me hubiera hecho mucho más largo todo este camino.

A mi tutora Miriam por haberse adaptado a la petición de un proyecto de las características que queríamos y por haberme ayudado cada vez que ha podido y haberme ido guiando a lo largo tanto del proyecto como en las asignaturas que me ha impartido.

Gracias a todos.

Índice de contenidos

CAPÍTULO 1: INTRODUCCIÓN GENERAL	1
1.1 OBJETIVOS.....	9
1.2 FASES Y MÉTODOS.....	10
1.3 MEDIOS	10
1.4 ESTRUCTURA DEL DOCUMENTO.....	11
CAPÍTULO 2: TECNOLOGÍAS.....	13
2.1 INTRODUCCIÓN A LAS GENERACIONES MÓVILES	13
2.1.1 <i>Generación 0</i>	13
2.1.2 <i>Primera Generación (1G)</i>	14
2.1.3 <i>Segunda Generación (2G)</i>	14
2.1.4 <i>Tercera Generación (3G)</i>	15
2.1.5 <i>Cuarta Generación (4G)</i>	16
2.2 SISTEMAS OPERATIVOS	17
2.2.1 <i>iOS</i>	17
2.2.1.1 <i>Evolución</i>	17
2.2.1.2 <i>Ventajas y desventajas</i>	19
2.2.2 <i>Android</i>	21
2.2.2.1 <i>Evolución</i>	22
2.2.2.2 <i>Ventajas y desventajas</i>	24
2.3 OTROS SISTEMAS OPERATIVOS	26
2.3.1 <i>Windows Phone</i>	26
2.3.2 <i>Ubuntu OS</i>	28
2.3.3 <i>Tizen</i>	28
2.3.4 <i>Firefox OS</i>	29
2.4 TECNOLOGÍAS DE PROGRAMACIÓN DEL LADO DEL SERVIDOR	30
2.4.1 <i>PHP</i>	30
2.5 BASES DE DATOS	32
2.5.1 <i>Bases de datos relacionales</i>	32
2.5.2 <i>Tipos</i>	33
2.6 SERVICIOS WEB	35
2.6.1 <i>REST</i>	35
2.6.2 <i>SOAP</i>	40
2.7 COMPARACIÓN Y ELECCIÓN	42
2.7.1 <i>Sistema operativo</i>	42
2.7.2 <i>Lenguaje de servidor y base de datos</i>	44
2.7.3 <i>Servicio web</i>	44
CAPÍTULO 3: ANDROID.....	46
3.1 ESTRUCTURA ANDROID	46
3.2 PROGRAMACIÓN Y DESARROLLO ANDROID	48
CAPÍTULO 4: DESCRIPCIÓN TÉCNICA DEL USUARIO	52
4.1 ESTRUCTURA DE LA BASE DE DATOS.....	52
4.1.1 <i>Tabla Clinics</i>	54
4.1.2 <i>Tabla Treatments</i>	55

4.1.3	<i>Tabla Clinics_Treatments</i>	56
4.1.4	<i>Tabla Users</i>	56
4.1.5	<i>Tabla Exercises</i>	57
4.1.6	<i>Tabla Users_Exercise</i>	58
4.1.7	<i>Tabla Professionals</i>	59
4.1.8	<i>Tabla Reservations</i>	60
4.1.9	<i>Tabla News</i>	61
4.2	RELACIÓN ENTRE LAS TABLAS DE LA BASE DE DATOS	62
4.2.1	<i>Clinics y Treatments</i>	62
4.2.2	<i>Professionals, Exercises y Users</i>	63
4.2.3	<i>Clinics, Treatments, Reservations y Professionals</i>	64
4.2.4	<i>Professionals, Reservations y Users_Exercise</i>	65
4.2.5	<i>Professionals, Reservations, Clinics y Users</i>	66
4.3	DESCRIPCIÓN DE FUNCIONALIDADES	68
4.3.1	<i>Login</i>	68
4.3.2	<i>Registrar</i>	70
4.3.3	<i>Usuario Principal</i>	71
4.3.4	<i>Ver Clínicas</i>	72
4.3.5	<i>Lista de Tratamientos</i>	74
4.3.6	<i>Reservar</i>	76
4.3.7	<i>Lista de Ejercicios</i>	81
4.3.8	<i>Mis Reservas</i>	82
CAPÍTULO 5: MANUAL DE USUARIO		85
5.1	LOGIN.....	85
5.2	REGISTRO	86
5.3	USUARIO PRINCIPAL	88
5.4	CLÍNICAS	89
5.4.1	<i>Lista de tratamientos</i>	91
5.4.2	<i>Cómo llegar</i>	93
5.4.3	<i>Compartir</i>	94
5.4.4	<i>Contactar con la clínica</i>	95
5.5	RESERVAR	95
5.6	LISTA DE EJERCICIOS.....	99
5.7	MIS RESERVAS	101
CAPÍTULO 6: PRESUPUESTO ECONÓMICO		103
CAPÍTULO 7: CONCLUSIONES Y LÍNEAS FUTURAS.....		105
7.1	CONCLUSIONES	105
7.2	LÍNEAS FUTURAS	107
CAPÍTULO 8: BIBLIOGRAFÍA.....		109
CAPÍTULO 9: ANEXOS.....		113
ANEXO I. PUESTA EN MARCHA DEL ENTORNO DE PROGRAMACIÓN.....		113
INSTALACIÓN Y PUESTA EN MARCHA DE INSTALACIÓN DE <i>ECLIPSE</i>		113
INSTALACIÓN DEL SDK DE <i>ANDROID</i>		114
INSTALACIÓN DEL <i>PLUING ADT ANDROID</i> PARA <i>ECLIPSE</i>		115
ANEXO II. WRAPPER.....		118

Índice de figuras

Figura 1. Ejemplo de la aplicación de prevención de cáncer.	6
Figura 2. Aplicación TE ACOMPAÑA (Movistar)	7
Figura 3. HandieTalkie H12-16	13
Figura 4. Teléfono de primera generación (1G).....	14
Figura 5. Teléfono de segunda generación (2G).....	15
Figura 6. Teléfono de tercera generación (3G)	16
Figura 7. Teléfono de cuarta generación (4G)	16
Figura 8. Evolución IOS.....	17
Figura 9. Versiones de Android.	21
Figura 10. Esquema del funcionamiento de las paginas PHP.	31
Figura 11. Servicio con estado.	37
Figura 12. Servicio sin estado	38
Figura 13. Ratings de los lenguajes de programación del índice TIOBE 2014.....	43
Figura 14. Gráfica del incremento del número de desarrolladores de los principales sistemas operativos móviles.	44
Figura 15. Estructura de sistema operativo Android.....	46
Figura 16. La compilación del código .java	49
Figura 17. Conversión de .class a .dex.....	50
Figura 18. Estructura de la base de datos.	53
Figura 19. Relación entre la tabla clinics y la tabla treatments (Verde).....	63
Figura 20. Relación entre las tablas professionals, exercises y users (Azul).	64
Figura 21. Relación entre las tablas professionals, clinics, treatments y reservations (Amarillo).	65
Figura 22. Relación entre las tablas professionals, reservations y users_exercise (Rojo).	66
Figura 23. Relación entre las tablas professionals, reservations, users y clinics (Naranja).....	67
Figura 24. Diagrama de flujo de la clase Login.	68
Figura 25. Diagrama de flujo de la funcionalidad Login.	69
Figura 26. Diagrama de flujo de la funcionalidad Registrar.	70
Figura 27. Diagrama de flujo de la funcionalidad UsuarioPrincipal.....	71
Figura 28. Diagrama de flujo de la clase UsuarioPrincipal.....	72
Figura 29. Diagrama de flujo de la clase VerClinicas.....	73
Figura 30. Diagrama de flujo de la funcionalidad VerClinicas.....	74
Figura 31. Diagrama de flujo de la clase VerTratamientos.....	75
Figura 32. Diagrama de flujo de la funcionalidad VerTratamientosClinicas.....	76

Figura 33. Diagrama de flujo de la clase Reservar	77
Figura 34. Diagrama de flujo de la clase ReservarCita.....	78
Figura 35. Diagrama de flujo de la funcionalidad ReservarCita.....	80
Figura 36. Diagrama de flujo de la clase VerEjercicios.....	81
Figura 37. Diagrama de flujo de la funcionalidad de VerEjercicios.....	82
Figura 38. Diagrama de flujo de la clase VerReservasUsuario.	83
Figura 39. Diagrama de flujo de funcionalidad de VerReservasUsuario.....	84

Índice de tablas

Tabla 1. Tabla clinics	55
Tabla 2. Tabla Treatments	56
Tabla 3. Tabla Clinics_Treatments	56
Tabla 4. Tabla Users	57
Tabla 5. Tabla excercises	58
Tabla 6. Tabla users_exercise	58
Tabla 7. Tabla Professionals	60
Tabla 8. Tabla Reservations.....	61
Tabla 9. Tabla News	62

Índice de layouts

Layout 1. Login.....	85
Layout 2. Login incorrecto.	86
Layout 3. Registro.....	87
Layout 4. Registro sin campos obligatorios.....	87
Layout 5. Registro usuario ya existente.....	88
Layout 6. Usuario_principal.	89
Layout 7. Ver_clinicas, lista rellena y lista vacía.....	90
Layout 8. Vista_clinica, con visión del menú	91
Layout 9. Ver_tratamientos , lista rellena y lista vacía.	92
Layout 10. Vista_tratamiento, con opciones de menú y vista opción compartir.	93
Layout 11. Cómo llegar a la clínica, usando Google Maps.	94
Layout 12. Compartir clínica.	95
Layout 13. Contactar con la clínica	95
Layout 14. Ver_clinicas y vista_clinica_reserva.	96

Layout 15. Comentario y lugar del dolor.....	96
Layout 16. Selección del día de la cita.....	97
Layout 17. Selección de la hora de la cita.....	98
Layout 18. Resumen de la reserva.....	98
Layout 19. Consulta de la hora disponible.....	99
Layout 20. Lista de ejercicios adjudicados al paciente y lista vacía.....	100
Layout 21. Vista del ejercicio y comentario del profesional.....	100
Layout 22. Vista de Mis Reservas.....	101
Layout 23. Aviso y confirmación de la anulación de la cita.....	102

Índice de anexos

Anexo 1. Lugar de descarga del programa de desarrollo.....	114
Anexo 2. Descarga del SDK desde la página de Android Developers.....	115
Anexo 3. Primer paso de la instalación ADT <i>plugin</i> en <i>Eclipse</i>	116
Anexo 4. Instalación del plugin ADT.....	117
Anexo 5. Configuración del ADT <i>plugin</i>	117
Anexo 6. Wrapper realizado para el Login.....	118
Anexo 7. Principio de la tarea en segundo plano.....	119
Anexo 8. Guardando variables en el wrapper.....	120
Anexo 9. Guardando variables en el wrapper II.....	120
Anexo 10. onPostExecute y error.....	121

Capítulo 1: Introducción general

La OMS [1], es la autoridad directiva y coordinadora de la acción sanitaria en el sistema de las Naciones Unidas. Es la responsable de desempeñar una función de liderazgo en los asuntos sanitarios mundiales, configurar la agenda de las investigaciones en salud, establecer normas, articular opciones de política basadas en la evidencia, prestar apoyo técnico a los países y vigilar las tendencias sanitarias mundiales. En el siglo XXI, la salud es una responsabilidad compartida, que exige el acceso equitativo a la atención sanitaria y la defensa colectiva frente a amenazas transnacionales.

Los sistemas sanitarios se pueden considerar variantes directos de los sistemas sociales, definidos estos como el conjunto de relaciones sociales, estructuras y estratificaciones sociales de la sociedad. La parte más destacada e importante de los sistemas sociales son las llamadas funciones sociales, que son el conjunto de actividades dirigidas a satisfacer las necesidades de la sociedad, y las instituciones o estructuras que tienen por finalidad ejecutarlas. Dentro de los sistemas sociales una de las funciones de mayor complejidad son aquellas relacionadas con la atención a la salud, las cuales conforman lo que podemos denominar el sistema sanitario. Los sistemas sanitarios, son, por lo tanto, sistemas sociales que la Organización Mundial de la Salud (OMS) los ha definido más recientemente como un conjunto de elementos interrelacionados que contribuyen a la salud en los hogares, los lugares de trabajo, los lugares públicos y las comunidades, así como en el medio ambiente físico y psicosocial, y en el sector de la salud y otros sectores afines. Según la OMS la definición del sistema sanitario en el año 1946 es: «estado de completo bienestar físico, mental y social, y no solamente la ausencia de afecciones o enfermedades» convirtiéndose de este modo los sistemas sanitarios en uno de los más importantes indicadores, y vital para poder analizar el Estado de Bienestar de una sociedad y su nivel de desarrollo [2].

Según la Organización Mundial de la Salud (OMS), la ciber salud (conocida también como e-Salud o e-Health) consiste "en el apoyo que la utilización costoeficaz y segura de las tecnologías de la información y las comunicaciones ofrece a la salud y a los ámbitos relacionados con ella, con inclusión de los servicios de atención de salud, la vigilancia y la documentación sanitarias, así como la educación, los conocimientos y las investigaciones en materia de salud". Algunos de los componentes fundamentales de e-Salud o e-Health son los siguientes:



CAPÍTULO 1

a) Registro médico electrónico (o historia clínica electrónica): es el registro en formato electrónico de información sobre la salud de cada paciente que puede ayudar a los profesionales de salud en la toma de decisiones y el tratamiento.

b) Telesalud (incluida la telemedicina): consiste en la prestación de servicios de salud utilizando las tecnologías de la información y la comunicación, especialmente donde la distancia es una barrera para recibir atención de salud.

c) mSalud (o salud por dispositivos móviles): es un término empleado para designar el ejercicio de la medicina y la salud pública con apoyo de los dispositivos móviles, como teléfonos móviles, dispositivos de monitoreo de pacientes y otros dispositivos inalámbricos.

d) eLearning (incluida la formación o aprendizaje a distancia): aplicación de las tecnologías de la información y la comunicación al aprendizaje. Puede utilizarse para mejorar la calidad de la educación, aumentar el acceso a la educación y crear formas nuevas e innovadoras de enseñanza al alcance de un mayor número de personas.

e) Educación continua en tecnologías de la información y la comunicación: consiste en el desarrollo de cursos o programas de salud profesionales (no necesariamente acreditados formalmente) que facilitan habilidades en tecnologías de la información y la comunicación de aplicación en la salud. Esto incluye los métodos actuales para el intercambio de conocimiento científico como la publicación electrónica, el acceso abierto, la alfabetización digital y el uso de las redes sociales.

f) Estandarización e interoperabilidad: la interoperabilidad hace referencia a la comunicación entre diferentes tecnologías y aplicaciones de software para el intercambio y uso de datos en forma eficaz, precisa y sólida. Esto requiere del uso de estándares, es decir, de normas, regulaciones, guías o definiciones con especificaciones técnicas para hacer viable la gestión integrada de los sistemas de salud en todos los niveles [3].

Los avances que proporcionan las nuevas tecnologías están presentes en diferentes facetas de nuestra rutina diaria: la educación, la comunicación o la producción industrial. La medicina es también un campo en el que las nuevas innovaciones tienen mucho que decir. Todos estos avances pueden ayudar a la hora del cuidado de nuestra salud.

Las aplicaciones móviles también aportan beneficios en el ámbito de la medicina, facilitando una mejor relación entre los profesionales sanitarios y los pacientes

Transformar un área tan “tradicional” a la vez que innovadora como la medicina no es tarea sencilla pero gracias a la intervención de los dispositivos



móviles se ha marcado un punto de partida para la expansión de los sistemas sanitarios en estos terminales. Hoy existen miles de aplicaciones móviles que podemos utilizar a través de nuestros smartphones o tablets. Algunas de ellas permiten monitorizar algunas constantes vitales, facilitando su registro y envío a nuestro médico personal [4].

Decir que aplicaciones para smartphones hay muchísimas es una obviedad, pero que existan casi 100.000 aplicaciones destinadas a la salud, da que pensar. Una «aplicación móvil» (app) es un programa de software que se ejecuta sobre un dispositivo móvil con tecnología inalámbrica, ya sean teléfonos inteligentes, tabletas, portátiles o cualquier dispositivo electrónico con conectividad sin cables (wearables, smartwatch...). Una «aplicación médica móvil» es una aplicación, que tiene como finalidad realizar una función médica (prevención, diagnóstico, curación, tratamiento o mitigación de una enfermedad, o bien a que afecten a las funciones o morfología del cuerpo). A este subconjunto de aplicaciones de la salud y su ecosistema de dispositivos específicos lo denominamos mHealth (Salud Móvil).

Según un estudio de Research2Guidance, en 2015 más de 500 millones de personas usarán aplicaciones médicas en sus teléfonos móviles. De estas aplicaciones, no sólo encontramos las que miden constantes vitales, sino que en las principales tiendas de aplicaciones, como Google Play o App Store, se distribuyen aplicaciones relacionadas con la gestión y control del bienestar y la alimentación o los atlas y las consultas médicas. También existen aplicaciones que recuerdan cuándo se debe tomar un determinado medicamento, o que ayudan a conocer factores externos que pueden afectar a la salud (por ejemplo, los niveles de polen en pacientes asmáticos).

Pero el desarrollo ha supuesto un desafío importante para las agencias reguladoras, como la norteamericana FDA [5] o la europea EMEA [6].

Estas entidades, son las encargadas de autorizar la producción de medicamentos. En este sentido, la mHealth Regulatory Coalition identificó en el Libro Blanco sobre la Regulación de la mHealth por la FDA tres lagunas importantes para ser discutidas, en torno a la creación y regulación de aplicaciones sanitarias:

- Por una parte, debería discernirse bien cuál es la intencionalidad de uso de la propia app, diferenciando si son productos dirigidos únicamente al bienestar del ciudadano o por contra, se utilizan como herramientas de manejo y diagnóstico de determinadas enfermedades.



CAPÍTULO 1

- Por otra parte, también es interesante saber si los accesorios conectados y la propia configuración de los componentes deberían ser supervisados de algún modo por estas agencias. ¿Deberían ser regulados los dispositivos móviles por su uso dando servicios de productos sanitarios? El debate, sin dudas, está servido.
- Lo mismo ocurriría con el software utilizado, ¿tendría que ser controlado por entidades como la FDA o la EMEA? El diseño de estas aplicaciones instaladas en smartphones, ordenadores, servidores o en la nube cambiaría radicalmente si tuviera que pasar por un proceso de supervisión.

Un ejemplo en este caso era MediCom, un microchip de 14 mm implantado en los pacientes, a través del cual se podían realizar análisis de sangre, y cuyos resultados se enviaban directamente al profesional médico de forma inmediata. Y aunque existen dudas respecto a la evidencia científica de algunas de estas aplicaciones, lo cierto es que la FDA cuenta con un protocolo de aprobación de aplicaciones móviles, mientras que la EMEA sigue con un registro más estándar.

Algunas cifras de mHealth que muestran cómo de grande es este mercado en el mundo:

- En EEUU, la monitorización de pacientes con dispositivos móviles generó en 2010 unos 6.100 millones de dólares. En 2017 está previsto que este mercado alcance los 8.000 millones de dólares [7].
- En Latinoamérica, la salud móvil generará un volumen de negocio de 1.600 millones de dólares con un crecimiento de un 50% anual para los próximos 6 años [8].
- En la UE, el uso de 'mHealth' permitiría ahorrar cerca de 99.000 millones de euros hasta 2017. Y además aportar 93.000 millones al PIB europeo [9].
- En Europa, el uso de tecnología móvil puede reducir el coste per cápita en sanidad un 18% y hasta un 35% en el tratamiento de pacientes crónicos [10].

Ahora se tiene la posibilidad de ejercer el autocontrol sobre nuestra salud, lo que se conoce como «empoderamiento del paciente», antes de tener que recurrir a centros médicos por urgencias gracias a los nuevos elementos que se incluyen en los teléfonos inteligentes.



En el Smartphone llevamos el teléfono, el reloj, la agenda, el correo, la música, las noticias, la cámara de fotos, el navegador de internet, GPS... y también podemos bajarnos apps para controlar el ejercicio físico, la alimentación sana, los recordatorios de medicamentos...

Las apps que registran parámetros corporales tales como tensión arterial, pulso, oximetría, espirometría, glucosa, peso... necesitan recibir información de sensores externos que tomen las biomedidas. Por tanto, además de los sensores integrados de movimiento que miden nuestra actividad física, un smartphone tiene la capacidad de asociar por bluetooth dichos sensores externos [11].

Uno de los mayores problemas económicos en el entorno de la salud son las enfermedades crónicas, que están aumentando en los últimos años. Este problema sanitario es responsable del 63 por ciento de los fallecimientos en el mundo y representa el 75 por ciento de los costes de salud en Estados Unidos y Europa.

España no escapa a esta tendencia. Con más de 20 millones de pacientes crónicos mayores de 50 años, las enfermedades crónicas son causa directa del 70 por ciento del gasto sanitario. La incorporación de soluciones e-Salud puede contribuir a solventar esta situación insostenible a largo plazo. El acceso del paciente a su propia historia clínica digital y la monitorización remota de pacientes crónicos con diabetes o enfermedades respiratorias mediante dispositivos y aplicaciones de m-Health son algunos ejemplos de lo que la tecnología ha conseguido en el ámbito sanitario. Así lo señalaron los expertos participantes en la sesión del congreso Mobile World Congress 2014: 'Tackling the global challenge of chronic disease: opportunities for mobile Health' [12].

El camino hacia la interoperabilidad del sistema sanitario no es fácil, pero los múltiples beneficios que aporta la e-Salud a la gestión y automatización de procesos clínicos justifican los esfuerzos en inversión [13].

Existen miles de aplicaciones, y cada vez más centradas en la salud o el bienestar de las personas. Las aplicaciones sobrepasan límites que hace unos años eran imposibles. Lo que se intenta es que estos avances tecnológicos lleguen a todos los ciudadanos del mundo, un ejemplo de ello es un nuevo dispositivo fabricado por científicos de la Universidad de Cornell que detecta algunos tipos de cáncer de manera más rápida y barata.

Uno de los grandes problemas sanitarios de los países del tercer mundo es que no cuentan con sistemas de diagnóstico precoz de enfermedades. Esta gran desventaja hace que los pacientes no puedan ser tratados en el momento que comienzan a sufrir una determinada condición patológica. En el caso del cáncer,



CAPÍTULO 1

este problema se agrava, puesto que cuanto antes se iniciara la administración de una terapia en estos individuos, mejor podrían responder a los tratamientos.

La utilización de la mHealth, sin embargo, podría cambiar en buena medida los sistemas de diagnóstico precoz de algunos tipos de cáncer. En particular, como se ha mencionado previamente, científicos de la Universidad de Cornell han estudiado un dispositivo construido a partir de un smartphone, que ayudaría a diagnosticar casos de pacientes afectados con sarcoma de Kaposi.

Este tipo de cáncer forma lesiones malignas en la piel, las membranas mucosas, los ganglios linfáticos y otros órganos. Con el dispositivo inventado por los científicos norteamericanos, se podría diagnosticar fácilmente este tipo de neoplasia maligna, disminuyendo considerablemente los tiempos de espera y los costes económicos asociados al diagnóstico del cáncer.

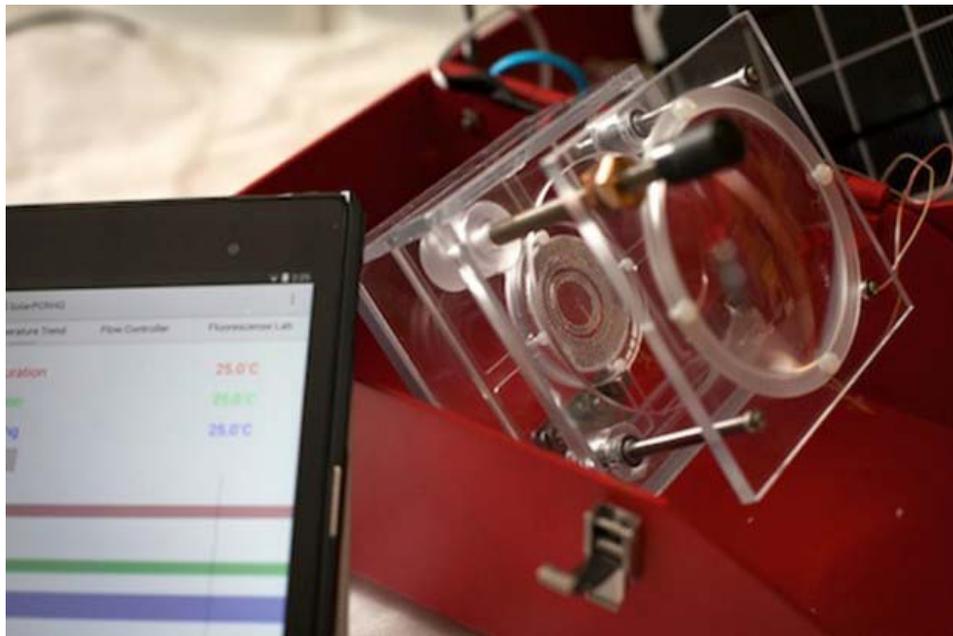


Figura 1. Ejemplo de la aplicación de prevención de cáncer.

El sistema, que cuesta algo menos de 500 dólares, es bastante sencillo. Cuenta con el propio smartphone, una aplicación móvil, unas lentes y un diminuto chip, que permiten detectar si los pacientes están afectados por este tipo de cáncer. El dispositivo funciona empleando nanopartículas de oro que son capaces de unirse a muestras de ADN viral, en el caso de que las personas sufran sarcoma de Kaposi.

Si la reacción química se produce, las partículas se combinan de manera conjunta y dejan de pasar menos luz de la habitual por el dispositivo, provocando un cambio de color que es visible a simple vista. Un sensor óptico acoplado al



smartphone puede “medir” ese cambio cromático para mostrar la severidad de la infección vírica que produce a la larga este tipo de cáncer.

El primer prototipo fue probado el año pasado en Uganda y Kenia, y dados los buenos resultados obtenidos, se prevé que pueda distribuirse en otras regiones del tercer mundo. Su bajo coste, asociado a la reducción de la energía que necesita para funcionar (alrededor de 100 veces menos que los métodos tradicionales, al estar alimentado por energía solar), convierten a este dispositivo en un candidato adecuado para mejorar la salud en estos países [14].

Por otro lado, y sin irnos tan lejos, Movistar, ha presentado "Te acompaña". Un servicio pensado para facilitar la vida a millones de personas en situación de dependencia, temporal o permanente, así como sus familias dándoles tranquilidad, movilidad y confianza.



Figura 2. Aplicación TE ACOMPAÑA (Movistar)

España es otro país que desde hace décadas muestra una clara tendencia hacia el envejecimiento de la población. Para el año 2020 un 19,2% de la población española tendrá más de 65 años. Por otro lado, estas personas se sienten, como es natural, mucho más a gusto viviendo en sus propias casas que en residencias o con familiares que se encargan de su cuidado.



CAPÍTULO 1

“Te Acompaña” nace fruto de esta búsqueda para ofrecer independencia y tranquilidad a millones de personas en situación de dependencia temporal o permanente, que pueden necesitar ayuda puntual de forma urgente y quieren tener una vida lo más plena posible.

Aprovechando los avances en las tecnologías de la comunicación, Te Acompaña es la evolución natural de los servicios de teleasistencia fija tradicional haciendo la vida mucho más sencilla y cómoda para todas las personas implicadas. Para las personas dependientes y sus familiares, un servicio como Te Acompaña implica un cambio radical en sus vidas ya que les ofrece tranquilidad, confianza y movilidad.

Este sistema busca transmitir tranquilidad gracias a que los usuarios saben que van a tener ayuda y asistencia siempre que la necesiten. El equipo que soporta la asistencia del servicio son todos profesionales que realizan una labor proactiva de comunicación y asistencia. En cualquier momento del día, el usuario se puede poner en contacto con el servicio para recibir el tipo de ayuda que requiera. Y dos veces al mes, el usuario recibe llamadas del servicio de asistencia para preguntar si está todo bien y si necesita algo [15].

Una vez mencionados los temas relacionados con la salud y visto que existen multitud de aplicaciones para ayudar al bienestar de la salud de las personas, el proyecto se va a centrar en los aspectos relacionados con la fisioterapia y acercar los aspectos tecnológicos a este ámbito de la medicina.

La fisioterapia es una disciplina de la Salud que ofrece una alternativa terapéutica no farmacológica que, en muchos casos, ayuda a paliar los síntomas de múltiples dolencias, tanto agudas como crónicas. Se caracteriza por buscar el desarrollo adecuado de las funciones que producen los sistemas del cuerpo, donde su buen o mal funcionamiento repercute en la cinética o movimiento corporal humano. Interviene, mediante el empleo de técnicas científicamente demostradas, cuando el ser humano ha perdido o se encuentra en riesgo de perder, o alterar de forma temporal o permanente, el adecuado movimiento, y con ello las funciones físicas.

La fisioterapia y, en concreto, el fisioterapeuta, como agente de salud, trabajan para la prevención, curación y paliación de los problemas musculoesqueléticos y posturales [16]. El fisioterapeuta desarrolla su labor en cuatro grandes campos:

- Asistencial: su labor consiste en promover, prevenir, curar y paliar la salud de los pacientes aplicando el abanico de conocimientos adquiridos en su formación continua.



- Docente: su labor consiste en formar y promover el conocimiento de la Fisioterapia en las escuelas universitarias públicas y privadas, así como en las propuestas formativas convocadas para la formación continua del fisioterapeuta.
- Investigador: su labor consiste en buscar evidencia científica de los modos de proceder de la fisioterapia, ofreciendo al paciente, en consecuencia, aquellas actuaciones con mayores garantías de éxito, demostradas científicamente por estudios con validez de la comunidad científica.
- Gestión y Dirección: realizando su labor en la dirección de centros asistenciales, educativos o en colegios profesionales.

Por otra parte, los avances tecnológicos están muy presentes en nuestra sociedad. La tecnología avanza día a día a pasos agigantados y está presente en la vida de todos. Gracias a ella se pueden realizar grandes avances en el tema relacionados con cualquiera de los ámbitos de nuestra vida cotidiana. Los últimos años han estado marcados por un gran desarrollo entorno a los dispositivos móviles, tanto en temas de hardware como software, con la aparición de los sistemas operativos más comunes hasta la fecha como son iOS y Android. Gracias a estos avances en la tecnología tenemos más facilidades a la hora de realizar nuestras tareas más comunes. Podemos afirmar que la aparición de los dispositivos móviles es una de las innovaciones más importantes de estos últimos años [17]. Empezó ofreciendo servicios de transferencia de voz y de datos y ahora prácticamente ha evolucionado tanto, que llevar un móvil contigo es como llevar un ordenador portátil.

1.1 Objetivos

Con este proyecto se pretende desarrollar una aplicación móvil de fisioterapia con sistema operativo Android, en la cual se permita a los usuarios (tanto pacientes como profesionales) realizar una serie de acciones. La aplicación contará con parte informativa para el usuario en la cual podrá obtener información sobre los aspectos relacionados con los servicios que ofrece la aplicación, y también tendrá una parte interactiva en la cual los usuarios y los profesionales podrán intercambiar información entre sí. Así mismo los pacientes podrán reservar y organizar sus citas desde el dispositivo móvil y ver los ejercicios que el especialista le manda.



1.2 Fases y métodos

Para lograr los objetivos presentados en el punto anterior, el proyecto consta de las siguientes fases:

- Fase de documentación, en la que he recogido toda la información acerca de la programación a partir del sistema operativo Android y realización y manipulación de bases de datos.
- Fase de iniciación, en la que he ido adquiriendo los conocimientos necesarios de la programación y metodología de Android gracias a un curso de Android de la Universidad Politécnica de Valencia [18] en el que, a base de ejemplos y ejercicios propuestos, iba aprendiendo todo lo relacionado con la programación en Android que posteriormente tendría que utilizar para la realización del proyecto.
- Desarrollo de la aplicación con el programa Eclipse, desarrollando tanto las labores de programación interna, como las distintas interfaces de la aplicación. También se ha utilizado un servidor independiente gracias al programa XAMPP que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.
- Integración de la aplicación desarrollada en un dispositivo móvil con sistema operativo Android. En este caso es un Samsung Galaxy Trend GT-s7560.
- Realización de las pruebas oportunas para comprobar que la aplicación cumple todos los requisitos estipulados en las bases del proyecto.
- Extracción de conclusiones y establecimiento de futuras líneas de desarrollo.

1.3 Medios

Para la realización del proyecto se han utilizado varios programas dedicados a la realización de códigos de programación, creación de bases de datos y servidores.

Para la implementación y las pruebas a realizar, se dispone de los siguientes equipos y programas:

- Ordenador Sony Vaio Intel® Core(TM) i3 CPU m370 @ 2.40GHz y 4 GB de RAM. Sistema Operativo Microsoft Windows X7 Home



Premium, versión 2009, Service Pack 1. Empleado para realizar las pruebas.

- ECLIPSE Juno 2013 v4.2.2
- ANDROID SDK Manager Tools.
- XAMPP v 3.2.1
- phpMyAdmin 2.10.3
- MySQLWorkbench
- Sublime Text 2 2012, v2.0.2.
- Dispositivo móvil Samsung Galaxy Trend GT-s7560.

1.4 Estructura del documento

El documento está organizado de la siguiente manera:

- En el primer capítulo, se realiza una introducción sobre el contenido del proyecto y los objetivos, fases y medios que se han empleado para la realización del mismo.
- En el segundo capítulo, se estudian las tecnologías, empezando con una breve introducción a las tecnologías móviles y siguiendo con la información de los sistemas operativos más destacados. También se habla de las tecnologías del lado del servidor, las bases de datos, servicios web y se concluye con la comparación y la elección del sistema utilizado.
- En el tercer capítulo, se detalla la estructura y el desarrollo de Android que es el sistema operativo utilizado.
- En el cuarto capítulo, se realiza la descripción técnica, en la que se muestra la estructura de la base de datos, las relaciones entre tablas y la descripción de las funcionalidades de la aplicación.
- En el quinto capítulo, se describe el manual de uso de la aplicación para el usuario, explicando paso a paso todas las pantallas de la aplicación.
- En el capítulo sexto, se detalla el presupuesto económico de la aplicación.
- En el capítulo séptimo, se exponen las conclusiones del proyecto y las líneas futuras.
- En capítulo octavo, se recoge la bibliografía empleada para la realización del proyecto.
- Por último, se exponen una serie de anexos explicativos para facilitar la comprensión de algunas partes del proyecto.



CAPÍTULO 1

Capítulo 2: Tecnologías

Los sistemas de telefonía móvil civil empezaron a desarrollarse a partir de finales de los años 40 en los Estados Unidos. Eran sistemas de radio analógicos que utilizaban en el primer momento modulación en amplitud (AM) y posteriormente modulación en frecuencia (FM). Se popularizó el uso de sistemas FM gracias a su superior calidad de audio y resistencia a las interferencias. El servicio se daba en las bandas de HF y VHF.

2.1 Introducción a las generaciones móviles

En base a esto podemos diferenciar la evolución de los dispositivos móviles por generaciones [19]:

2.1.1 Generación 0

En los años de la Segunda Guerra Mundial, la compañía Motorola lanzó el HandieTalkie H12-16, el cual permitía comunicarse a distancia entre las tropas, era un dispositivo que se basaba en la transmisión mediante ondas de radio.

Esta tecnología se aprovechó entre los años 50 y 60 para crear diversos aparatos de radio y comunicación a distancia (Walkie-Talkies), que eran utilizados en mayor parte por taxis, ambulancias o bomberos.



Figura 3. HandieTalkie H12-16

Estos dispositivos no se pueden considerar como teléfonos móviles pero su implementación supuso el comienzo de la evolución de los dispositivos que conocemos actualmente.



CAPÍTULO 2

Los primeros estándares más utilizados de esta generación fueron:

- Estándar PTT (PushToTalk): Pulsa para hablar
- Estándar IMTS (Improved Mobile TelephoneSystem): Sistema de telefonía móvil mejorado

2.1.2 Primera Generación (1G)

A partir de 1973 surgieron los móviles de primera generación los cuales eran de gran tamaño y peso. Funcionaban de manera analógica, lo que conllevaba que la transmisión y recepción de datos se apoyaba sobre un conjunto de ondas de radio que cambiaban de modo continuo.



Figura 4. Teléfono de primera generación (1G)

La desventaja de que fueran analógicos era que solo podían ser usados para la transmisión de voz, tenían muy baja seguridad y esto implicaba que una persona pudiera escuchar llamadas ajenas con un sintonizador de radio o incluso hacer uso de las frecuencias cargando el importe de las llamadas a otras personas.

Los estándares más utilizados en esta generación fueron:

- NMT: Nordic Mobile Telephone
- AMPS: Advanced Mobile PhoneSystem

2.1.3 Segunda Generación (2G)



Esta generación marca el paso de la telefonía analógica a la digital alrededor de 1990 lo cual mejoró el manejo de llamadas, se pudieron hacer más enlaces al mismo tiempo en el mismo ancho de banda e integrar otros servicios adicionales aparte de la voz, como el servicio de mensajes cortos (Short MessageService , SMS).



Figura 5. Teléfono de segunda generación (2G)

Los estándares más utilizados en esta generación fueron:

- GSM: Global Systemfor Mobile Communications - Sistema Global para Comunicaciones Móviles
- CDMA: Code Division Multiple Acces - Acceso Múltiple por División de Código
- GPRS: General Packet Radio Service - Servicio General de Radio por Paquetes

2.1.4 Tercera Generación (3G)

A partir de 2011 los dispositivos móviles sufrieron grandes cambios debido a la aparición principalmente de las pantallas LCD a color.

También nacieron dispositivos que se creían como mínimo futuristas como móviles con cámara fotográfica digital, grabación de vídeos que se podían enviar por mensajería instantánea, juegos en 3D, sonido mp3, conversaciones por videoconferencia gracias a una tasa de transferencia de datos más aceptable y a un soporte para internet correctamente implementado (correo electrónico, descargas, etc.).



CAPÍTULO 2



Figura 6. Teléfono de tercera generación (3G)

Los estándares más utilizados en esta generación fueron:

- UMTS: Universal Mobile Telecommunications System - Servicios Universales de Comunicaciones Móviles.

2.1.5 Cuarta Generación (4G)

En el año 2010 se lanzaron los primeros servicios 4G basados en la tecnología LTE en Tokyo, Nagoya y Osaka, la red 4G está basada en el protocolo IP. Esta tecnología puede ser utilizada por módems inalámbricos, celulares inteligentes y otros dispositivos móviles. La principal característica de esta red de esta generación es que tiene la capacidad de proveer velocidades de acceso mayores a los 100 Mbps en movimiento y 1 Gbps en reposo manteniendo una calidad de servicio (QoS) de punta a punta de alta seguridad que permitirá ofrecer servicios de cualquier clase en cualquier momento, en cualquier lugar.



Figura 7. Teléfono de cuarta generación (4G)



2.2 Sistemas Operativos

Una vez introducida una breve historia sobre la evolución de los dispositivos móviles, pasamos a mencionar los principales sistemas operativos que están actualmente en uso.

2.2.1 iOS

Este apartado comenzará con una pequeña introducción a la historia de iOS, así como sus avances [20] [21] y actualizaciones desde la primera versión en 2007:



Figura 8. Evolución IOS

2.2.1.1 Evolución

iPhone OS 1 (2007)

Este iPhone fue portada de la revista TIME como el invento del año bajo calificativos tan difíciles de discutir como “*El teléfono que ha cambiado los teléfonos para siempre*”. El iPhone original cayó como una bomba en 2007 con innovaciones como la pantalla táctil, la integración del navegador Safari, YouTube y Google Maps.



CAPÍTULO 2

iPhone OS 2 (2008)

Esta nueva versión introdujo la iTunes App Store, en la que los usuarios pueden descargar aplicaciones gratuitas o no. Se podía almacenar los datos de las tarjetas de crédito, bastante útil a la hora de pagar las apps y permitía la visualización de correos, contacto y calendario.

iPhone OS 3 (2009)

Con más de 100 nuevas características el gran avance en esta versión del sistema operativo fue la función cortar, copiar y pegar. También, se introdujo la búsqueda en Spotlight. Igualmente, se incluyeron las Notificaciones, la Brújula digital, Notas de voz y Buscar mi iPhone. Otras características que se incluyeron fueron la grabación de vídeos y las opciones para editarlos y compartirlos

iPhone OS 4 (2010)

Una de las funcionalidades que Apple presentó en esta versión es la multitarea, es decir, que el usuario pueda tener varias aplicaciones abiertas al mismo tiempo en su dispositivo sin que éste le genere problemas. Otras de las características que se incluyeron fueron el autocorrector y las carpetas en las que el usuario puede agrupar las aplicaciones en distintas categorías. Además, soportó la primera pantalla Retina. En esta versión de iOS también se introdujo el FaceTime que permite realizar llamadas de vídeo entre los usuarios.

iPhone OS 5 (2011)

Fue el último año en el que el creador, Steve Jobs, presenció la presentación antes de su muerte. En esta versión, Apple reemplazó al control por voz con Siri, un asistente virtual que ayuda al usuario a realizar tareas más rápidamente. También se incluyó –herencia de Android– el Centro de Notificaciones, y la idea del almacenamiento en la nube conocido como iCloud.

iPhone OS 6 (2012)

Google Maps fue retirado de los dispositivos y Apple creó su propia aplicación llamada Mapas, que se estrenó en esta versión. También fue incluido el Passbook.

A partir de esta versión los dispositivos también se podían integrar con Facebook. Por esto Siri permitía que los usuarios actualizaran sus estados en Facebook y Twitter. También, se empezó a poder compartir fotos vía streaming a través de iCloud.



iPhone OS 7 (2013)

El cambio más visible en esta versión de iOS fue el cambio del diseño del sistema operativo. Apple introdujo una interfaz renovada, mucho más sencilla y basada en un diseño plano y minimalista. Pero esa no fue la única novedad: con iOS 7 también llegó el Centro de Control, al que se puede acceder cuando el usuario realiza un gesto en el que desliza su dedo de abajo hacia arriba.

En iOS 7, las aplicaciones se actualizan automáticamente. Otra característica es el Touch ID, que puede detectar las huellas digitales, ya que cuenta con un sensor biométrico que debutó con el iPhone 5s.

iPhone OS 8 (2014)

Con iOS 8 se ha conseguido una mejor sincronización entre todos los dispositivos de Apple. Este nuevo sistema operativo cuenta con la funcionalidad ‘Handoff’ con la que los usuarios pueden pasar información de sus dispositivos móviles a sus ordenadores con AirDrop a través de una red inalámbrica.

Además, en los ordenadores los usuarios pueden recibir y enviar mensajes de texto, así como mensajes enviados desde iMessages y realizar llamadas telefónicas.

2.2.1.2 Ventajas y desventajas

Una vez introducida una breve historia de la evolución de iOS, se verán las principales ventajas y desventajas de este sistema operativo para luego poder comparar con los demás y justificar la opción elegida.

Analizando las características detalladas por varios expertos en este sistema operativo en Internet, se puede concluir que los pros y contras de iOS en función de la información almacenada de dichos sitios son los siguientes [22] [23]:



CAPÍTULO 2

Ventajas:

- Existe una gran comunidad de desarrolladores en torno a iOS, y hay mucha información disponible en la red para programadores noveles.
- Hay muchas APIs no oficiales bien implementadas que pueden ser reutilizadas en nuestro código.
- El hecho de que Apple restrinja y supervise las aplicaciones subidas al AppStore asegura un mínimo de calidad y de seguridad.
- El sistema operativo está diseñado para una arquitectura particular, la del pequeño número de dispositivos de Apple. El resultado es que la optimización de código es mejor que en Android.
- Buen sistema de sincronización entre equipos y capacidad de contestar mensajes desde cualquier equipo sin causar duplicados.
- Grandes defensas contra el malware, ya que iOS tiene un proceso de aprobación y selección de las apps que son revisadas antes de que se publiquen para el uso de los usuarios, con lo cual las aplicaciones que se encuentran en la App Store son 100% seguras.
- Una de las cualidades de iOS es que tiene un ecosistema bien planteado ya que independientemente de la plataforma Apple que utilices, la apariencia es la misma lo cual ayuda al manejo del terminal y es más fácil para la gente que no es muy hábil con los dispositivos.
- El asistente Siri es una herramienta muy útil ya que puedes realizar acciones solo con tu voz, e incluso puedes fijar alarmas y eventos sin necesidad de acceder a ningún menú apropiado para ello.

Desventajas

- Es bastante robusto en cuanto a temas de personalización visual, a diferencia de lo que se puede hacer con Android.
- No está incluido el flash en el navegador.
- Limitado en cuanto a la transferencia de archivos.
- Objective-C: el lenguaje de programación de iOS no está tan extendido como el empleado por Android, esto es, Java.
- Apple no permite modificar la API de cualquier componente de su framework, lo que resta libertad y capacidad de innovación al desarrollador, con lo cual es un sistema operativo cerrado.
- Para el desarrollo de aplicaciones y que estas puedan ser subidas a la App Store es necesario un Mac.



- Los desarrolladores necesitan abonar actualmente una cuota de 99\$ anuales para contar con las herramientas de desarrollo y poder subir aplicaciones a la tienda.
- La gestión multitarea. iOS no lleva a cabo una multitarea real. En el momento en que una aplicación deja de ser utilizada y pasa a segundo plano esta se congela y no recibe un solo ciclo de reloj para refrescar su estado.

2.2.2 Android

En octubre de 2003, hace ya 11 años, Android Inc. era fundada por Andy Rubin, Rich Miner, Nick Sears y Chris White. En 2005 Google adquiere la empresa para seguir trabajando en el mismo proyecto que después conociera la luz como un S.O. para móviles denominado finalmente como Android.

Dos años después, el 5 de noviembre de 2007, era lanzaba la primera versión de Android, aún ni siquiera catalogada como beta. A partir de entonces, Android se ha conseguido alzar como sistema operativo móvil más extendido a nivel global.



Figura 9. Versiones de Android.

Fue el 23 de septiembre de 2008 cuando se lanzó la primera versión estable de Android. Android tiene una característica peculiar: las versiones tienen nombre de postres en inglés y cada versión que cambia, continúa de forma incremental en el alfabeto, es decir que si el primer nombre inicio con A, el siguiente con B, el siguiente C y así sucesivamente.



2.2.2.1 Evolución

CUPCAKE (versión 1.5)

El 30 de abril de 2009, Android comenzó su curiosa denominación con postres en orden alfabético. La versión 1.5 llegó bajo el apellido Cupcake y trajo consigo interesantes novedades como la actualización de la tienda de aplicaciones Android Market Place, una interfaz más simple y organizada para el sistema, así como un renovado teclado táctil.

DONUTS (versión 1.6)

Posteriormente, el día 15 de septiembre del mismo año se liberó Android 1.6 Donut, que supuso la llegada de una mejora fundamental en la experiencia de uso de la introducción táctil. Y es que, al mismo tiempo que se mejoraba el control de la introducción táctil, esta versión de Android se actualizaba con un nuevo teclado táctil en pantalla, más sensible y preciso. Por otra parte, se optimizó la conversión de texto a voz -accesibilidad- y se introdujo el gestor de consumo energético casi como lo conocemos actualmente.

ECLAIR (versión 2.0)

Por último, el 26 de octubre de 2009 se lanzaba una tercera actualización del joven sistema operativo móvil bajo el nombre Android 2.0 Eclair. Con esta actualización, los usuarios vieron llegar un menú rápido de contactos, software para el control de la cámara, nuevas mejoras en el teclado, soporte para el estándar HTML5 en el navegador y una optimización de los gráficos.

Durante el segundo trimestre de este año 2009, Android alcanzaba una aún ridícula cuota de mercado correspondiente al 2,8% a nivel global.

FROYO (versión 2.2) Y GINGERBREAD (versión 2.3)

De nuevo, el 20 de mayo de 2010 Android se actualiza con la versión 2.2 Froyo, que supondría la llegada de soporte para múltiples idiomas en el teclado, la marcación de llamadas por comandos de voz a través de Bluetooth, el uso del smartphone como router WiFi, mejoras en la cámara y la galería y el soporte para memorias RAM de mayor capacidad.

También durante este año, pero el día 6 de diciembre, se vuelve a actualizar el sistema operativo móvil a la versión 2.3 Gingerbread, introduciendo así una importante renovación del aspecto a nivel interfaz y teclado, mejorando la



gestión del consumo energético, introduciendo las llamadas de voz sobre el protocolo IP -VoIP- y, además, añadiendo soporte para la conectividad NFC.

Y en 2010, durante el cuarto trimestre del año, Android ya había alcanzado una cuota de mercado del 33% a nivel global, el crecimiento experimentado en poco más de un año fue realmente sorprendente.

HONEYCOMB (versión3.0) y ICE CREAM SANDWICH (versión4.0)

Manteniendo el ritmo de actualizaciones del año anterior, Android se actualiza en dos ocasiones. La primera versión de 2011 fue Android 3.0 Honeycomb , que fue además la primera en estar diseñada exclusivamente para tabletas. Con ella llegó el nuevo sistema de notificaciones, el teclado con tabulador, mejoras en la selección de texto, widgets en la pantalla de inicio y una mejor navegación web. Además, fue la primera en introducir soporte para procesadores multicore.

El 19 de octubre, sin embargo, llegó Android 4.0 Ice Cream Sandwich, que unió tabletas y teléfonos inteligentes bajo un mismo software. Las novedades de esta versión fueron orientadas a la redimensión de widgets, la personalización de la pantalla de bloqueo y la introducción de accesos directos a aplicaciones y contactos. Ahora bien, también se introdujo la respuesta rápida de llamadas, un mejorado teclado y Android Beam NFC.

En mayo de 2011, Android estaba ya presente en más de la mitad de teléfonos inteligentes, con una cuota de mercado a nivel global del 53%.

JELLY BEAN(versión 4.1)

Entre mayo y junio de 2012, Google lanzó Android 4.1 JellyBean. Con esta versión, se optimizó notablemente el rendimiento de los dispositivos móviles y se introdujo “Project Butter”, con la intención de dar mayor fluidez a los terminales con Android. Además, Google introdujo su asistente virtual Google Now, en el que ahora centran gran parte de sus esfuerzos. Se mejoró el sistema de notificaciones y se simplificó el sistema de multitarea. Esta fue, sin duda, una de las versiones de Android que mayor evolución han dado al ecosistema.

A estas alturas, Android ya alcanzaba el 75% de cuota de mercado a nivel global.



CAPÍTULO 2

KITKAT(versión 4.4)

Por primera vez, Google se alía con otra compañía para el nombre de su nueva versión. Así tenía comienzo la alianza entre Nestlé y Google para poner “KitKat” como nombre a esta versión, lo que vino asociado de una llamativa campaña comercial con el postre correspondiente. Esta versión, que llegó el 31 de octubre, introdujo la impresión de forma remota, los ajustes rápidos para la modificación de las pantallas de inicio, el modo inmersivo para aplicaciones y la vista previa de carátulas de películas y álbumes de música en la pantalla de bloqueo, así como la botonera de reproducción. Además, también llegaron cambios importantes al gestor de correo electrónico y la aplicación de descargas.

El año 2013, Android ya alcanzaba el 78,6% de cuota de mercado a nivel global. Evidentemente, el crecimiento ha continuado, aunque mucho más lento que en años anteriores.

LOLLIPOP (versión 5.0)

Tras haber podido conocer una “vista previa” el pasado mes de junio, Android 5.0 Lollipop ha llegado ya de forma oficial. Con este último lanzamiento, y este último año de Android, el ecosistema ha crecido para llegar a los dispositivos wearables con Android Wear, a la consola integrada de los coches con Android Auto y a nuestros televisores con Android TV.

Mientras la cuota de mercado de Android continúa creciendo imparable con un último valor del 81.4% [24], sus competidores ya toman referencias clave de este sistema operativo. Y es que, aunque Apple llegó antes, Android ha tomado la delantera. Así, mientras que llevamos ya varios años con teléfonos inteligentes de más de 4 pulgadas de diagonal para su pantalla, ha sido este año cuando Apple ha decidido sumarse a la inevitable tendencia del mercado.

2.2.2.2 Ventajas y desventajas

Ventajas

Las características principales de Android que hace ventajoso este sistema a otros son:

- Personalizable: Sin duda esta es una de las grandes ventajas de Android, poder personalizar el escritorio con widgets (por ejemplo:



de tiempo, de hora, de rss...) e instalar decenas de alternativas presentes en la Google Play [25].

- Asequible: El triunfo de Android podría atribuirse en gran parte a esta característica, Android está presente en los teléfonos de última generación (gama alta) así como los más cómodos para el bolsillo (gama media o gama baja).
- El código de Android es abierto: Google liberó Android bajo licencia apache. Gracias a esto cualquier persona puede realizar una aplicación para Android. Cuando apareció Android ya había un consorcio de 78 importantes empresas especializadas en diseño de software para teléfonos móviles listas para diseñar aplicaciones [26].
- Comunidad: al ser Android un sistema operativo de código abierto permite que personas externas a una empresa desarrollen ROM's customizadas que mejoran la velocidad, entre otras características, del terminal.
- Gran cantidad de aplicaciones: A día de hoy hay más de 100.000 aplicaciones disponibles para teléfonos Android, gran parte de ellas gratuitas. Además la libertad de código permite adaptar Android a bastantes dispositivos además de teléfonos móviles (tablets, gps, relojes, microondas...).
- Varias aplicaciones al mismo tiempo : El sistema Android es capaz de hacer funcionar a la vez varias aplicaciones y además se encarga de gestionarlas, dejarlas en modo suspensión si no se utilizan e incluso cerrarlas si llevan un periodo determinado de inactividad. De esta manera se evita un consumo excesivo de batería. Esta es una de sus mayores ventajas por la rapidez con la que carga una aplicación abierta previamente
- [27].
- Lenguaje Java: El lenguaje de programación que exige este sistema operativo está basado en JAVA. Uno de los lenguajes de programación más usados y más implantados en el mundo.

Desventajas

Como desventajas se puede reseñar las siguientes:

Vulnerable: El hecho de que sea de código abierto trae esta gran desventaja

- [28].



CAPÍTULO 2

- Poco intuitivo: Las configuraciones del dispositivo móvil mediante Android no son tan sencillas de realizar. Tiene funcionalidades que pueden resultar difícil de configurar si no se sabe mucho de tecnología.

Fragmentado: Android está totalmente fragmentado: tiene bastantes versiones dentro de las versiones oficiales. Cada modelo de teléfono móvil se ha de adaptar a Android de manera que no es exactamente la misma versión de Android la que lleva una HTC One que un Samsung Galaxy 5. Esto provoca problemas de incompatibilidad con algunas aplicaciones de la market que funcionan en determinadas versiones de Android

- [29].
- Consumo de la batería: El permitir muchas aplicaciones abiertas hace que el consumo de la batería se dispare y que dure mucho menos de lo previsto. Esto a la larga, hace que el usuario tenga que tener invariablemente su cargador a mano
- [30].

2.3 Otros sistemas operativos

También se encuentran en el mercado otros sistemas operativos menos influyentes que iOS y Android ya que estos dos son los más importantes a nivel de mercado.

2.3.1 Windows Phone

Windows Phone (abreviado WP) es un sistema operativo móvil desarrollado por Microsoft, como sucesor de Windows Mobile.

A diferencia de su predecesor está enfocado en el mercado de consumo en lugar de en el mercado empresarial. Con Windows Phone; Microsoft ofrece una nueva interfaz de usuario que integra varios de sus servicios propios como OneDrive, Skype y Xbox Live en el sistema operativo. Compite directamente contra Android de Google e iOS de Apple. Su última versión disponible y definitiva es Windows Phone 8.1, lanzado el 14 de abril de 2014.

Investigando las características de Windows Phone, se pueden extraer las principales ventajas y desventajas de este sistema operativo

[31]

[32]:

Ventajas:



- Microsoft ha diseñado un sistema operativo con la estabilidad, fluidez y seguridad como principales rasgos de identidad independientemente de la categoría del dispositivo.
- Gracias a la multitud de widgets las posibilidades de personalización de su pantalla de bloqueo y su pantalla de inicio son numerosas.
- Colabora con Nokia para temas de plataforma y desarrollo de software y hardware.
- Posee una gran gama de terminales que se adaptan a todo tipo de precios, esto favorece a que cualquier usuario puede acceder a ellos.
- Incluye Office Mobile para acceder y editar documentos de Microsoft Word, Excel y PowerPoint. Dispone de integración completa con los servicios de Microsoft como SkyDrive, Bing, Xbox o Skype y la Windows PhoneStore.
- Los programadores de aplicaciones pueden portar su software de Windows 8 a Windows Phone 8 dado que ambos están basados en el Kernel NT.
- Soporta los lenguajes de programación en C y Visual Basic.

Desventajas:

- Es un sistema operativo joven.
- No habrá un centro de notificaciones completo hasta la actualización a Windows Phone 8.1. Actualmente en la pantalla de bloqueo sólo se muestran notificaciones de 5 aplicaciones (a elegir) y para el resto hay que ver el *tile* en la pantalla de inicio.
- Cada vez son menos las aplicaciones destacadas en otros sistemas operativos de importancia con iOS y Android que están disponibles en el market de Windows Phone.
- Es un sistema cerrado que busca estabilidad y fiabilidad por lo que no deja prácticamente margen para personalizar su *software*.
- El navegador del que dispone es Internet Explorer, prácticamente en desuso para todos los usuarios de cualquier dispositivo.
- Temas como la sincronización multimedia o la multitarea están muy atrasados en comparación con iOS y Android.
- El código de Windows Phone, denominado código transparente, no puede realizar llamadas a bibliotecas, tipos y métodos críticos para la seguridad
- [33].



2.3.2 Ubuntu OS

Ubuntu Mobile fue desarrollado para dispositivos MID (Mobile Internet Device), es decir para dispositivos que cuenten con la capacidad de conectarse a internet; en este caso no es necesario que el dispositivo tenga una pantalla táctil.

Este sistema operativo ha sido desarrollado por Canonical, una empresa del Reino Unido dedicada al desarrollo de software de computadoras; una de sus creaciones más populares es Ubuntu, otro software basado en Open Source de código abierto.

Ubuntu Mobile fue presentado el 4 de marzo de 2004, entonces ejecutado mediante una microarquitectura de CPU. En cuanto a la interfaz de usuario, este OS es ejecutado por GNOME, un gestor de escritorio –el mismo con el que se ha creado toda la interfaz de Linux- que el usuario final puede personalizar bastante en términos de apariencia y algunos detalles del diseño.

Ubuntu OS es todavía un proyecto de futuro que usará aplicaciones nativas, lo que significa que los desarrolladores pueden crear una sola aplicación, tanto para escritorio como gadgets móviles, que permitan a los usuarios moverse fácilmente entre dispositivos.

Entre sus características más destacadas encontramos:

- Larga vida de la batería.
- Disponibilidad de numerosas aplicaciones, esto es posible gracias al enorme empuje que está teniendo actualmente las llamadas aplicaciones Webs hechas con HTML5.

2.3.3 Tizen

Tizen es un sistema operativo móvil creado con base en Linux y Linux Foundation. Muy en sus inicios fue la combinación de Noblin, creado por Intel, y de Maemo, creado por Nokia. Este equipo se popularizó bajo el nombre de Meego y pretendían competir con Android.

El producto de este proyecto llegó materializarse con el celular Nokia N9, presentado en 2011 y tenían planes para que fuera implementado en algunas



notebooks, diversos dispositivos portátiles, televisores e incluso vehículos. Con el tiempo Nokia cedió la estafeta a Samsung, quien junto con Intel y algunos ex integrantes de Meego, decidieron seguir adelante con este proyecto, al cual llamaron Tizen.

En cuanto a desarrollo, Tizen está basado en HTML5 y otros estándares web. Entre sus características cuenta con:

- Su desarrollo es completamente abierto, al igual que el de Android.
- El SDK –el software para su desarrollo- está disponible para todo el público desde 2012, así como algunos smartphones en el mercado.
- La más reciente actualización (estable) de sistema fue lanzada en julio de 2013.
- La ventaja de Tizen es que también puede personalizarse hasta cierto punto, así como Android, y esto es gracias a que sigue la idea de "open source" de Linux.
- Combina la versatilidad de las aplicaciones HTML5 con las de herramientas nativas basadas en un SDK

2.3.4 Firefox OS

Este proyecto nace con el apoyo de Mozilla Corp, Telefónica (Movistar) y el trabajo voluntario de diversos ingenieros en sistemas y desarrollo. FireFox OS está basado en Linux y está disponible actualmente para algunos dispositivos móviles como smartphones y tablets (primer teléfono con este sistema operativo: ZTE Open “2012”).

Este sistema operativo está hecho para que opere en equipos de gama baja. Entre sus características encontramos:

- Está pensado para soportar aplicaciones desarrolladas en HTML5 para que puedan comunicarse de manera directa con el hardware del dispositivo haciendo uso de JavaScript.
- Al hacer las aplicaciones bajo este lenguaje, los navegadores nativos interpretan el código y acomodan el diseño de cada aplicación para que se vea bien en cualquier tamaño o resolución de pantalla.



CAPÍTULO 2

- El entorno gráfico es sencillo de utilizar y rápido en la ejecución, debido a que este sistema operativo no necesita grandes requerimientos de hardware.
- El núcleo de este sistema operativo está basado en el código abierto del sistema operativo Linux, como sucede con Android.

Los teléfonos inteligentes que se encuentran a la venta corriendo este sistema operativo son:

- ZTE Open
- Alcatel One Touch Fire

Son equipos de gama baja y son muy económicos, por lo tanto su venta está orientada a un público muy específico.

2.4 Tecnologías de programación del lado del servidor

2.4.1 PHP

PHP

[34] es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente

[35].

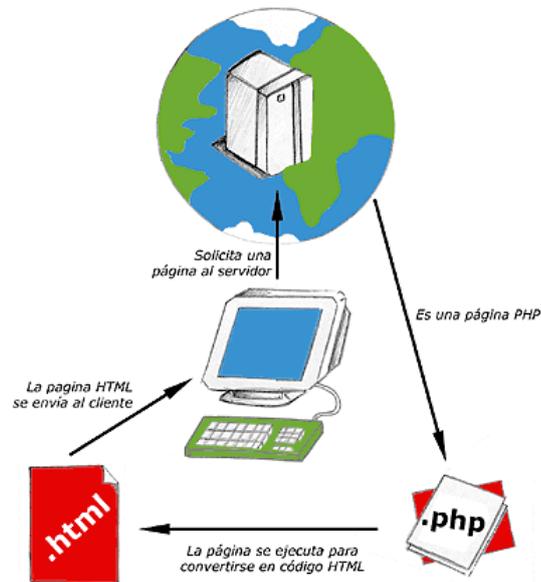


Figura 10. Esquema del funcionamiento de las paginas PHP.

PHP tiene una serie de características más importantes a tener en cuenta, que mencionaremos a continuación:

- Gratuito. Cualquiera puede descargar a través de la página principal de PHP www.php.net y de manera gratuita, un módulo que hace que nuestro servidor web comprenda los scripts realizados en este lenguaje.
- Independiente de plataforma. Puesto que existe un módulo de PHP para casi cualquier servidor web. Esto hace que cualquier sistema pueda ser compatible con el lenguaje y significa una ventaja importante, ya que permite portar el sitio desarrollado en PHP de un sistema a otro sin prácticamente ningún trabajo.
- De código abierto. PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparan rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP
- [36].
- Bibliotecas incorporadas. Como se ha diseñado para su uso en la Web, PHP incorpora una gran cantidad de funciones integradas para realizar útiles tareas relacionadas con la Web. Puede generar imágenes GIF al instante, establecer conexiones a otros servicios de red, enviar correos electrónicos, trabajar con cookies y generar documentos PDF, todo con unas pocas líneas de código.



CAPÍTULO 2

Rapidez: PHP, en el caso de estar montado sobre un servidor Linux u Unix, es más rápido que ASP, dado que se ejecuta en un único espacio de memoria y esto evita las comunicaciones entre componentes COM que se realizan entre todas las tecnologías implicadas en una página ASP

- [37].
- Seguridad: el hecho de que en muchas ocasiones PHP se encuentra instalado sobre servidores Unix o Linux, que son de sobra conocidos como más veloces y seguros que el sistema operativo donde se ejecuta las ASP, Windows NT o 2000. Además, PHP permite configurar el servidor de modo que se permita o rechacen diferentes usos, lo que puede hacer al lenguaje más o menos seguro dependiendo de las necesidades de cada cual.
- Interfaces para una gran cantidad Integración de base de datos. PHP dispone de una conexión propia a todos los sistemas de base de datos. Puede conectarse directamente a las bases de datos de MySQL, PostgreSQL, mSQL, Oracle, dbm, filePro, Hyperwave, Informix, Internase y Sybase, entre otras. Esto se debe a que PHP utiliza ODBC (Open Database Connectivity Standard).
- Sencillez en el aprendizaje: PHP es relativamente sencillo de aprender, además se pueden encontrar infinidad de manuales en internet que permiten un aprendizaje rápido de este lenguaje de programación.

2.5 Bases de datos

2.5.1 Bases de datos relacionales

El modelo de datos relacional organiza y representa los datos en forma de tablas o relaciones. Relación es un término que viene de la matemática y representa una simple tabla de dos dimensiones, consistente en filas y columnas de datos.

Tiene los componentes siguientes:

- Estructura de datos: Es una colección de objetos abstractos formados por datos. Dominios, tuplas, atributos y relaciones.
- Operadores: Conjunto de operadores, con reglas bien definidas, que permiten manipular las estructuras de datos. Además del cambio de esquema, los primitivos del álgebra relacional para manipulación de datos, es decir, unión, diferencia, producto cartesiano, proyección y selección.



- Definiciones de integridad: Colección de conceptos y reglas que permite expresar qué valores de datos pueden aparecer válidamente en nuestro esquema. Las claves y la posibilidad de tener valores nulos [38]. También se incluyen aquí dos reglas de integridad, llamadas:
 - Integridad de claves primarias.
 - Integridad referencial.

En el modelo de datos relacional la forma en que se almacenan los datos no importa, por lo que es más fácil para un usuario entender y utilizar la BD. La información puede ser recuperada o almacenada mediante consultas que ofrecen una amplia flexibilidad y poder para administrar la información. Además, durante el diseño de una BD relacional se realiza un proceso de normalización, en el que cada relación se describe en términos de dependencia. Este proceso se realiza para evitar la redundancia de los datos, evitando así problemas de actualización de los datos, y para proteger la integridad de los datos [39].

2.5.2 Tipos

SQL

El SQL es el lenguaje estándar ANSI/ISO de definición, manipulación y control de bases de datos relacionales

[40]. Es un lenguaje declarativo: sólo hay que indicar qué se quiere hacer. En cambio, en los lenguajes procedimentales es necesario especificar cómo hay que hacer cualquier acción sobre la base de datos. El SQL es un lenguaje muy parecido al lenguaje natural; concretamente, se parece al inglés, y es muy expresivo. Por estas razones, y como lenguaje estándar, el SQL es un lenguaje con el que se puede acceder a todos los sistemas relacionales comerciales

[41].

El lenguaje SQL tiene varios aspectos diferentes:

- Lenguaje de manipulación de datos (LMD). Este subconjunto de SQL permite a los usuarios formular consultas e insertar, eliminar y modificar filas.



CAPÍTULO 2

- Lenguaje de definición de datos (LDD). Este subconjunto de SQL soporta la creación, eliminación y modificación de definiciones de tablas y vistas. Se pueden definir restricciones de integridad para las tablas, bien en el momento de crearlas, bien posteriormente.
- Disparadores y restricciones de integridad avanzadas. SQL: 1999 incluye soporte para los disparadores, que son acciones ejecutadas por el SGBD siempre que las modificaciones de la base de datos cumplen las condiciones especificadas en el disparador.
- SQL incorporado y SQL dinámico. Las características de SQL incorporado permiten llamar al código SQL desde lenguajes anfitriones como C o COBOL. Las características de SQL dinámico permiten que se creen (y ejecuten) consultas en el momento de la ejecución.
- Ejecución cliente-servidor y acceso a bases de datos remotas. Estas órdenes controlan el modo en que los programas de aplicación clientes pueden conectarse con los servidores de bases de datos de SQL o tener acceso a los datos de las bases de datos a través de la red.
- Gestión de transacciones. Diversas órdenes permiten que los usuarios controlen de manera explícita aspectos del modo en que se deben ejecutar las transacciones.
- Seguridad. SQL ofrece mecanismos para control de acceso de los usuarios a los objetos de datos, como tablas y vistas.

MySQL

MySQL es un sistema de bases de datos que se puede usar tanto en la web como en el servidor, fácil de usar, y sirve tanto para pequeñas aplicaciones como para aplicaciones más grandes y potentes.

Es un sistema de bases de datos multiplataforma, y además es gratuito y distribuido oficialmente por Oracle.

La estructuración de las diferentes bases de datos en MySQL es muy sencilla, la información se almacena mediante tablas, donde cada uno de los campos que contiene la tabla se estructura en forma de columnas, y cada elemento que se introduzca en la base de datos en forma de filas.

Además, determinadas sentencias de MySQL pueden ser embebidas en código PHP y HTML para diseñar aplicaciones Web dinámicas que incorporan la información de las tablas de MySQL a páginas Web

[42]



[43].

Por esta razón MySQL es uno de los sistemas de bases de datos más conocidos y utilizados en la actualidad, webs como Facebook, o Google utilizan este sistema de bases de datos

[44]

[45].

Características MySQL:

- Velocidad. MySQL es rápido.
- Facilidad de uso. Es un sistema de base de datos de alto rendimiento pero relativamente simple y es mucho menos complejo de configurar y administrar que sistemas más grandes.
- Coste. Es gratuito.
- Capacidad de gestión de lenguajes de consulta. MySQL comprende SQL, el lenguaje elegido para todos los sistemas de bases de datos modernos.
- Capacidad. Pueden conectarse muchos clientes simultáneamente al servidor. Los clientes pueden utilizar varias bases de datos simultáneamente. Además, está disponible una amplia variedad de interfaces de programación para lenguajes como C, Perl, Java, PHP y Python.
- Conectividad y seguridad. MySQL está completamente preparado para el trabajo en red y las bases de datos pueden ser accedidas desde cualquier lugar de Internet. Dispone de control de acceso.
- Portabilidad. MySQL se puede utilizar en una gran cantidad de sistemas Unix diferentes así como bajo Microsoft Windows.
- Distribución abierta. Puede obtener y modificar el código fuente de MySQL.

2.6 Servicios Web

2.6.1 REST

REST define un set de principios arquitectónicos por los cuales se diseñan servicios web haciendo foco en los recursos del sistema, incluyendo cómo se accede al estado de dichos recursos y cómo se transfieren por HTTP hacia clientes escritos en diversos lenguajes. REST emergió en los últimos años como el modelo predominante para el diseño de servicios. De hecho, REST



CAPÍTULO 2

logró un impacto tan grande en la web que prácticamente logró desplazar a SOAP y las interfaces basadas en WSDL por tener un estilo bastante más simple de usar.

Una implementación concreta de un servicio web REST sigue cuatro principios de diseño fundamentales.

REST utiliza los métodos HTTP de manera explícita

Una de las características claves de los servicios web REST es el uso explícito de los métodos HTTP, siguiendo el protocolo definido por RFC 2616. Por ejemplo, HTTP GET se define como un método productor de datos, cuyo uso está pensado para que las aplicaciones cliente obtengan recursos, busquen datos de un servidor web, o ejecuten una consulta esperando que el servidor web la realice y devuelva un conjunto de recursos.

REST hace que los desarrolladores usen los métodos HTTP explícitamente de manera que resulte consistente con la definición del protocolo. Este principio de diseño básico establece una asociación uno-a-uno entre las operaciones de crear, leer, actualizar y borrar y los métodos HTTP. De acuerdo a esta asociación:

- POST para crear un recurso en el servidor
- GET para obtener un recurso
- PUT para cambiar el estado de un recurso o actualizarlo
- DELETE para eliminar un recurso

REST no mantiene estado

Los servicios web REST necesitan escalar para poder satisfacer una demanda en constante crecimiento. Se usan clusters de servidores con balanceadores de carga y alta disponibilidad, proxies y gateways, que permita transferir peticiones de un equipo a otro para disminuir el tiempo total de respuesta de una invocación al servicio web. El uso de servidores intermedios para mejorar la escalabilidad hace necesario que los clientes de servicios web REST envíen peticiones completas e independientes; es decir, se deben enviar peticiones que incluyan todos los datos necesarios para cumplir el pedido, de manera que los componentes en los servidores intermedios puedan redireccionar y gestionar la carga sin mantener el estado localmente entre las peticiones.

Una petición completa e independiente hace que el servidor no tenga que recuperar ninguna información de contexto o estado al procesar la petición. Una



aplicación o cliente de servicio web REST debe incluir dentro del encabezado y del cuerpo HTTP de la petición todos los parámetros, contexto y datos que necesita el servidor para generar la respuesta. De esta manera, el no mantener estado mejora el rendimiento de los servicios web y simplifica el diseño e implementación de los componentes del servidor, ya que la ausencia de estado en el servidor elimina la necesidad de sincronizar los datos de la sesión con una aplicación externa.

- Servicios con estado vs. sin estado

La siguiente ilustración nos muestra un servicio con estado, del cual una aplicación realiza peticiones para la página siguiente en un conjunto de resultados multi-página, asumiendo que el servicio mantiene información sobre la última página que pidió el cliente. En un diseño con estado, el servicio incrementa y almacena en algún lugar una variable pagina

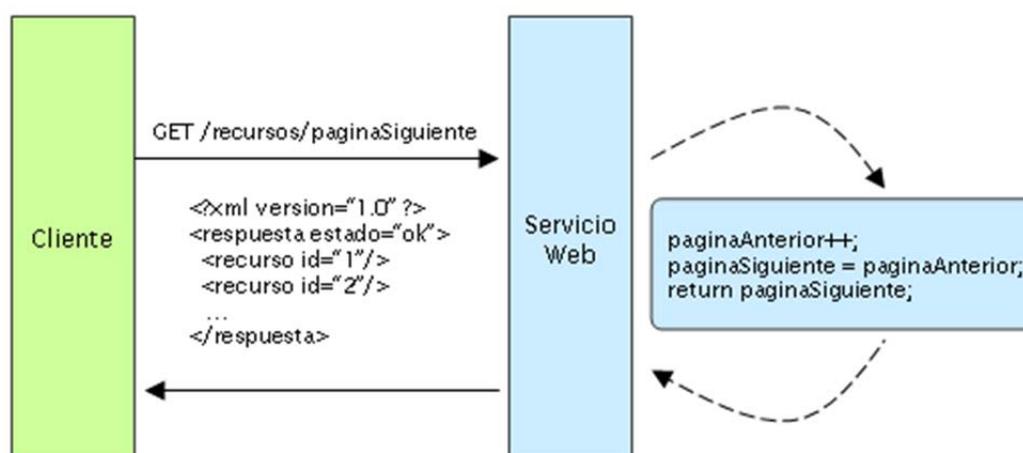


Figura 11. Servicio con estado.

Los servicios con estado tienden a volverse complicados. En la plataforma Java Enterprise Edition (Java EE), un entorno de servicios con estado necesita bastante análisis y diseño desde el inicio para poder almacenar los datos eficientemente y poder sincronizar la sesión del cliente dentro de un clúster de servidores. En este tipo de ambientes, ocurre un problema que le resulta familiar a los desarrolladores de servlets/JSP y EJB, quienes a menudo tienen que revolver buscando la causa de una `java.io.NotSerializableException` cuando ocurre la replicación de una sesión. Puede ocurrir tanto sea en el contenedor de servlets al intentar replicar la `Http Session` o por el contenedor de EJB al replicar un EJB con estado; en todos los casos, es un problema que puede costar mucho esfuerzo resolver, buscando el objeto que no implementa `Serializable` dentro de un grafo



CAPÍTULO 2

complejo de objetos que constituyen el estado del servidor. Además, la sincronización de sesiones es costosa en procesamiento, lo que impacta negativamente en el rendimiento general del servidor.

Por otro lado, los servicios sin estado son mucho más simples de diseñar, escribir y distribuir a través de múltiples servidores. Un servicio sin estado no sólo funciona mejor, sino que además mueve la responsabilidad de mantener el estado al cliente de la aplicación. En un servicio web REST, el servidor es responsable de generar las respuestas y proveer una interfaz que le permita al cliente mantener el estado de la aplicación por su cuenta. Por ejemplo, en el mismo ejemplo de una petición de datos en múltiples páginas, el cliente debería incluir el número de página a recuperar en vez de pedir "la siguiente", tal como se muestra en la siguiente figura:

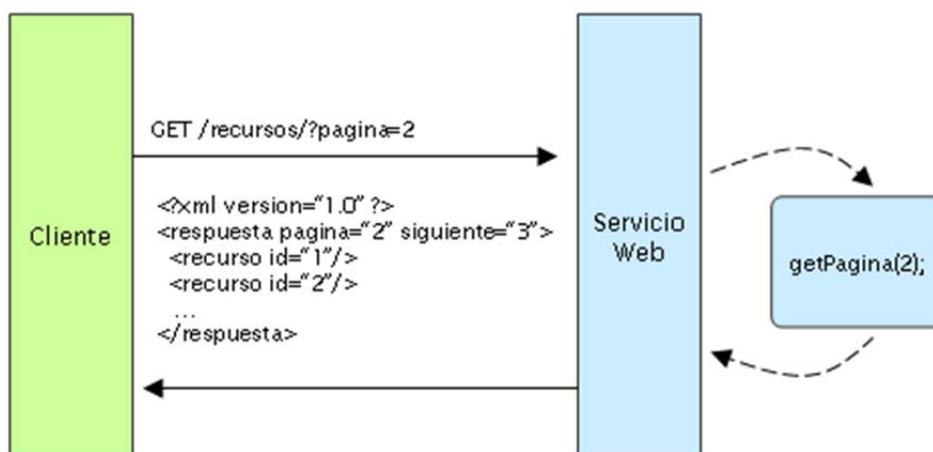


Figura 12. Servicio sin estado

- Responsabilidad del servidor

Genera respuestas que incluyen enlaces a otros recursos para permitirle a la aplicación navegar entre los recursos relacionados. Este tipo de respuestas tiene enlaces embebidos. De la misma manera, si la petición es hacia un padre o un recurso contenedor, entonces una respuesta REST típica debería también incluir enlaces hacia los hijos del padre o los recursos subordinados, de manera que se mantengan conectados.

Genera respuestas que indican si son susceptibles de caché o no, para mejorar el rendimiento al reducir la cantidad de peticiones para recursos duplicados, y para lograr eliminar algunas peticiones completamente. El servidor utiliza los atributos Cache-Control y Last-Modified de la cabecera en la respuesta HTTP para indicarlo.



- Responsabilidades del cliente de la aplicación

Utiliza el atributo Cache-Control del encabezado de la respuesta para determinar si debe cachear el recurso (es decir, hacer una copia local del mismo) o no. El cliente también lee el atributo Last-Modified y envía la fecha en el atributo If-Modified-Since del encabezado para preguntarle al servidor si el recurso cambió desde entonces. Esto se conoce como GET Condicional, y ambos encabezados van de la mano con la respuesta del servidor 304 (No Modificado) y se omite al recurso que se había solicitado si no hubo cambios desde esa fecha. Una respuesta HTTP 304 significa que el cliente puede seguir usando la copia local de manera segura, evitando así realizar las peticiones GET hasta tanto el recurso no cambie.

Envía peticiones completas que pueden ser serviciadas en forma independiente a otras peticiones. Esto implica que el cliente hace uso completo de los encabezados HTTP tal como está especificado por la interfaz del servicio web, y envía las representaciones del recurso en el cuerpo de la petición. El cliente envía peticiones que hacen muy pocas presunciones sobre las peticiones anteriores, la existencia de una sesión en el servidor, la capacidad del servidor para agregarle contexto a una petición, o sobre el estado de la aplicación que se mantiene entre las peticiones.

Esta colaboración entre el cliente y el servicio es esencial para crear un servicio web REST sin estado. Mejora el rendimiento, ya que ahorra ancho de banda y minimiza el estado de la aplicación en el servidor.

REST expone URIs con forma de directorios

Desde el punto de vista del cliente de la aplicación que accede a un recurso, la URI determina qué tan intuitivo va a ser el web service REST, y si el servicio va a ser utilizado tal como fue pensado al momento de diseñarlo. La tercera característica de los servicios web REST es justamente sobre las URIs.

Las URI de los servicios web REST deben ser intuitivas, hasta el punto de que sea fácil adivinarlas. Pensemos en las URI como una interfaz auto-documentada que necesita de muy poca o ninguna explicación o referencia para que un desarrollador pueda comprender a lo que apunta, y a los recursos derivados relacionados.



CAPÍTULO 2

Una forma de lograr este nivel de usabilidad es definir URIs con una estructura al estilo de los directorios. Este tipo de URIs es jerárquica, con una única ruta raíz, y va abriendo ramas a través de las subrutas para exponer las áreas principales del servicio. De acuerdo a esta definición, una URI no es solamente una cadena de caracteres delimitada por barras, sino más bien un árbol con subordinados y padres organizados como nodos.

REST transfiere XML, JSON, o ambos

La representación de un recurso en general refleja el estado actual del mismo y sus atributos al momento en que el cliente de la aplicación realiza la petición. La representación del recurso son simples "fotos" en el tiempo. Esto podría ser una representación de un registro de la base de datos que consiste en la asociación entre columnas y tags XML, donde los valores de los elementos en el XML contienen los valores de las filas. O, si el sistema tiene un modelo de datos, la representación de un recurso es una fotografía de los atributos de una de las cosas en el modelo de datos del sistema. Estas son las cosas que serviciamos con servicios web REST.

La última restricción al momento de diseñar un servicio web REST tiene que ver con el formato de los datos que la aplicación y el servicio intercambian en las peticiones/respuestas. Acá es donde realmente vale la pena mantener las cosas simples, legibles por humanos, y conectadas.

Los objetos del modelo de datos generalmente se relacionan de alguna manera, y las relaciones entre los objetos del modelo de datos (los recursos) deben reflejarse en la forma en la que se representan al momento de transferir los datos al cliente.

Es un protocolo que permite la comunicación entre aplicaciones a través de mensajes por medio de Internet. Es independiente de la plataforma, y del lenguaje. Está basado en XML y es la base principal de los Web Services [46].

2.6.2 SOAP

Es un protocolo que permite la comunicación entre aplicaciones a través de mensajes por medio de Internet. Es independiente de la plataforma, y del lenguaje. Está basado en XML y es la base principal de los Web Services.



SOAP es el primer protocolo de su tipo que ha sido aceptado prácticamente por todas las grandes compañías de software del mundo. Compañías que en raras ocasiones cooperan entre sí están ofreciendo su apoyo a este protocolo. Algunas de las mayores Compañías que soportan SOAP son Microsoft, IBM, SUN, Microsystems, SAP y Ariba.

Algunas de las Ventajas de SOAP son:

- No está asociado con ningún lenguaje: los desarrolladores involucrados en nuevos proyectos pueden elegir desarrollar con el último y mejor lenguaje de programación que exista pero los desarrolladores responsables de mantener antiguas aflicciones heredadas podrían no poder hacer esta elección sobre el lenguaje de programación que utilizan. SOAP no especifica una API, por lo que la implementación de la API se deja al lenguaje de programación.
- No se encuentra fuertemente asociado a ningún protocolo de transporte: La especificación de SOAP no describe como se deberían asociar los mensajes de SOAP con HTTP. Un mensaje de SOAP no es más que un documento XML, por lo que puede transportarse utilizando cualquier protocolo capaz de transmitir texto.
- No está atado a ninguna infraestructura de objeto distribuido La mayoría de los sistemas de objetos distribuidos se pueden extender, y ya lo están alguno de ellos para que admitan SOAP.
- Aprovecha los estándares existentes en la industria: Los principales contribuyentes a la especificación SOAP evitaron, intencionadamente, reinventar las cosas. Optaron por extender los estándares existentes para que coincidieran con sus necesidades. Por ejemplo, SOAP aprovecha XML para la codificación de los mensajes, en lugar de utilizar su propio sistema de tipo que ya están definidas en la especificación esquema de XML. Y como ya se ha mencionado SOAP no define un medio de transporte de los mensajes; los mensajes de SOAP se pueden asociar a los protocolos de transporte existentes como HTTP y SMTP.
- Permite la interoperabilidad entre múltiples entornos: las aplicaciones se ejecutan en plataformas con estándares que pueden comunicarse mediante mensaje SOAP con aplicaciones que se ejecutan en otras plataformas. Por ejemplo, una aplicación de escritorio que se ejecute en una PC puede comunicarse con una aplicación del back-end ejecutándose en un mainframe capaz de enviar y recibir XML sobre HTTP
- [47].



2.7 Comparación y elección

2.7.1 Sistema operativo

Finalmente este capítulo se concluye con la justificación de porque se ha elegido este método para la realización del proyecto, con ayuda de las características descritas en los apartados anteriores.

La aplicación del proyecto es realizada para el sistema operativo Android. La elección de este sistema operativo se hace con la ayuda de la comparación de las características de los dos sistemas operativos más importantes hasta la fecha (iOS y Android), los sistemas del apartado 2.3 fueron descartados de inmediato por su baja afluencia de usuarios. A la hora de buscar información sobre las características de uno y otro sistema, se llega a la conclusión de que realizar el proyecto en Android sería una opción más fiable y menos costosa. Los motivos son los siguientes:

- Android está basado en Java e iOS está basado en Objective C. Por la proximidad con los conocimientos que se adquieren en la carrera de Sistemas de Telecomunicaciones se eligió el sistema operativo Android.
- Otro factor a tener en cuenta es el precio a nivel de desarrollo software. La programación para iOS requiere el uso de un Mac así como una cuota anual de \$99 para poder contar con las herramientas y soportes necesarios para poder programar.
- Aunque iOS lleva más tiempo en el mercado, el sistema operativo Android se ha consolidado muy bien en el mercado y ha sufrido un crecimiento exponencial, en cuanto a niveles de usuarios se refiere. Y como se puede ver en la siguiente gráfica, el lenguaje Java es más utilizado que Objective C según el índice TIOBE.

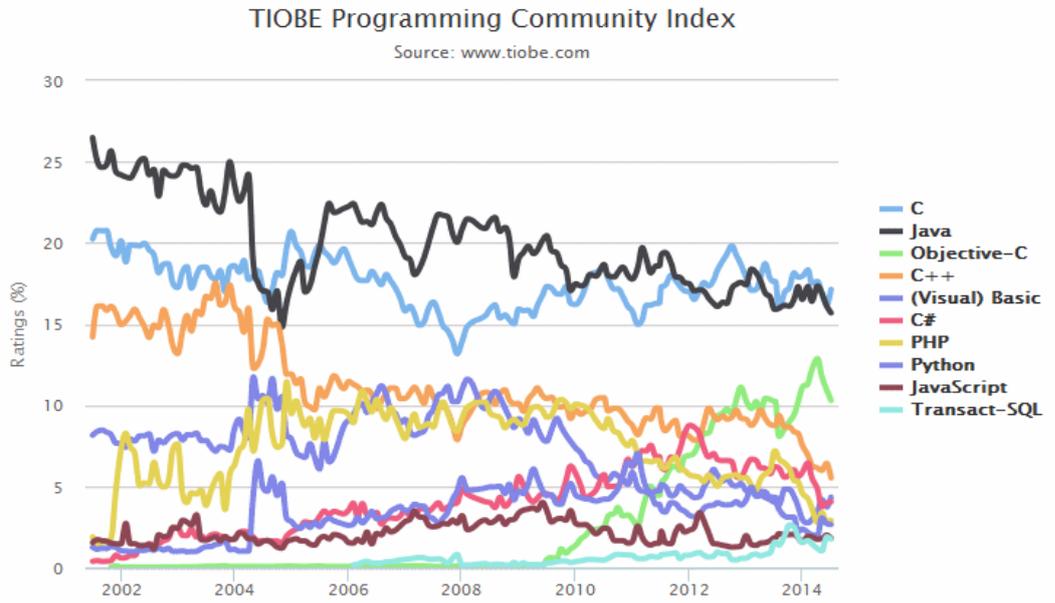


Figura 13. Ratings de los lenguajes de programación del índice TIOBE 2014.

- En cuanto a las aplicaciones, los dos sistemas principales, han incrementado el número de aplicaciones en los últimos años, pero es Android el que se ha alzado sobre iOS en este crecimiento. Android cuenta con cerca de 400 mil desarrolladores frente a los menos de 300 mil de iOS.

Total Number of Developers by App Store

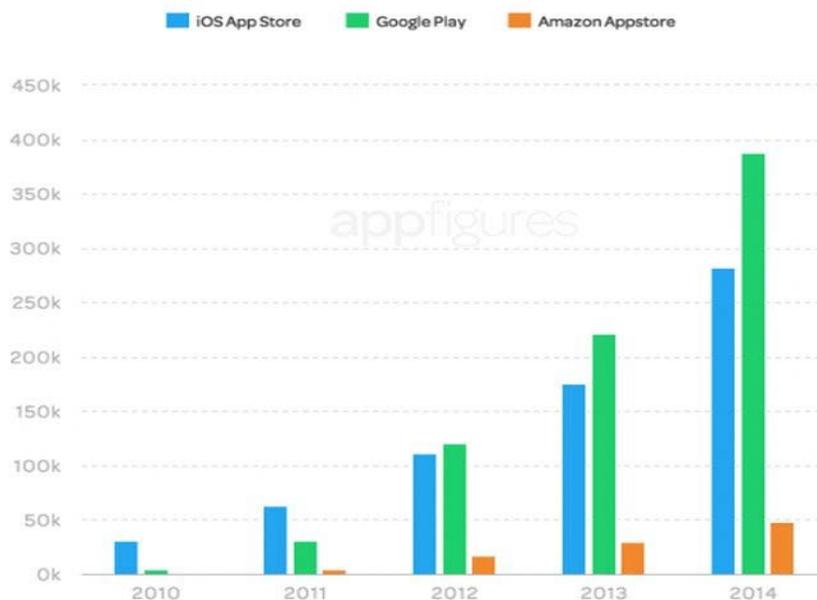




Figura 14. Gráfica del incremento del número de desarrolladores de los principales sistemas operativos móviles.

2.7.2 Lenguaje de servidor y base de datos

Tras haber realizado un estudio de las diferentes tecnologías existentes, finalmente se ha optado por el uso PHP junto a MySQL. El hecho de que ASP tenga una fuerte dependencia de Microsoft, debido a que requiere de un servidor web de Microsoft, ha descartado su uso.

PHP es un lenguaje, totalmente extendido y fiable, además es gratuito, y existe documentación y manuales en su web que hace que sea muy fácil encontrar información sobre funcionalidades, además es un lenguaje relativamente sencillo e intuitivo, por lo que finalmente ha sido la opción utilizada.

En cuanto a la elección de la base de datos, se ha optado por la elección de MySQL ya que es la base de datos más extendida en la actualidad, además como ya se ha mencionado, tiene una serie de ventajas, que hacen que sea una elección más que fiable y recomendada, es de código abierto, y se puede obtener de forma gratuita, además está respaldada por web tan importantes como Google y Facebook que usan esta tecnología como motor de base de datos en sus respectivos sitios web.

2.7.3 Servicio web

A continuación se mostrará una breve comparación entre REST y SOAP:

- REST, al utilizar HTTP es mucho más sencillo, desarrollar APIs, crear clientes, y la documentación es más fácil de entender.
- REST permite inúmeros formatos de datos, dando por ejemplo al desarrollador la posibilidad de utilizar JSON que normalmente es más rápido y como permite la utilización de JSON, permite también un mejor soporte a los clientes del explorador. SOAP solamente permite XML.
- REST tiene mejor escalabilidad y rendimiento.
- Las lecturas del REST se pueden cachear, las lecturas basadas en SOAP no se pueden.
- SOAP es interesante a la hora de hablar de seguridad, pues si REST soporta SSL, SOAP también lo hace, pero también soporta WS-Security lo que añade características de seguridad Enterprise.
- SOAP Proporciona una implementación estándar de integridad de datos y privacidad de datos.



- REST no tiene un sistema de mensajería estándar y no puede lidiar con la comunicación de fallos. SOAP proporciona fiabilidad en este sentido, incluso a través de intermediarios SOAP
- [48].

Viendo cada una de estas características, se ha optado por la utilización de REST, ya que al ser una Web pequeña y relativamente sencilla, REST resulta más sencillo y práctico. Se ha utilizado la tecnología JSON, por lo que finalmente se ha optado por la utilización de REST que soporta esta tecnología.

Capítulo 3: ANDROID

Una vez elegido Android como sistema operativo para realizar la aplicación, se detalla a continuación varios aspectos de este sistema operativo tales como la estructura, el lenguaje de programación y su desarrollo y el programa utilizado para compilar y ejecutar la aplicación.

3.1 Estructura Android

Ya se ha mencionado que Android está basado en Linux. Para ser más específicos, se habla del kernel. Android utiliza como base el kernel de Linux. Esto no significa que por estar basado en él, algo que se desarrolló en Linux funcione para Android, ya que por ejemplo Android no tiene soporte glibc.

La estructura consta de una serie de partes diferenciadas por colores. A continuación se detallará cada subestructura:

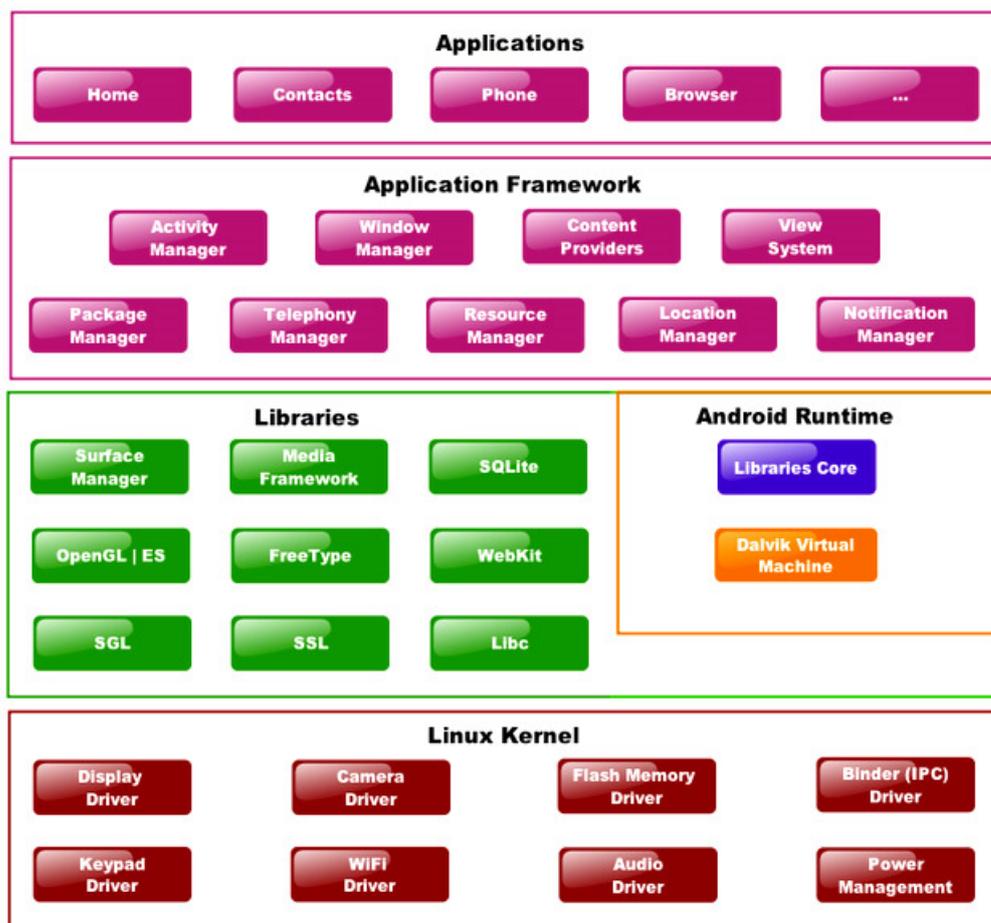


Figura 15. Estructura de sistema operativo Android.



Capa del Kernel (Roja)

Aquí se encuentra el corazón de Android: el manejo de memoria, procesos, drivers, etc. Es donde se da la comunicación con el hardware. Esto sirve para no depender de los fabricantes de cada móvil. También se administran los recursos del dispositivo, memoria, energía...

Capa Librerías (Verde)

Esta capa tiene las librerías nativas de Android, están escritas en C o C++ y tienen tareas específicas.

- Surface manager: Gestión dedicada al acceso a la pantalla.
- Media Framework: Realiza las acciones de reproducción de imágenes, audio y vídeo.
- SQLite: En este punto está incluida la base de datos (BD).
- Webkit: tarea dedicada a la navegación.
- SGL: Gráficos 2D.
- OpenGL: Gráficos 3D.
- Freetype: Su tarea es enderezar vectores o imágenes.

Android Runtime (Capa naranja)

Esta capa naranja no se considera al 100% una capa. Lo que es muy importante comentar es que aquí se encuentra Dalvik, la máquina virtual de Android, que no es lo misma que la Java Virtual Machine. Esto quiere decir que cuando se compila en Java lo que se genera solamente va a funcionar en la JVM, porque Dalvik es la máquina virtual de Android, así que el ByteCode que genera Java es inservible para Dalvik.

Algunas de las características de Dalvik son:

- Trabaja en entorno con restricción de memoria y procesador.
- Ejecuta el formato .dex.
- Convierte .class en .dx.

Application Framework (Capa rosa)

Esta capa es la es más visible para el desarrollador, ya que la mayoría de los componentes que forman parte del desarrollo se vamos a encontrar aquí.



CAPÍTULO 3

- Activity Manager: Administra las actividades de nuestra aplicación y el ciclo de vida.
- Windows Manager: Administra lo que se muestra en la pantalla.
- Content Provider: Administra dependiendo de cómo le indiquemos algunos contenidos, puede ser información que necesitamos la encapsule para enviar o compartir.
- View: Las vistas de elementos que son parte de la interfaz gráfica, como los mapas, cuadros de texto, etc.
- Notification Manager: Administra las notificaciones.
- PackageManager: Administra los paquetes y nos permite el uso de archivos en otros paquetes.
- Telephony Manager: Administra lo que tiene que ver con la telefonía, llamadas, mensajes.
- Resource Manager: Administra recursos de la aplicación, como los xml, imágenes, sonido.
- Location Manager: Gestiona la posición geográfica.
- Sensor Manager: Gestiona los sensores que tenga el dispositivo.
- Cámara: Administra la cámara.
- Multimedia: Administra lo referente a audio, video y fotos.

Aplicaciones (Última Capa Rosa)

Aquí tenemos las aplicaciones que vienen en el dispositivo, por ejemplo: el gestor de correos, los mensajes, el market, etc.

3.2 Programación y desarrollo Android

Java es un lenguaje orientado al objeto que fue diseñado para funcionar de forma independiente a la arquitectura sobre la que trabaja. La compilación del código fuente (archivos .java) genera archivos de clases (archivos .class), i.e. bytecode, los cuales serán ejecutados por la máquina virtual JVM (Java Virtual Machine), utilizada como abstracción entre el hardware de la máquina y los programas Java. Los distintos archivos .class suelen ser compilados en un único archivo JAR (Java Archive). En Android sucede algo parecido.

La Open Handset Alliance optó por diseñar un sistema similar teniendo en cuenta las limitaciones de los dispositivos móviles, que en principio eran una baja capacidad de almacenamiento y procesamiento y poca potencia de cálculo.

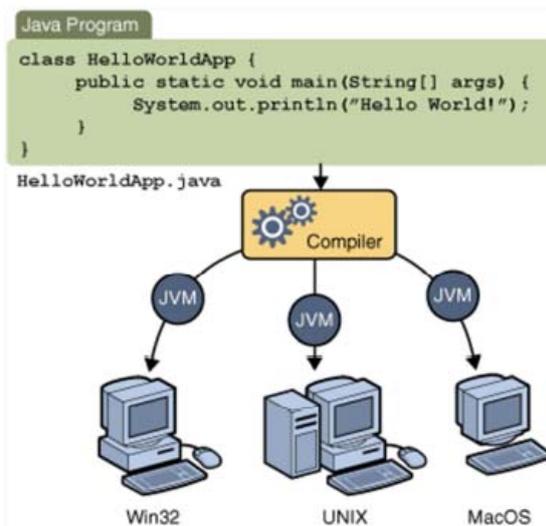


Figura 16. La compilación del código .java .

La máquina virtual empleada por Android recibe el nombre de DVM (DalvikVirtual Machine)

[49]. Los programas generalmente son escritos en Java y compilados a bytecode. Posteriormente son convertidos a archivos compatibles con la DVM (archivos.dex) antes de su instalación en el dispositivo. Estos archivos son comprimidos en archivos APK (Application Package), los cuales pueden ser instalados en dispositivos Android compatibles (a la hora de crear una aplicación es necesario especificar la versión para la que está diseñada por lo que el terminal donde se instalará ha de tener una versión de Android igual o superior). Se puede observar cómo se crean los archivos con extensión .dex a partir de los archivos de clase .class. La principal diferencia radica en la forma de empaquetar la información. Los archivos de clase son transformados en un único archivo .dex en el que se intenta desprestigiar la información repetida. Es más que evidente el enfoque de Android hacia dispositivos con memoria pequeña y almacenamiento similar.



CAPÍTULO 3

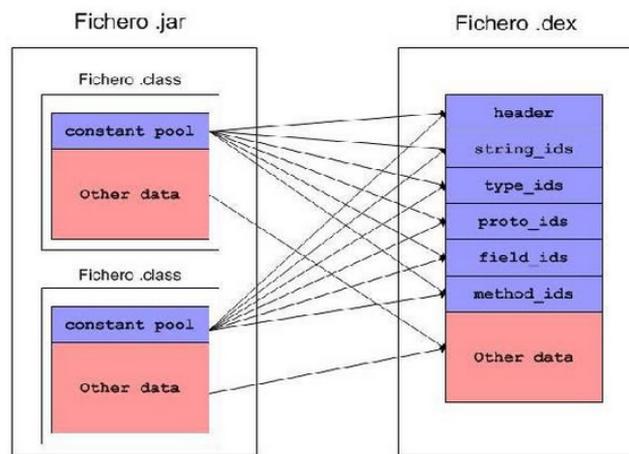


Figura 17. Conversión de .class a .dex

En cuanto al entorno de desarrollo, programar para Android tiene bastantes menos requisitos que hacerlo para iOS. Prueba de ello es la posibilidad de elección del sistema sobre el que se quiere programar (Unix, Windows, Mac OS,...) y del entorno de desarrollo a utilizar. Existen varias opciones, entre las que la más destacada es sin duda alguna Eclipse. Otro programa igual de válido que Eclipse es Android Studio, surgió más tarde que Eclipse pero que desde hace unos años es los nuevos programadores e incluso los que ya programaban Android con Eclipse se están pasando es este nuevo entorno de desarrollo como se puede ver en las opiniones de programadores de varios foros de Internet. Aquí se enumeran unas de las principales características por la cual los usuarios están decidiendo programa en un entorno como Android Studio:

- Es el futuro.
- En poco tiempo será lo único que el equipo de Android recomiende.
- Está basado en IntelliJ IDEA, uno de los IDE para java de primer nivel (entre los mejores, con Eclipse, netbeans)
- Nueva forma de construir los apk. Más serio, más versátil, más potente, más actual, y más parecido a un proyecto en java.
- Utiliza Gradle, con ventajas como:
 - Facilita muchísimo rehusar código y recursos.
 - Facilita configurar, extender y personalizar el proceso.
 - Facilita la distribución del código y por tanto trabajar en equipos.
 - Gestiona las dependencias de una forma cómoda y potente (está basado en Maven).
 - Permite compilar desde línea de comandos.
 - Hace muy fácil crear distintas versiones de la aplicación, por ejemplo para hacer una distribución multi-apk, para

ANDROID



distintos dispositivos, o una versión gratis y otra de pago, o una versión de prueba que carga distintos recursos, apunta a webservices distintos, usa estadísticas distintas, etc.

Otras opciones menos convencionales pero igual de válidas pasarían por el uso de NetBeans IntelliJ Idea

[50]

[51].

Capítulo 4: Descripción técnica del usuario

En este apartado se realizará una descripción técnica de la aplicación en la que se hablará de la base de datos en la que se comentará con la ayuda de imágenes y tablas de la base de datos todas las variables de las tablas y el uso que tiene cada una de ellas en la aplicación. Después con ayuda de diagramas de flujo se explicarán el orden de las clases usadas en la aplicación y la funcionalidad que se realiza en cada parte de la aplicación. Con ello se tratará de esclarecer lo máximo posible la parte interna de la aplicación para un mayor entendimiento de la misma.

Como se ha mencionado anteriormente, a continuación se explicará todo lo referente a la base de datos externa que se ha utilizado así como las relaciones entre tablas.

4.1 Estructura de la base de datos

A continuación se muestra la estructura de la base de datos utilizada, tanto para la aplicación para Smartphone como para la Web, la base de datos sigue una estructura racional, y se vinculan tablas entre sí. En los siguientes apartados se describirá cada una de las tablas detalladamente, y las relaciones más importantes entre cada una de ellas, para entender mejor el funcionamiento.

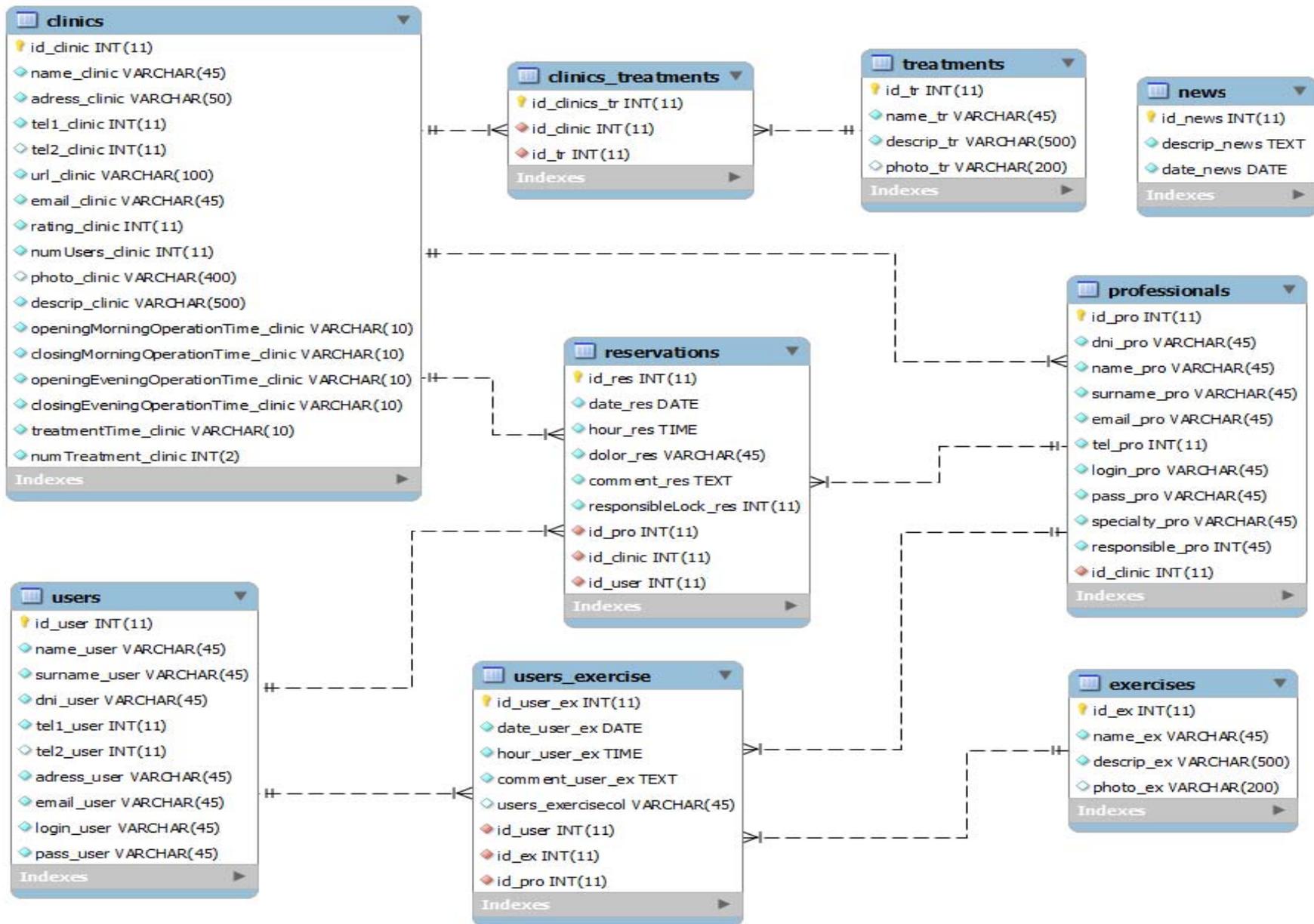


Figura 18. Estructura de la base de datos.

4.1.1 Tabla Clinics

Esta tabla contiene toda la información de las clínicas, habrá datos de interés para el usuario como pueden ser los campos de teléfono, url o email, y otros campos importantes para el funcionamiento de la aplicación, como pueden ser el tiempo de cada tratamiento, o el número de tratamientos de la clínica, que sirven de base en algunos de los algoritmos de la aplicación.

A continuación se describe con detalle alguno de los campos de la tabla:

- id_clinic: es el identificador de cada clínica, el campo que va a diferenciar una clínica de otra.
- name_clinic: indica el nombre de la clínica.
- adress_clinic: indica la dirección de la clínica.
- tel1_clinic: da la información del primer teléfono de la clínica.
- tel2_clinic: da la información del segundo teléfono de la clínica.
- url_clinic: campo utilizado para guardar la página web de la clínica.
- email_clinic: en este campo se guarda el correo electrónico de contacto de la clínica.
- rating_clinic: campo utilizado en la aplicación Android, que aparece en forma de RatingBar, para indicar la valoración que tiene cada clínica.
- numUsers_clinics: guarda el número de usuarios que han reservado en la clínica, necesario para establecer el rating.
- photo_clinic: es un campo de texto, debido a que no se guarda, la imagen como tal en la base de datos, sino la url donde está alojada la fotografía.
- descrip_clinic: campo utilizado para mostrar una breve descripción de los servicios que ofrecen las clínicas a los pacientes.
- opening(Morning, Evening)OperationTime_clinic: indica la hora en la que la clínica inicia la apertura, tanto por la mañana (Morning), como por la tarde (Evening).
- closing(Morning, Evening)OperationTime_clinic: indica la hora en la que la clínica cierra, tanto por la mañana (Morning), como por la tarde (Evening).
- treatmentTime_clinic: campo que indica el tiempo aproximado que tarda un profesional en realizarle un tratamiento a un cliente.
- numTreatment_clinic: indica el número de tratamientos que realiza una clínica en una misma franja horaria.



Tabla Clinicas				
Campo	Tipo	Nulo	Predeterminada	Extra
id_clinic	int (11)	No	ninguna	auto_increment
name_clinic	varchar (45)	No	ninguna	
adress_clinic	varchar (50)	No	ninguna	
tel1_clinic	int (11)	No	ninguna	
tel2_clinic	int (11)	No	<i>Null</i>	
url_clinic	varchar (100)	No	ninguna	
email_clinic	varchar (45)	No	ninguna	
rating_clinic	int (11)	No	ninguna	
numUsers_clinic	int (11)	No	ninguna	
photo_clinic	varchar (400)	No	<i>Null</i>	
descrip_clinic	varchar (500)	No	ninguna	
openingMorningOperationTime_clinic	varchar (10)	No	ninguna	
closingMorningOperationTime_clinic	varchar (10)	No	ninguna	
openingEveningOperationTime_clinic	varchar (10)	No	ninguna	
ClosingEveningOperationTime_clinic	varchar (10)	No	ninguna	
treatmentTime_clinic	varchar (10)	No	ninguna	
numTreatment_clinic	int (11)	No	ninguna	

Tabla 1. Tabla clinics

4.1.2 Tabla Treatments

Esta tabla contiene los campos necesarios para los tratamientos, a continuación se describen detalladamente cada uno de ellos:

- id_tr: es el identificador de cada tratamiento, cada tratamiento tiene uno diferente e irrepetible.
- name_tr: indica el nombre del tratamiento.
- descrip_tr: campo para la descripción del tratamiento (en qué consiste, aplicaciones...).
- photo_tr: al igual que en la anterior tabla, este campo contiene la url donde está alojada la fotografía.



CAPÍTULO 4

Tabla treatments				
Campo	Tipo	Nulo	Predeterminada	Extra
id_tr	int (11)	No	ninguna	auto_increment
name_tr	varchar (45)	No	ninguna	
descrip_tr	varchar (500)	No	ninguna	
photo_tr	varchar (200)	No	<i>Null</i>	

Tabla 2. Tabla Treatments

4.1.3 Tabla Clinics_Treatments

La Tabla clinics_treatments, ha sido creada porque se necesita una relación de varios a varios entre clínicas y tratamientos, es decir, una clínica puede tener varios tratamientos, y un tratamiento puede estar implantado en varias clínicas, de ahí la necesidad de crear una nueva tabla que interrelacione las dos anteriores, a continuación se muestra la descripción detallada de los campos:

- id_clinic_tr: campo clave, con el que se idéntica a cada elemento perteneciente a esta tabla.
- id_clinic: es el identificador de la clínica al que pertenece.
- id_tr: identifica al tratamiento.

Tabla clinics_treatments				
Campo	Tipo	Nulo	Predeterminada	Extra
id_clinic_tr	int (11)	No	ninguna	auto_increment
id_clinic	int (11)	No	ninguna	
id_tr	int (11)	No	ninguna	

Tabla 3. Tabla Clinics_Treatments

4.1.4 Tabla Users

Contiene los campos más relevantes de un usuario que se detallan a continuación:

- id_user: es el identificador de usuario, es único para cada usuario.
- name_user: este campo almacena el nombre del paciente.
- surname_user: este campo almacena el apellido del paciente.



- dni_user: se almacena el documento nacional de identidad del paciente.
- tel1_user: se almacena el primer teléfono de contacto del usuario.
- tel2_user: se almacena el segundo teléfono de contacto del usuario.
- adress_user: se recoge la dirección del domicilio del paciente.
- email_user: se recoge la información del correo electrónico del paciente.
- login: es un campo que va a permitir distinguir a un usuario u otro, es irrepetible, no puede haber dos usuarios con el mismo login.
- password: es la clave de ingreso a la aplicación de cada usuario, la que le va a permitir acceder a la aplicación.

Tabla users				
Campo	Tipo	Nulo	Predeterminada	Extra
id_user	int (11)	No	ninguna	auto_increment
name_user	varchar (45)	No	ninguna	
surname_user	varchar (45)	No	ninguna	
dni_user	varchar (45)	No	ninguna	
tel1_user	int (11)	No	ninguna	
tel2_user	int (11)	No	<i>Null</i>	
adress_user	varchar (45)	No	ninguna	
email_user	varchar (45)	No	ninguna	
login_user	varchar (45)	No	ninguna	
pass_user	varchar (45)	No	ninguna	

Tabla 4. Tabla Users

4.1.5 Tabla Exercises

Contiene los datos necesarios para describir cada uno de los ejercicios que tienen que realizar los usuarios de la aplicación:

- id_ex: al identifica a cada ejercicio, cada ejercicio tiene su propio identificador, es único e irrepetible.
- name_ex: campo en el que se va registrar el nombre de cada ejercicio.
- descrip_ex: pequeña descripción de cada ejercicio (en qué consiste, número de repeticiones...)
- photo_ex: contiene la url donde está alojada la fotografía.



CAPÍTULO 4

Tabla exercises				
Campo	Tipo	Nulo	Predeterminada	Extra
id_ex	int (11)	No	ninguna	auto_increment
name_ex	varchar (45)	No	ninguna	
descrip_ex	varchar (500)	No	ninguna	
photo_ex	varchar (200)	No	<i>Null</i>	

Tabla 5. Tabla excercises

4.1.6 Tabla Users_Exercise

Al igual que la tabla `clinics_treatments`, esta tabla ha sido creada por la necesidad de relacionar de varios a varios los usuarios y los ejercicios, es decir un usuario puede tener varios ejercicios asignados, y cada ejercicio puede estar asignado a diferentes usuarios, por este motivo, es necesario la creación de esta tabla que los relacione.

Por lo tanto esta tabla va a tener un campo que sea `id_user` y otro `id_ex`, que identifique a que usuario y a qué ejercicio pertenece. Los campos son:

- id_user_ex: identificador de relación entre usuario y ejercicio.
- id_user: identificador de usuario.
- id_ex: identificador de ejercicio.
- date_user_ex: indica la fecha del ejercicio asignado.
- hour_user_ex: indica la hora del ejercicio asignado.
- comment_user_ex: indica información que el profesional puede considerar interesante, como se debería de realizar el ejercicio, aclaraciones...
- users_exerciseCol: indica si el ejercicio está realizado.

Tabla users_exercise				
Campo	Tipo	Nulo	Predeterminada	Extra
id_user_ex	int (11)	No	ninguna	auto_increment
id_user	int (11)	No	ninguna	
id_ex	int (11)	No	ninguna	
date_users_ex	date	No	ninguna	
hour_users_ex	time	No	ninguna	
id_pro	int (11)	No	ninguna	
comment_user_ex	text	No	ninguna	
users_exerciseCol	varchar (45)	No	ninguna	

Tabla 6. Tabla users_exercise



4.1.7 Tabla Professionals

Contiene la información de cada profesional, almacena el nombre, email, teléfono...

Un campo que hay que describir con claridad de esta tabla es el campo `responsable_pro` que permite distinguir entre un profesional, es decir un fisioterapeuta, y un encargado de la clínica, que también puede ser un fisioterapeuta, pero además tiene una serie de privilegios que solo posee él, y le permite realizar una serie de funcionalidades a mayores, que serán explicadas más adelante. Las variables son:

- `id_pro`: identificador del profesional, único para cada especialista.
- `name_pro`: recoge el nombre del profesional.
- `surmane_pro`: recoge el apellido del profesional.
- `email_pro`: almacena el correo electrónico de contacto con el especialista.
- `tel_pro`: almacena el teléfono de contacto del profesional.
- `login_pro`: almacena el login de inicio de sesión de cada profesional.
- `pass_pro`: almacena la clase de acceso a los servicios de la parte de profesional o encargado.
- `specialty_pro`: hace referencia a la especialidad de cada profesional, puede haber profesionales expertos en fisioterapia deportiva, otros expertos en una técnica en concreto...
- `id_clinic`: número entero que hace referencia a la clínica en la que está trabajando el profesional o encargado.
- `resposable_pro`: si tiene el valor 0, será un profesional, en cambio si el valor de ese campo es 1, se podrá loguear como encargado, y tener los privilegios anteriormente mencionados.



CAPÍTULO 4

Tabla professionals				
Campo	Tipo	Nulo	Predeterminada	Extra
id_pro	int (11)	No	ninguna	auto_increment
dni_pro	varchar (45)	No	ninguna	
name_pro	varchar (45)	No	ninguna	
surname_pro	varchar (45)	No	ninguna	
email_pro	varchar (45)	No	ninguna	
tel_pro	int (11)	No	ninguna	
login_pro	varchar (45)	No	ninguna	
pass_pro	varchar (45)	No	ninguna	
specialty_pro	varchar (45)	No	ninguna	
id_clinic	int (11)	No	ninguna	
responsible_pro	varchar (45)	No	ninguna	

Tabla 7. Tabla Professionals

4.1.8 Tabla Reservations

Esta tabla contiene información de las reservas que realizan los usuarios e intervienen bastantes relaciones entre tablas. Cada usuario, va a poder realizar reservas en las diferentes clínicas, según el interés del usuario en un determinado tratamiento que ofrece la clínica, o por cualquier otra razón que resulte atractiva al usuario.

A continuación se describen los campos:

- id_res: identificador de la reserva, único para cada reserva realizada con éxito.
- date_res: se almacena la fecha de la reserva.
- hour_res: se almacena la hora de la reserva.
- id_clinic: contiene el identificador de la clínica en la que quiere reservar el usuario.
- id_user: es el campo en el que se va a guardar el identificador de usuario, y va a permitir al profesional identificar a cada usuario.
- dolor_res: se almacena el lugar del dolor que ha especificado el paciente a la hora de realizar la reserva.
- comment_res: se almacena el comentario que el paciente le escribe al paciente a la hora de la realización de la reserva.



- id_pro: sirve para almacenar el profesional que va a ser asignado por el encargado (o administrador) a cada reserva, es decir a cada reserva se le asigna un profesional.
- responsibleLock_res: el encargado puede bloquear ciertas horas o días enteros para impedir que los pacientes puedan reservar en esas franjas horarias, por ejemplo, festivos u horas libres. Esto se hace poniendo este campo a 1.

Tabla reservations				
Campo	Tipo	Nulo	Predeterminada	Extra
id_res	int (11)	No	ninguna	auto_increment
date_res	date	No	ninguna	
hour_res	time	No	ninguna	
id_clinic	int (11)	No	ninguna	
id_user	int (11)	No	ninguna	
dolor_res	varchar (45)	No	ninguna	
comment_res	text	No	ninguna	
id_pro	int (11)	No	ninguna	
responsibleLock_res	int (11)	No	ninguna	

Tabla 8. Tabla Reservations

4.1.9 Tabla News

Esta tabla es utilizada en el sitio web para ofrecer las últimas noticias a los usuarios y contiene los siguientes campos:

- id_news: identifica a cada noticia con un número diferente
- descrip_news: es el campo donde se guarda la información, o la descripción de la noticia
- date_news: es la fecha en la que se genera la noticia



CAPÍTULO 4

Tabla news				
Campo	Tipo	Nulo	Predeterminada	Extra
id_news	int (11)	No	ninguna	auto_increment
descrip_news	text	No	ninguna	
date_news	date	No	ninguna	

Tabla 9. Tabla News

4.2 Relación entre las tablas de la base de datos

Con los siguientes gráficos, se aporta información de cómo se relacionan unas tablas con otras, las relaciones más importantes son las descritas a continuación.

4.2.1 Clinics y Treatments

En verde se puede observar de forma gráfica que existe una relación entre la tabla clinics y la tabla treatments, dado que una clínica puede tener varios tratamientos, y un tratamiento puede estar implantado en varias clínicas, se crea la tabla clinics_treatments, que relaciona una con otra. Cuando se realicen consultas a la base de datos, para obtener los tratamientos que tiene una clínica, es necesaria esta tabla.

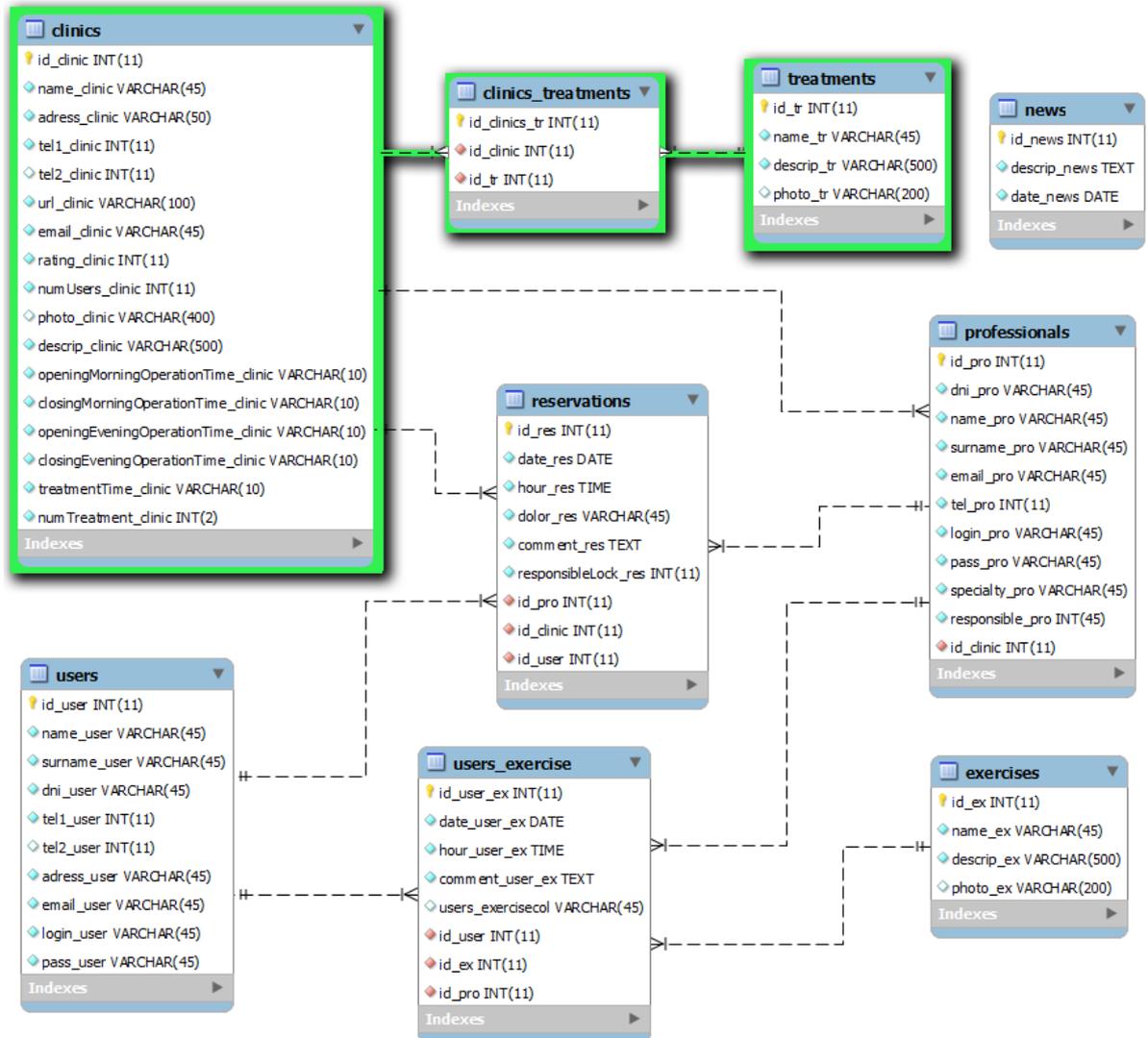


Figura 19. Relación entre la tabla clinics y la tabla tratamientos (Verde).

4.2.2 Professionals, Exercises y Users

Al igual que ocurre con el caso anterior, también hay una relación de varios a varios, con la consiguiente creación de la tabla users_exercise, ya que un usuario puede tener varios ejercicios, pero también un mismo ejercicio puede ser asignado a diferentes usuarios.

También está relacionada con la tabla professionals ya que cada ejercicio es asignado por un determinado profesional.



CAPÍTULO 4

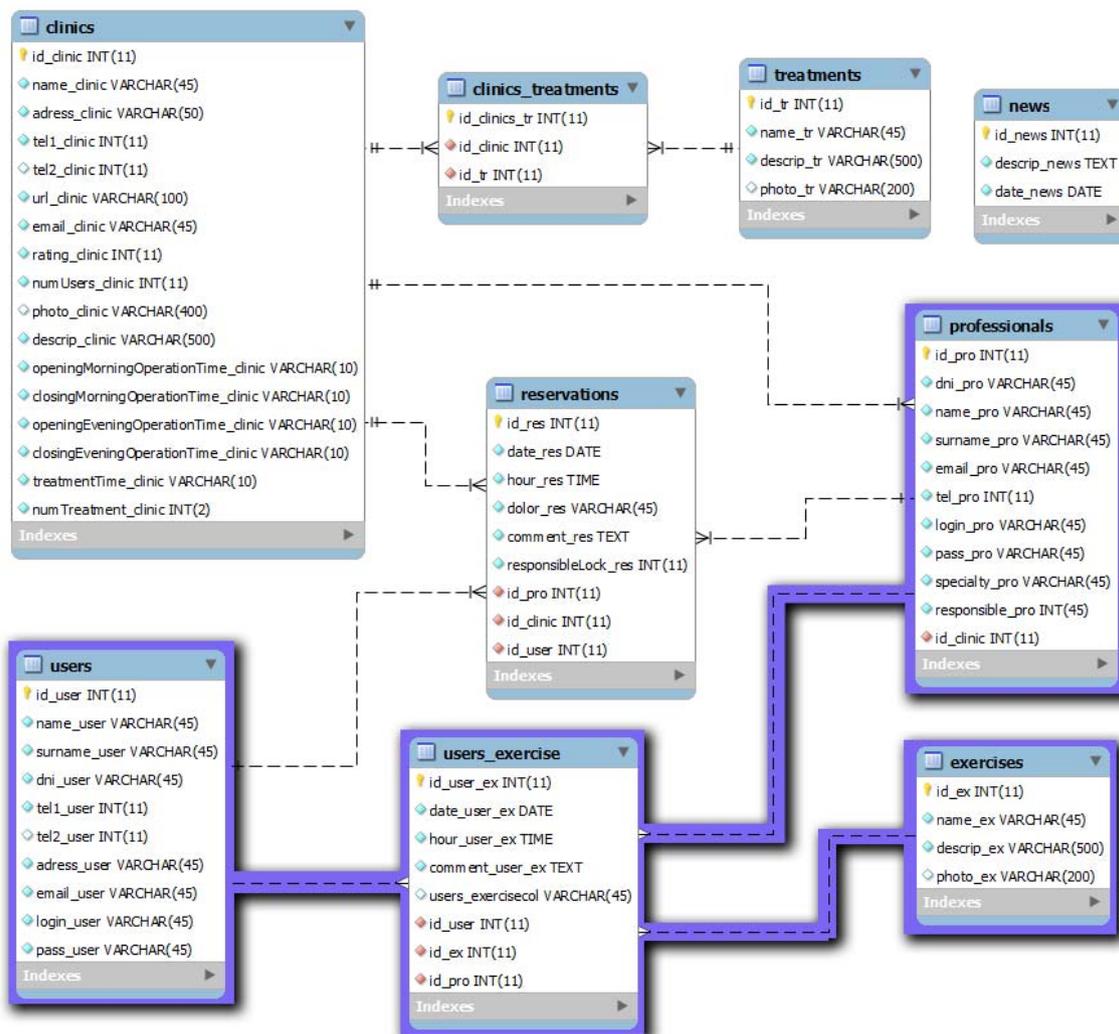


Figura 20. Relación entre las tablas professionals, exercises y users (Azul).

4.2.3 Clinics, Treatments, Reservations y Professionals

En este caso se ve que hay una relación de uno a varios entre la tabla de clínica y las tablas de profesionales y reservas; hay que añadir la anteriormente descrita de varios a varios entre clínicas y tratamientos.

En este caso se ve que la tabla professionals, contiene el id_clinic de la tabla clinics, debido que en una clínica puede haber diferentes profesionales trabajando, es necesario hacer una relación de uno a varios que relacione ambas.

Lo mismo ocurre con la tabla reservations, que también contiene id_clinic, ya que en una clínica se pueden realizar varias reservas.

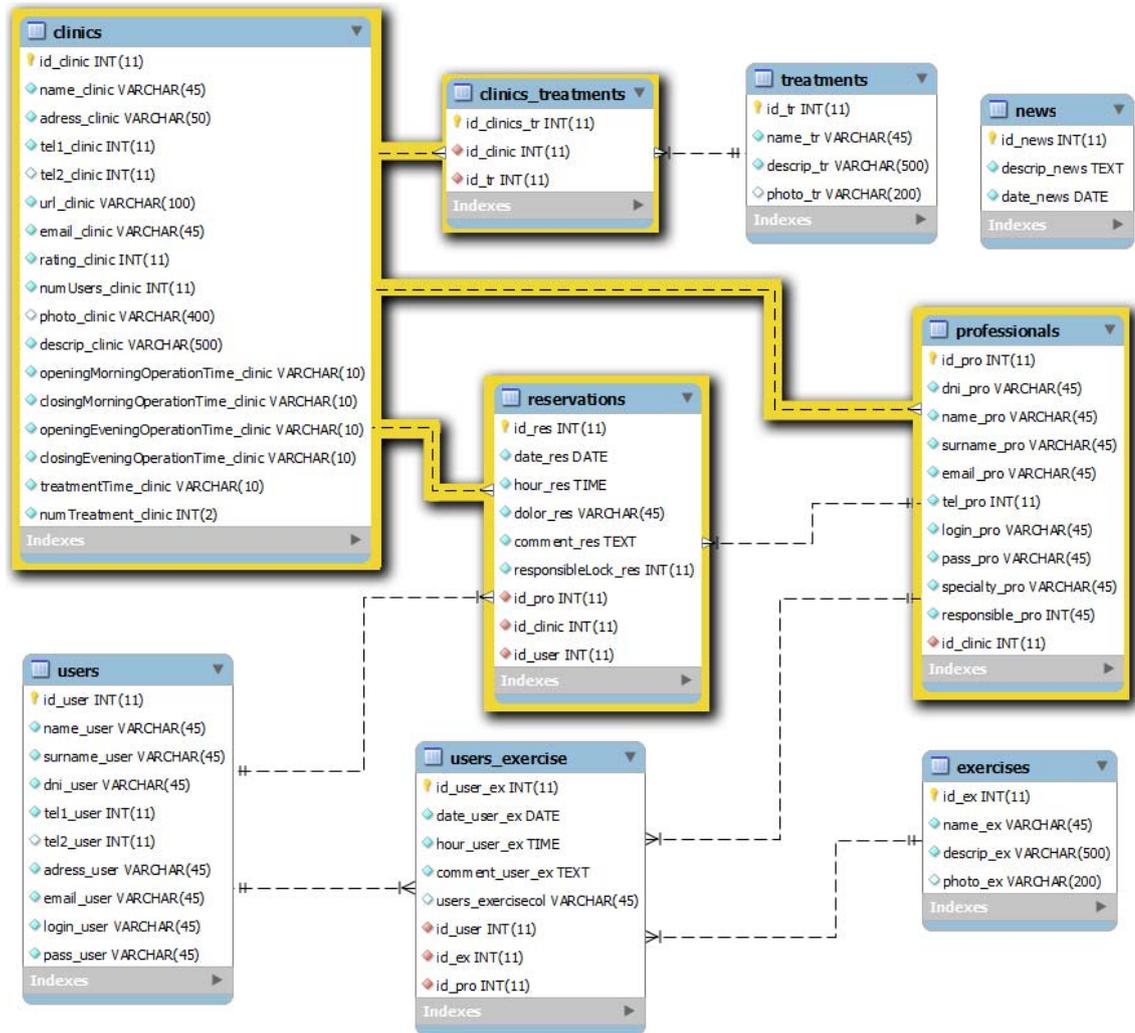


Figura 21. Relación entre las tablas professionals, clinics, treatments y reservations (Amarillo).

4.2.4 Professionals, Reservations y Users_Exercise

Como se puede ver, la tabla professionals, está relacionada con reservations y users_exercises, en forma de uno a varios, esto se debe a que a la hora de realizar una reserva, esa reserva es vista por un profesional que el que asigna un determinado ejercicio al usuario, de esta forma un profesional puede tener varias reservas, pero una reserva sólo puede ser gestionada por un profesional.

Por otro lado, a cada usuario se le asignan diferentes ejercicios, y esos ejercicios son asignados por un profesional, de aquí viene esta última relación, cada ejercicio es asignado por un sólo profesional, pero un profesional puede asignar varios ejercicios.



CAPÍTULO 4

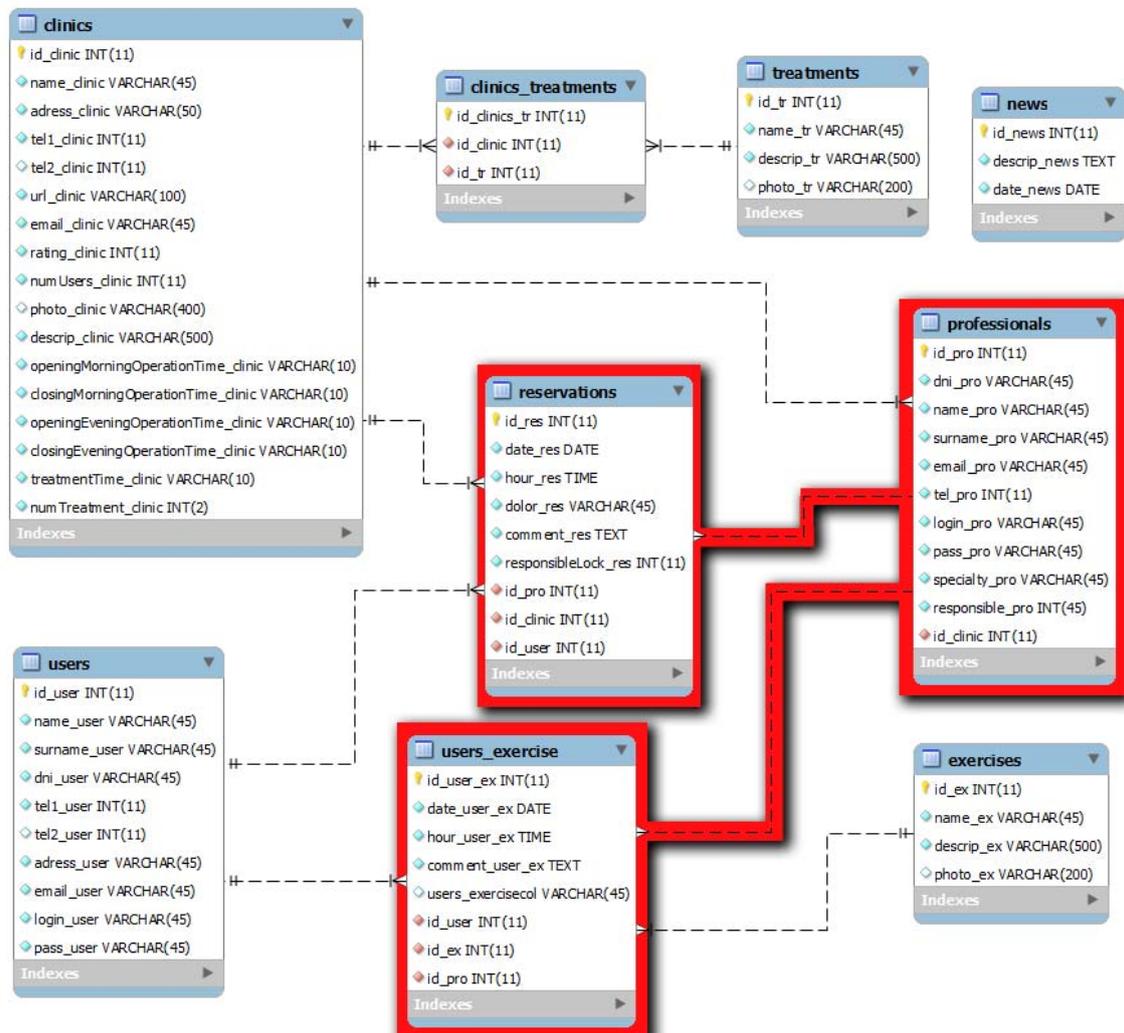


Figura 22. Relación entre las tablas professionals, reservations y users_exercise (Rojo).

4.2.5 Professionals, Reservations, Clinics y Users

Como se ve en la figura, la tabla reservations, necesita información de las tablas clinics, professionals, y users. Esta implementado de esta manera ya que cuando un usuario realiza una reserva, el id_user queda guardado en la reserva para identificar de qué usuario se trata, el id_clinic es necesario también ya que el usuario decide la clínica en la que quiere realizar la reserva y por lo tanto también tendremos que tener un campo para almacenar ese dato.

Una vez que se realiza la reserva, esta es gestionada por el encargado de la clínica y asignada a un profesional. Para realizar esta asignación necesitaremos el último campo (id_pro), para saber a qué profesional le ha sido asignada la reserva.

DESCRIPCIÓN TÉCNICA DEL USUARIO



Todas ellas son relaciones de uno a varios, ya que una reserva solo puede realizarse por un único usuario, y solo se le es asignado a una clínica y a un profesional.

En cambio un usuario, clínica o profesional, puede tener varias reservas.

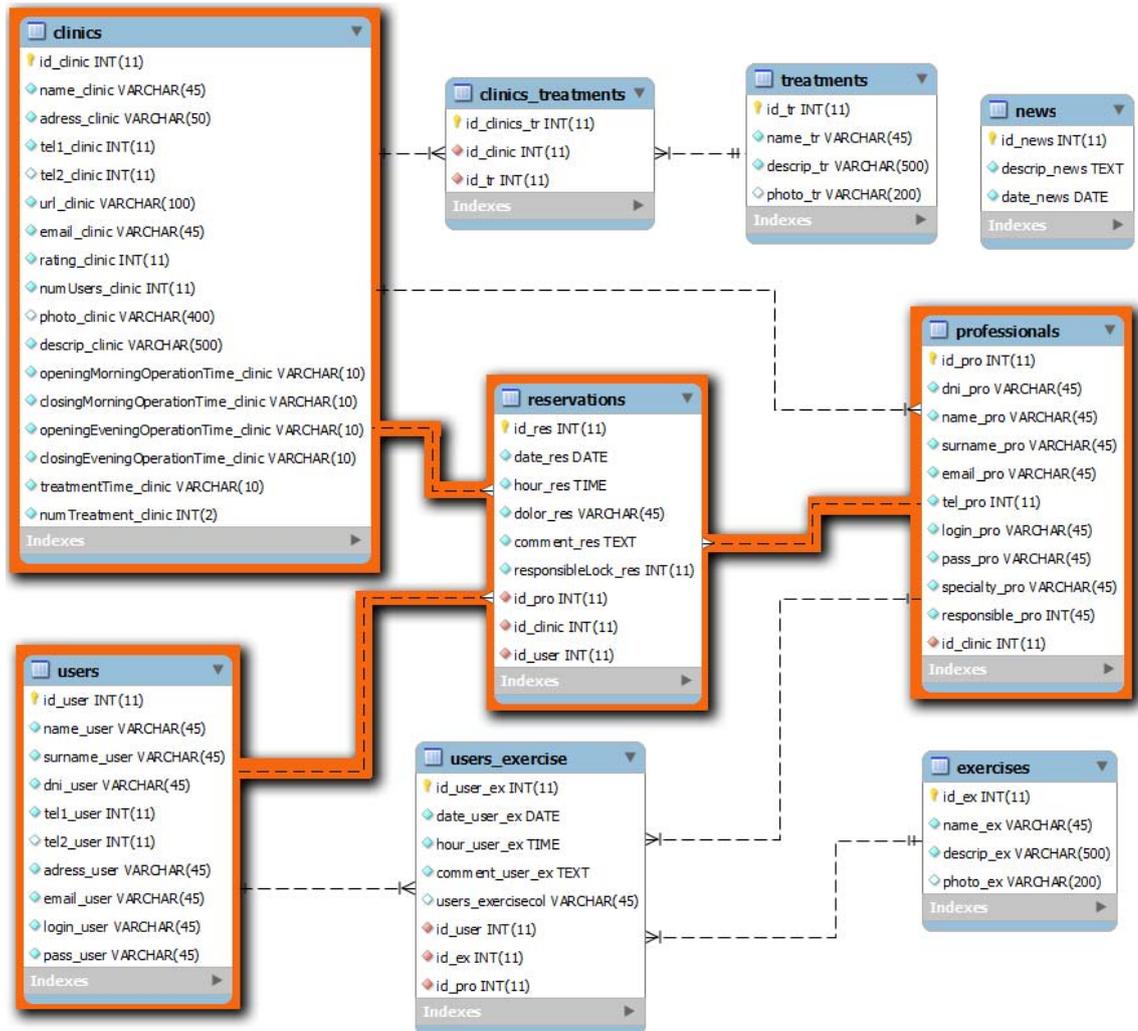


Figura 23. Relación entre las tablas professionals, reservations, users y clinics (Naranja).



4.3 Descripción de funcionalidades

En este apartado se va a describir las funcionalidades internas de la aplicación con respecto al paciente. Para ello, se tomarán como apoyo los diagramas de flujo de la aplicación. Por cada pantalla se mostrará dos diagramas de flujo, uno para explicar las clases que van apareciendo, y otro en el que se detallan las funcionalidades internas de esa parte de la aplicación.

4.3.1 Login

En la siguiente figura se ven las clases a las que se pueden acceder cuando se inicia la aplicación. Como inicio de la aplicación se tiene la clase Login. En esta clase se puede realizar dos opciones, la primera es registrarse, en la cual dando al botón de registro se mostrará el layout de la clase Registrar; la segunda opción es la del logueo, para poder realizar esta opción es necesario haber sido registrado previamente. Para el logueo es necesario introducir el usuario y la contraseña correctas y en función de esos datos se mostrará la el layout de la clase Usuario, Profesional o Encargado. En el caso de que el inicio de sesión no fuera correcto se mostraría de nuevo la pantalla de Login.

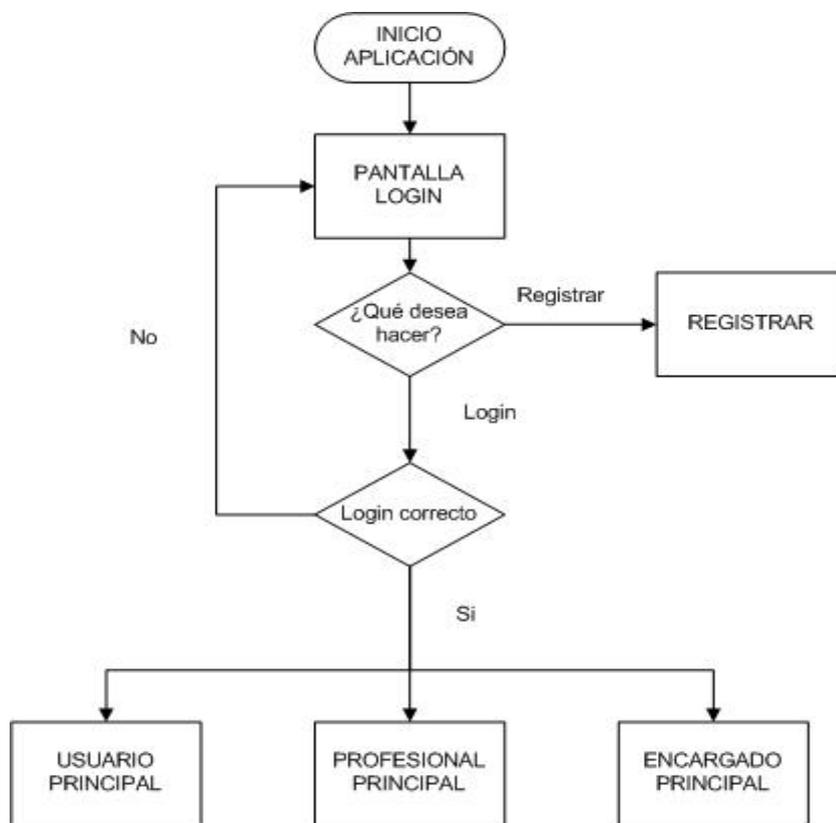


Figura 24. Diagrama de flujo de la clase Login.



En la siguiente figura se ve el diagrama de flujo de la funcionalidad de Login. En la pantalla de esta clase se pide dos datos (usuario y contraseña). Una vez introducido los datos, el programa se encarga de comprobar esos datos con la base de datos. Primero mira en la tabla de users, si los datos corresponden con algún usuario registrado en la aplicación, se pasaría al layout de la clase Usuario. En el caso de no encontrar ninguna coincidencia en la tabla de users, el programa realiza la misma acción con la tabla professionals y si encuentra alguna coincidencia saltaría el layout de la clase Profesional. Por último, sino es un profesional, el programa miraría si es un encargado, de acuerdo a lo mencionado en los apartados anteriores donde se menciona las variables de las tablas de la base de datos, el programa mira tan bien en la base de datos la tabla de professionals, pero en este caso atendiendo a las filas en las que la variable responsable_lock esté a 1 (ver apartado 4.1).

Una vez recorridas las tablas, si el programa no encuentra ninguna coincidencia en ellas, o alguno de los campos (usuario o contraseña) no han sido introducidos, la aplicación mostrará un mensaje de error por pantalla con un aviso de que los campos introducidos no son correctos.

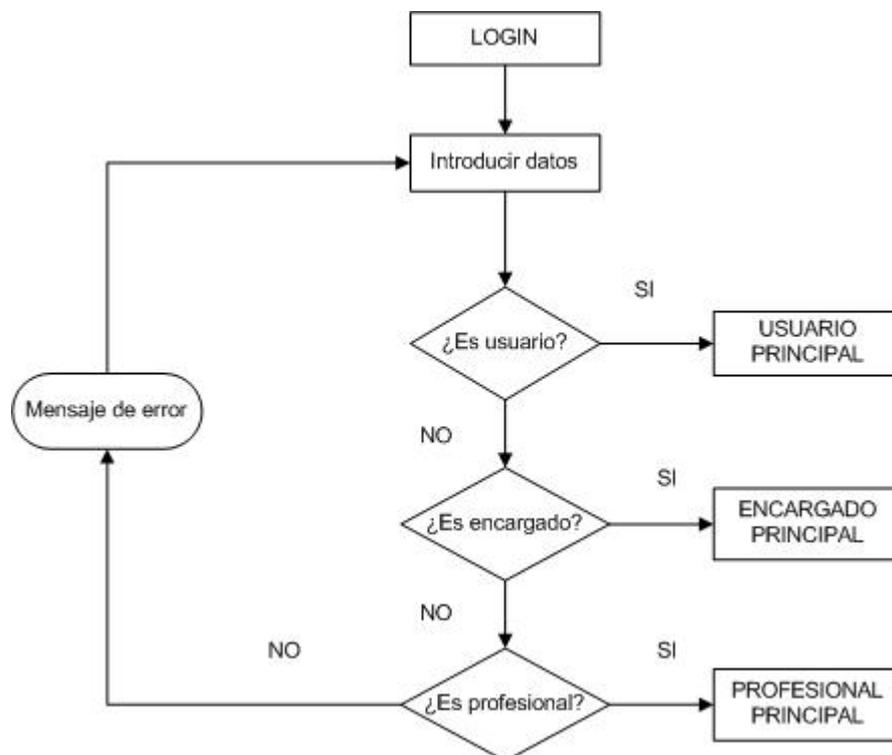


Figura 25. Diagrama de flujo de la funcionalidad Login.



4.3.2 Registrar

En este apartado se comentará la funcionalidad de la clase Registrar. Una vez se muestre el layout de esta clase, se pedirá introducir una serie de campos necesarios para el registro. De todos los campos que se piden hay 5 campos obligatorios (Nombre, Apellidos, DNI, Login y Password). La primera acción del programa es comprobar si los campos obligatorios para el registro están introducidos, en caso de no estar se mostraría un mensaje de error en el que se informa de que no todos los campos están introducidos y se mostraría de nuevo el layout de la clase Registrar.

Si los campos obligatorios están introducidos, la aplicación pasa a comprobar si el Login que ha usado el paciente para la aplicación existe o no ya en la tabla users de la base de datos. Si existe, se muestra por pantalla un mensaje advirtiendo del error que ha surgido (Login ya existente); en el caso de que el Login estuviera libre, el programa procederá a la inserción del paciente en la base de datos y se mostraría de nuevo el layout de la clase Login para que el nuevo usuario pueda proceder con el logueo.

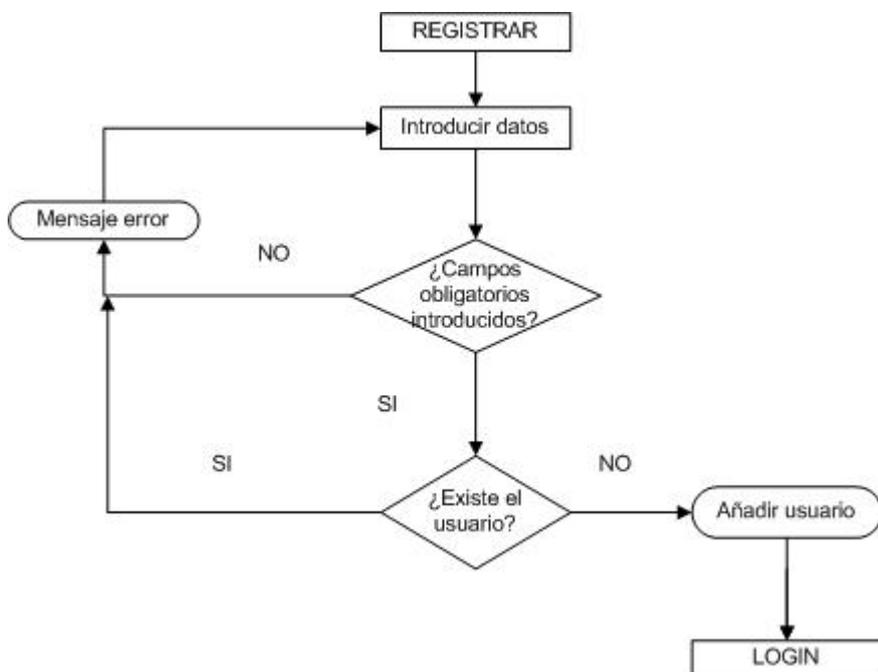


Figura 26. Diagrama de flujo de la funcionalidad Registrar.



4.3.3 Usuario Principal

Cuando el paciente se loguea, la aplicación comprueba si hay conexión a Internet. En el caso de haber conexión, el programa accederá a la base de datos externa que utiliza el programa, cogerá los datos del paciente, se introducen en la base de datos interna y se muestra el layout de la clase UsuarioPrincipal. En el caso de no conexión, el programa busca en la base de datos interna si en la tabla users se encuentra algún paciente con los campos de Login y Password introducidos. Apuntar que para que un usuario esté en la base de datos interna, al menos una vez, el paciente se ha tenido de loguear con conexión a Internet para que se puede realizar el traspaso de la base de datos externa a la interna.

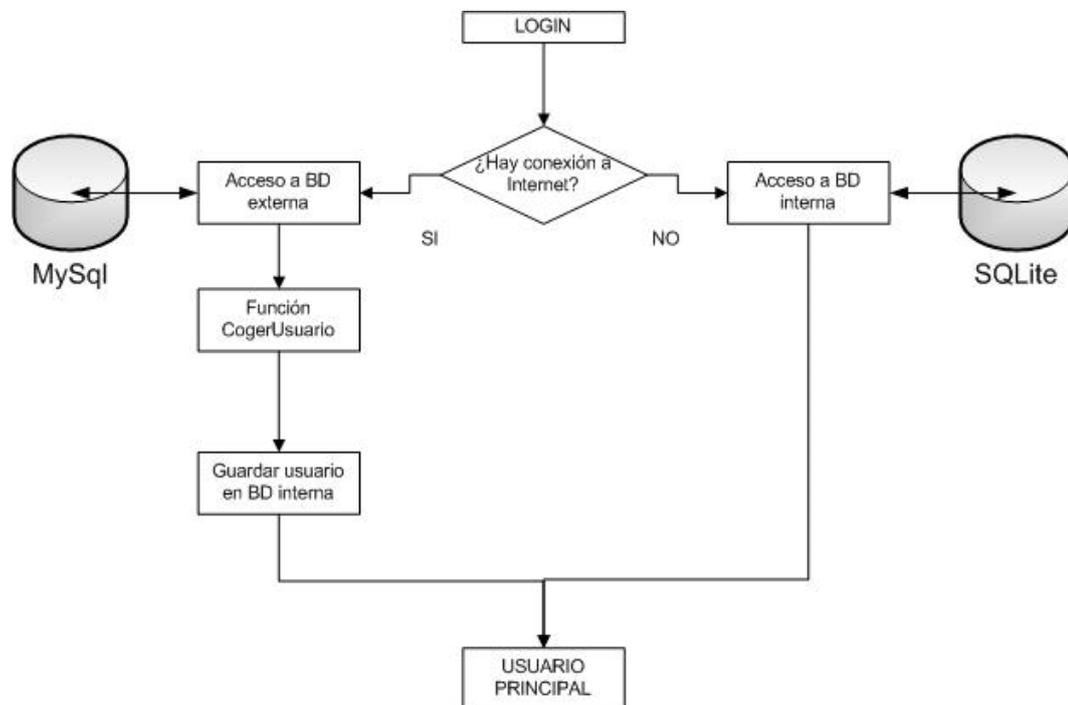


Figura 27. Diagrama de flujo de la funcionalidad UsuarioPrincipal.

Cuando se accede como paciente a la aplicación, aparece el layout de la clase UsuarioPrincipal. En esta pantalla de la aplicación, el paciente puede acceder a la vista de las clínicas que están registradas en la base de datos y puede reservar cita en cualquiera de ellas. El paciente también puede ver los ejercicios que el fisioterapeuta le ha asignado y tiene un botón que permite al paciente ver las reservas que tiene pendientes. Por último el paciente puede cerrar sesión y será llevado a la pantalla de Login.



CAPÍTULO 4

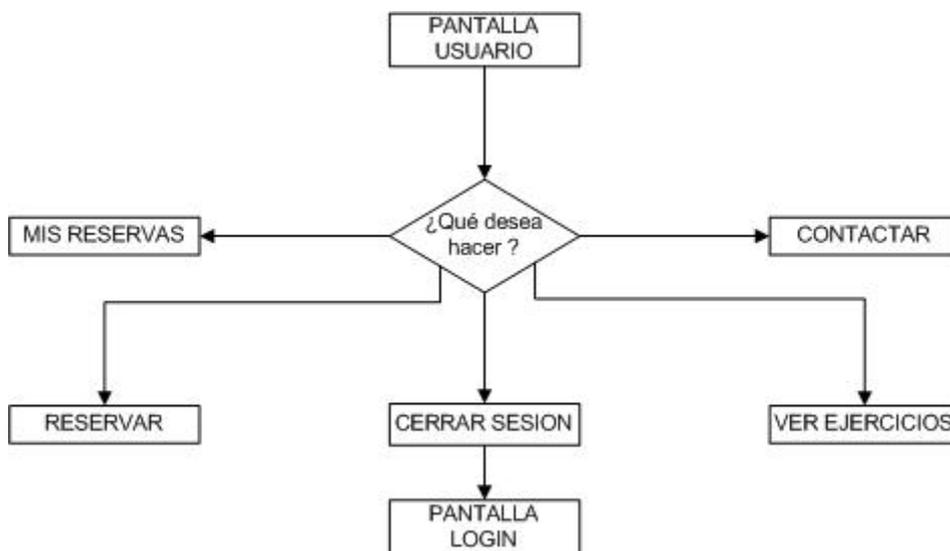


Figura 28. Diagrama de flujo de la clase UsuarioPrincipal.

4.3.4 Ver Clínicas

En la siguiente figura se muestran las clases derivadas de VerClinicas. Cuando el paciente pulsa en el botón ver clínicas, aparece un layout con la lista de todas las clínicas que usan la aplicación. El paciente puede seleccionar cualquiera de ellas y se muestra el layout de la VistaClinica. En este apartado el paciente puede realizar diversas acciones (llamar a la clínica, como llegar a la clínica, visitar la página web, ver la lista de tratamiento de la clínica, compartir la información de la clínica o volver al usuario principal). Todas las acciones exceptuando la de volver y la lista de tratamientos son ejecuciones por defecto de Android y no es necesario la creación de una clase para realizarlas.

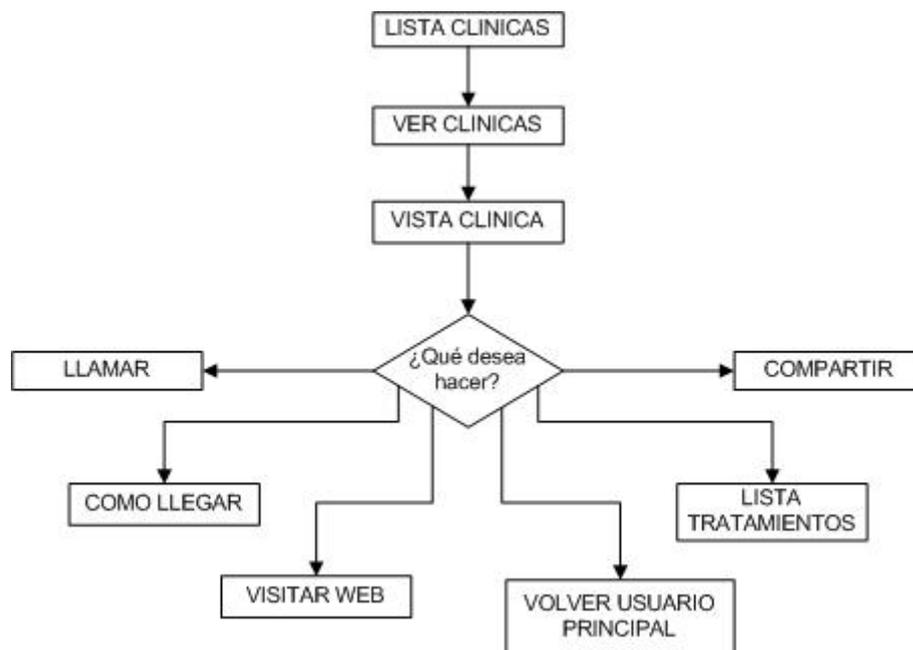


Figura 29. Diagrama de flujo de la clase VerClinicas.

En la siguiente figura se muestra el diagrama de funcionalidad de la clase VerClinicas. Cuando el paciente accede a esta clase, el programa realiza una comprobación de la conexión como la mencionada para el login mencionada en el apartado 4.3.3. Si hay conexión, el programa accede a la tabla clinics de la base de datos externa y copia toda la tabla de clinics en la base de datos interna. En el caso de no haber conexión, la aplicación consulta la tabla clinics de la base de datos interna, como antes, para que la tabla de la base de datos interna este rellena, previamente al menos una vez el usuario ha tenido que realizar una petición de ver clínicas con conexión a Internet para que la tabla clinics de la base de datos interna tenga datos.

Una vez recogido los datos de la tabla clinics, se comprueba si la tabla contiene alguna clínica. En el caso de no haber ninguna clínica se mostrará por pantalla un mensaje de que no hay clínicas disponibles. Si hay clínicas en la base de datos aparecerá una pantalla con la lista de las clínicas, la cual el usuario puede seleccionar cualquiera de ellas, y seguidamente una vez seleccionada una de las clínicas de la lista, se mostrará la información de la clínica con el layout de la clase VistaClinica.



CAPÍTULO 4

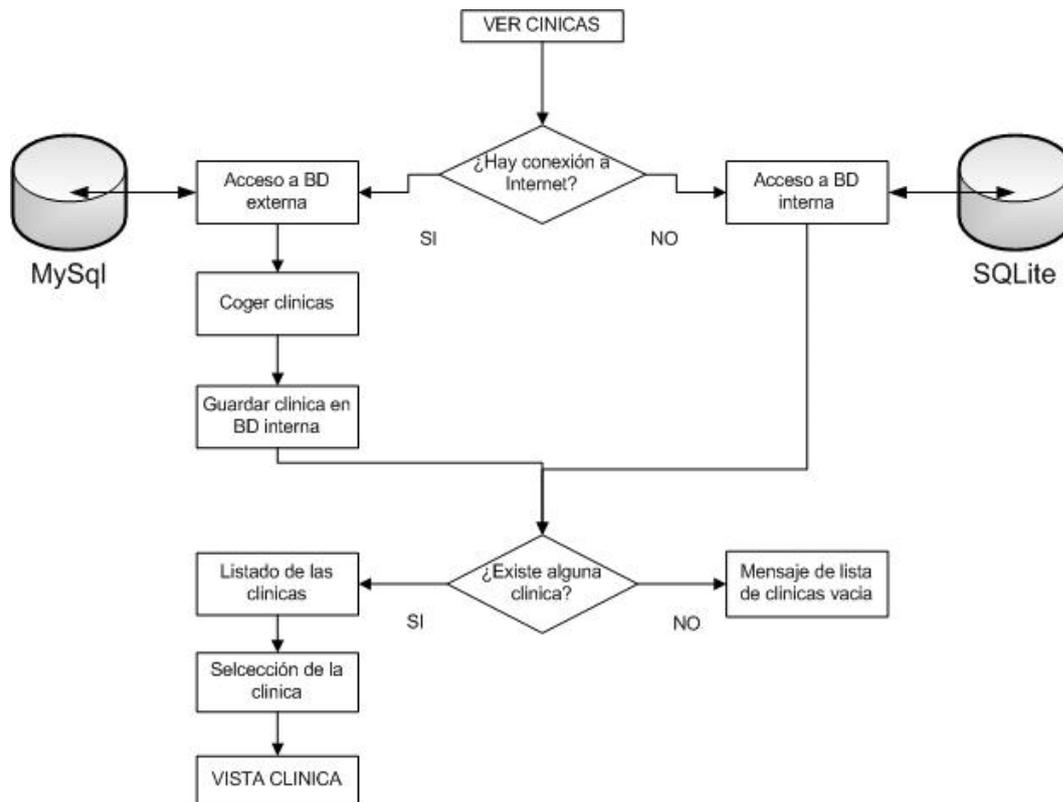


Figura 30. Diagrama de flujo de la funcionalidad VerClinicas.

4.3.5 Lista de Tratamientos

Para llegar a esta clase, se accede a través del menú de la clase VistaClinica detallado en el apartado 4.3.4. Cuando el paciente pulsa el botón del menú se muestra el layout de la clase VerTratamientos donde aparecerá una lista con los tratamientos de la clínica y al pulsar en uno de ellos la aplicación muestra la vista de ese tratamiento. Este layout es informativo para el paciente y tiene dos opciones en el menú, se puede compartir el tratamiento o volver al layout de la clase UsuarioPrincipal.

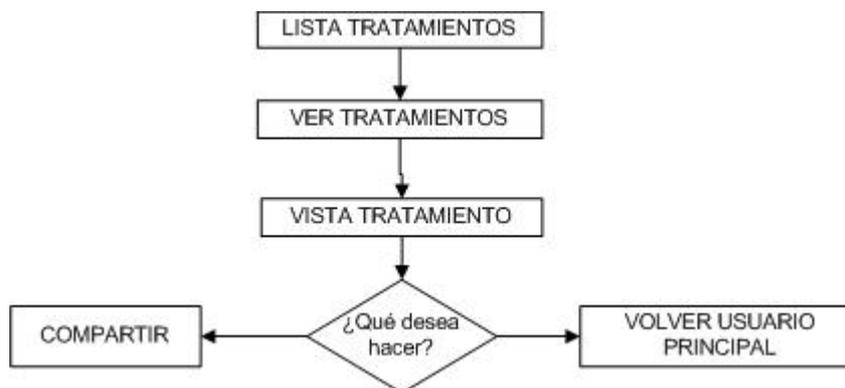


Figura 31. Diagrama de flujo de la clase VerTratamientos.

En esta figura se explica la funcionalidad de la opción del menú del listado de tratamientos desde la clase VistaClinica. Cuando el paciente pulsa esta opción del menú, la aplicación realiza de nuevo la comprobación para ver si hay conexión a Internet o no. En el caso de que sí haya conexión, se procederá a recoger los valores de las tablas `clinics_treatments`, `clinics` y `treatments`, y posteriormente se añadirán a la base de datos interna. Si no hay conexión, se mostrarán los datos que haya en las tablas correspondientes de la base de datos interna. Igual que en los casos anteriores, para que la base de datos tenga alguna de esas tablas rellenas, previamente el paciente ha tenido que realizar esta acción.

Una vez recogidos y almacenados los datos, se comprueba si en la tabla `clinics_treatments` hay alguna relación entre la clínica seleccionada por el paciente y los tratamientos. En caso de no haber ninguna relación se mostrará un layout advirtiendo de que no hay ningún tratamiento. Por el contrario, si existe alguna relación, se mostrará el listado de los tratamientos, y al seleccionar cualquiera de ellos se mostrará el layout de la clase VistaTratamiento.



CAPÍTULO 4

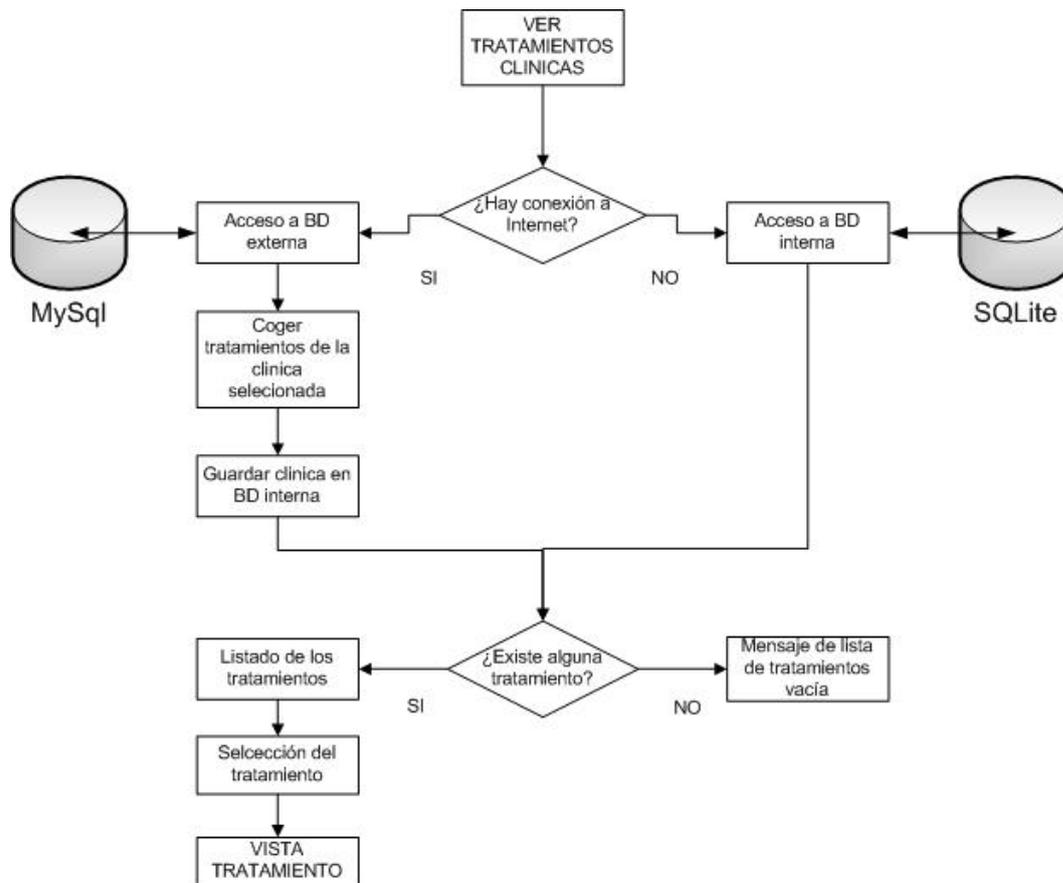


Figura 32. Diagrama de flujo de la funcionalidad VerTratamientosClinicas.

4.3.6 Reservar

En la siguiente figura se detalla la estructura de clases cuando el paciente accede a la clase Reservar. Esta estructura es muy similar a la de ver clínicas pero tiene cambios en las opciones del menú que se ofrecen al paciente. Cuando el paciente selecciona una clínica de la lista se accede al layout de la clase VistaClinicaReserva. Es en este punto donde se diferencia del apartado 4.3.4 ya que el menú muestra las opciones de reservar cita en la clínica seleccionada, contactar con la clínica vía gmail y la opción de volver al layout de la clase UsuarioPrincipal.



Figura 33. Diagrama de flujo de la clase Reservar

Cuando el paciente selecciona en el menú la opción Reservar Cita, se abre el layout de la clase ComentarioDolorReserva en el que el paciente escribe el lugar del dolor y un comentario acerca de su dolencia, en los siguientes layouts el paciente podrá elegir tanto el día como la fecha de la reserva. Por último, en el layout de la clase DateDayActivty, se muestra un resumen de la reserva y se consultará si la fecha y hora elegidas están disponibles, en caso de no estar disponible se le ofrecerá al paciente otra hora alternativa que será la siguiente hora disponible en esa clínica más cercana a la que el usuario haya sugerido. En el caso de que al usuario le convenga esa hora, se aceptará y se volverá al layout de UsuarioPrincipal, y en el caso de que el paciente no quiera la cita sugerida se le devolverá al layout de VistaClinicaReserva.



CAPÍTULO 4

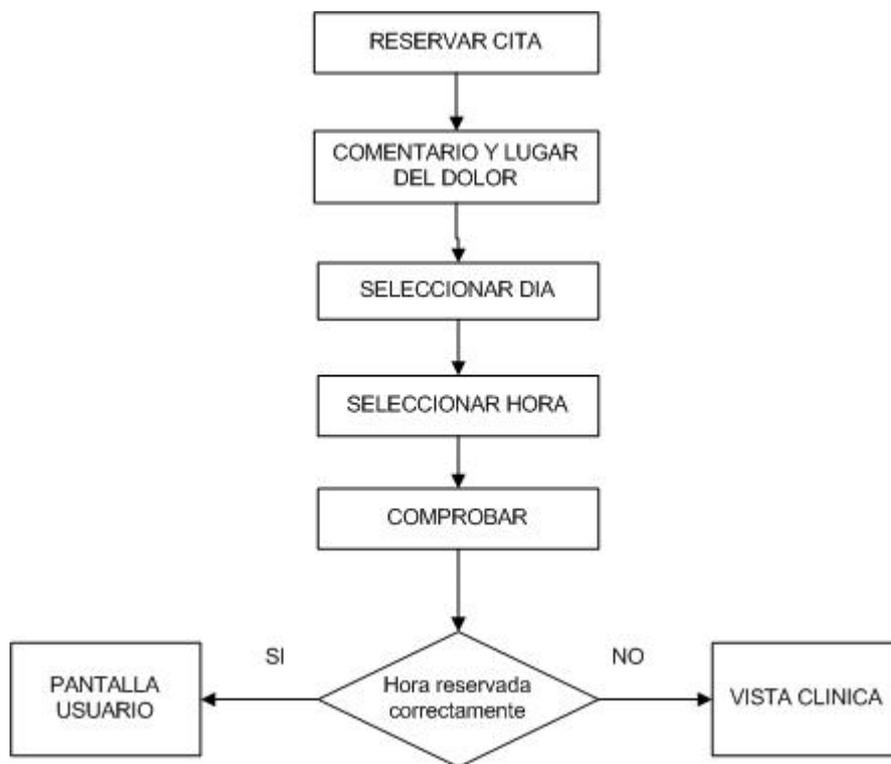


Figura 34. Diagrama de flujo de la clase ReservasCita.

Para terminar con la funcionalidad de reservar, con la siguiente figura se detalla más detalladamente el proceso que se sigue internamente cuando un usuario quiere reservar una cita.

Una vez ya en el layout de DateDayActivity, cuando el paciente ve el resumen de la reserva, al dar botón de Consultar, la aplicación recoge los datos de la fecha y la hora.

Primero comprueba si la fecha que ha introducido existe y si es posterior a la actual. En el caso de que no se cumpliera, se actualizan los valores a los actuales en el momento de la consulta.

Una vez comprobado el día, se comprueba que la clínica tenga definido su horario, en el caso de no tener definido el horario, se actualizan los valores de la clínica a los valores por defecto (10:00-14:00, 17:00-20:30).

Después se comprueba si el día escogido por el usuario cae en fin de semana (contando desde el viernes al cierre de la clínica) y en ese caso se llamaría a la función addOneDay las veces necesarias para poner el día en la siguiente hora disponible a partir del siguiente lunes.



Comprobado el día se pasa a mirar la hora, y la hora de la cita tendrá que estar dentro de la franja horaria de apertura y cierre de la clínica definida en los pasos previos. En el caso de que no entre en el rango se procederá al cambio de la hora al siguiente rango disponible con la consecuente comprobación de si es fin de semana.

Después, con la información recogida al principio del archivo sobre la clínica seleccionada, se crea un array con slots en función del horario de apertura y cierre de la clínica y el tiempo que conlleva cada tratamiento. Este array fijará las horas que se pueden reservar y cada cuanto tiempo.

A continuación se genera otro array de slots en el que se ponen a uno las horas que están ocupadas. Esto también va en función del número de tratamientos que la clínica pueda realizar a la vez. Por ejemplo si la clínica cuenta con tres fisioterapeutas, a una misma hora, podrán reservar en esa misma clínica 3 pacientes diferentes, y hasta que no se llenen las tres, el slot no se pondrá a 1, con lo cual indicará que sigue libre al menos una cita más en esa hora.

Esta última funcionalidad se hace gracias a un bucle en función de la variable NumTr que está definida en el apartado 4.1 cuando se mencionan las variables de la base de datos. Por último, si según la información del array, hay sitio libre, se procede con la reserva de la cita, actualizando la tabla reservations en la base de datos; y si por el contrario no está disponible, se procede a incrementar la hora o el día en caso de que fuera necesario.



CAPÍTULO 4

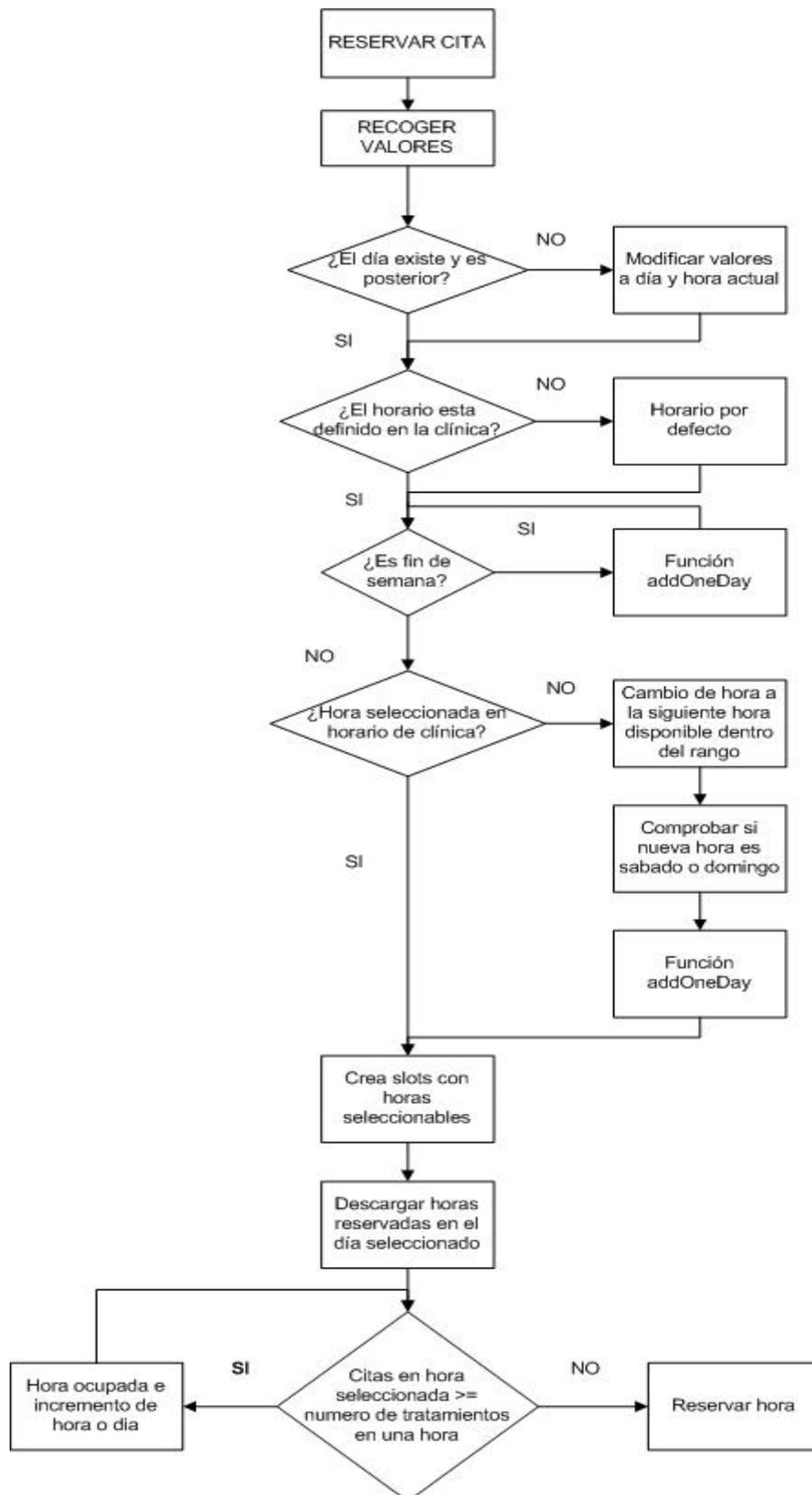


Figura 35. Diagrama de flujo de la funcionalidad ReservarCita.



4.3.7 Lista de Ejercicios

En este diagrama, se ven las clases derivadas del layout de VerEjercicios. Cuando el paciente desde el layout de UsuarioPrincipal pica sobre el botón ejercicios, se abre el layout de la clase VerEjercicios, en el cual se muestra una lista con los ejercicios que su fisioterapeuta le ha mandado después de la cita previa que hayan tenido.

Cuando el paciente selecciona uno, se abre el layout de VistaEjercicio, donde se puede ver la descripción del ejercicio, acompañado de una foto, y un comentario personal que el profesional de la clínica quiere resaltar al paciente para que lo tenga en cuenta a la hora de realizar dicho ejercicio.

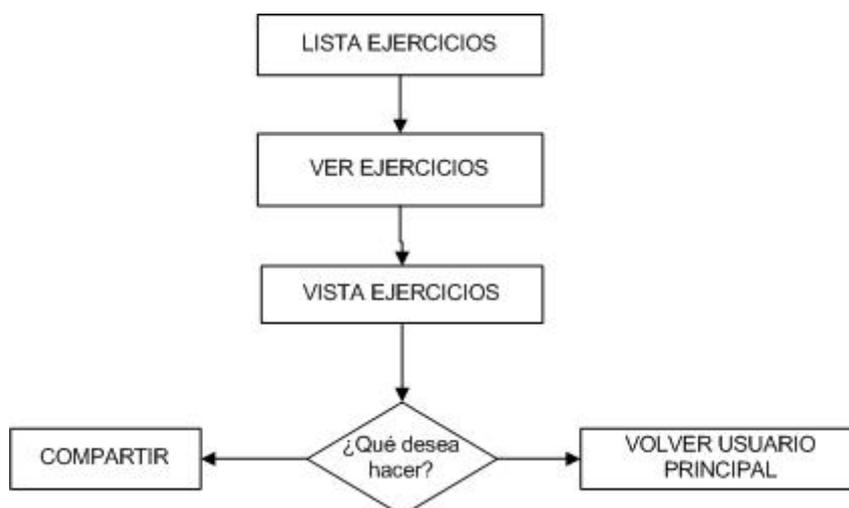


Figura 36. Diagrama de flujo de la clase VerEjercicios.

En este último diagrama, se aprecia cómo funciona la clase VerEjercicios en conjunto con las demás clases que derivan de ellas.

Al igual que en algunos caso anteriores, se comprueba si hay conexión y si la hay se cogen los valores de las tablas users, users_exercises y exercise y se guardan en la base de datos internas para en el caso de que no haya conexión en algún momento, se pueden ver los ejercicios descargados de la última vez que hubo conexión.

Una vez comprobado esto, se mira si hay algún ejercicio adjudicado al paciente que realiza la petición, en el caso de no haberlo se muestra un layout con un aviso de que la lista está vacía porque no hay ningún ejercicio asignado al paciente. Si por el contrario sí que hay algún ejercicio asignado se mostrará la



CAPÍTULO 4

lista de todos los ejercicios que el paciente tiene asignados y se podrá acceder a cualquiera de ellos para ver la foto, la descripción y el comentario que el fisioterapeuta le ha escrito al paciente para la correcta realización del ejercicio.

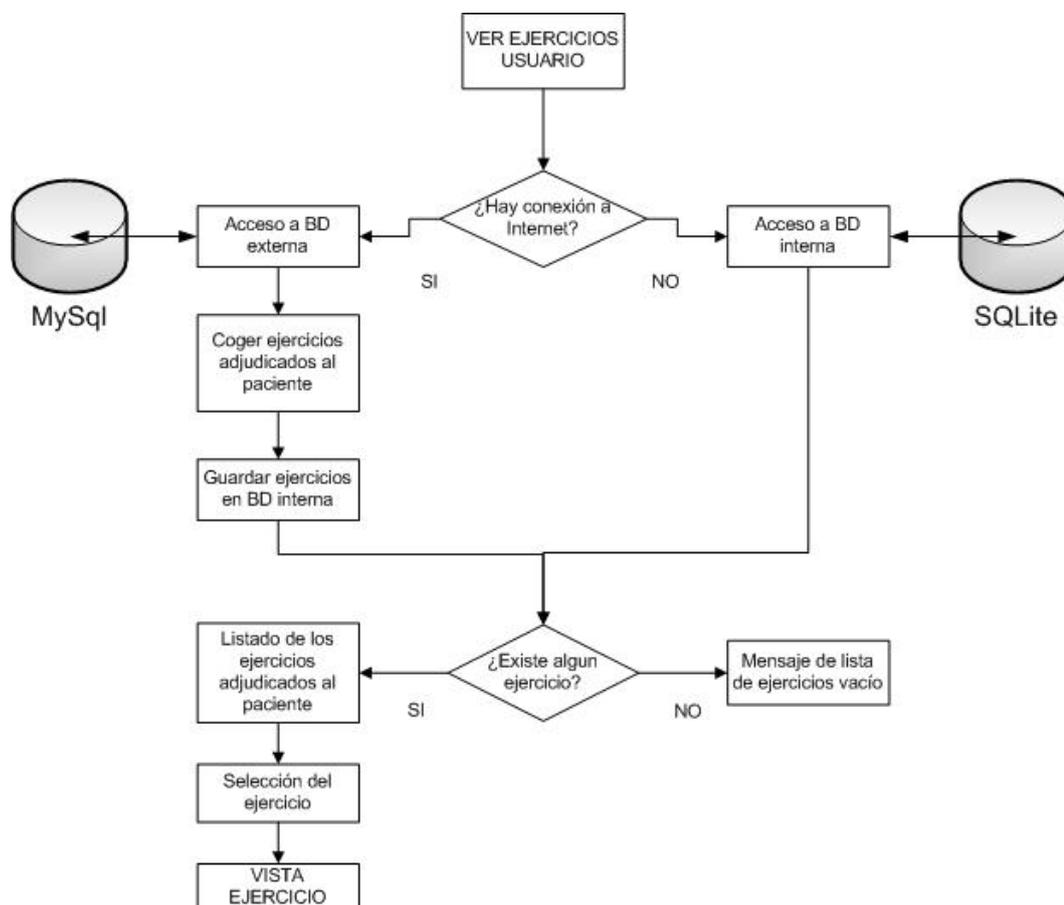


Figura 37. Diagrama de flujo de la funcionalidad de VerEjercicios.

4.3.8 Mis Reservas

En el diagrama se ven las clases derivadas de la clase de VerReservarUsuario. El paciente ve la lista con las reservas pendientes en ese momento y al clicar en una reserva se va al layout de la clase VistaReservaUsuario. En este último layout se puede volver al menú de usuario desde el menú del layout o anular la reserva. Tanto si se anula la reserva como no, se lleva al paciente al menú principal con la diferencia de si se ha seleccionado SI o NO a la hora de anular la reserva.

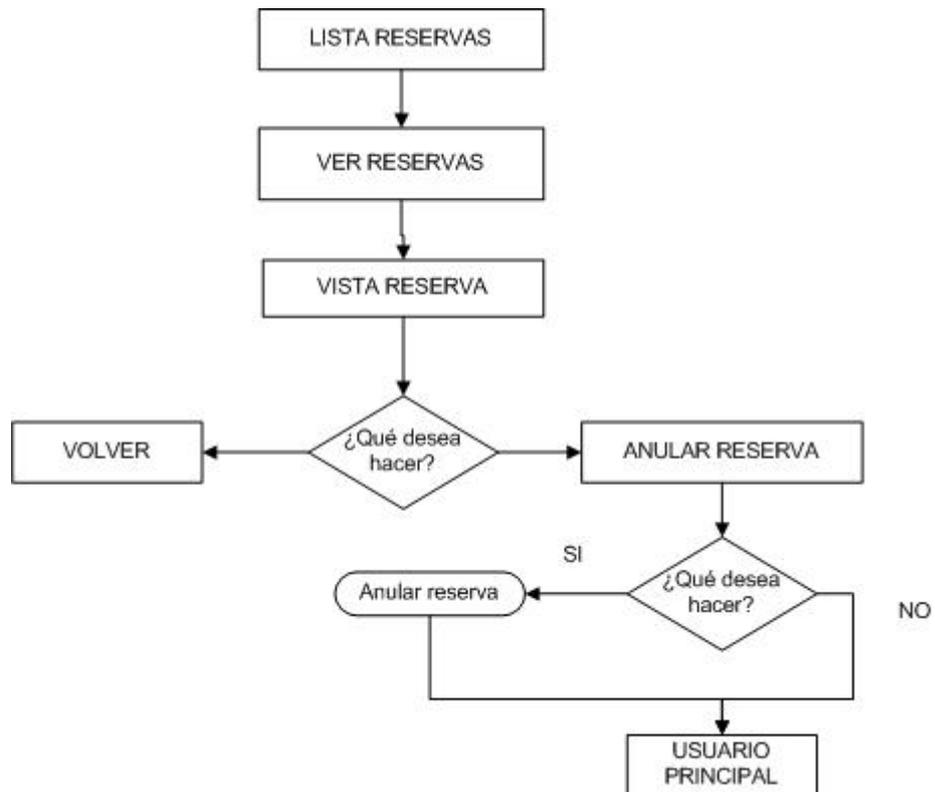


Figura 38. Diagrama de flujo de la clase VerReservasUsuario.

En el siguiente diagrama se ven las funcionalidades de la clase VerReservasUsuario. Se accede a la base de datos para descargar las reservas del usuario a las que tiene que acudir. Se guardan en la base de datos interna y se muestran por pantalla. Cuando se pulsa sobre una reserva se la a la vista de la reservar en la cual se puede anular la reserva o no.



CAPÍTULO 4

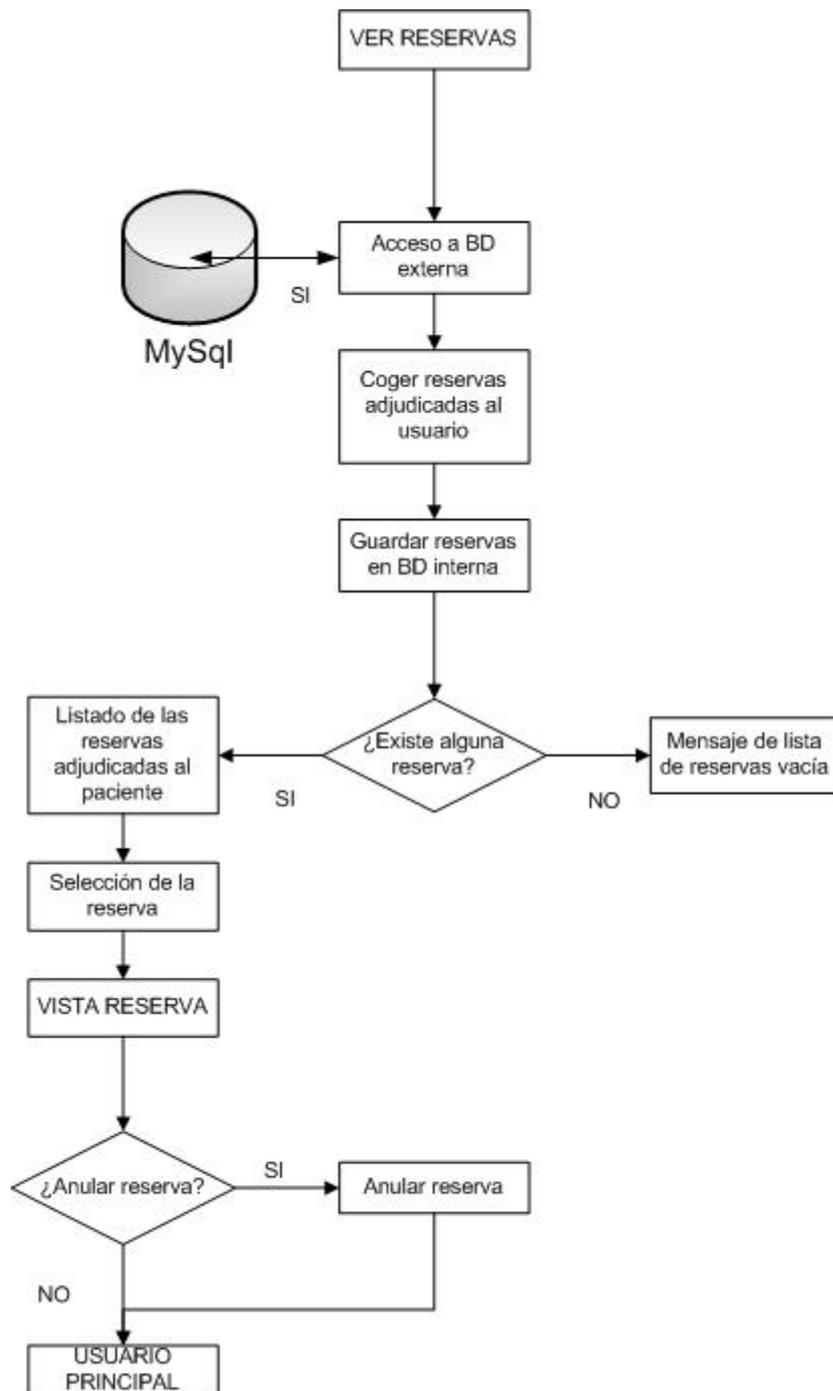


Figura 39. Diagrama de flujo de funcionalidad de VerReservasUsuario.

Capítulo 5: Manual de Usuario

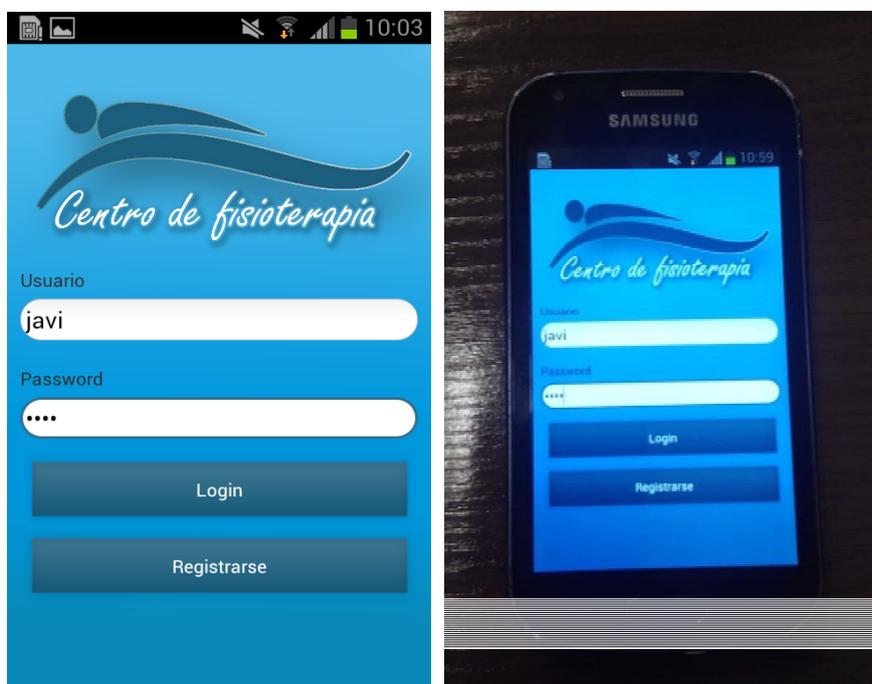
En este capítulo se va a comentar la guía del usuario para que se pueda realizar un buen uso de la aplicación. El manual se centrará en la parte del paciente el cual con ayuda de diversas imágenes tomadas directamente del dispositivo móvil, ayudará al paciente a manejar la aplicación. Se irán mostrando capturas de pantalla de la aplicación y debajo de cada captura un texto explicativo del layout que se visualizaría por parte del paciente en esa parte de la aplicación.

Constará de varias partes entre las cuales estarán el login, el registro, clínicas, reservas, lista de ejercicios y una sección para contactar con atención al cliente.

5.1 Login

Este es el layout principal que se muestra al usuario de la aplicación en cuanto se inicia la aplicación. Para que el paciente se pueda loguear se requiere la información de usuario (login) y contraseña (password).

Una vez introducido los campos, el usuario deberá presionar el botón LOGIN para que el programa pueda proceder con la acción de autenticación.



Layout 1. Login.



CAPÍTULO 5

Cuando el paciente se logea con un usuario que no está registrado previamente en la base de datos (o usuario o contraseña están mal escritas), la aplicación muestra un mensaje de error “Error: Nombre de usuario o password incorrectos” y se volverá a pedir los datos de autenticación de usuario.



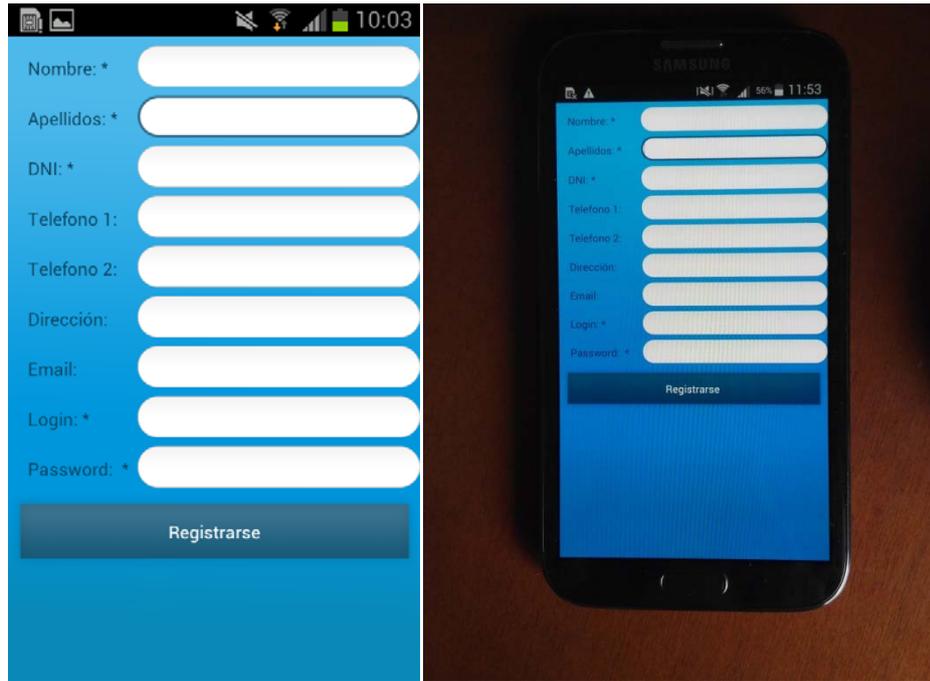
Layout 2. Login incorrecto.

5.2 Registro

Para que un paciente pueda registrarse como nuevo usuario de la aplicación, desde el layout de login se deberá pulsar en el botón REGISTRARSE.

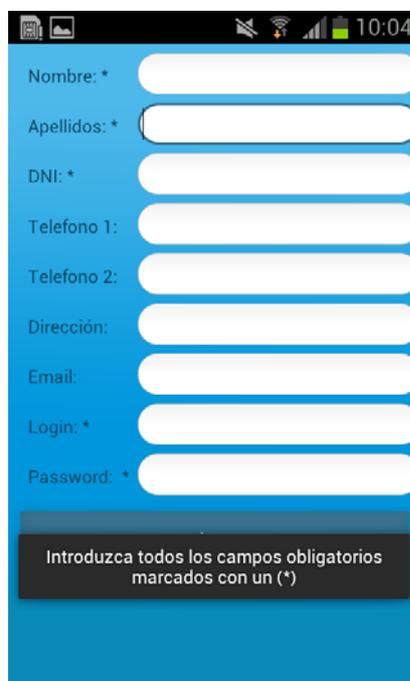
Una vez pulsado se mostrará una nueva pantalla en la que se piden varios campos para que se pueda realizar el registro adecuadamente (Nombre, Apellidos, DNI, Telefono 1, Telefono 2, Dirección, Email, Login y Password).

Cuando el paciente haya sido registrado en la base de datos, se volverá a mostrar el layout de Login, para que el usuario pueda acceder a los servicios de la aplicación como usuario.



Layout 3. Registro

A la hora de proceder con el registro, el paciente deberá introducir los campos marcados con un asterisco que indican que esos son los campos indispensables para poder realizar el registro en la base de datos de la aplicación (Nombre, Apellidos, DNI, Login y Password).



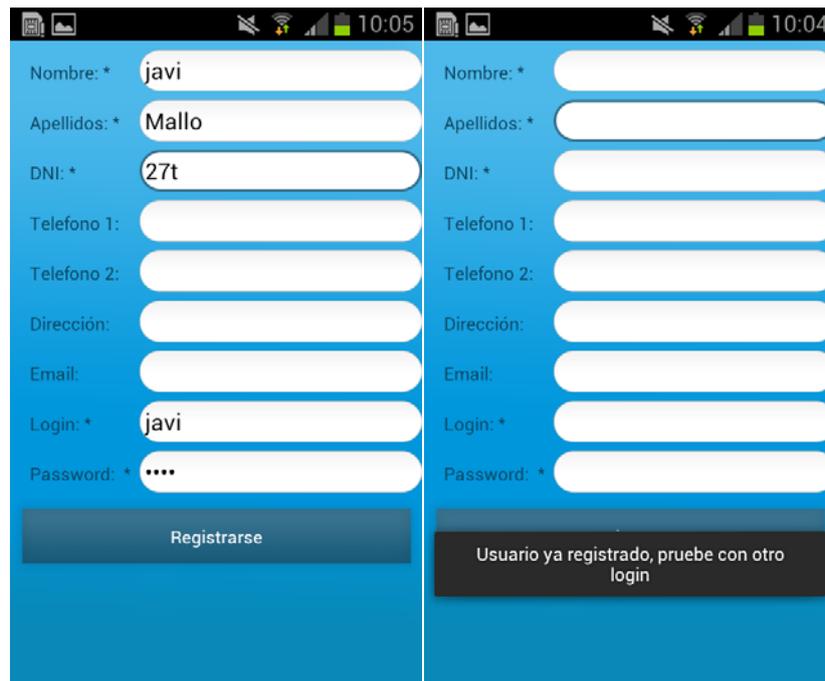
Layout 4. Registro sin campos obligatorios.



CAPÍTULO 5

Si el paciente intenta registrarse con un login ya existente, la aplicación mostrará un mensaje de error “Usuario ya registrado, pruebe con otro login”.

Esto avisa al usuario que debe escoger otro nombre de usuario (Login) para poder realizar el registro en la base de datos de la aplicación.



Layout 5. Registro usuario ya existente

5.3 Usuario Principal

En la siguiente imagen se muestra el layout de la clase UsuarioPrincipal que se mostrará cuando el paciente haya iniciado sesión correctamente en la aplicación.

Se muestran cinco botones, el primero es el botón de CLINICAS, que deberá pulsar el paciente para poder ver la información de las clínicas que tiene registradas el sistema.

El segundo botón es el botón RESERVAR, cuando el paciente quiera reservar una cita en laguna de las clínicas disponibles deberá acceder mediante este botón.

El tercer botón es el botón de EJERCICIOS, cuando el paciente haya acudido previamente a una cita con el especialista de una clínica, el fisioterapeuta



podrá mandar ejercicios personalizados al paciente una vez realizado el tratamiento si la situación así lo requiere. El paciente puede ver toda la información de los ejercicios mandados por el especialista en esta sección.

El cuarto botón que aparece en el menú es el de MIS RESERVAS, aquí el paciente podrá ver las reservas que tiene pendientes en cualquiera de las clínicas. Aquí se puede ver los detalles de la reserva (clínica, hora y fecha) y se podrá anular la reserva.

Por último, fuera del menú, arriba a la derecha, se encuentra el botón de CERRAR SESIÓN, junto con el nombre del usuario que se ha registrado. Cuando el paciente haya realizado las acciones y se quiera salir de la aplicación, u otro paciente quiera iniciar sesión con su cuenta, el paciente deberá pulsar sobre el icono de desconexión para que se muestre de nuevo el layout de la clase Login para poder volver a realizar el logueo o el registro.



Layout 6. Usuario_principal.

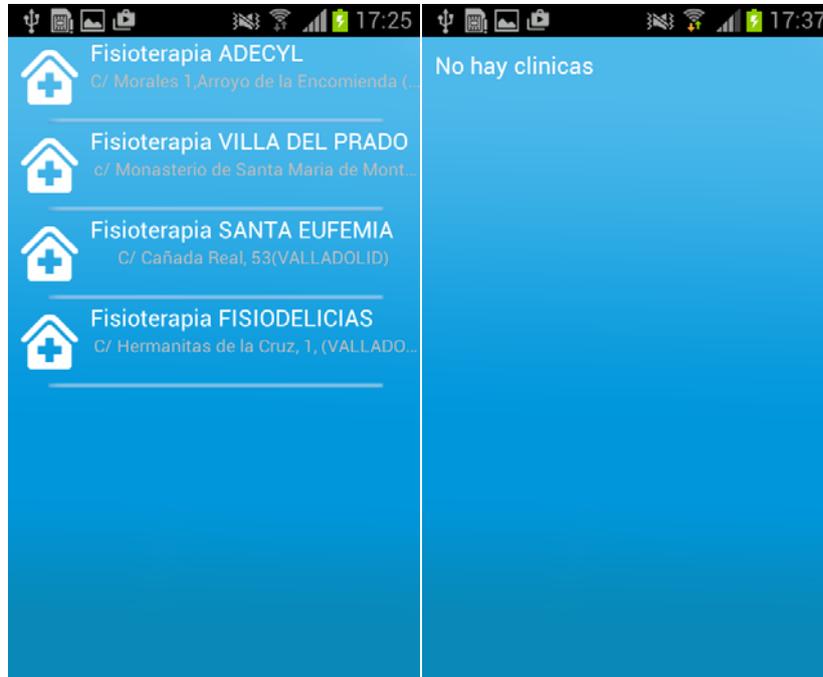
5.4 Clínicas

Cuando el paciente hace clic en el botón CLINICAS, se muestra la lista de las clínicas que están registradas en la base de datos de la aplicación, con una pequeña información sobre cada una de ellas como es el nombre de la clínica y la dirección.



CAPÍTULO 5

En el caso de que no haya ninguna clínica registrada, se muestra un aviso de que no hay ninguna clínica registrada y por lo tanto la lista aparece vacía.

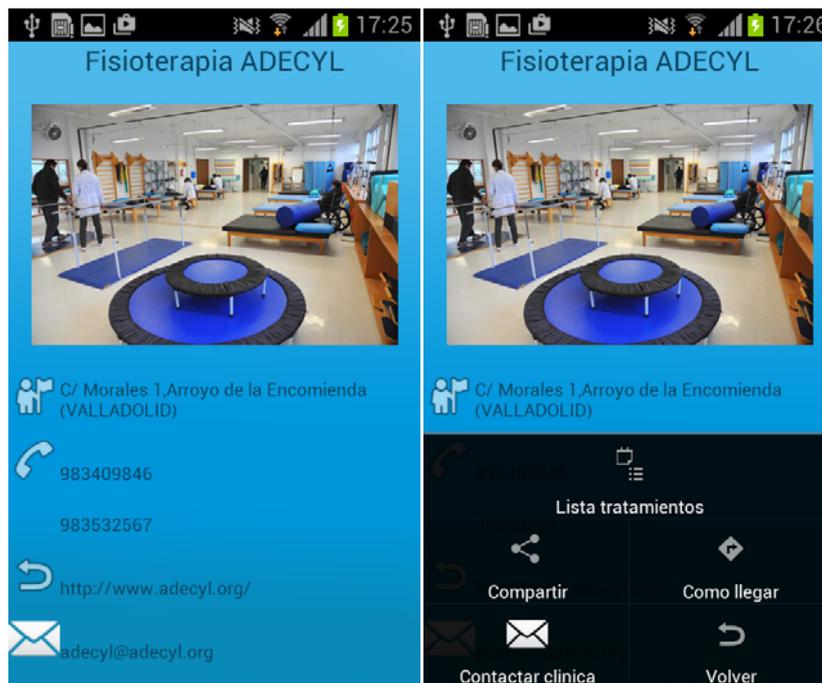


Layout 7. Ver_clinicas, lista rellena y lista vacía

Cuando el paciente selecciona una de las clínicas de la lista, se muestra por pantalla una vista de la clínica, en la que el paciente puede ver información como el nombre de la clínica, dirección, teléfonos, email, pagina web, horario y una breve descripción de la clínica.

Si el paciente pulsa sobre la información de la clínica, por ejemplo hace clic en el teléfono, la aplicación lanza la acción de llamar y permitirá al usuario llamar directamente a la clínica.

Si se pulsa sobre el botón del menú del móvil, aparecerá por pantalla un menú desplegable en el que se pueden realizar una serie de acciones como ver la lista de tratamientos que tiene la clínica seleccionada, compartir la información de la clínica, cómo llegar a la clínica, contactar con la clínica y volver al menú principal.



Layout 8. Vista_clinica, con visión del menú

A continuación se detallarán las pantallas que se muestran cuando el usuario hace clic en algunas de las opciones del menú anterior.

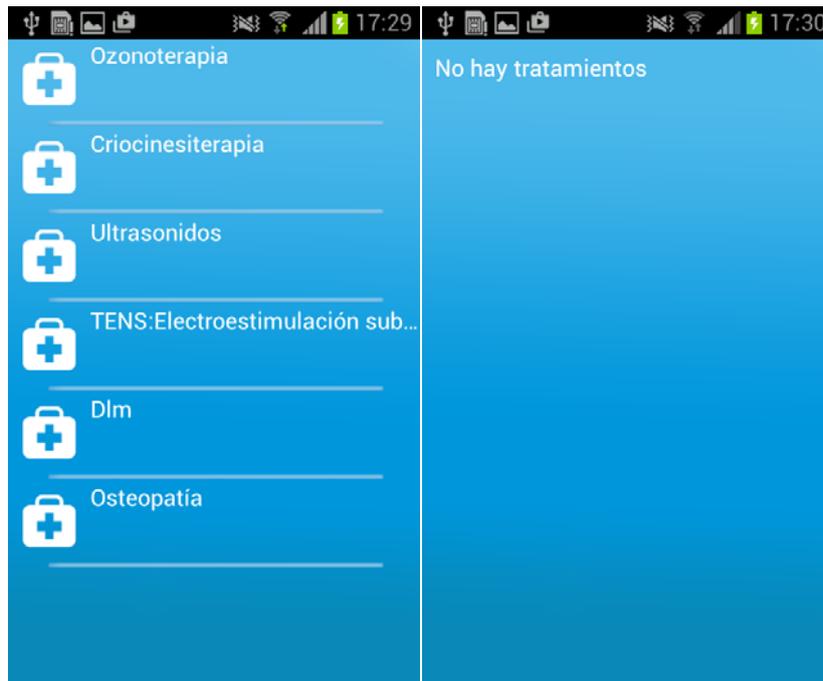
5.4.1 Lista de tratamientos

Cuando el paciente hace clic en el botón del menú de lista de tratamientos, se visualizará una lista con los tratamientos que se pueden realizar en esa clínica.

Al igual que con la lista de las clínicas de la aplicación, si la clínica que ha seleccionado el paciente no posee ningún tratamiento aparecerá un mensaje advirtiéndolo al usuario que la clínica no tiene tratamientos y se mostrará la lista vacía.



CAPÍTULO 5



Layout 9. Ver_tratamientos , lista rellena y lista vacía.

Si el usuario pulsa sobre cualquier tratamiento de la lista, se mostrará por pantalla la información del tratamiento seleccionado (nombre, foto y descripción).

Esta pantalla también consta de un menú al que se accede al pulsar el botón del menú del teléfono que da dos opciones al usuario.

La primera opción es COMPARTIR, que como se puede ver en la imagen de la derecha, el paciente podrá compartir la información del tratamiento de la manera que desee.

La segunda opción es la de VOLVER, que permitirá al usuario con un solo clic volver al menú principal del usuario.



Layout 10. Vista_tratamiento, con opciones de menú y vista opción compartir.

5.4.2 Cómo llegar

Para la opción de cómo llegar, se utiliza el servicio del programa Google Maps. Como se ha mencionado anteriormente, para llegar a esta opción se puede llegar de varias formas, haciendo clic sobre la calle en la vista de la clínica, o desde el menú desplegable haciendo clic en COMO LLEGAR.

Seguidamente se abrirá la aplicación seleccionada para la visualización del mapa que en este caso es Google Maps y como se puede observar en la imagen que se muestra a continuación se visualiza la localización del paciente con una flecha azul, y sobre el mapa la calle escrita correspondiente a la clínica seleccionada. Para poder visualizar el recorrido el usuario lo único que tiene que hacer es hacer clic en la parte inferior donde como “Cómo llegar” y el programa le guiará a través del mapa.



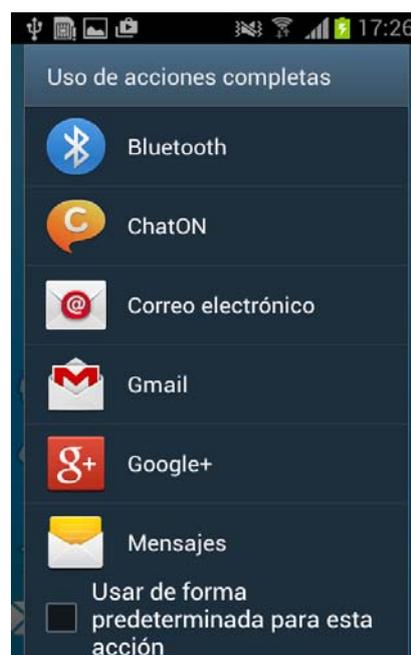
CAPÍTULO 5



Layout 11. Cómo llegar a la clínica, usando Google Maps.

5.4.3 Compartir

Esta opción del menú tiene la misma funcionalidad mencionada anteriormente en el apartado de lista de tratamiento, pero en vez de compartir la información de un tratamiento, el paciente podrá compartir con quien desee la información de la clínica.

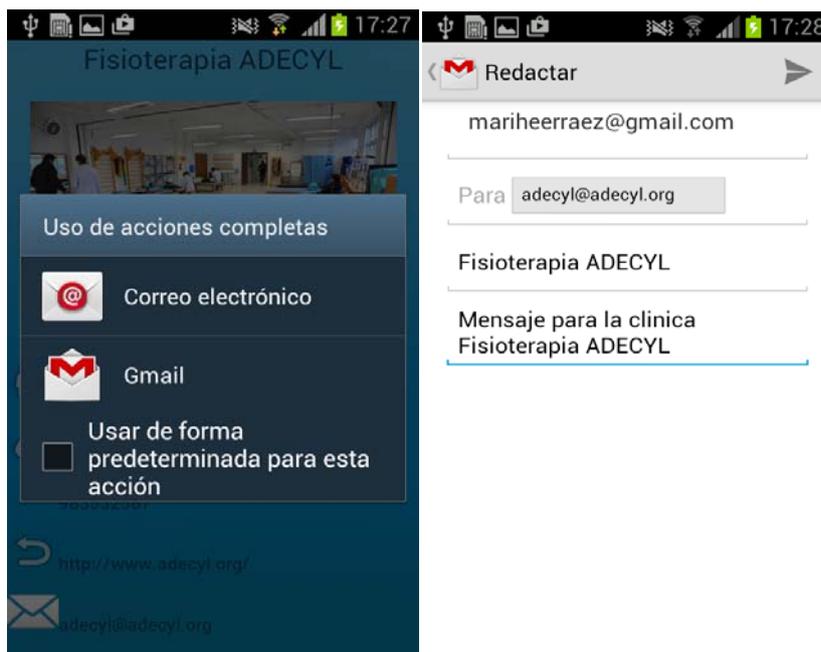




Layout 12. Compartir clínica.

5.4.4 Contactar con la clínica

Cuando el paciente haga clic en la opción del menú contactar, se mostrarán las opciones de servidores de correo que tenga el teléfono instaladas y se abrirá la aplicación, con los campos rellenos de para, asunto y una referencia de la clínica a la que se va a enviar el mensaje al principio del mismo.



Layout 13. Contactar con la clínica

5.5 Reservar

A continuación se detallará cómo funciona el botón RESERVAR. Cuando se hace clic sobre este botón, se muestra por pantalla una lista de clínicas (en caso de que haya clínicas en la base de datos) y el paciente puede seleccionar cualquiera de ellas.

Una vez seleccionada una clínica, se accederá a la vista de la clínica. Para poder reservar es preciso pulsar sobre el botón del menú del teléfono en el cual aparecerán tres opciones, volver al menú principal, contactar con la clínica mediante el servicio de correo y reservar la cita en la clínica selección.

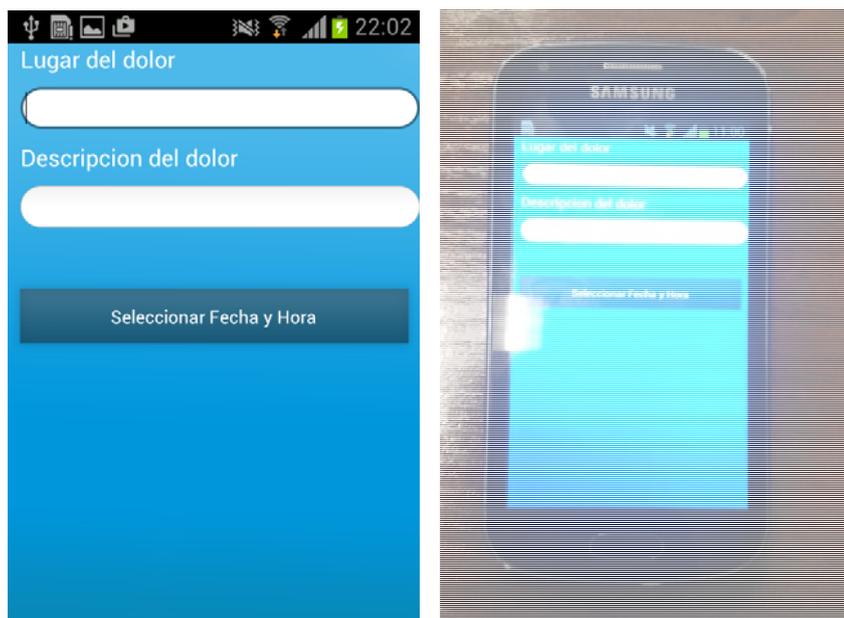


CAPÍTULO 5



Layout 14. Ver_clinicas y vista_clinica_reserva.

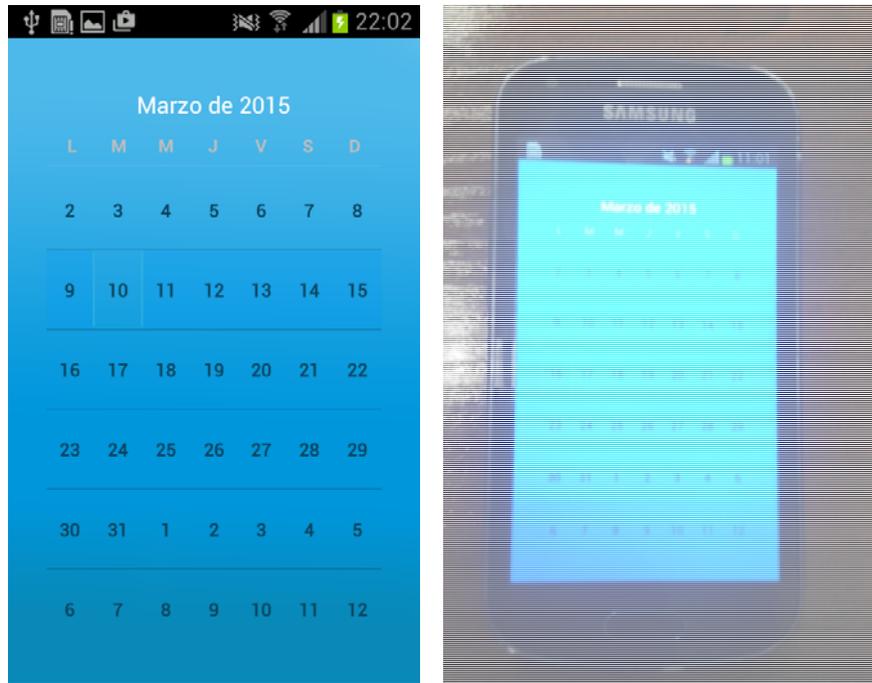
Para que el paciente pueda proceder con la cita deberá presionar sobre el botón del menú RESERVAR CITA. Una vez presionado se mostrará por pantalla dos cuadros de texto editables en los cuales el paciente deberá proporcionar la información para el especialista informando del lugar del dolor y un pequeño comentario sobre como es el dolor o cualquier cosa que ayude al especialista a entender mejor su situación. Una vez hecho esto se deberá pulsar el botón SELECCIONAR FECHAY Y HORA.



Layout 15. Comentario y lugar del dolor.



Una vez presionado el botón, se mostrará por pantalla un calendario en el cual el paciente deberá seleccionar el día que quiere acudir a la cita con el especialista.



Layout 16. Selección del día de la cita.

Cuando el usuario selecciona el día, la siguiente pantalla que se le muestra al usuario es la de selección de la hora. Aquí el usuario por medio del timepicker puede seleccionar la hora (hora y minutos) a la que quiere acudir a la cita.

Una vez ajustada la hora deberá pulsar el botón SELECCIONAR HORA para poder visualizar la siguiente pantalla.

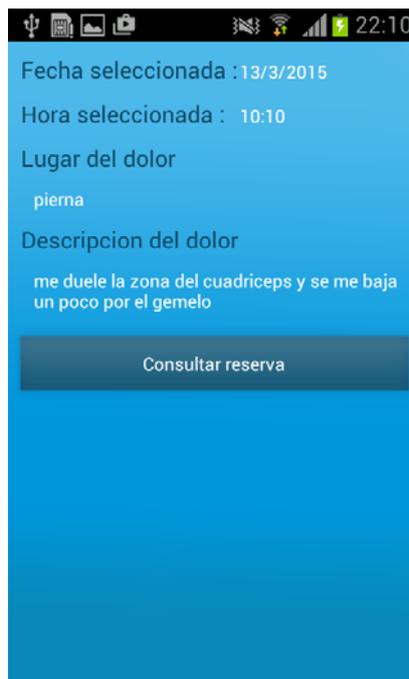


CAPÍTULO 5



Layout 17. Selección de la hora de la cita

Cuando el paciente haya completado todos los pasos, aparecerá por pantalla un resumen de la reserva con todos los datos que ha ido proporcionando el paciente. Cuando revise todos los datos deberá pulsar el botón CONSULTAR RESERVA para que la aplicación le muestre si hay hueco en el día y la hora seleccionada.



Layout 18. Resumen de la reserva.



Por último al pulsar el botón, se muestra por pantalla la hora de la cita disponible, que puede ser la misma que el usuario ha escogido u otra diferente en función de las reservas que tenga la clínica en ese momento.

Se muestran dos opciones, si el paciente está conforme con la hora que se le ofrece deberá pulsar OK y la aplicación le mostrará menú principal. Si por el contrario no está conforme con la hora seleccionada se mostrará de nuevo la vista de la clínica seleccionada para que pueda iniciar al proceso de reserva de nuevo.



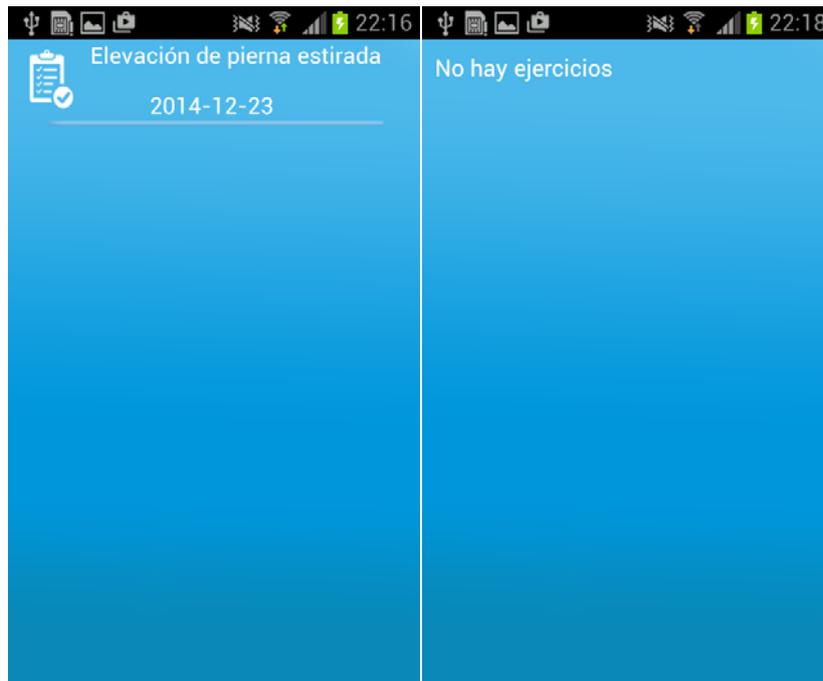
Layout 19. Consulta de la hora disponible.

5.6 Lista de ejercicios

Para acceder a la lista de ejercicios adjudicados el paciente debe pulsar el botón EJERCICIOS desde el menú principal de usuario. Seguidamente aparecerán por pantalla, en caso de haber algún ejercicio asignado, la lista de ejercicios del paciente, mostrando la información del nombre del ejercicio y la fecha de cuando el especialista le ha asignado el ejercicio.

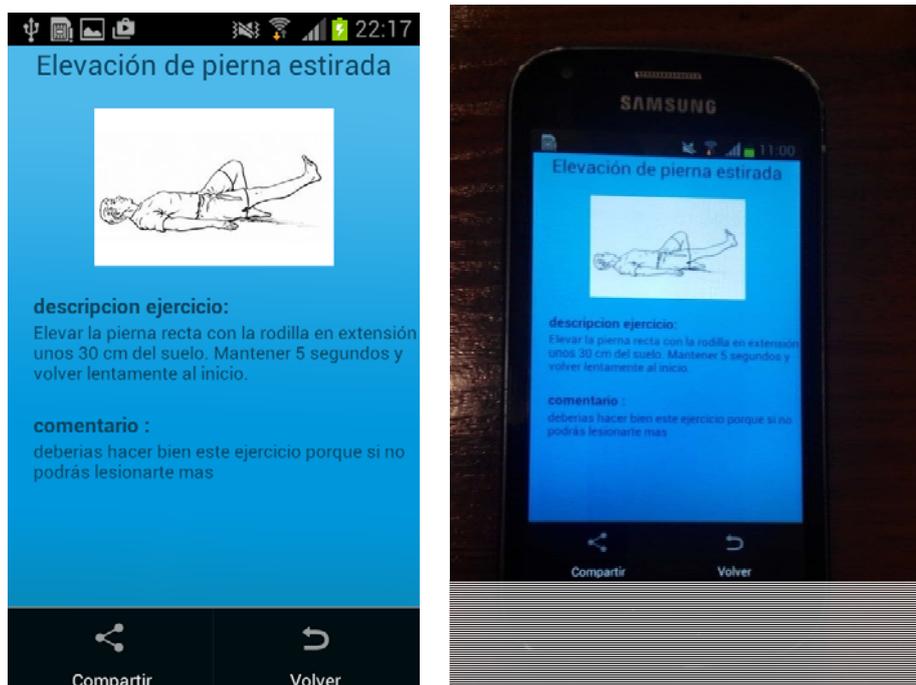


CAPÍTULO 5



Layout 20. Lista de ejercicios adjudicados al paciente y lista vacía.

Cuando el paciente haga clic en uno de los ejercicios que le aparecen en la pantalla se muestra la información del ejercicio asignado. Por otra parte esta pantalla tiene un menú que permite compartir el ejercicio y volver al menú principal.



Layout 21. Vista del ejercicio y comentario del profesional.



5.7 Mis Reservas

Por último, el botón MIS RESERVAS del menú principal del usuario, permite al paciente ver sus reservas pendientes en cualquiera de las clínicas disponibles en la aplicación. Las reservas salen por orden de fecha. Una vez elegida una reserva el paciente podrá ver los detalles de la reserva.

Como opciones del menú sale la opción de volver y saldrá por pantalla el menú principal.



Layout 22. Vista de Mis Reservas

Cuando el paciente ve por pantalla los detalles de la reserva se muestra un botón que permite anular la reserva seleccionada. Cuando se presiona el botón sale un aviso por pantalla “¿Está seguro de anular la cita seleccionada?”, si se pulsa SI, la cita es anulada, saliendo un aviso por pantalla y se redirige a la pantalla del menú principal. Si por el contrario se elige NO, no se realizará la acción de anular y saldrá por pantalla el menú de inicio.



CAPÍTULO 5



Layout 23. Aviso y confirmación de la anulación de la cita

Capítulo 6: Presupuesto Económico

En este capítulo se estimará el cálculo del presupuesto económico que ha supuesto este trabajo. Se analizarán todos los factores que han influido en la aplicación y se calculará un presupuesto final.

Para la realización de este trabajo se ha hecho uso de programas de software libre y soporte informático ya adquirido previamente, esto ha implicado que el gasto inicial de la aplicación en material de trabajo haya sido nulo. Por lo que el único coste de la aplicación será el de la mano de obra.

Hay muchos factores que influyen en el coste de una aplicación. En primer lugar, se ven los parámetros más importantes que influyen en el precio de una aplicación:

- Complejidad de la aplicación: existen aplicaciones sencillas que se desarrollan en pocas horas y juegos o redes sociales complejas que requieren decenas de miles de horas de trabajo y un equipo amplio multidisciplinar.
- Tipos de contenidos: estáticos o que se puedan actualizar dinámicamente.
- Acceso a datos: ¿va a necesitar la aplicación conectarse a servidores para realizar búsquedas, actualizar su información en tiempo real y mostrar los resultados?
- Geoposicionamiento: ¿incluye información dependiente de la localización del usuario? Por ejemplo, se podría mostrar información de la tienda más cercana.
- Realidad aumentada: por ejemplo, ¿queremos mostrar una imagen en 3D de una televisión cuando estamos viéndola en un catálogo impreso?
- Pasarela de pago: la aplicación, ¿tiene opciones de pago o se utiliza para vender productos? ¿Necesita integrar el pago a través de las tiendas de apps u otras pasarelas de pago?
- Registro de usuarios: ¿es necesario incluir un registro de usuarios? ¿Qué datos se deben recoger?
- Envío de notificaciones push: en general, es interesante incluir la posibilidad del envío de mensajes personalizados a los usuarios a través de la aplicación que han descargado. ¿Es un requisito? ¿Qué tipo de notificaciones? ¿Es necesario crear una interfaz de gestión de notificaciones o se debe integrar con un sistema existente?



CAPÍTULO 6

- Integración con otros sistemas: la complejidad del desarrollo aumenta considerablemente en el caso de tener que integrar la aplicación con sistemas existentes de la empresa (gestor de contenidos, bases de datos, gestión de usuarios, envío de notificaciones push, etc.) y puede haber mucha variación en función del sistema existente.
- Diseño gráfico: evidentemente no es lo mismo un diseño sencillo con menús y páginas tipo ficha informativa que aplicaciones que incluyan juegos.
- Plataformas: el número de plataformas en las que deba funcionar la aplicación es determinante para calcular el coste.
- Aplicaciones nativas/híbridas: las aplicaciones se pueden hacer de forma nativa o de forma híbrida que permiten la generación de aplicaciones multiplataforma con un único desarrollo. En el caso de aplicaciones nativas se consigue una mayor calidad y rendimiento con un coste mayor, mientras que las aplicaciones híbridas ofrecen menor rendimiento pero el coste es sensiblemente inferior.
- Coste/hora del programador de apps móviles: el coste de una hora de programación puede variar desde los 20 € de un programador junior sin experiencia hasta los más de 120 € que cobran las consultoras por programadores experimentados.

En función de los factores descritos anteriormente y atendiendo a las características de la aplicación, se puede estimar que el coste total de la aplicación es de 6600 €, ya que se han estimado 264 horas de trabajo real de programación, en las que no se reflejan la fase previa de aprendizaje del lenguaje Android. Se ha tomado como referencia el salario por hora de un programador junior (entre 20 y 30 €/hora) y se ha fijado en 25 €/hora.

Capítulo 7: Conclusiones y líneas futuras

En este capítulo se abordan las conclusiones obtenidas del desarrollo de este Proyecto Fin de Carrera, así como algunas orientaciones para su perfeccionamiento en un futuro.

7.1 Conclusiones

El sector de las comunicaciones móviles y en concreto los dispositivos móviles están en una constante evolución que está creciendo exponencialmente y de manera vertiginosa. Gracias a estos numerosos avances, este sector de la tecnología móvil se está haciendo cada vez más importante en el día a día de las personas y adquiriendo un mayor peso en sus vidas. Todos estos nuevos elementos se están convirtiendo en elementos indispensables y cada vez aumenta la dependencia de ellos en la vida cotidiana.

A esto se le puede añadir el empeño de las compañías telefónicas de aumentar su número de usuarios para obtener mayores beneficios lo cual también favorece enormemente al desarrollo de nuevas aplicaciones y terminales cada más potentes y modernos, con un mayor número de funcionalidades que ayudan a un mejor entendimiento de las aplicaciones.

Sin duda, este desarrollo de las aplicaciones favorece enormemente al desarrollo de las aplicaciones del campos de la mHealth ya que cada vez cuentan con mejores sensores, más velocidad y mayor transparencia para el usuario, características que están propiciando su uso y su despegue dentro del campo de la salud móvil, que para el caso de las aplicaciones de la mHealth (eHealth) requieren un desarrollo y diseño específicos.

En este proyecto se ha realizado una primera versión de una aplicación móvil de fisioterapia para el paciente. El objetivo era realizar una implementación de varios servicios para que los pacientes pudieran interactuar con la aplicación, intercambiando información y servicios entre la aplicación móvil y el servidor web. Con la aplicación el usuario puede obtener información de las clínicas de fisioterapia apuntadas en la aplicación así como reservar cita para acudir la clínica que el paciente desee. También puede comunicarse tanto con la clínica como con el especialista y recibir los ejercicios que éste le manda después de la sesión.



CAPÍTULO 7

La aplicación se ha decidido hacer para el sistema operativo Android, ya que se realizó un estudio de mercado antes del inicio del proyecto y se vio que Android es el sistema operativo dominante y más usado actualmente. El aspecto económico a la hora de poder realizar la aplicación (medios, equipos...) también jugó un papel importante en la toma de la decisión.

Para esta primera versión de la aplicación se ha utilizado servicios web basado es REST para el intercambio de información con la aplicación ya que es relativamente sencillo, y era suficiente para lo que la aplicación requiere. Para la codificación de los datos se ha preferido usar JSON en vez de XML, aunque no hay una diferencia tan significativa, sí que es cierto que produce menos sobrecarga que XML, y con la ayuda de los tutoriales de Internet fue bastante fácil implementarlo con JSON.

En mi opinión, creo que lo más importante de la realización del proyecto es la experiencia que con ello he ganado. Gracias a ello he adquirido conocimientos nuevos sobre Android que a lo largo de la carrera no he aprendido. Creo que la elección de Android ha sido la mejor opción para el desarrollo del proyecto ya que es el sistema operativo que más abunda actualmente en los dispositivos móviles y que se utilice Java es algo muy beneficioso a la hora de encontrar trabajo como programador en el mundo laboral.

La realización del proyecto me abre muchas puertas dentro del campo de las aplicaciones móviles, y con vistas a seguir programando en un futuro ha sido un buen inicio ya que he podido comprobar cuál es mi capacidad de ser autodidacta. He ido adquiriendo los conocimientos necesarios para llevar a cabo la aplicación por mí mismo a base de cursos y tutoriales y gracias a que he resuelto los problemas surgidos he podido afianzar los conocimientos y asentando bien las bases del proyecto. La aplicación no tiene mucha envergadura pero sí que toca de una manera u otra casi todas las opciones que Android ofrece a los usuarios.

Por otro lado creo que es importante destacar el trabajo en grupo para el proyecto. He trabajado con dos compañeros Jairo Méndez y David Gómez lo cuál ha sido una experiencia muy gratificante a la vez que enriquecedora. Pienso que el trabajo en grupo es bastante importante a la hora de salir al mundo laboral y este proyecto ha sido una gran oportunidad para ver cómo reacciono antes las diferente opciones que mis compañeros me ofrecían y cómo resolver los problemas de la programación de la aplicación, incluso ayudar a mis compañeros con algún problema en su parte del proyecto. Son aspectos de los que en mi opinión salgo reforzado.



7.2 Líneas futuras

En este apartado se presentan las líneas futuras del proyecto de la aplicación Android de fisioterapia en el que se comentan los cambios que puede tener la aplicación en un futuro que puedan ayudar a dar más envergadura y sencillez a la aplicación.

Establecimiento de sesiones: El hecho de implementar los servicios web con REST ha hecho que el desarrollo haya sido bastante simple. Sin embargo, es característico de este tipo de servicios no recordar el estado de las comunicaciones, es decir, cada mensaje de petición o respuesta contiene la información necesaria para realizar ese propio intercambio. Por este motivo no se pueden establecer sesiones en esta primera versión de la aplicación. Una posible migración a SOAP podría permitir establecer estas sesiones desde la aplicación móvil, pudiendo de esta manera asemejarse aún más a la versión web. A si al registrarse como usuario o como paciente y salir de la aplicación en cualquiera de los casos, al volver a entrar en la aplicación se mantendrá la sesión con las que se ha accedido por última vez.

Actualizar acorde a los cambios de versiones pertinentes: debido a la continua actualización de las versiones de Android cada poco tiempo se presentan nuevas opciones que pueden ayudar a que la aplicación sea más completa y que sea más fácil y sencilla para el usuario.

Uso de fragments: respecto a los menús que aparecen en la aplicación, son bastante sencillos ya que la aplicación está hecha para una versión de Android baja y no se han podido hacer uso de ellos. Al actualizar a una nueva versión de Android para que corra sobre ella la aplicación se podrá hacer uso de los fragments para presentar al usuario los menús de una forma más actual y dinámica ya que con versiones de la aplicación más actuales se podrá hacer uso del ActionBarActivity necesario para el uso de fragments.

Poder borrar los ejercicios realizados por los pacientes: un cambio interesante en la parte de usuarios sería darle la posibilidad al usuario de borrar un ejercicio de los que el especialista le manda después de la cita en la clínica una vez ya ha realizado o cumplido el tiempo de realización que el fisioterapeuta a estipulado.

Añadir apartado noticias: como se puede ver en el proyecto en conjunto, en el sitio web hay un apartado de noticias en la que los encargados de las clínicas



CAPÍTULO 7

ponen en el tablón anuncios sobre las clínicas como pueden ser cambios de horarios de las clínicas o tratamientos nuevos que se ofrecen en las clínicas.

Recordar contraseña: para los usuarios que olviden su contraseña que aparezca un botón que les permita que les envíen un email a su correo electrónico con los pasos para recuperar la contraseña.

Capítulo 8: Bibliografía

[1] OMS, (2015), Acerca de la OMS. Recuperado en Marzo de 2015 de: <http://www.who.int/es/>

[2] Appleby, J. y Harrison, A. (2005). Spending on health care. How much is enough? King's Fund. Recuperado en Marzo de 2015 de : <http://www.gestion-sanitaria.com/1-sistemas-sanitarios-union-europea.html>

[3] OMS, 51º Consejo directivo, 63º Sesión del comité regional (2011)

[4] Bernardo, A.(2013). E-Health: Tecnología móvil que revoluciona la medicina. Recuperado en Marzo de 2015 de: <http://blogthinkbig.com/ehealth-tecnologia-movil-medicina/>

[5] FDA, (2015). Recuperado en Marzo de 2015 de: <http://www.fda.gov/>

[6] EMEA, (2015). About us. Recuperado en marzo de 2015 de: <http://www.ema.europa.eu/ema/>

[7] Dolan B. (2012). Patient monitoring device market to hit \$8B in 2017. Recuperado en Marzo de 2015 de: <http://mobihealthnews.com/16747/patient-monitoring-device-market-to-hit-8b-in-2017/>

[8] Vishwanath S. (2012). Touching lives through mobile health Assessment of the global market opportunity. Recuperado en Marzo de 2015 de: http://www.pwc.in/assets/pdfs/telecom/gsma-pwc_mhealth_report.pdf

[9] Shankar R. (2013). Socio-economic impact of mHealth .An assessment report for the European. Pag 6. Recuperado en Marzo de 2015 de: http://www.gsma.com/connectedliving/wp-content/uploads/2013/06/Socio-economic_impact-of-mHealth_EU_14062013V2.pdf

[10] Telefónica (2014). Informe Sociedad de la Información en España 2014' SIE. Recuperado en Marzo de 2015 de: http://www.fundaciontelefonica.com/artes_cultura/sociedad-de-la-informacion/informe-sociedad-de-la-informacion-en-espana-2014/

[11] Muñoz V. (2015, 20 de Febrero). mHealth: Tecnología inalámbrica que salva vidas. El Mundo. Recuperado en Marzo de 2015 de: <http://www.elmundo.es/economia/2015/02/20/54e712b022601df4728b4588.html>

[12] Figge H. (2014). Tackling the Global Challenge of Chronic. Disease: Opportunities for Mobile Health. Recuperado en Marzo de 2015 de: https://www.b2match.eu/system/echalliance-at-mobileworldcongress2014/files/13.Helen_Figge_2014Final.pdf?1393850466



BIBLIOGRAFIA

[13] Gaceta medica (2014, 26 de febrero). La e-Salud es esencial en el abordaje de los crónicos. Recuperado en Marzo de 2015 de: <http://www.gacetamedica.com/noticias-medicina/2014-02-26/politica/la-e-salud-es-esencial-en-el-abordaje-de-los-cronicos/pagina.aspx?idart=819956>

[14] Bernardo. A (2014, 25 de Marzo). Un simple smartphone ayudaría a detectar casos de cáncer en el tercer mundo. Recuperado en Marzo de 2015 de: <http://blogthinkbig.com/smartphone-deteccion-cancer/>

[15] Amate C. (2015, 20 de Enero). Te Acompaña facilita la vida a personas en situación de dependencia y sus familias. Recuperado en Marzo de 2015 de: <http://blogthinkbig.com/te-acompana/>

[16] Colegio Profesional de Fisioterapeutas de la Comunidad de Madrid (2015). Recuperado en Marzo de 2015 de: https://www.cfisiomad.org/pages/informacion_fisioterapia.aspx

[17] Rafael Anta, Shireen El-Wahab y Antonino Giuffrida, “Salud Móvil. El potencial de la telefonía celular para llevar la salud a la mayoría”, Banco Interamericano de Desarrollo, Febrero 2009.

[18] Tomás Gironés, J., (2015), Android: Introducción a la programación. Recuperado en Marzo de 2015 de: <http://cursointroduccionandroid.upvx.es/course>

[19] Dispositivos móviles (2012). Recuperado en Marzo de 2015 de: <http://dispositivosmobilesits.blogspot.com.es/2012/02/evolucion-de-moviles.html>

[20] Arias, X., (2014), Del 1 al 8: La evolución del sistema operativo IOS. Recuperado en Marzo de 2015: <http://www.enter.co/especiales/vida-digital/del-1-al-8-la-evolucion-del-sistema-operativo-ios/>

[21] Michan, M., (2013), La evolución de iOS desde 2007 hasta la actualidad. Recuperado en Marzo de 2015: <http://www.applesfera.com/ios/la-evolucion-de-ios-desde-2007-hasta-la-actualidad-especial-historia-wwdc>

[22] Team-Soporte (2013). Recuperado en Marzo de 2015 de: <http://team-soporte.es.tl/Android,-iOS-y-Windows-Phone-Ventajas-y-Desventajas.htm>

[23] Balcázar Alanís, F., (2014), Ventajas e inconvenientes de los sistemas operativos móviles: iOS. Recuperado en Marzo de 2015 de: <http://xombit.com/2013/08/ventajas-inconvenientes-ios-2>

[24] Poder Pda (2014). Recuperado en Marzo de 2015 de: <http://www.poderpda.com/plataformas/android/android-supera-los-mil-millones-de-smartphones-en-2014/>

[25] Team-Soporte (2013). Recuperado en Marzo de 2015 de: <http://team-soporte.es.tl/Android,-iOS-y-Windows-Phone-Ventajas-y-Desventajas.htm>



[26] Ounae (2014). Recuperado en Marzo de 2015 de: <http://ounae.com/ventajas-inconvenientes-android-google/>

[27] Blog de WordPress, Sistema Android (2015). Recuperado en Marzo de 2015 de: <https://scoello12.wordpress.com/ventajas-y-desventajas/>

[28] Gigatecno (2015). Recuperado en Marzo de 2015 de: <http://gigatecno.blogspot.com.es/2014/05/ventajas-y-desventajas-de-android.html>

[29] About, Tabletas y Ipad (2015). Recuperado en Marzo de 2015 de: http://tabletas.about.com/od/Glosario/ss/Las-distintas-versiones-de-Android-y-el-problema-de-la-fragmentacion_2.html

[30] Santa-María, F. (2015). Recuperado en Marzo de 2015 de: <http://blog.staffcreativa.pe/android-ventajas-desventajas/>

[31] Microsoft, (2015), Programación en Windows Phone. Recuperado en Marzo de 2015 de: <https://msdn.microsoft.com/es-es/library/ff827803%28v=xnagamestudio.40%29.aspx>

[32] Windows, (2015). Centro de desarrollo. Recuperado en Marzo de 2015 de: <http://dev.windows.com/es-es/develop/download-phone-sdk>

[33] Hurtado, C., (2014). Firefox OS, Ubuntu y Tizen: La vida más allá de iOS y Android.”. Recuperado en Marzo de 2015 de: http://parentesis.com/noticias/software_aplicaciones/Firefox_Ubuntu_y_Tizen

[34] Php (2014), ¿Que es PHP?. Recuperado en marzo de 2015 de: <http://php.net/manual/es/intro-whatish.php>

[35] Álvarez M.A (2001, 09 de mayo). Qué es PHP. Recuperado en Marzo de 2015 de: <http://www.desarrolloweb.com/articulos/392.php>

[36] Van Der Henst S C. (2001, 23de mayo).¿Qué es PHP?. Recuperado en Marzo de 2015 de: <http://www.maestrosdelweb.com/phpintro/>

[37] Welling L. & Thomson L. (2003). Desarrollo web con PHP y MySQL. Madrid: Anaya

[38] Rivero-Cornelio E., Martínez-Fuentes L. & Alonso-Martínez I. (2005). Bases de datos relacionales: fundamentos y diseño lógico. Madrid: Universidad Pontificia Comillas.

[39] De la Torre-Díez, I., Díaz-Pernas, F.J., Antón-Rodríguez, M., Gutiérrez-Díez, P., Martínez-Zarzuola, M., González-Ortega, D. & Díez-Higuera, J.F. (Diciembre 2010). Requisitos y evolución de los sistemas de e-Health, en: Actas de la Conferencia IADIS Ibero Americana WWW/INTERNET 2010 (CIAWI 2010), Algarve (Portugal). ISBN: 978-972-8939-34-2



BIBLIOGRAFIA

[40] Martín-Escofet C. El lenguaje SQL. Recuperado en Marzo de 2015 de: http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02149.pdf

[41] Ramakrishnan R. & Gehrke J. (2007). Sistemas de gestión de bases de datos (3ª Ed.). Madrid: McGraw-Hill.

[42] MySQL (2014), WhyMySQL?. Recuperado en Marzo de 2015 de: <http://php.net/manual/es/intro-what-is.php>

[43] LukeWelling y Laura Thomson, Desarrollo web con PHP y MySQL, Anaya Multimedia, 2003.

[44] Paul Dubois, MySQL, Prentice Hall, 2001

[45] Hugh E. Williams y David Lane, Web Database Applications with PHP and MySQL, O'Reilly, 2002

[46] Fabián Brea O (2005, 03 de marzo).SOAP (Simple Object Access Protocol). Recuperado en Marzo de 2015 de: <http://www.desarrolloweb.com/articulos/1853.php>

[47] Benjamín-González C. (2004, 07 de julio).SOAP (Simple Object Access Protocol). Recuperado en Marzo de 2015 de: <http://www.desarrolloweb.com/articulos/1557.php>

[48] Simoes C. (2015, 31 de enero). REST vs SOAP al servicio de la web. Recuperado en Marzo de 2015 de: <http://inusual.com/articulos/rest-vs-soap-al-servicio-de-la-web/>

[49] AndroidDevMx (2011). Recuperado en Marzo de 2015 de: <http://www.androiddevmx.net/java-y-la-maquina-virtual-dalvik>

[50] TodoAndroid (2015). Recuperado en Marzo de 2015 de: <http://www.todoandroid.es/index.php/faq-de-android/65-versiones/1698-android-studio-o-eclipse-opinion-de-un-desarrollador-de-aplicaciones.html>

[51] Catalan, M. (2014). ¿Es el momento de saltar a Android Studio?. Recuperado en Marzo de 2015 de: <https://geekytheory.com/es-el-momento-de-saltar-a-android-studio/>



Capítulo 9: Anexos

Anexo I. Puesta en marcha del entorno de programación.

Instalación y puesta en marcha de Instalación de *Eclipse*

El primer paso consiste en la instalación del IDE Eclipse. Para ello, es necesario acudir a la página web de Eclipse, <http://www.eclipse.org/downloads/>, y elegir en función de la plataforma y arquitectura sobre la que se trabaje, una versión de Eclipse adecuada. En la Figura siguiente se puede apreciar el aspecto de la sección de descargas comentada a fecha actual.



ANEXOS

The screenshot displays the Eclipse Downloads website interface. At the top, there are navigation tabs for 'Packages', 'Developer Builds', and 'Projects'. Below this, a sub-header reads 'Eclipse Indigo (3.7.2) Packages for' followed by a dropdown menu currently set to 'Windows'. The main content area lists several Eclipse IDE packages, each with a download icon, package name, size, download count, and a 'Details' link. The packages listed are:

Package Name	Size	Download Count	Available Downloads
Eclipse IDE for Java EE Developers	212 MB	2,116,656 Times	Windows, Linux, Mac OS X (Cocoa)
Eclipse Classic 3.7.2	174 MB	1,236,277 Times	Windows 32 Bit, Windows 64 Bit
Eclipse IDE for Java Developers	128 MB	749,160 Times	Windows 32 Bit, Windows 64 Bit
Eclipse IDE for C/C++ Developers (includes Incubating components)	108 MB	317,089 Times	Windows 32 Bit, Windows 64 Bit
Eclipse Modeling Tools	272 MB	102,659 Times	Windows 32 Bit, Windows 64 Bit
Eclipse IDE for JavaScript Web Developers	110 MB	92,867 Times	Windows 32 Bit, Windows 64 Bit
Eclipse for RCP and RAP Developers	181 MB	90,509 Times	Windows 32 Bit, Windows 64 Bit
Eclipse IDE for Java and Report Developers	243 MB	84,973 Times	Windows 32 Bit, Windows 64 Bit
Eclipse for Testers	90 MB	63,222 Times	Windows 32 Bit, Windows 64 Bit
Eclipse for Scout Developers	175 MB	52,841 Times	Windows 32 Bit, Windows 64 Bit
Eclipse IDE for Parallel Application Developers (includes Incubating components)	181 MB	52,151 Times	Windows 32 Bit, Windows 64 Bit

Anexo 1. Lugar de descarga del programa de desarrollo

Tanto Eclipse Classic, como Eclipse IDE for Java EE Developers, como Eclipse IDE for Java Developers son válidas para el fin propuesto. La descarga es en un archivo ZIP que lleva comprimido en su interior el entorno de desarrollo elegido. No es necesaria ninguna instalación. La versión de Eclipse utilizada durante la elaboración del proyecto es Eclipse Juno.

Instalación del SDK de *Android*

Para instalar el SDK de Android es necesario acceder a la siguiente página web: <http://developer.android.com/sdk/index.html>. En la Figura siguiente se puede ver el aspecto de la sección desde la cual se descargará un SDK u otro en función del sistema con el que se trabaje.



Download the Android SDK

Welcome Developers! If you are new to the Android SDK, please read the steps below, for an overview of how to set up the SDK.

If you're already using the Android SDK, you should update to the latest tools or platform using the *Android SDK and AVD Manager*, rather than downloading a new SDK starter package. See [Adding SDK Components](#).

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r17-windows.zip	37417953 bytes	3af1baeb39707e54df068e939aaa5a79
	installer_r17-windows.exe (Recommended)	37410775 bytes	5afaf6511ebaa52bd8d1dba4afc61e41
Mac OS X (intel)	android-sdk_r17-macosx.zip	33867836 bytes	52639aae036b7c2e47cf291696b23236
Linux (i386)	android-sdk_r17-linux.tgz	29706368 bytes	14e99dfa8eb1a8fadd2f3557322245c4

Anexo 2. Descarga del SDK desde la página de Android Developers

Suponiendo que la máquina donde se instale el SDK trabaje con Windows, se procederá a la descarga del ejecutable. Éste, al ser ejecutado, comprobará primeramente que hay alguna versión instalada del JDK (Java Development Kit) en el sistema. De no ser así, será necesaria su descarga. La instalación es muy similar a la de cualquier programa convencional. A lo largo de la instalación, se le requerirá al usuario la introducción de una ruta o path hacia la dirección de destino de la instalación.

Instalación del *pluing* ADT *Android* para *Eclipse*

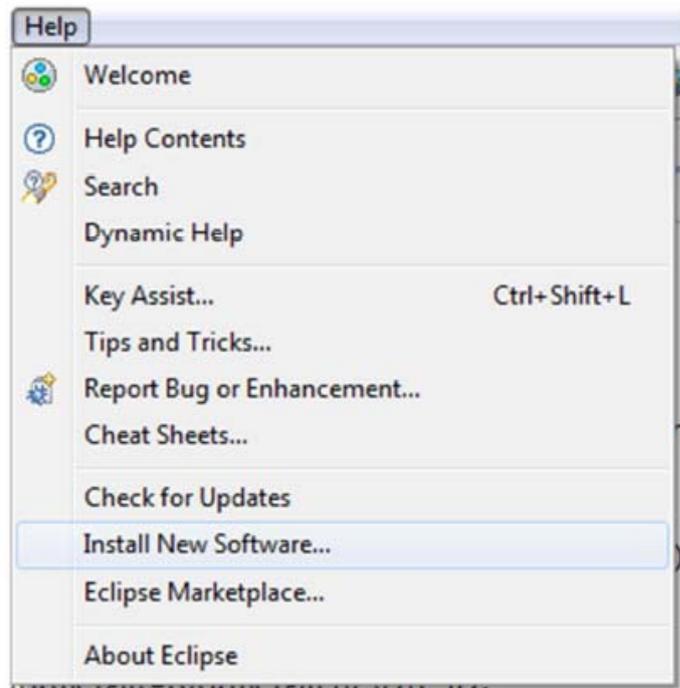
ADT (Android Development Tools) es un plugin para el entorno de desarrollo Eclipse diseñado para proporcionar al programador un conjunto de herramientas y facilidades para crear aplicaciones Android.

ADT extiende las capacidades de Eclipse y permite configurar rápidamente nuevos proyectos, crear interfaces de usuario de una forma intuitiva, hacer debug de la aplicación usando el Android SDK y hasta exportar la aplicación a formato APK para su distribución.

Para instalar el plugin en Eclipse, tal y como se puede ver en la Figura siguiente, es necesario abrir Eclipse, y en la sección de ayuda, seleccionar la opción de Install new Software...



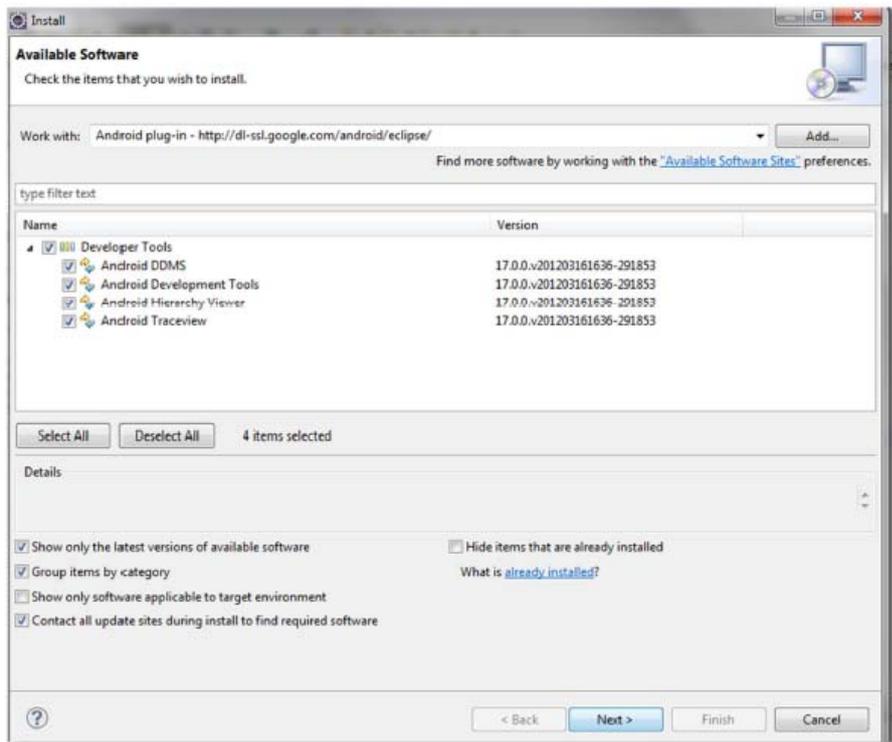
ANEXOS



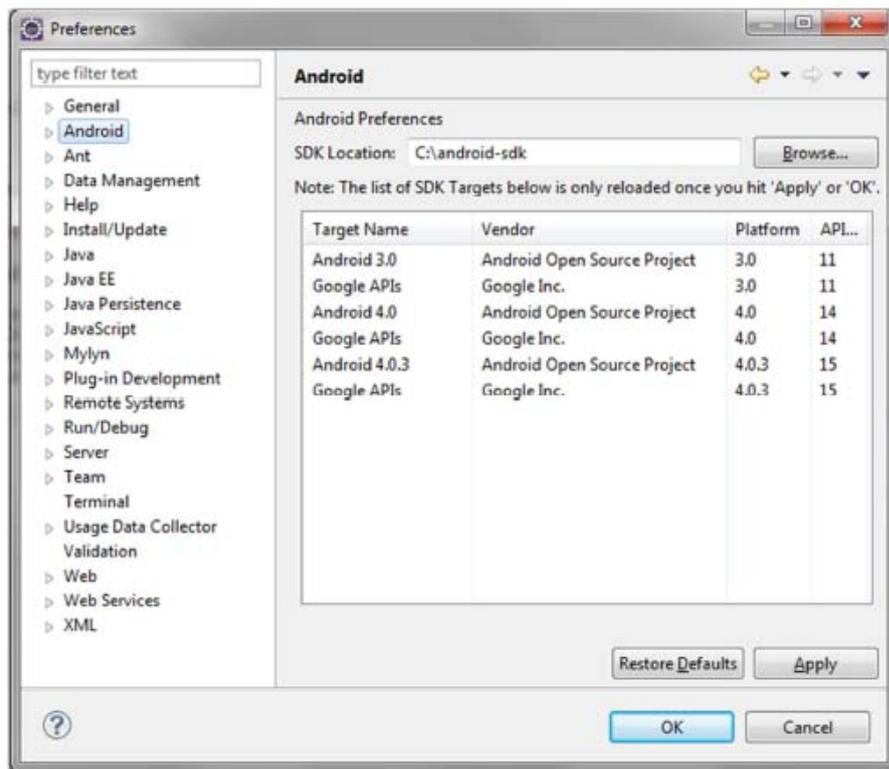
Anexo 3. Primer paso de la instalación ADT *plugin* en *Eclipse*

A continuación, será necesario añadir un nuevo repositorio presionando el botón Add de la parte superior derecha. El nombre es indiferente, pero la dirección tiene que coincidir con <https://dl-ssl.google.com/android/eclipse/>. En el software disponible se deberá marcar el checkbox que se encuentra al lado de Developers Tools y presionar el botón Next dos veces. En la siguiente figura se muestra una captura de pantalla de este proceso. A continuación se muestran los términos de la licencia, los cuales deberán ser leídos y aceptados.

Una vez instalado el plugin, será necesario un reinicio del entorno de desarrollo para poder configurarlo. Para ello, será necesario entrar en las preferencias de Eclipse seleccionando Windows y la opción Preferences. En el menú de la izquierda, tal y como se puede ver en la figura siguiente se deberá de seleccionar Android y en SDK Location será necesario introducir la ruta o path donde se instaló el Android SDK.



Anexo 4. Instalación del plugin ADT.



Anexo 5. Configuración del ADT *plugin*.



ANEXOS

Con este paso, Eclipse está preparado para desarrollar aplicaciones Android.

Anexo II. Wrapper.

En este proyecto, se tuvieron que retornar varias variables primitivas desde una misma función y de diferentes tipos, para ello se recurrió a la utilización del Wrapper.

El Wrapper es un tipo de encapsulamiento, en el que podemos declarar el número de variables y del tipo que se requiera. En nuestro caso a la hora realizar el Login, la función deber retornar el resultado, nombre de usuario y el id del usuario. En la siguiente imagen se puede apreciar como declaramos 3 variables:

- String *resultado*: esta variable resuelve el Login, determinando si es paciente, profesional o encargado.
- String *login_user*: esta variable devolverá el usuario logueado.
- Long *id*: esta variable devuelve el id del usuario logueado.

```
1 package proyecto;
2
3 public class Wrapper {
4
5     public String resultado;
6     public String login_user;
7     public long id;
8 }
9
```

Anexo 6. Wrapper realizado para el Login

Una vez declarados en la clase *Wrapper* ya estarán disponibles en todas las clases creadas donde se quieran utilizar.

En la clase donde queramos utilizarlo tendremos que inicializarlo para poder usarlo. Como se puede apreciar en la imagen siguiente, se quiere que la función *doInBackground* devuelva el *Wrapper* por lo que la función es declarada con este tipo de variable. También se declara la variable *Wrapper* mediante la sentencia:

```
Wrapper w = new Wrapper();
```



```

class asynclogin extends AsyncTask< String, String, Wrapper > {

    String user,pass;
    protected void onPreExecute() {
        //para el progress dialog
        pDialog = new ProgressDialog(Login.this);
        pDialog.setMessage("Autenticando...");
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(false);
        pDialog.show();
    } //fin onPreexecute

    protected Wrapper doInBackground(String... params) {
        //obtnemos usr y pass
        user=params[0];
        pass=params[1];
        Wrapper w = new Wrapper();

        //enviamos y recibimos y analizamos los datos en segundo plano.
        if (loginstatus(user,pass)==1){

            try {
                // Llamamos al servicio web para recuperar los datos
                HttpGet httpGet = new HttpGet("http://" +Funciones.direccion() +"/proyecto/cogerUsuario.php");
                HttpClient httpClient = new DefaultHttpClient();
                HttpResponse response = (HttpResponse)httpClient.execute(httpGet);
                HttpEntity entity = response.getEntity();
                BufferedHttpEntity buffer = new BufferedHttpEntity(entity);
                InputStream iStream = buffer.getContent();

                String aux = "";
            }
        }
    }
}

```

Anexo 7. Principio de la tarea en segundo plano

Una vez declarado y creado, se deberá introducir los valores que se quieran. Por ejemplo, en el caso de este proyecto, se descargaban todos los usuarios, pero únicamente la función se quedaba con el que correspondía con el usuario introducido en el login. De esta manera se mete en el *Wrapper* los valores *Login_user* e *Id_user*, el primero en la variable *w.login_user* y el segundo en la variable *w.id*.



ANEXOS

```
for(int i = 0; i < usuario1.length(); i++) {  
  
    JSONObject usuario = usuario1.getJSONObject(i);  
  
    Usuario u = new Usuario(usuario.getInt("id_user"), usuario.getString("name_user"), usuario.getString("surname_user"),  
        usuario.getString("dni_user"), usuario.getString("tel1_user"), usuario.getString("tel2_user"), usuario.getString("adress_user"),  
        usuario.getString("email_user"), usuario.getString("login_user"), usuario.getString("pass_user"));  
  
    String usuarioBueno=u.getLogin_user();  
    if(user.equals(usuarioBueno))  
    {  
        w.login_user= u.getLogin_user();  
        w.id=u.getId_user();  
    }//fin del if usuarioBueno
```

Anexo 8. Guardando variables en el wrapper

La otra variable que se introduce en el *Wrapper* es la del resultado, que almacenará el tipo de usuario de la aplicación, Paciente, Profesional o Encargado. Si el nombre del usuario y la contraseña coinciden con el de algún usuario de las tablas de la base de datos, a la variable *w.resultado* se le asignará el valor "*usuario*", "*profesional*" o "*encargado*" como se muestra en la siguiente imagen.

```
for(int i = 0; i < encargado1.length(); i++) {  
    JSONObject encargado = encargado1.getJSONObject(i);  
  
    Profesional e = new Profesional(encargado.getInt("id_pro"), encargado.getString("dni_pro"), encargado.getString("name_pro"), encargado.getString("surname_ pro"),  
        encargado.getString("email_pro"), encargado.getInt("tel_pro"), encargado.getString("login_pro"), encargado.getString("pass_pro"),  
        encargado.getString("specialty_pro"), encargado.getInt("id_clinic"), encargado.getInt("responsible_pro"));  
  
    String encargadoBueno=e.getLogin_pro();  
    if(user.equals(encargadoBueno))  
    {  
        w.login_user= e.getLogin_pro();  
        w.id=e.getId_pro();  
    }//fin del if encargadoBueno  
  
    }//fin del for  
  
}//fin del try  
  
catch(Exception e) {  
  
    Log.e("WebService", e.getMessage());  
  
}//fin del catch  
  
w.resultado="encargado";  
}//fin del if encargado
```

Anexo 9. Guardando variables en el wrapper II

De no existir en ninguna de las tablas se le asignará el valor "*err*" a la variable *w.resultado*. Una vez almacenados todos los datos en la variable *w* (Wrapper) la retornamos y la recibimos en el *onPostExecute* como se puede apreciar en la siguiente imagen. Pudiendo utilizar estas variables en todo momento como se



puede apreciar en una última línea en la que se comprueba que la variable *w.resultado* sea igual a "usuario" para comprobar que es un usuario.

```
else
{
    w.resultado="err";

} //fin del else general

return w;

} //fin doInBackground

protected void onPostExecute(Wraper w) {

    pDialog.dismiss(); //ocultamos progress dialog.
    Log.e("onPostExecute=", ""+w.resultado);

    if (w.resultado.equals("usuario")){
```

Anexo 10. onPostExecute y error