

**Universitat Oberta
de Catalunya**



Ingeniería Técnica Informática de Gestión

Trabajo fin de carrera

**Agenda de fisioterapeutas
(PlanningFisio): Gestión de
pacientes y citas para
dispositivos móviles Android.**

Ricardo Carrillo Cobo

Consultor: Roberto Ramírez Vique

Mayo de 2013

Agradecimientos

Gracias a mis padres y hermanas por su apoyo durante todos estos años y por creer en mí.

Gracias a mi esposa por los paseos que no nos dimos y por las cosas que no pudimos hacer, por tu paciencia conmigo en los ratos de agobio y por darme luz en los días nublados.

Contenido

1. INTRODUCCIÓN Y OBJETIVOS	9
1.1 Introducción	10
1.2 Motivación	10
1.3 Objetivos	10
1.4 Posibles aplicaciones del sistema	11
1.5 Fase de desarrollo	11
1.6 Medios utilizados	12
1.7 Contenidos	13
2. ESTADO DEL ARTE	15
2.1 Introducción	16
2.2 Elección de la plataforma	16
2.2.1 Android	17
2.2.2 Historia de Android	19
2.2.3 Arquitectura de Android	23
2.3 JSON (JavaScript Object Notation)	27
2.4 WEB SERVICE	27
2.4.1 Introducción	27
2.4.2 Ventajas e inconvenientes	28
2.5 PHP (Hypertext Pre-processo)	28
2.6 MySQL	28
3. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN	30
3.1 Introducción	31
3.2 Planificación	31
3.3 Descripción general	33
3.3.1 Requisitos de usuario	33
3.3.2 Entorno operacional	33
3.3.3 Requisitos de uso	34
3.3.4 Actores del sistema	39

3.3.5 Casos de uso	40
3.4 Diseño	47
3.4.1 Arquitectura del sistema	48
3.4.2 Diseño de interfaces de usuario	51
3.4.3 Diagrama de clases.	55
3.4.4 Servicio web y base de datos	57
3.4.5 Diagrama de bases de datos	58
3.4.6 Gestión de comunicaciones	58
3.5 Implementación	59
4. CONCLUSIONES	60
5. LINEAS FUTURAS	62
6. REFERENCIAS	64

Tablas

Tabla 1. Entorno operacional del administrador	33
Tabla 2. Entorno operacional del fisioterapeuta	34
Tabla 3: RUC-01	35
Tabla 4: RUC-02	35
Tabla 5: RUC-03	35
Tabla 6: RUC-04	35
Tabla 7: RUC-05	36
Tabla 8: RUC-06	36
Tabla 9: RUC-07	36
Tabla 10: RUC-08	36
Tabla 11: RUC-09	36
Tabla 12: RUC-10	37
Tabla 13: RUC-11	37
Tabla 14: RUC-12	37
Tabla 15: RUC-13	37
Tabla 16: RUR-01	38
Tabla 17: RUR-02	38
Tabla 18: RUR-03	38
Tabla 19: RUR-04	38
Tabla 20: RUR-05	38
Tabla 21: RUR-06	39
Tabla 22: A01	39
Tabla 23: A02	39
Tabla 24: CU-01	42
Tabla 25: CU-02	42
Tabla 26: CU-03	43
Tabla 27: CU-04	43
Tabla 28: CU-05	44
Tabla 29: CU-06	44
Tabla 30: CU-07	44
Tabla 31: CU-08	45
Tabla 32: CU-09	45
Tabla 33: CU-10	46
Tabla 34: CU-11	46
Tabla 35: CU-12	47
Tabla 36: CU-13	47

Ilustraciones

<i>Ilustración 1. Android vs iOS.....</i>	<i>16</i>
<i>Ilustración 2 Evolución de Android.....</i>	<i>23</i>
<i>Ilustración 3. Arquitectura de Android</i>	<i>26</i>
<i>Ilustración 4: Planificación</i>	<i>32</i>
<i>Ilustración 5: Diagrama de casos de uso – Administrador.....</i>	<i>40</i>
<i>Ilustración 6: Diagrama de casos de uso – Fisioterapeuta</i>	<i>41</i>
<i>Ilustración 7: Modelo vista controlado (MVC).....</i>	<i>48</i>
<i>Ilustración 8: Modelo Cliente-Servidor</i>	<i>49</i>
<i>Ilustración 9: Arquitectura del sistema</i>	<i>50</i>
<i>Ilustración 10: Acceso al sistema.....</i>	<i>52</i>
<i>Ilustración 11: Menú del usuario “fisioterapeuta”.....</i>	<i>52</i>
<i>Ilustración 12: Menú del usuario “administrador”.....</i>	<i>52</i>
<i>Ilustración 13: Listado de pacientes/fisioterapeutas.....</i>	<i>53</i>
<i>Ilustración 14: Alta de pacientes/fisioterapeutas</i>	<i>53</i>
<i>Ilustración 15: Selección de fecha.....</i>	<i>53</i>
<i>Ilustración 16: Citas de una fecha.....</i>	<i>53</i>
<i>Ilustración 17: Diagrama de navegación</i>	<i>54</i>
<i>Ilustración 18: Diagrama de clases</i>	<i>56</i>
<i>Ilustración 19: Diagrama de clases – Conexión del web service a base de datos.....</i>	<i>57</i>
<i>Ilustración 20: Diagrama de la base de datos</i>	<i>58</i>

1. INTRODUCCIÓN Y OBJETIVOS

1.1 Introducción

Los motivos por los cuales he decidido realizar el trabajo de fin de carrera en el área de **“Desarrollo de aplicaciones de dispositivos móviles”**, es debido a la gran evolución que están teniendo las nuevas tecnologías aplicadas a los citados dispositivos, así como la gran variedad de los mismos y de sistemas operativos dedicados, por lo que creo que será interesante ir aprendiendo nuevos lenguajes de programación, los cuales, pueden ser muy útiles en mi vida laboral.

1.2 Motivación

El trabajo de fin de carrera, es la culminación de la carrera de Ingeniería Técnica Informática de Gestión. En el mismo, se ponen en práctica la mayoría de los conocimientos adquiridos durante el desarrollo de la misma, tales como definición de requerimientos, análisis, implementación...

A nivel personal supone el reto de ampliar mis conocimientos de Java y el desarrollo de una aplicación destinada a fisioterapeutas que en un futuro podría ser adaptada a otros entornos.

1.3 Objetivos

El presente Trabajo Fin de Carrera, tiene el objetivo el realizar el análisis, diseño e implementación de una aplicación móvil destinada a la gestión de pacientes y citas para una clínica de fisioterapia, pero con la característica especial de que la citada clínica no tiene una sede como tal, sino que los fisioterapeutas se desplazarán a los domicilios de sus clientes.

La aplicación logra que un fisioterapeuta pueda acceder de manera rápida y sencilla desde su dispositivo a su cartera de pacientes, con lo cual podrá

conseguir una gestión eficiente de los mismos, así como la posibilidad de crear citas para los citados pacientes y llevar un control de su trabajo.

1.4 Posibles aplicaciones del sistema

Existen varios escenarios a los que se puede aplicar el uso de este sistema a desarrollar y que dependen de varios factores, tales como el tipo de cliente que quiera usar la aplicación, así como el presupuesto final que se destine a desarrollar el proyecto.

La aplicación principal será la sustitución de la tradicional agenda de citas, por una más moderna, gestionada a través de un teléfono móvil, lo que supondrá una serie de ventajas para la clínica fisioterapéutica así como para los fisioterapeutas que trabajan en la misma.

El fisioterapeuta también puede gestionar a cada uno de los pacientes pudiendo dar de alta y baja a los mismos, y además la aplicación le proporciona un calendario de citas en el que el fisioterapeuta podrá gestionar las mismas.

Por último, como la aplicación está destinada a clínicas fisioterapéuticas en las que los fisioterapeutas atienden a domicilio, la aplicación permite la posibilidad de guiar al fisioterapeuta desde su posición actual hasta la dirección del paciente que seleccione.

1.5 Fase de desarrollo

El presente trabajo fin de carrera consta de una serie de fases las cuales se detallan a continuación.

Documentación previa: Al elegir este trabajo fin de carrera no se tenían conocimientos sobre Android, de manera que ha sido necesario documentarse sobre esta plataforma y como desarrollar aplicaciones para ella.

Análisis de requisitos: Ha sido preciso evaluar inicialmente los requisitos necesarios para la elaboración de este trabajo fin de carrera y tener en cuenta las funcionalidades que se querían desarrollar.

Diseño: El diseño del sistema se ha realizado valorando que se quería lograr en cada una de las pantallas y como se quería mostrar al usuario, de manera que sirviera de guía para la implementación posterior.

Implementación: Tras el análisis y diseño del sistema, se prosigue con la fase de implementación, en la que se supone que a medida que se profundice en el desarrollo se irán implementando las diferentes funcionalidades de la aplicación.

1.6 Medios utilizados

Para el desarrollo del presente trabajo fin de carrera, en su fase de desarrollo, así como en la elaboración de ésta memoria, se han utilizado una serie de elementos tanto de hardware y software, los cuales son detallados a continuación.

Elementos hardware

Ordenador portátil: Se ha utilizado tanto para la programación de la aplicación PlanningFisio, como para la elaboración de la memoria. El modelo usado ha sido un Asus modelo M50 con 4Gb de RAM y sistema operativo Windows 8.

Teléfono Smartphone: Utilizado para las pruebas durante el desarrollo de la aplicación, la marca y el modelo concreto ha sido un Sony Xperia S, con un procesador de doble núcleo Qualcomm de 1,5 GHz una pantalla de 4,3 pulgadas y resolución de 1280x720 pixeles.

Elementos software

Paquete ADT: Este paquete suministrado por Android, incluye los componentes esenciales de Android SDK y una versión del IDE de Eclipse con una función ADT (Android Developer Tools) para agilizar el desarrollo de aplicaciones Android. Entre sus funciones se encuentran configurar rápidamente nuevos proyectos, crear interfaces gráficos e incluso exportar aplicaciones firmadas para su distribución.

El SDK de Android proporciona tanto las bibliotecas API como las herramientas de desarrollo para crear, probar y depurar aplicaciones para Android.

Microsoft Office Professional Plus 2010: Dentro del paquete office, se han utilizado herramientas como Microsoft Project para la elaboración de los diagramas y Microsoft Word y Excel para la elaboración de la presente memoria y gráficas de la misma.

Gimp 2.8: Usado para el retoque de imágenes, así como para la creación de parte de la interfaz gráfica de la aplicación.

1.7 Contenidos

La presente memoria se ha estructurado en varios capítulos en los cuales se profundiza más detalladamente sobre los diferentes aspectos que componen éste trabajo fin de carrera.

En el primer capítulo, “**Introducción y objetivos**”, se muestra la motivación para la elección del trabajo fin de carrera, los objetivos que se ha buscado perseguir, así como las posibles aplicaciones del sistema y las diferentes fases del proyecto y los medios que se han requerido para la elaboración del mismo.

El segundo capítulo, “**Estado del arte**”, llevará a cabo un estudio de las diferentes tecnologías móviles existentes hasta el momento y por qué la elección de Android. Igualmente se mostrará una breve historia de Android y su evolución desde su inicio hasta día de hoy.

El tercer capítulo, “**Análisis, diseño e implementación**”, se trata del capítulo más importante de la memoria, y núcleo central de la misma. En él se analiza la estructura de la aplicación, se realiza el diseño y se explica su posterior implementación.

El cuarto capítulo, “**Conclusiones**”, se analiza el resultado final del trabajo fin de carrera y se medita sobre la consecución o no de los objetivos iniciales.

El quinto capítulo, “**Líneas futuras**”, se indican las posibles mejoras implementables al sistema para añadirle nuevas funcionalidades o evolucionarlo.

2. ESTADO DEL ARTE

2.1 Introducción

En este punto se llevará a cabo un estudio de las diferentes tecnologías móviles existentes hasta el momento y por qué la elección de desarrollar la aplicación para dispositivos móviles Android.

El objetivo final de este estudio es obtener una mayor visión del entorno tecnológico y así valorar otros productos similares evaluando si el producto desarrollado es viable en la actualidad.

2.2 Elección de la plataforma

Para la elección de la plataforma sobre la que se desarrollará el trabajo fin de carrera, se ha tenido que tomar una serie de decisiones entre las que se encuentra la necesidad de poseer un hardware con el sistema operativo sobre el que se vaya a ejecutar la aplicación desarrollada, así como lenguaje de programación utilizado para el desarrollo.

En la actualidad, existen multitud de sistemas operativos móviles, entre los que se encuentran BlackBerry, Symbian, Windows Phone, Linux, Firefox, etc., pero la realidad es que los dos más desarrollados y utilizados son iOS y Android, y es por ello, por lo que la elección se basó en usar uno de ellos.



Ilustración 1. Android vs iOS

Las dos plataformas se integran perfectamente con bases de datos y todo tipo de comunicaciones. Tienen un interfaz muy bueno, lo que las hace ser atractivas para el usuario y la mayoría de los móviles que se venden hoy en día poseen uno de estos dos sistemas operativos, lo que hace que nuestro sistema pueda llegar a un mayor número de usuarios, así como permitir dotarlo en un futuro de mayores y mejores características y funcionalidades.

Entre las características de los dispositivos iOS y Android, podemos mencionar que son dispositivos con interfaces táctiles, con unas capacidades de Hardware bastante avanzadas, con procesadores potentes que incluso por dispositivo alcanzan los 8 núcleos.

El problema que tiene iOS en comparación con Android es que tan sólo se puede utilizar en un dispositivo de Apple, mientras que Android se utiliza en múltiples dispositivos, existiendo multitud de marcas y modelos, lo que permite que se cree una gran competencia de mercado.

Por todo ello y por la disponibilidad que tiene el desarrollador de acceder a un dispositivo Android, se ha elegido realizar la aplicación para el sistema operativo Android, utilizando un teléfono marca Sony, modelo Xperia S y desarrollando la misma con la versión de Android 2.3

2.2.1 Android

Android es un sistema operativo para móviles basado en Linux y diseñado única y exclusivamente para teléfonos móviles (o Smartphone) y tablets. En la actualidad, también hay otros dispositivos que funcionan con Android como pueden ser: relojes, ordenadores, televisores...

El sistema operativo Android fue adquirido por Google en Julio de 2005, pero no fue hasta el año 2007, al adquirir la compañía Open Handset Alliance, la cual es un consorcio de varias compañías líderes en tecnología móvil, entre las que se

encuentran Texas Instruments, Broadcom, Nvidia, Qualcomm, Samsung, Intel, LG, Motorola y T-Mobile. El objetivo del consorcio respecto a Android, era el de desarrollar estándares abiertos para dispositivos móviles.

Android permite controlar dispositivos por medio de bibliotecas desarrolladas o adaptadas por Google mediante el lenguaje de programación Java. Además es una plataforma de código abierto. Esto quiere decir, que cualquier desarrollador puede crear y desarrollar aplicaciones escritas con lenguaje C u otros lenguajes y compilarlas a código nativo de ARM (API de Android). El responsable del mayor despliegue del código fuente de Android bajo licencia de Software Apache ha sido Google, haciendo uso de una licencia de software libre y de código abierto a cualquier desarrollador.

El sistema permite programar aplicaciones en una variación de Java llamada Dalvik. El sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (como el GPS, las llamadas, la agenda, etc.) de una forma muy sencilla en un lenguaje de programación muy conocido como es Java. Esta sencillez, junto a la existencia de herramientas de programación gratuitas, hace que una de las cosas más importantes de este sistema operativo sea la cantidad de aplicaciones disponibles, que extienden casi sin límites la experiencia del usuario. Una de las mejores características de este sistema operativo es que es completamente libre.

Es decir, ni para programar en este sistema ni para incluirlo en un teléfono hay que pagar nada. Y esto lo hace muy popular entre fabricantes y desarrolladores, ya que los costes para lanzar un teléfono o una aplicación son muy bajos. Cualquiera puede bajarse el código fuente, inspeccionarlo, compilarlo e incluso cambiarlo. Esto da una seguridad a los usuarios, ya que algo que es abierto permite detectar fallos más rápidamente. Y también a los fabricantes, pues pueden adaptar mejor el sistema operativo a los terminales.

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el

núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (surface manager), un framework OpenCore, una base de datos relacional SQLite, una Interfaz de programación de API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL,SSL y una biblioteca estándar de C Bionic.

2.2.2 Historia de Android

A lo largo de su historia, Android ha desarrollado multitud de versiones, las cuales van aportando mejoras con el fin de optimizar el potencial de los dispositivos móviles actuales. Desde su lanzamiento las versiones han sido las siguientes:

- **Android 1.1 y 1.0 (Septiembre de 2008)**

No fue utilizada en ningún dispositivo comercialmente accesible. Su versión mejorada llegó al primer dispositivo T-mobile G1 el mes siguiente del mismo año.

- **Android 1.5 (Lanzado en abril de 2009. Cupcake by Google)**

- ✓ Soporte para capturar y reproducir video
- ✓ Carga de imágenes y videos a Picasa y YouTube
- ✓ Mejoras en Bluetooth
- ✓ Se pueden colocar folders y widgets en pantalla de inicio
- ✓ Animación en varias transiciones de animación
- ✓ Predicción de texto

- **Android 1.6 (Lanzado en septiembre de 2009. Donut.)**

- ✓ Cámara nueva, camcoder e interfaz para galería
- ✓ Búsqueda por voz mejorada
- ✓ Soporte para pantallas más grandes
- ✓ Navegación de Google
- ✓ Convertidor de voz a texto
- ✓ Soporte multi tacto

- ✓ Soporte para VPN
- **Android 2.0 y 2.1 (Lanzado en octubre de 2008 y enero de 2010. Eclair.)**
 - ✓ Interfaz mejorada
 - ✓ Mejoría en sincronización de cuentas y contactos
 - ✓ Soporte de correo para Exchange, soporte para múltiples cuentas
 - ✓ Mejoras en la cámara como flash, zoom digital, balance de blancos y macro focus
 - ✓ Teclado virtual mejorado
 - ✓ Explorador mejorado con funciones clave de HTML5
 - ✓ Más rápido
 - ✓ Mejoras a calendario y Google Maps
 - ✓ Soporte para bluetooth 2.1
 - ✓ Wallpapers con movimiento
- **Android 2.2 (Lanzado en mayo 2010. Froyo)**
 - ✓ Soporte para hasta 8 conexiones WiFi o por USB
 - ✓ Teclado multi lenguaje soporta cambios de idioma rápidos
 - ✓ Mejoramientos en velocidad de acceso a buscador y aplicaciones
 - ✓ Mejora en llamada por voz, compartir contactos, soporte para bluetooth
- **Android 2.3 (Lanzamiento en diciembre de 2010. Gingerbread.)**
 - ✓ Interfaz mejorada para simplicidad y velocidad
 - ✓ Texto y teclado virtual más veloz e intuitivo
 - ✓ Selección de palabras y copiar/pegar con un toque
 - ✓ Administración de energía mejorada
 - ✓ Llamadas por Internet
 - ✓ Soporte NFC
 - ✓ Nuevo administrador de descargas
 - ✓ Soporte para cámaras traseras y delanteras
 - ✓ Soporte para barómetros y sensores de vectores

- **Android 3.0 (Lanzado en febrero de 2011. Honeycomb.)**
 - ✓ Incluye dispositivos más grandes, como tabletas
 - ✓ Interfaz optimizada incluye barra de sistema, de acciones, pantalla de inicio personalizable y lista de apps recientes
 - ✓ Selección de texto mejorada, copiar y pegar
 - ✓ Sincronización de archivos por USB sin montar un dispositivo de almacenamiento
 - ✓ Soporte para teclados por USB o Bluetooth
 - ✓ Soporte para gráficos de alta calidad en 2D y 3D

- **Android 3.1 (Lanzado en Junio de 2011. Honeycomb.)**
 - ✓ Mejorías de interfaz en navegación y animaciones
 - ✓ Soporte para dispositivos USB y accesorios como micrófonos, teclados, cámaras digitales y más
 - ✓ Lista de apps recientes expandible
 - ✓ Soporte para gamepads y joysticks
 - ✓ Estabilidad para redes Wi-Fi mejorada, incluyendo buscador, calendario, contactos y correo electrónico

- **Android Versión 3.2 (Lanzado en julio de 2011. Honeycomb.)**
 - ✓ Mejor desempeño en tabletas
 - ✓ Zoom compatible para aplicaciones de tamaño fijo
 - ✓ Acceso directo al sistema de almacenaje SD card
 - ✓ API de pantalla extendida para manejar distintos tamaños de pantalla

- **Android 4.0 (Lanzado en Octubre de 2011. Ice-Cream Sandwich.)**
 - ✓ Fusionó el diseño de 3.x y el v2.x en una sola versión
 - ✓ Interfaz mejorada
 - ✓ Folders de home y favoritos
 - ✓ Widgets de tamaño ajustable
 - ✓ Bloqueo de pantalla activado
 - ✓ Respuesta rápida a llamadas
 - ✓ Control de datos en red

- ✓ Mejora en redes sociales
- ✓ Calendario unificado
- ✓ Cámara: detección de rostro, estabilización de imagen, captura de fotos a la par de captura de video, app de galería y editor de fotos.
- ✓ Buscador despliega la versión para computadora de sitios web
- ✓ Correo electrónico mejorado
- ✓ Soporte directo de Wi-Fi

- **Android 4.1 (Lanzado en Julio de 2012. Jelly Bean.)**
 - ✓ Respuesta al tacto mejorada y transiciones más ágiles
 - ✓ notificaciones expansibles y accionables
 - ✓ Teclado intuitivo adaptable
 - ✓ Vista inmediata de fotografías capturadas
 - ✓ Braille vía USB
 - ✓ Búsqueda de voz mejorada
 - ✓ Envío y recepción de imágenes mejorada NFC
 - ✓ Audio USB
 - ✓ Google Wallet

- **Android 4.2 (Octubre de 2012. Jelly Bean.)**
 - ✓ Usuarios múltiples en tabletas
 - ✓ Captura de imágenes en 360°
 - ✓ Teclado activado por deslizamiento (como Swype)
 - ✓ Información desplegada cuando no está en uso

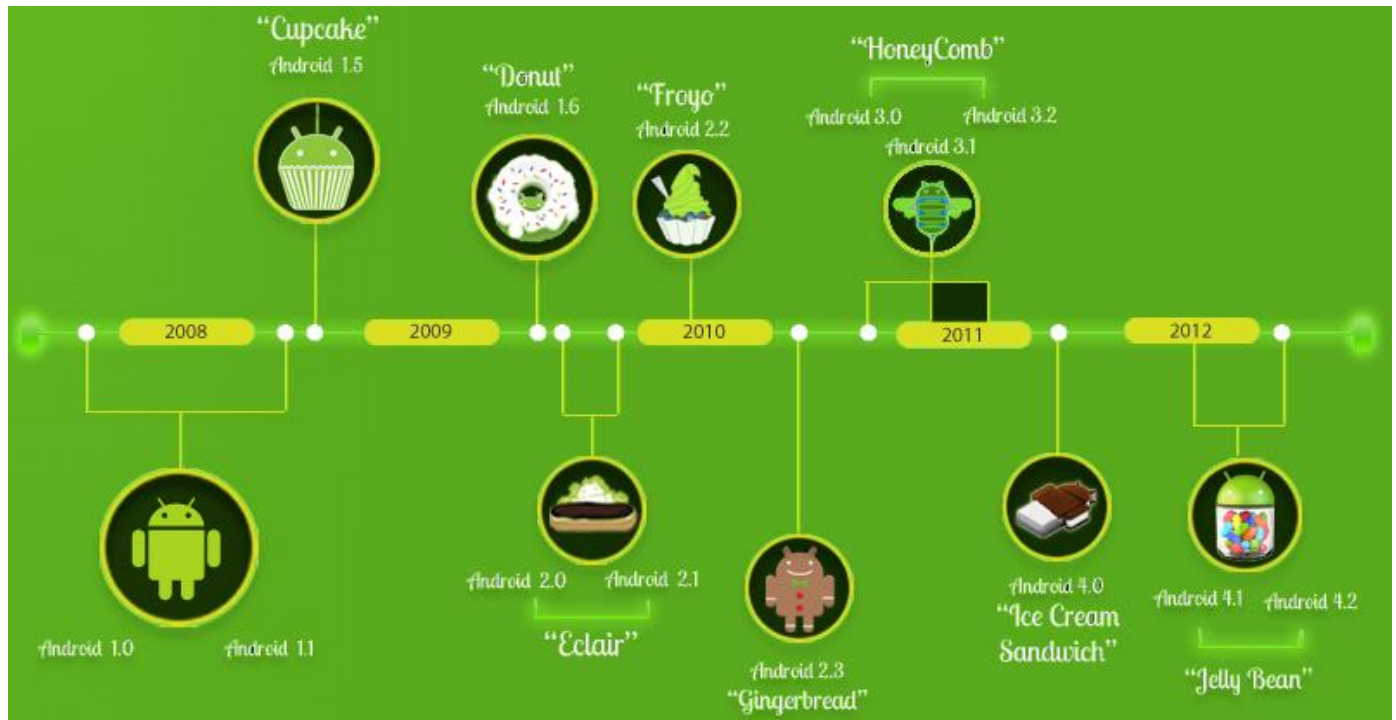


Ilustración 2 Evolución de Android

2.2.3 Arquitectura de Android

La arquitectura interna de la plataforma Android, está formada por 5 capas: Aplicaciones y Widgets, Framework de Aplicaciones, librerías, Runtime de Android y Kernel de Linux.

A continuación se detallan las citadas capas:

❖ Capa de aplicaciones y widgets

La capa está en la parte superior Android, es la capa que los usuarios finales verán, sin saber de todas las acciones que pasan por debajo de la citada capa.

En esta capa nos encontramos con las aplicaciones creadas con la plataforma Android, como mapas, programas fotográficos, programas de redes sociales, los

contactos, un cliente de email, el calendario, mapas, el navegador y algunos otros servicios mínimos.

Las citadas aplicaciones están escritas en el lenguaje de programación Java y se encuentran cargadas ya en el dispositivo cuando lo compramos o podemos hacer uso del Play Store para descargar nuevas aplicaciones.

❖ **Capa marco de trabajo (framework) de aplicaciones**

Debajo de la capa de Aplicaciones y Widgets, y encima de la capa de bibliotecas nativas y tiempo de ejecución, se sitúa la capa del Framework de aplicaciones.

La capa nos brinda los bloques de construcción de alto nivel que vamos a utilizar para crear las aplicaciones. El Framework viene preinstalado con Android, pero también se puede ampliar con sus propios componentes según sea necesario. Todos los desarrolladores de aplicaciones Android, tienen acceso total al código fuente, así de esta forma, no se generan cientos de componentes de aplicaciones distintas, que respondan a la misma acción, dando la posibilidad de que los programas sean modificados o reemplazados por cualquier usuario sin tener que empezar a programar sus aplicaciones desde el principio.

Las partes más importantes de esta capa son las siguientes:

- Activity Manager: es quien controla el ciclo de vida de las aplicaciones y mantiene un "historial" para la navegación del usuario.
- Content providers: Son los proveedores de contenido, que encapsulan los datos que necesita ser compartidos entre las aplicaciones, como por ejemplo, las listas de los contactos.

- Resource Manager: es el gerente de Recursos, recordando que los recursos son cualquier cosa que acompañe a una aplicación que no sea código fuente.
- Manager de Notificaciones: Nos permite administrar como Eventos los mensajes que llegan y las alertas.

❖ Capa de librerías

Debajo del Framework de aplicaciones está la capa de bibliotecas nativas de Android, que incluye un set de librerías C/C++, que son expuestas a todos los desarrolladores a través del Framework de las aplicaciones. Estas bibliotecas compartidas están escritas en C/C++, compiladas según la arquitectura de hardware particular utilizado por el teléfono, y preinstalado por el proveedor de teléfono.

❖ Capa de runtime de Android

Sobre el Kernel de Linux, encontramos el runtime de Android, que incluye la máquina virtual Dalvik y el núcleo de las bibliotecas Java. Es la capa de tiempo de ejecución Android, incorpora un set de librerías que aportan la mayor parte de las funcionalidades disponibles en las librerías base del lenguaje de programación Java. La Máquina Virtual está basada en registros, y corre clases compiladas por el compilador de Java que anteriormente han sido transformadas al formato .dex (Dalvik Executable) por la herramienta "dx".

Aclarar que el núcleo de las bibliotecas Java que viene con Android son diferentes de las librerías Java Standard Edition (Java SE) y de las bibliotecas Java Mobile Edition (Java ME).

❖ Capa del kernel de linux

El Kernel de Linux es utilizado en Android, es decir, en la base de la arquitectura está el kernel de Linux, proporcionando una capa de abstracción de hardware para Android y el resto del software.

Internamente, Android utiliza Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, red y controladores. El usuario final nunca se entera de que está Linux en la base, gracias a una buena arquitectura en capas.

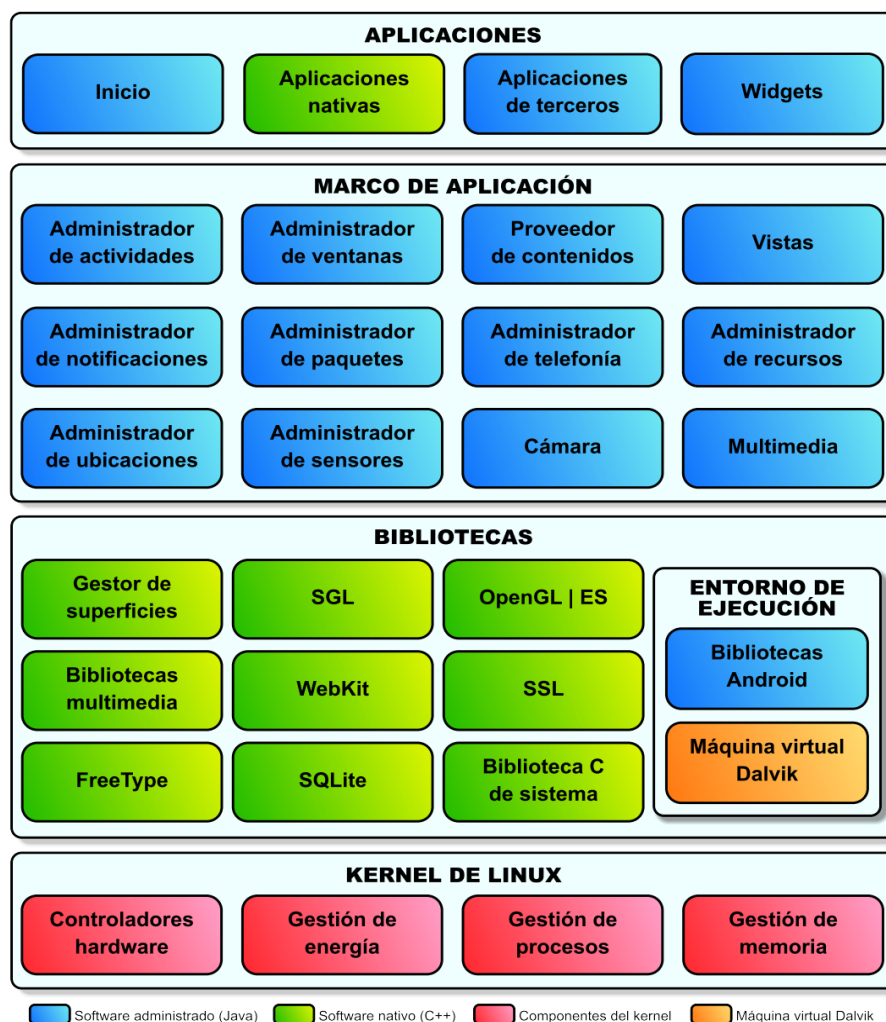


Ilustración 3. Arquitectura de Android

2.3 JSON (JavaScript Object Notation)

JSON es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

En la actualidad, el uso de JSON se ha generalizado como alternativa a XML, incluso Android posee una librería nativa que facilita su uso. Además JSON ahorra considerablemente en el uso de recursos.

Una vez recibidos los datos como una cadena Json procederemos a su procesamiento a través de un **webservice**, basándonos en un par de clases que son **JSONArray** y **JSONObject**, que nos proporcionarán la colección de datos y los datos uno a uno respectivamente.

Al crear una instancia de JSONArray, a partir del JSON leído, obtendríamos un array de JSONObject con todos los elementos anidados en el JSON. Una vez obtenido esto deberemos ir recorriendo el array obteniendo cada JSONObject, pudiendo acceder a sus atributos y procesándolos a nuestro antojo.

2.4 WEB SERVICE

2.4.1 Introducción

Un Web Service es un sistema software diseñado para soportar la interoperabilidad máquina - máquina a través de una red. Este tiene una interfaz descrita en un formato que puede ser procesado por una máquina. Otros sistemas interactúan con el Web Service utilizando mensajes JSON los cuales se encuentran establecidos previamente.

Se podría decir que un Web Service es una comunicación por medio de mensajes JSON entre diferentes equipos a través de una red, en nuestro caso, entre un dispositivo móvil y un ordenador.

2.4.2 Ventajas e inconvenientes

- ✓ Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- ✓ Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- ✓ Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- ✓ El principal inconveniente que tiene es que al apoyarse en HTTP, pueden esquivar medidas de seguridad.

2.5 PHP (Hypertext Pre-processo)

PHP es un lenguaje de código abierto muy popular especialmente adecuado para desarrollo web y que puede ser incrustado en HTML. Lo que distingue a PHP de un lenguaje del lado cliente, como Javascript, es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá los resultados de ejecutar el script, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido.

Su funcionamiento es el siguiente: se realiza una petición de página web al servidor, el servidor recibe la petición, reúne la información necesaria consultando a bases de datos o a otras páginas webs, otros servidores y posteriormente el servidor responde enviando una página web “normal” (estática) pero cuya creación ha sido dinámica (realizando procesos de modo que la página web devuelta no siempre es igual).

2.6 MySQL

MySQL es un sistema gestor de bases de datos relacional, multihilo y multiusuario, más popular de código abierto. Es válido para cualquier sistema operativo. Nace en Suecia en el año 1995 cumpliendo el estándar SQL pero sin sacrificar velocidad, fiabilidad o usabilidad.

Unido al lenguaje PHP, una base de datos en MySQL, permite dinamizar una web, puesto que ofrece la posibilidad de usar el propio contenido de la base de datos en la web.

3. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN

3.1 Introducción

Una vez establecidas las tecnologías que se van a usar en el desarrollo del sistema, se van a definir de forma detallada cada una de las funcionalidades requeridas del mismo, de forma que se va a dar una descripción general de la aplicación para que el lector tenga una visión aproximada de cómo será el sistema final desarrollado.

Por tanto, en este punto se especificarán cuáles son las condiciones generales que tienen el sistema y también sus capacidades. Por último se detallarán todos los requisitos de usuario seguidos, de los casos de uso, en los cuales se mostrarán los distintos usuarios con sus posibles acciones al interactuar con el sistema.

Para finalizar, se detallaran los requisitos software que se han recogido y analizado a partir de lo expuesto anteriormente.

3.2 Planificación

En todo proyecto es muy importante una buena planificación del trabajo, por ello a continuación se muestra un diagrama de Gantt con las fases del mismo.



Ilustración 4: Planificación

3.3 Descripción general

3.3.1 Requisitos de usuario

En el sistema hay dos tipos de usuarios. El primero es el usuario administrador y el segundo es el fisioterapeuta, usuario que va a usar de manera habitual la aplicación. Cada uno de ellos tiene un rol y unos privilegios distintos que hacen que el uso que cada usuario hace de la aplicación sea distinto.

- **Administrador:** Usuario encargado de gestionar y administrar la aplicación. Puede añadir o eliminar tanto administradores como pacientes, en el caso de los pacientes deberá asignar a éstos, alguno de los fisioterapeutas ya dados de alta en el sistema. Es el usuario con más privilegios capaz de acceder a la aplicación.
- **Fisioterapeutas:** Usuarios finales del sistema que mediante la aplicación destinada a este colectivo pueden realizar gestiones de mantenimiento de sus pacientes, así como la posibilidad de dar de alta citas en su agenda. Igualmente tendrán la posibilidad de ver la localización de un paciente determinado y mostrar el camino hasta él.

3.3.2 Entorno operacional

Se deben distinguir entre las necesidades de cada uno de los roles anteriores:

ADMINISTRADOR

Teléfono móvil con sistema operativo Android y conexión a internet de forma inalámbrica, ya sea mediante Wifi o conexión 3G. Se requiere tener instalada la aplicación en el dispositivo desde el que se va a usar. El administrador usará la aplicación de manera eventual, con el fin de realizar el mantenimiento de fisioterapeutas y pacientes.

Tabla 1. Entorno operacional del administrador

FISIOTERAPEUTA

Teléfono móvil con sistema operativo Android y conexión a internet de forma inalámbrica, ya sea mediante Wifi o conexión 3G y GPS. Se requiere tener instalada la aplicación en el dispositivo desde el que se va a usar. El usuario fisioterapeuta hará un uso frecuente de la aplicación. Puesto que las citas serán gestionadas diariamente.

Tabla 2. Entorno operacional del fisioterapeuta

3.3.3 Requisitos de usuario

Tras el análisis de la propuesta inicial, se realiza la evaluación de los requisitos de usuario. En dichos requisitos, el propio usuario indica un conjunto de restricciones o funcionalidades del programa. Se diferencian dos tipos de requisitos:

- Requisitos de capacidad: son aquellos requeridos por el usuario para resolver un problema o determinar un objetivo.
- Requisitos de restricciones: son aquellos dados por los usuarios sobre cómo debe ser resuelto el problema o logrado el objetivo.

Los requisitos de usuario están compuestos por los siguientes campos:

- **Identificador:** Código único que identifica a cada requisito. El requisito de usuario puede ser de capacidad o de restricción y va acompañado de un valor numérico.
- **Título:** Nombre único del requisito.
- **Descripción:** Es un breve comentario, en el que se detalla la especificación del requisito.
- **Necesidad:** Muestra la importancia del requisito desde el punto de vista del cliente. Los valores pueden ser esencial, conveniente u opcional.

Requisitos de capacidad

Identificador	RUC-01	Necesidad	Esencial
Título	Acceso a la aplicación		
Descripción	Un usuario puede acceder a la aplicación introduciendo su nombre de usuario y contraseña, accediendo a la misma dependiendo de sus credenciales a la parte administrador o fisioterapeuta.		

Tabla 3: RUC-01

Identificador	RUC-02	Necesidad	Esencial
Título	Consulta paciente por nombre y apellidos		
Descripción	Un usuario puede consultar sus pacientes filtrándolo por su nombre, nombre y apellidos o solamente los apellidos del paciente.		

Tabla 4: RUC-02

Identificador	RUC-03	Necesidad	Esencial
Título	Alta de un paciente		
Descripción	Un usuario puede dar de alta un paciente introduciendo sus datos en el formulario de alta.		

Tabla 5: RUC-03

Identificador	RUC-04	Necesidad	Esencial
Título	Baja de un paciente		
Descripción	Un usuario puede dar de baja un paciente seleccionado		

Tabla 6: RUC-04

Identificador	RUC-05	Necesidad	Esencial
Título	Modifica un paciente		
Descripción	Un usuario puede modificar los datos de un paciente seleccionado,		

	introduciendo los nuevos datos en el formulario.
--	--

Tabla 7: RUC-05

Identificador	RUC-06	Necesidad	Esencial
Título	Consulta fisioterapeuta por nombre y apellidos		
Descripción	Un usuario puede consultar los fisioterapeutas del sistema filtrándolos por su nombre y apellidos o solamente el nombre o los apellidos del paciente.		

Tabla 8: RUC-06

Identificador	RUC-07	Necesidad	Esencial
Título	Alta de un fisioterapeuta		
Descripción	Un usuario puede dar de alta un fisioterapeuta introduciendo sus datos en el formulario de alta.		

Tabla 9: RUC-07

Identificador	RUC-08	Necesidad	Esencial
Título	Baja de un fisioterapeuta		
Descripción	Un usuario puede dar de baja un fisioterapeuta seleccionado		

Tabla 10: RUC-08

Identificador	RUC-09	Necesidad	Esencial
Título	Modifica un fisioterapeuta		
Descripción	Un usuario puede modificar los datos de un fisioterapeuta seleccionado, introduciendo los nuevos datos del fisioterapeuta en el formulario.		

Tabla 11: RUC-09

Identificador	RUC-10	Necesidad	Esencial
Título	Alta de una cita		
Descripción	Un usuario puede dar de alta una cita para un paciente seleccionado en una fecha y hora también seleccionados		

Tabla 12: RUC-10

Identificador	RUC-11	Necesidad	Esencial
Título	Baja de una cita		
Descripción	Un usuario puede dar de baja la cita de un paciente en una fecha y hora concreta		

Tabla 13: RUC-11

Identificador	RUC-12	Necesidad	Esencial
Título	Consultar una cita		
Descripción	Un usuario puede consultar las citas de una fecha seleccionada en el calendario.		

Tabla 14: RUC-12

Identificador	RUC-13	Necesidad	Esencial
Título	Guiar hasta el paciente		
Descripción	Un usuario puede ser guiado desde su posición actual hasta la posición del paciente seleccionado.		

Tabla 15: RUC-13

Requisitos de restricciones

Identificador	RUR-01	Necesidad	Esencial
Título	Acceso a la aplicación de usuarios registrados		
Descripción	Únicamente podrán acceder a la aplicación aquellos usuarios registrados previamente		

Tabla 16: RUR-01

Identificador	RUR-02	Necesidad	Esencial
Título	Idioma de la aplicación		
Descripción	El idioma de la aplicación será el castellano		

Tabla 17: RUR-02

Identificador	RUR-03	Necesidad	Esencial
Título	Funcionamiento en dispositivos móviles Android		
Descripción	La aplicación PlanningFisio solo funcionará en dispositivos que tengan el sistema operativo Android.		

Tabla 18: RUR-03

Identificador	RUR-04	Necesidad	Esencial
Título	Funcionamiento a través de internet		
Descripción	La aplicación funcionará cuando el dispositivo tenga acceso a Internet, ya sea a través de Wifi o de la red móvil.		

Tabla 19: RUR-04

Identificador	RUR-05	Necesidad	Esencial
Título	Registro manual		
Descripción	El registro de los usuarios del sistema será realizado únicamente por el administrador del mismo.		

Tabla 20: RUR-05

Identificador	RUR-06	Necesidad	Esencial
Título	Guardar usuario y contraseña al inicio		
Descripción	Los datos de usuario y contraseña son almacenados por la aplicación tras el primer acceso, evitando así que el usuario tenga que introducirlos cada vez que quiera acceder a la misma.		

Tabla 21: RUR-06

3.3.4 Actores del sistema

La aplicación tiene dos tipos de actores, los cuales poseerán la facultad de realizar una serie de operaciones en función de su tipo.

Actor	Administrador	A01
Descripción	Usuario que utiliza la aplicación móvil con el fin de gestionar su perfil así como los fisioterapeutas y los pacientes del sistema	
Tipo	Actor principal	
Comentarios	El administrador de la aplicación será uno de los fisioterapeutas	

Tabla 22: A01

Actor	Usuario gestor (fisioterapeuta)	A02
Descripción	Usuario que utiliza la aplicación móvil con el fin de gestionar su perfil, así como sus pacientes y las citas	
Tipo	Actor principal	
Comentarios		

Tabla 23: A02

3.3.5 Casos de uso

Los casos de uso son secuencias de transacciones desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios, es decir, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema.

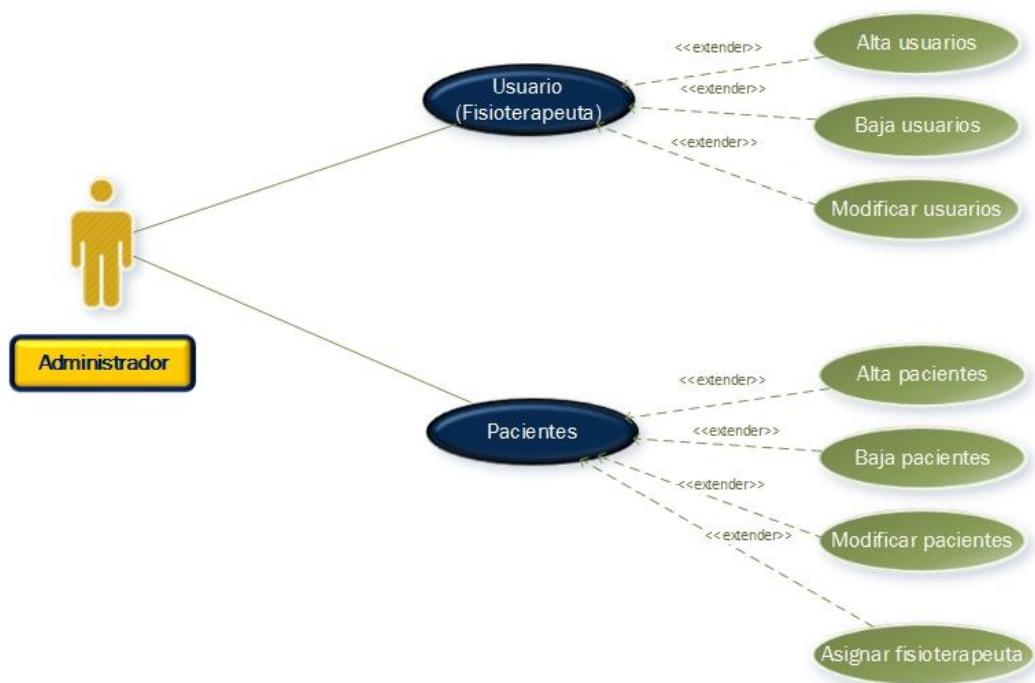


Ilustración 5: Diagrama de casos de uso – Administrador

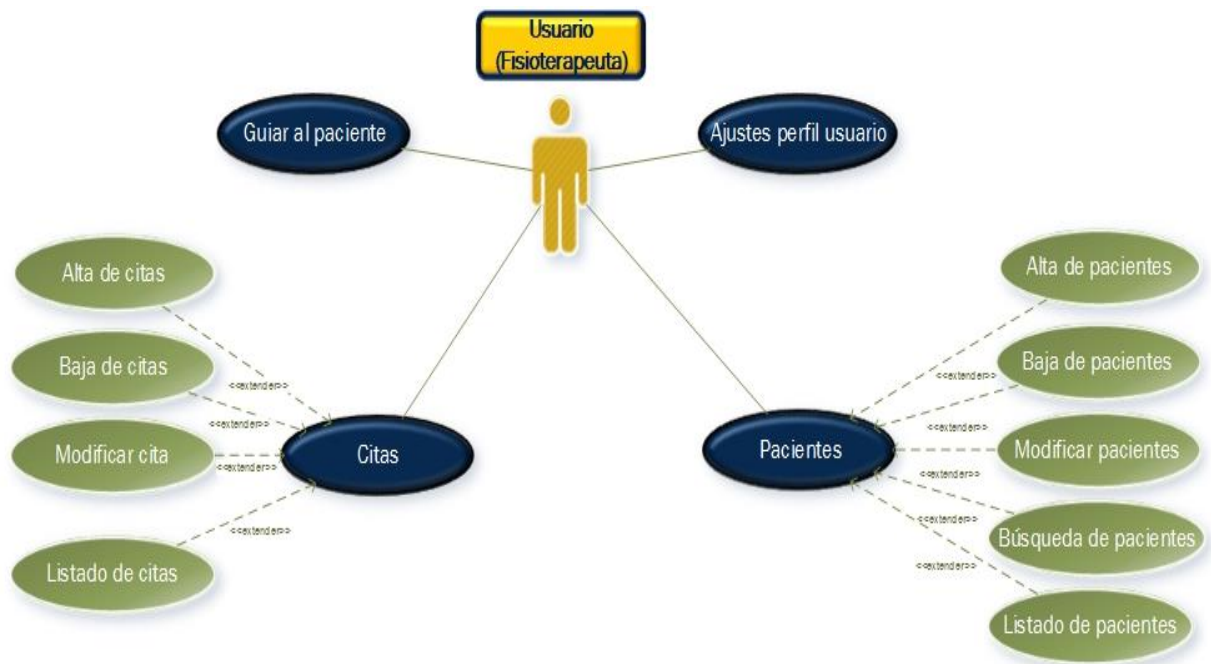


Ilustración 6: Diagrama de casos de uso – Fisioterapeuta

A continuación se muestra una representación textual de los casos de uso, mediante los siguientes atributos:

- **Identificador:** Muestra el caso de uso (CU) acompañado de un valor numérico único.
- **Título:** Frase breve que identifica el nombre del caso de uso.
- **Actores:** Principales actores del caso de uso.
- **Resumen:** Pequeño resumen pretendido por el caso de uso.
- **Precondición:** Condiciones previas que se deben cumplir para que ocurra el caso de uso.
- **Postcondición:** Condiciones que suceden a posteriori de que produzca el caso de uso.
- **Secuencia normal:** Secuencia de ejecución del caso de uso.

Identificador	CU-01
Título	Alta fisioterapeuta
Actores	Administrador
Resumen	Da de alta a una persona como fisioterapeuta
Precondición	La persona no esté dada de alta en la aplicación como fisioterapeuta
Postcondición	La persona está dada de alta en la aplicación como fisioterapeuta
Secuencia normal	La aplicación solicita al usuario los datos del fisioterapeuta El usuario introduce los datos El sistema valida que el fisioterapeuta no está dado de alta en el sistema y lo guarda en la base de datos

Tabla 24: CU-01

Identificador	CU-02
Título	Modificar fisioterapeuta
Actores	Administrador
Resumen	Modifica los datos del fisioterapeuta
Precondición	La persona está dada de alta en la aplicación como fisioterapeuta
Postcondición	Los datos del fisioterapeuta son modificados
Secuencia normal	La aplicación muestra un listado con los fisioterapeutas del sistema y solicita al usuario que seleccione el fisioterapeuta a modificar Una vez seleccionado, el sistema recupera la información del fisioterapeuta El usuario modifica la información del fisioterapeuta El sistema guarda las modificaciones

Tabla 25: CU-02

Identificador	CU-03
Título	Baja de fisioterapeuta
Actores	Administrador
Resumen	Da de baja a una persona como fisioterapeuta
Precondición	La persona está dada de alta en la aplicación como fisioterapeuta
Postcondición	La persona es dada de baja en la aplicación como fisioterapeuta
Secuencia normal	<p>La aplicación muestra un listado con los fisioterapeutas del sistema.</p> <p>La aplicación solicita al usuario que seleccione el fisioterapeuta a eliminar.</p> <p>Una vez seleccionado, el sistema elimina el fisioterapeuta.</p>

Tabla 26: CU-03

Identificador	CU-04
Título	Alta pacientes
Actores	Administrador, Usuario (fisioterapeuta)
Resumen	Da de alta a una persona como paciente
Precondición	La persona no esté dada de alta en la aplicación como paciente
Postcondición	La persona está dada de alta en la aplicación como paciente
Secuencia normal	<p>La aplicación solicita al usuario los datos del paciente</p> <p>El usuario introduce los datos</p> <p>El sistema valida que el paciente no está dado de alta en el sistema y lo guarda en la base de datos</p>

Tabla 27: CU-04

Identificador	CU-05
Título	Modificar pacientes
Actores	Administrador, Usuario (fisioterapeuta)
Resumen	Modifica los datos del paciente
Precondición	La persona está dada de alta en la aplicación como paciente
Postcondición	Los datos del paciente son modificados

Secuencia normal	<p>La aplicación solicita al usuario que seleccione el paciente a modificar</p> <p>Una vez seleccionado, el sistema recupera la información del paciente</p> <p>El usuario modifica la información del paciente</p> <p>El sistema guarda las modificaciones</p>
-------------------------	---

Tabla 28: CU-05

Identificador	CU-06
Título	Baja de pacientes
Actores	Administrador, Usuario (fisioterapeuta)
Resumen	Da de baja a una persona como paciente
Precondición	La persona está dada de alta en la aplicación como paciente
Postcondición	La persona es dada de baja en la aplicación como paciente
Secuencia normal	<p>La aplicación solicita al usuario que seleccione el paciente a eliminar</p> <p>Una vez seleccionado, el sistema elimina el paciente</p>

Tabla 29: CU-06

Identificador	CU-07
Título	Búsqueda de pacientes
Actores	Administrador, Usuario (fisioterapeuta)
Resumen	Realiza una búsqueda de pacientes en base a unos parámetros
Precondición	La persona está dada de alta en la aplicación como paciente
Postcondición	Se recupera un listado con el resultado de la búsqueda
Secuencia normal	<p>La aplicación solicita al usuario que introduzca una serie de parámetros de búsqueda</p> <p>Una vez introducidos, el sistema muestra un listado con los pacientes que cumplen esos parámetros</p>

Tabla 30: CU-07

Identificador	CU-08
Título	Alta citas
Actores	Usuario (fisioterapeuta)
Resumen	Da de alta a una cita de un paciente
Precondición	La fecha y hora de la cita no está ocupada
Postcondición	La cita está dada de alta en la agenda del fisioterapeuta
Secuencia normal	La aplicación solicita al fisioterapeuta la fecha y hora de la cita Selecciona el paciente El sistema valida que la cita no está dada de alta en la agenda y la guarda en la base de datos

Tabla 31: CU-08

Identificador	CU-09
Título	Modificar citas
Actores	Usuario (fisioterapeuta)
Resumen	Modifica los datos de la cita
Precondición	La cita está dada de alta en la agenda del fisioterapeuta
Postcondición	Los datos de la cita son modificados
Secuencia normal	La aplicación solicita al usuario que seleccione la cita del paciente a modificar Una vez seleccionada, el sistema recupera la información de la cita El usuario modifica la información de la cita El sistema guarda las modificaciones en la agenda del fisioterapeuta

Tabla 32: CU-09

Identificador	CU-10
Título	Baja de citas
Actores	Usuario (fisioterapeuta)
Resumen	Da de baja una cita de un paciente
Precondición	La cita está dada de alta en la agenda del fisioterapeuta
Postcondición	La cita es dada de baja en la agenda del fisioterapeuta
Secuencia normal	La aplicación solicita al usuario que seleccione la cita a eliminar Una vez seleccionada, el sistema elimina la cita de la agenda del fisioterapeuta

Tabla 33: CU-10

Identificador	CU-11
Título	Búsqueda de citas
Actores	Usuario (fisioterapeuta)
Resumen	Realiza una búsqueda de citas en la agenda del fisioterapeuta en base a unos parámetros
Precondición	La agenda del fisioterapeuta tiene citas
Postcondición	Se recupera un listado con el resultado de la búsqueda
Secuencia normal	La aplicación solicita al usuario que introduzca una serie de parámetros de búsqueda Una vez introducidos, el sistema muestra un listado con las pacientes que cumplen esos parámetros

Tabla 34: CU-11

Identificador	CU-12
Título	Ajuste Perfil usuario
Actores	Administrador, Usuario (fisioterapeuta)
Resumen	Permite modificar los datos del perfil de usuario

Precondición	El usuario está dado de alta en el sistema
Postcondición	Los datos del usuario son modificados
Secuencia normal	La aplicación muestra los datos del usuario El usuario modifica los datos Una vez introducidos, el sistema guarda los nuevos datos del usuario

Tabla 35: CU-12

Identificador	CU-13
Título	Guiar hasta el paciente
Actores	Usuario (fisioterapeuta)
Resumen	Permite guiar al fisioterapeuta desde su posición al domicilio del paciente seleccionado
Precondición	Existen usuarios fisioterapeutas en el sistema y pacientes
Postcondición	El sistema muestra un mapa con la ruta desde el fisioterapeuta al paciente
Secuencia normal	El usuario accede a la opción Selecciona el paciente hasta el que quiere ir El sistema muestra en un mapa la localización del fisioterapeuta y la ruta hasta el paciente

Tabla 36: CU-13

3.4 Diseño

Tras el análisis y evaluar que se quiere hacer, pasamos a la fase de diseño, en la cual se va a mostrar el diseño arquitectónico que consiste en un diseño detallado de la aplicación, explicando cada una de las decisiones tomadas a la hora de realizar el diseño.

3.4.1 Arquitectura del sistema

Para el desarrollo de la aplicación, además de usar una arquitectura Cliente-Servidor, también nos basamos en el patrón de arquitectura llamado Modelo Vista Controlador (MVC).

A continuación pasamos a ver en que consiste cada uno de los modelos utilizamos.

❖ Modelo Vista Controlador (MVC)

El Modelo Vista Controlador (MVC) consiste en separar los datos de una aplicación, la interfaz de usuario y la lógica de negocios en tres componentes distintos que se relacionarán para al final tener como resultado nuestra aplicación.

Modelo: Nos referimos con modelo a las representaciones que construiremos basadas en la información con la que operará nuestra aplicación, en nuestro caso Base de datos y Web services.

Vista: La vista no es más que la interfaz con la que va a interactuar el usuario. En Android, las interfaces se construyen a través de archivos XML.

Controlador: Son todas esas clases que nos ayudarán a darle vida a las acciones de usuario, realizando una serie de eventos.

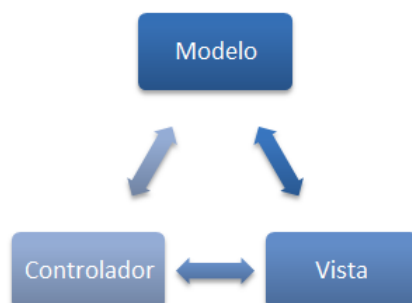


Ilustración 7: Modelo vista controlado (MVC)

❖ Modelo Cliente-Servidor

Este modelo de arquitectura se divide en dos partes claramente diferenciadas. Por un lado el cliente realiza una serie de peticiones a otro programa y el servidor es el encargado de darle la respuesta a las mismas.

Cliente: El cliente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de una red.

Servidor: El servidor es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se le conoce con el término back-end. Normalmente el servidor maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos.

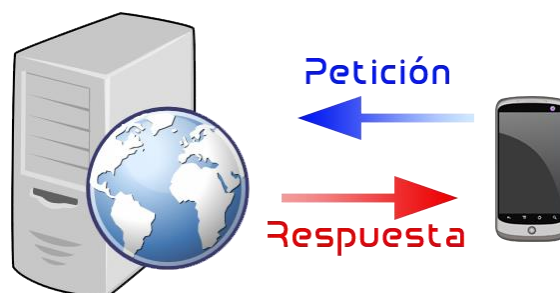


Ilustración 8: Modelo Cliente-Servidor

El sistema se comporta como un Cliente-Servidor puesto que es un sistema distribuido, en que los clientes realizan peticiones mediante JSON en los terminales móviles Android, mientras que la parte del servidor es el servicio web junto con la base de datos que reciben éstas y devuelven respuestas con la información solicitada por los mismos.

El sistema final se compone de un servidor que almacena los datos y gestiona las peticiones y una aplicación cliente utilizada por el administrador y los fisioterapeutas, que realizarán peticiones al servidor con el fin de obtener datos.

En cuanto a la arquitectura MVC el sistema nos indica que la parte de la Vista serán los dispositivos móviles, que en este caso es el cliente Android. Por el contrario, la parte del Servidor contiene al Controlador que será el Servicio Web que se encarga de realizar todas las peticiones y respuestas entre la base de datos y los clientes, y al Modelo que va a ser la base de datos que almacena la información privada de los usuarios y las citas.

Por ello, se ha creado un conjunto de funciones en el Controlador, mediante las cuales se va a gestionar las peticiones que la Vista realiza para obtener datos del Modelo. El modelo resultante sería el siguiente:

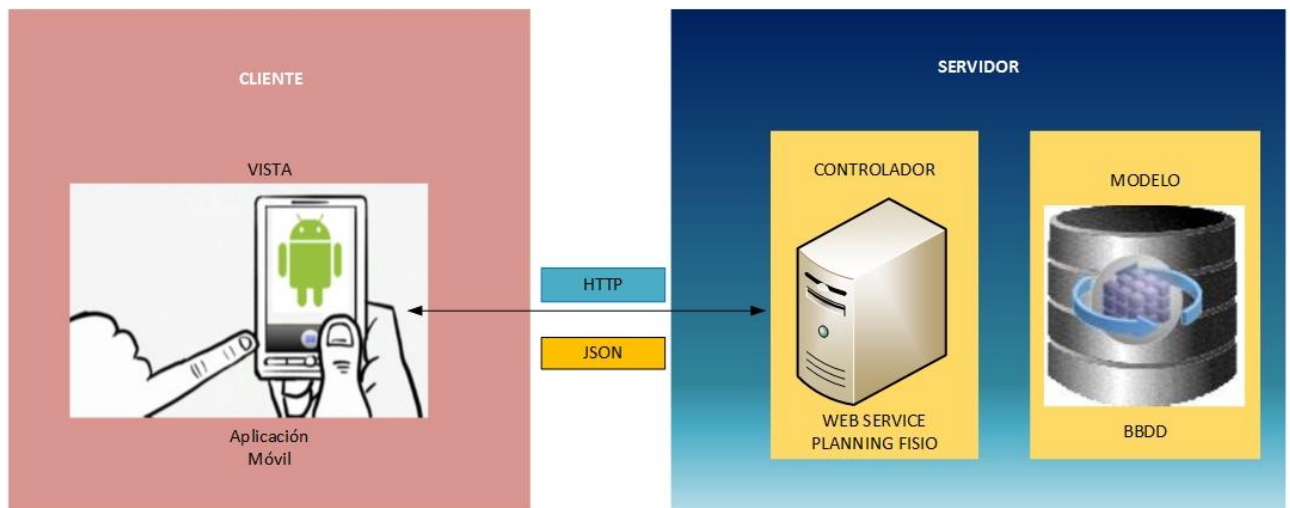


Ilustración 9: Arquitectura del sistema

Por tanto como se puede observar en la ilustración anterior, los pasos a seguir para poder enviar peticiones al servidor y obtener así respuestas son los que se muestran a continuación:

- Primero, el usuario desde la aplicación móvil, enviará una petición al servicio Web.
- La aplicación crea una solicitud HTTP y la envía a la URL donde está alojado el servidor, encapsulada en una petición JSON.
- La petición JSON llega al Servicio Web (Controlador) que la gestiona y dependiendo del tipo de petición, hace una consulta a la base de datos (Modelo).
- El servidor crea una respuesta HTTP que incluye la respuesta JSON del Servicio Web, enviándolo a la dirección desde la que recibió la petición, que en este caso es el dispositivo móvil que envió la petición JSON.
- A continuación, el dispositivo móvil recibe la respuesta del servidor, decodificándola y procesando los datos obtenidos.
- Para finalizar, mostrar los datos.

3.4.2 Diseño de interfaces de usuario

En este apartado se muestra de forma detallada el diseño de la interfaz de la aplicación, que será usada por los usuarios de la aplicación. Se ha intentado que la interfaz de usuario sea limpia y simple de usar, puesto que en caso contrario producirá que la aplicación no sea utilizada.

Para el desarrollo de la interfaz, se ha tenido en cuenta el análisis de requisitos realizado previamente.

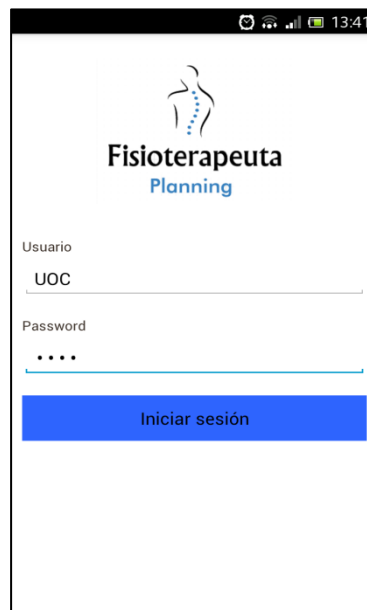


Ilustración 10: Acceso al sistema



Ilustración 11: Menú del usuario "fisioterapeuta"



Ilustración 12: Menú del usuario "administrador"

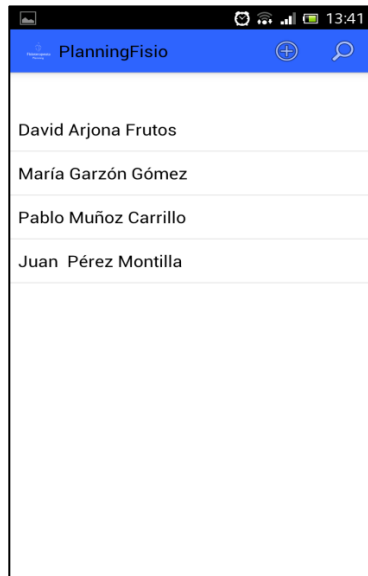


Ilustración 13: Listado de pacientes/fisioterapeutas

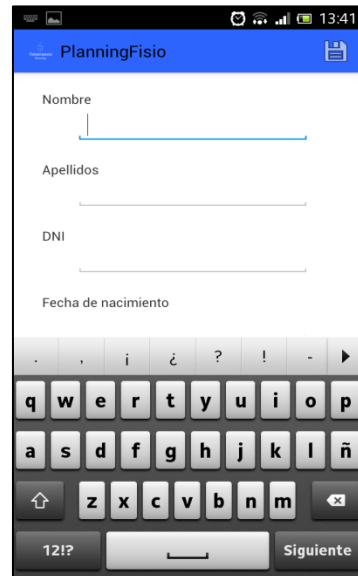


Ilustración 14: Alta de pacientes/fisioterapeutas



Ilustración 15: Selección de fecha

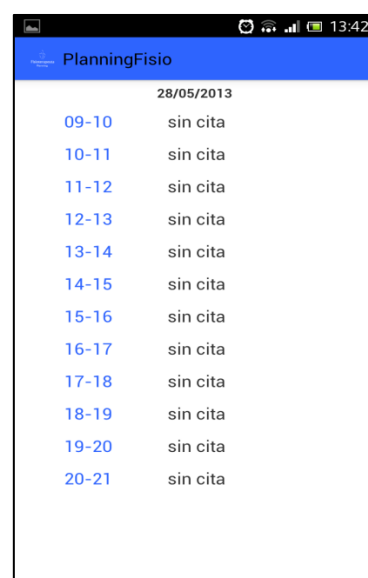


Ilustración 16: Citas de una fecha

A continuación se muestra el diagrama de navegación de la aplicación, con el objetivo de mostrar la navegabilidad entre las distintas pantallas.

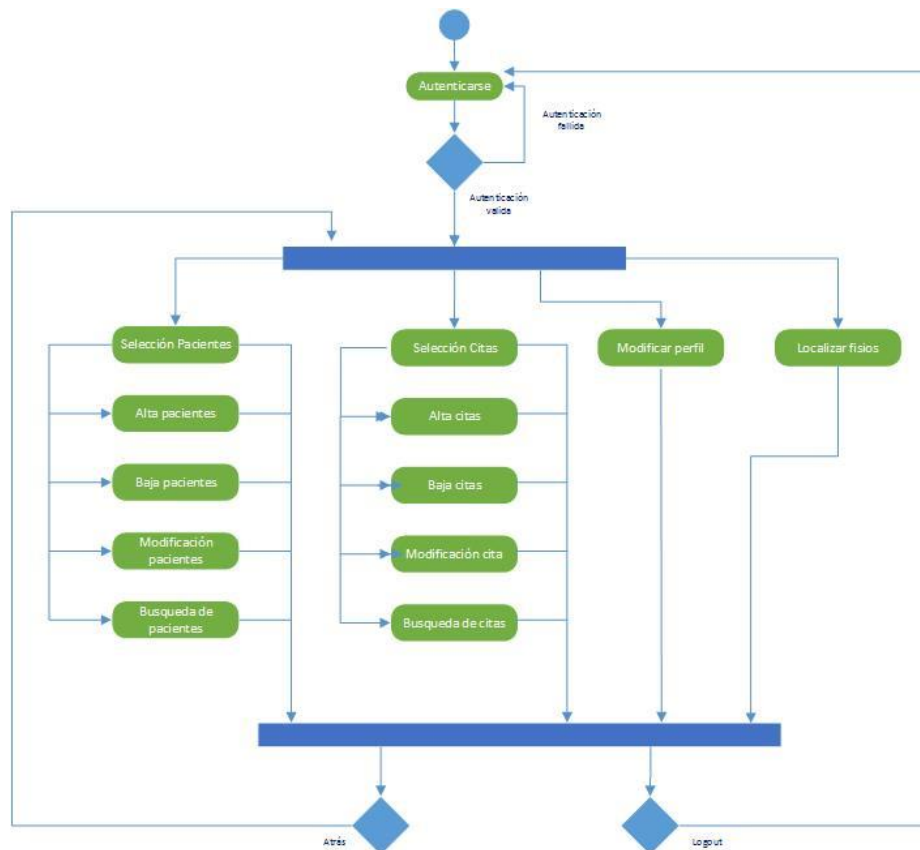


Ilustración 17: Diagrama de navegación

Las diferentes pantallas de una aplicación Android se definen mediante objetos de tipo Activity.

Cada una de las clases que contienen interfaz gráfica y permiten interacción con el usuario extiende de la clase Activity. Estas clases deben declararse en el fichero AndroidManifest.xml. Una Activity es un componente de aplicación que proporciona una interfaz con la cual el usuario puede interactuar y hacer algo. Una aplicación puede tener definidas muchas actividades pero el usuario solo puede interactuar con una a la vez.

La comunicación entre los distintos componentes y aplicaciones en Android se realiza mediante intents, Existen varias variantes del constructor de la clase Intent, cada una de ellas dirigida a unas determinadas acciones, pero en nuestro caso particular vamos a utilizar el intent para llamar a una actividad desde otra de la misma aplicación, para lo que pasaremos al constructor una referencia a la propia actividad llamadora (Login.this), y la clase de la actividad llamada (MFisioterapeuta.class). Luego añadiremos la información al intent mediante el método putExtras.

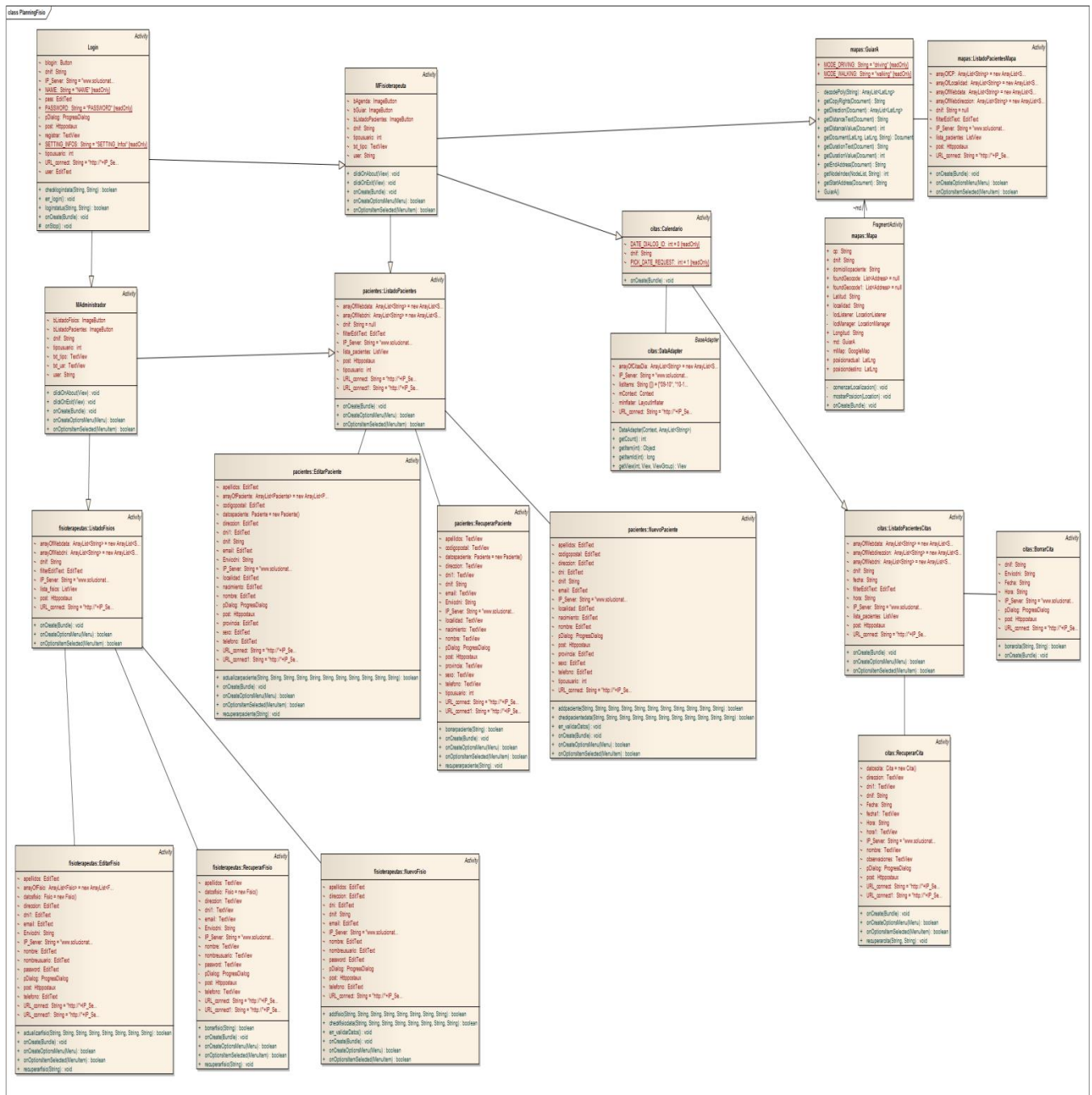
A continuación se muestra el código de la aplicación usado para pasar de la pantalla Login a la pantalla de menú del fisioterapeuta, pasando una serie de parámetros.

```
Intent i=new Intent(Login.this, MFisioterapeuta.class);  
  
    i.putExtra("user",user);  
    i.putExtra("tipo_usuario",tipousuario);  
    i.putExtra("dni_f",dnif);  
    startActivity(i);
```

3.4.3 Diagrama de clases.

Los diagramas de clases son un tipo de diagramas que muestran las clases de un sistema, sus atributos y las relaciones entre ellas.

A continuación se muestra el diagrama de clases del sistema desarrollado:



3.4.4 Servicio web y base de datos

Para realizar la parte web se ha utilizado como lenguaje de programación PHP en la parte de servidor PHP y como Base de Datos MySQL.

Se ha creado un fichero llamado **config.php** en el que se ha definido el nombre de la base de datos, su usuario, password y la ip del servidor donde está alojada.

Igualmente se han creado otra serie de funciones que se comentan a continuación:

- **connectbd.php** → Esta función provee los métodos para conectarse y desconectarse a la bbdd.
- **funciones_bd.php** → Función que provee los procedimientos para interactuar con la bbdd.

Además existen otra serie de funciones que son las encargadas de devolver los datos en notación json.

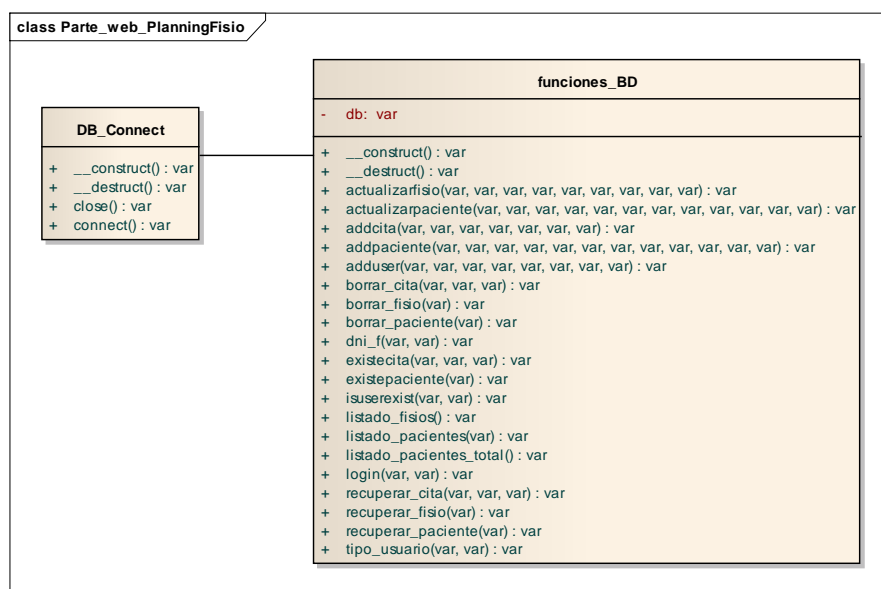


Ilustración 19: Diagrama de clases – Conexión del web service a base de datos

Se significa, que si bien para simular el sistema se han realizado pruebas en modo local utilizando XAMPP, el sistema final se ha ubicado en un servidor web al que se tiene acceso a través de Internet.

3.4.5 Diagrama de bases de datos



Ilustración 20: Diagrama de la base de datos

3.4.6 Gestión de comunicaciones

Se he creado una clase auxiliar con el fin de manejar con mayor facilidad el envío de peticiones a nuestro web service.

La petición de conexión se realizará a través de la función httpconnect, la citada función recibe un array de datos que son los que se usarán para realizar la consulta en la base de datos y la dirección donde se encuentra el web service.

```
httpconnect(parameters,urlwebserver)
```

En nuestro caso la url de conexión quedaría de la siguiente manera:

```
URL_connect="http://" + IP_Server + "/campus/planningfisio/archivo.php"
```

3.5 Implementación

A continuación se muestra un pequeño resumen de los pasos seguidos en la implementación del sistema.

- Creación del servidor: se crea un servidor Apache con el software XAMPP, desde el que se puede crear posteriormente el servicio web y la base de datos que alojará, con el fin, de que se puedan obtener datos a partir de la aplicación cliente.
- Implementación del servicio web: se crea un Servicio Web escrito en lenguaje de programación php, el cual incluye todas las funciones necesarias para comunicarse correctamente a través de peticiones realizadas con la base de datos.
- Creación de la base de datos: se crea una base de datos alojada en el servidor. Esta base de datos, ha sido creada en el sistema de gestión de base de datos MySQL.
- Implementación de la aplicación Android: este paso trata de implementar la aplicación que será usada por el usuario, tanto administrador como fisioterapeuta. Aquí se desarrollará la parte gráfica de la aplicación y además mediante JSON se realizará una serie de solicitudes a partir de una URL para comunicarse con la parte del servidor y se gestionará el resultado de las mismas.

4. CONCLUSIONES

A continuación, en este punto se muestran las conclusiones alcanzadas al desarrollar el trabajo fin de carrera. Dichas conclusiones son las que se muestran a continuación:

Puesto que actualmente el uso de dispositivos móviles y en el caso concreto que nos ocupa de dispositivos con sistema operativo Android está tan extendido, ha sido muy interesante el aprendizaje alcanzado sobre cómo desarrollar aplicaciones para éste tipo de dispositivos.

En el caso que nos ocupa, la herramienta facilitará a la clínica de fisioterapeutas, poder llevar una gestión de pacientes y citas que al ser combinada con la tecnología móvil se convierte en una herramienta muy potente puesto que los fisioterapeutas tratan a los pacientes a domicilio y no siempre disponen de un ordenador en sus visitas.

En el desarrollo del trabajo fin de carrera se han encontrado multitud de problemas, los cuales han sido resueltos con esfuerzo:

- El primero de ellos ha estado centrado en la creación del interfaz de usuario, puesto que ha sido difícil obtener una interfaz atractiva para el usuario y a la vez funcional que permitiese gestionar la aplicación de manera fluida y eficaz y que fuese compatible con múltiples dispositivos.
- El segundo problema ha sido conseguir entender la forma en la que Android se comunica con la parte Servidor, es decir la parte Cliente – Servidor, el uso de JSON, como enviar peticiones y una vez recibida la respuesta por parte del servidor tratar la información, controlando los posibles errores de comunicación entre ambas partes.

No obstante, creo haber cumplido mis objetivos, puesto que he aprendido bastante en el desarrollo del proyecto, desde montaje de servidores, al desarrollo de la aplicación Android propiamente dicha, así como la combinación de ambas.

5. LINEAS FUTURAS

Para completar el sistema, en un futuro se pueden realizar una serie de mejoras las cuales se detallan a continuación:

- Se puede crear un Sistema de Gestión de Contenidos (CMS) dentro del Servicio Web de forma que se pueda gestionar las citas, pacientes y además otra serie de funciones, como puede ser contabilidad de empresa, o página web corporativa en la que se informe a los clientes de noticias y novedades de la clínica.

Igualmente en lo referente a la aplicación Android, también se podría realizar una serie de mejoras:

- Respecto a los pacientes, se podría generar un historial de tratamientos realizados, que daría información al fisioterapeuta sobre las lesiones del paciente.
- Se podría crear una parte cliente, en la que los pacientes pudieran solicitar citas en la agenda de su fisioterapeuta y que estas citas fueran recibidas por el fisioterapeuta mediante una notificación push.

6. REFERENCIAS

- [1] El gran libro de Android. 3ª edición 2013.
Autor: Jesús Tomas Gironés
Editorial: Marcombo
- [2] Android Wikipedia. 2013. Disponible [Internet]:
<http://es.wikipedia.org/wiki/Android>
- [3] Google Developers. 2013. Disponible [Internet]:
<https://developers.google.com/android/>
- [4] JSON Wikipedia. 2013. Disponible [Internet]:
<http://es.wikipedia.org/wiki/JSON>
- [5] Introducing JSON. 2013. Disponible [Internet]:
<http://www.json.org/>
- [6] Tutorial JSON. 2013. Disponible [Internet]:
<http://www.w3schools.com/json/default.asp>
- [7] MVC Wikipedia. 2013. Disponible [Internet]:
http://es.wikipedia.org/wiki/Modelo_Vista_Controlador
- [8] Cliente-Servidor Wikipedia. 2013. Disponible [Internet]:
<http://es.wikipedia.org/wiki/Cliente-servidor>
- [9] Arquitectura Android. 2013. Disponible [Internet]:
<http://www.muchoandroid.com.ar/2011/04/plataforma-android-su-arquitectura.html>