



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Escola Tècnica
Superior d'Enginyeria
Informàtica



etsinf

ESCOLA TÈCNICA SUPERIOR D'ENGINYERÍA INFORMÀTICA
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Aplicación móvil para dar soporte a un sistema de aprendizaje automático para la prevención de la depresión postparto

Trabajo fin de grado

Grado en ingeniería informática

Autor: Germán Molina Francés

Tutor: José Luis Poza Luján

José Alberto Conejero Casares

Curso 2019-2020

Resumen

La depresión postparto afecta a un número elevado de mujeres (entre un 10 y un 15 por ciento). Consecuentemente, esto también afecta al desarrollo del infante y a sus relaciones personales, por ello, es importante contar con un buen método para su detección temprana. Estos métodos van desde los cuestionarios básicos hasta la evaluación por parte de un psiquiatra. Este proyecto consiste en el desarrollo de una aplicación Android que permita monitorizar el estado de pacientes que sufran depresión postparto. Mediante esta aplicación el personal especializado puede acceder a la lista de pacientes que tratan y consultar variada información: cuestionarios realizados, expediente clínico, tratamientos en curso, estadísticas, etc. El objetivo es desarrollar una herramienta que permita tanto consultar información sobre pacientes como configurar los elementos de esta.

Palabras clave: Depresión postparto, DPP, Cuestionarios automatizados, HL-7, Aplicación móvil, Android, Escala de Depresión Postnatal de Edimburgo, EPDS.

Resum

La depressió postpart afecta a un nombre elevat de dones (entre un 10 i un 15 per cent). Conseqüentment, això també afecta al desenvolupament de l'infant i a les seues relacions personals, per això, és important comptar amb un bon mètode per a la seua detecció primerenca. Aquestos mètodes van des dels qüestionaris bàsics fins a l'avaluació per part d'un psiquiatre. Aquest projecte consisteix en el desenvolupament d'una aplicació Android que permeta monitorar l'estat de pacients que patisquen depressió postpart. Mitjançant aquesta aplicació el personal especialitzat pot accedir a la llista de pacients que tracten i consultar variada informació: qüestionaris realitzats, expedient clínic, tractaments en curs, estadístiques, etc. L'objectiu desenvolupar una eina que permeti tant consultar informació sobre pacients com configurar els elements d'aquesta.

Paraules clau: Depressió postpart, DPP, Qüestionaris automatitzats, HL-7, Aplicació mòbil, Android, Escala de Depressió Postnatal d'Edimburg, EPDS.

Abstract

Postpartum depression affects a large number of women (between 10 and 15 percent). Consequently, this also affects the development of the infant and his personal relationships, so it is important to have a good method for its early detection. These methods range from basic questionnaires to evaluation by a psychiatrist. This project consists in the development of an Android application that allows to monitor the status of patients suffering from postpartum depression. Through this application, specialized staff can access the list of patients they treat and consult various information: questionnaires completed, clinical records, treatments in progress, statistics, etc. The objective is to develop a tool that allows both consulting information about patients and configuring its items.

Keywords: Postpartum depression, PPD, Automated questionnaires, HL-7, Mobile app, Android, Edinburgh Postnatal Depression Scale, EPDS.

Índice general

I	MEMORIA	12
1.	Introducción	13
2.	Objeto del proyecto	14
3.	Antecedentes	15
4.	Descripción de la situación actual	17
4.1.	Sistemas similares	17
4.1.1.	eMoods	17
4.1.2.	Moodpath	19
4.1.3.	Youper	19
4.1.4.	Mind Diagnostics	20
4.1.5.	MoodTools	20
4.2.	Análisis de las características	20
4.3.	Conclusión	22
5.	Normas y referencias	23
5.1.	Disposiciones legales y normas aplicadas	23
5.2.	Métodos, Herramientas, Modelos, Métricas y Prototipos	23
5.2.1.	Métodos y Herramientas	23
5.2.2.	Modelos, Métricas y Prototipos	25
5.3.	Mecanismos de control de calidad aplicados durante la redacción del proyecto	26
6.	Definiciones y abreviaturas	27
7.	Requisitos iniciales	29
7.1.	Requisitos funcionales	29
7.2.	Requisitos no funcionales	30
8.	Alcance	31
8.1.	Alcance de la aplicación móvil	31
8.2.	Alcance del sistema	32
9.	Hipótesis y restricciones	34
10.	Estudio de alternativas y viabilidad	35
10.1.	Viabilidad económica	35
10.2.	Viabilidad técnica	35
10.3.	Viabilidad legal	35
11.	Descripción de la solución propuesta	36

12. Análisis de Riesgos	37
13. Organización y gestión del proyecto	39
13.1. Organización	39
13.1.1. Actores del proyecto y relaciones entre los mismos	39
13.1.2. Estructura interna	40
13.1.3. Interfaces externas	40
13.2. Gestión del proyecto	42
14. Planificación temporal	43
15. Resumen del presupuesto	44
15.1. Costes hardware	44
15.2. Costes software	44
15.3. Costes mano de obra	44
15.4. Costes totales	45
16. Conclusiones y trabajo futuro	46
16.1. Dificultades encontradas	46
16.2. Aportaciones del proyecto	46
16.3. Ampliaciones futuras	47
II PROYECTO TÉCNICO	48
17. ANEXOS	49
17.1. Anexo - Análisis del sistema	49
17.1.1. Actores	49
17.1.2. Diagramas de casos de uso	49
17.2. Anexo - Diseño del sistema	59
17.2.1. Capa de presentación	59
17.2.2. Capa de negocio	60
17.2.3. Capa de persistencia	61
17.3. Anexo - Implementación	63
17.3.1. Aplicación móvil	63
17.3.2. API	66
17.3.3. Base de datos	69
17.4. Anexo - Fase de validación y pruebas	70
17.4.1. Pruebas con Postman	70
17.4.2. Pruebas unitarias	70
17.5. Anexo - Despliegue	77
17.5.1. Software necesario	77
17.5.2. Puesta en marcha	77

Índice de figuras

4.1.	Capturas de pantalla de eMood	18
4.2.	Capturas de pantalla de Moodpath	19
8.1.	Diagrama EDT	31
8.2.	Esquema conceptual de sistema completo	33
11.1.	Esquema conceptual	36
13.1.	Estructura interna	40
13.2.	Interfaz externa	41
14.1.	Diagrama de Gantt	43
17.1.	Diagrama de casos de uso de médico	49
17.2.	Diagrama de casos de uso de administrador	53
17.3.	Diagrama de casos de uso de administrador	57
17.4.	<i>Mockups</i> de la aplicación	59
17.5.	Diagrama de secuencia general	61
17.6.	Esquema BD	62
17.7.	Estructura de la aplicación móvil	63
17.8.	Implementación del diseño de la aplicación	64
17.9.	Estructura de la API	66
17.10.	Ejemplo de datos de la BD	69

Índice de tablas

4.1.	Análisis cuantitativo	21
4.2.	Análisis cualitativo	22
5.2.	Análisis de puntos de función	25
12.1.	Análisis de riesgos	38
13.1.	Integrantes del proyecto	39
15.1.	Costes hardware	44
15.2.	Coste total	45
17.1.	Caso de uso M-1	50
17.2.	Caso de uso M-2	50
17.3.	Caso de uso M-3	50
17.4.	Caso de uso M-4	50
17.5.	Caso de uso M-5	51
17.6.	Caso de uso M-6	51
17.7.	Caso de uso M-7	51
17.8.	Caso de uso M-8	51
17.9.	Caso de uso M-9	52
17.10.	Caso de uso M-10	52
17.11.	Caso de uso A-1	53
17.12.	Caso de uso A-2	53
17.13.	Caso de uso A-3	54
17.14.	Caso de uso A-4	54
17.15.	Caso de uso A-5	54
17.16.	Caso de uso A-6	54
17.17.	Caso de uso A-7	55
17.18.	Caso de uso A-8	55
17.19.	Caso de uso A-9	55
17.20.	Caso de uso A-10	55
17.21.	Caso de uso A-11	56
17.22.	Caso de uso A-12	56
17.23.	Caso de uso A-13	57
17.24.	Caso de uso A-14	57
17.25.	Caso de uso A-15	58
17.26.	Caso de uso A-16	58
17.27.	Caso de uso A-17	58
17.28.	Caso de uso A-18	58
17.29.	Caso de uso A-19	59
17.30.	Pruebas unitarias RF01	71
17.32.	Pruebas unitarias RF02	71
17.33.	Pruebas unitarias RF03	71
17.34.	Pruebas unitarias RF04	71

17.35. Pruebas unitarias RF05	72
17.36. Pruebas unitarias RF06	72
17.37. Pruebas unitarias RF07	73
17.38. Pruebas unitarias RF08	74
17.39. Pruebas unitarias RF09	74
17.40. Pruebas unitarias RF10	77
17.41. Pruebas unitarias RF11	77

Índice de código de interés

1.	Invocación del método <code>listarPacientes()</code>	65
2.	Método <code>listarPacientes()</code>	65
3.	<i>Endpoint</i> de la API	67
4.	Función <code>paciente_listarPacientes()</code>	68
5.	Ejemplo de <i>token</i>	68
6.	Ejemplo de tabla de BD	69

Prefacio

Esta memoria ha sido desarrollada en función de la Norma Técnica «Norma CCII-N2016-02 Norma Técnica para la realización de la Documentación de Proyectos en Ingeniería Informática», cumpliendo el estándar AENOR UNE 157001: «Criterios generales para la elaboración formal de los documentos que constituyen un proyecto técnico». Esta Norma Técnica ha sido creada para la documentación de los proyectos de ingeniería informática como elemento esencial para su concepción, dirección y gestión.

La situación extraordinaria vivida durante estos días debido a la pandemia de la COVID-19 ha influido negativamente en el desarrollo de este proyecto. Debido a ello, han surgido problemas durante el desarrollo que han impedido que el proyecto se desarrollara adecuadamente.

Parte I

MEMORIA

1. Introducción

Según un estudio epidemiológico, [7] entre el 10 y el 15 por ciento de mujeres que han dado a luz sufren depresión en el periodo postparto (DPP), afectando al desarrollo del infante y a sus relaciones personales. Además, en la mitad de los casos este desorden persiste durante el año posterior al parto, aumentando el riesgo de volver a sufrir episodios depresivos en el futuro.

Para detectar la DPP, en 1987 Cox et al crearon un cuestionario de 10 preguntas conocido como Edinburgh Postnatal Depression Scale (EPDS) [4]. En 2003 la versión española fue validada con un alto nivel de acierto, convirtiéndose en una herramienta útil para prevenir la DPP [7].

Las preguntas del cuestionario corresponden a varios síntomas de depresión clínica. La evaluación se realiza considerando la puntuación total: una mayor puntuación indica una mayor cantidad de síntomas depresivos. La EDPS puede usarse durante las primeras 8 semanas después del embarazo y también durante este. Esta escala se ha convertido en una herramienta ampliamente usada y ha sido adaptada y validada en muchos idiomas [12].

Tradicionalmente, la paciente debe desplazarse hasta el centro sanitario para reunirse con el médico y poder realizar el cuestionario. Esto supone un gasto de recursos valiosos para las dos partes, destacando aquellos relacionados con el material y el tiempo. Este problema evidencia la falta de herramientas que estén validadas clínicamente, las cuales faciliten estos actos. Es aquí, donde entra en juego la informática y aporta una solución, en la cual se basa este proyecto.

En este documento se presentará el desarrollo de una aplicación Android que permite al personal sanitario llevar un seguimiento de distintas pacientes, consultar información sobre estas y asignar cuestionarios y tratamientos. También será posible acceder a una parte de configuración, solo accesible por usuarios autorizados, donde es posible crear y configurar distintos elementos, como puede ser la creación de nuevos cuestionarios o editar la información de los usuarios.

Esta aplicación es parte de una colaboración con el Hospital Clínico de València, formado por proyectos individuales de varios compañeros con el fin de mejorar la prestación sanitaria a los pacientes.

2. Objeto del proyecto

El objetivo de este trabajo es proporcionar una solución a la gestión automática del proceso de detección de depresión postparto que permita monitorizar el estado de pacientes y gestionar sus elementos más relevantes. Más concretamente, los objetivos específicos a alcanzar son los siguientes:

- Crear cuestionarios personalizados, además de cuestiones y respuestas personalizadas.
- Asignarles cuestionarios para evaluar su estado y tratamientos en respuesta.
- Observar el desarrollo de pacientes a lo largo del tiempo.
- Automatizar y sistematizar el proceso.
- Generar simulaciones que permitan analizar diferentes patrones.
- Gestionar los elementos almacenados en el sistema (usuarios, cuestionarios, tratamientos...).
- Facilitar el almacenamiento de la información y su disponibilidad de forma segura.

3. Antecedentes

La depresión es una enfermedad grave y común que nos afecta física y mentalmente en nuestro modo de sentir y de pensar. La depresión nos puede provocar deseos de alejarnos de nuestra familia, amigos, trabajo, y escuela. Puede además causarnos ansiedad, pérdida del sueño, del apetito, y falta de interés o placer en realizar diferentes actividades. Entre sus variantes encontramos la depresión postparto.

La depresión postparto es la presencia de síntomas depresivos que duran una vez se ha producido el parto, más de dos semanas, y que interfieren en las actividades de la vida diaria. Como hemos mencionado en la introducción, la depresión postparto aparece en el 10 al 15 por ciento de las mujeres después del parto. Aunque puede suceder a cualquier madre, las de riesgo más elevado son las que presentan los siguientes antecedentes:

- Tristeza postparto o *Baby blues*.
- Episodios previos de depresión postparto.
- Diagnóstico previo de depresión.
- Antecedentes familiares de depresión.
- Factores estresantes significativos de la vida diaria.
- Falta de apoyo de la pareja o de los miembros de la familia.
- Antecedentes de alteraciones del estado de ánimo asociados temporalmente con los ciclos menstruales o el uso de anticonceptivos orales.
- Malos resultados obstétricos previos o actuales.
- Ambivalencia previa o vigente sobre el embarazo actual.

La etiología exacta de la depresión postparto es desconocida; sin embargo, la depresión previa es un riesgo mayor, y los cambios hormonales durante el puerperio, la privación del sueño y la susceptibilidad genética pueden contribuir a su desarrollo.

Los síntomas de la depresión postparto pueden ser similares a los de la depresión mayor y pueden incluir:

- Tristeza extrema.
- Culpa.
- Llanto incontrolable.
- Insomnio o sueño excesivo.
- Pérdida o exceso del apetito.

- Irritabilidad e ira.
- Cefaleas y dolores corporales.
- Cansancio extremo.
- Preocupaciones poco realistas sobre el bebé o desinterés.
- Una sensación de ser incapaz de cuidar el bebé o de no cumplir las expectativas de los que se espera de una *buena* madre.
- Temor de dañar al bebé.
- Ideación suicida.
- Ansiedad o ataques de pánico.

Normalmente, los síntomas aparecen de forma insidiosa durante 3 meses, pero el inicio puede ser más brusco. La depresión postparto interfiere en la capacidad de la mujer de cuidar de sí misma y del bebé. La psicosis es rara, pero la depresión postparto aumenta el riesgo de suicidio y de infanticidio, que son las complicaciones más graves. Las mujeres pueden no sentirse unidas con sus bebés, lo que produce problemas emocionales, sociales y cognitivos en el niño más adelante. Sin tratamiento, la depresión postparto puede resolverse espontáneamente o volverse crónica [9]. El riesgo de prevalencia es de 1 en 3, variando significativamente según los ingresos familiares. [8].

Como hemos mencionado en la introducción, los cuestionarios de salud se vienen realizando por personal cualificado en centros sanitarios de manera presencial. Con el desarrollo de las TIC, existe la oportunidad de poder liberar recursos de atención sanitaria automatizando y sistematizando procesos, como sería el paso de estos cuestionarios. Además de los ahorros en coste de personal, se dispondría de la información antes, se facilitaría su almacenamiento y posterior acceso, así como la protección de los datos contenidos, mediante el control del acceso.

Los médicos manejan gran cantidad de información de diferentes pacientes que se va generando a lo largo del seguimiento. El esfuerzo adicional para conseguir organizarla total y correctamente se traduce en una pérdida de eficiencia.

Por parte de la paciente, el seguimiento tradicional conlleva esfuerzos que podría ahorrarse, como por ejemplo el desplazamiento al centro.

Con el desarrollo del sistema presentado en este trabajo, se pretende aportar una alternativa que permita a todas las partes una mejor eficiencia y facilidad.

4. Descripción de la situación actual

Si buscamos en Play Store aplicaciones relacionadas con la depresión, encontraremos desde algunas que ofrecen información sobre cómo sobrellevarla y tratarla, hasta otras más profesionales que permiten llevar un seguimiento sobre la evolución de una persona.

En estos subapartados expondremos aplicaciones similares a la que vamos a desarrollar, describiendo sus principales funcionalidades y posteriormente realizaremos una comparativa entre estos, tanto cuantitativa como cualitativa. Observando las características que poseen las herramientas actuales podremos establecer cuáles son interesantes y cuáles no para nuestra aplicación.

La búsqueda de estas aplicaciones se realizó en Play Store (plataforma de distribución de aplicaciones Android). Los términos de búsqueda eran depresión, embarazo, postparto y EPDS.

4.1. Sistemas similares

4.1.1. eMoods

eMoods se trata de una aplicación que permite a una persona llevar un registro sobre su estado psíquico a lo largo del tiempo. El usuario indica datos sobre el paso del día: estados, horas de sueño y anotaciones. Las características más destacables son:

- **Registros personalizados:** la aplicación permite crear registros a medida donde se pueden establecer qué estados se quieren recoger en el registro e información relevante.
- **Anotaciones:** dentro de un registro es posible añadir notas diarias, notas con fecha y hora e incluso medicamentos, indicando si se tomaron ese día o no.
- **Historial de registros:** la aplicación guarda todos estos registros para más tarde poder visualizarlos. Se guarda un registro por día.
- **Gráficas:** los datos recogidos en los registros a lo largo del tiempo, son visualizados en unas gráficas, mostrando datos de los últimos 30 días o de un mes en concreto.

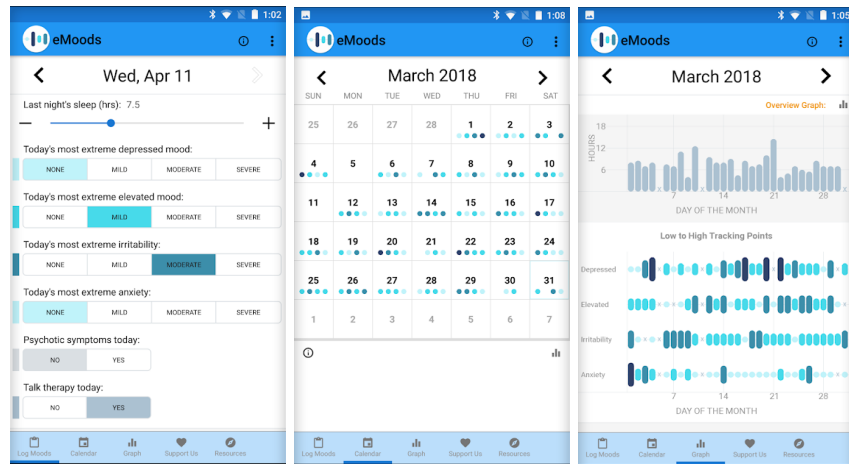


Figura 4.1: Capturas de pantalla de eMood

Tiene una versión gratuita que prácticamente permite utilizar todas sus funcionalidades. Los registros personalizados y las notas con fecha y hora forman parte de la versión de pago.

4.1.2. Moodpath

Moodpath realiza un seguimiento del estado del ánimo a través del tiempo. Consiste en responder una serie de preguntas e indicar emociones que se han sentido o experiencias. También permite añadir como notas pensamientos que has tenido. La aplicación recoge estas respuestas por la mañana, por la tarde y al final del día, pueden responderse varias veces pero la aplicación solo contará la última del periodo en el cual se responden. Después de catorce días se generará un informe a partir de los datos recogidos en las respuestas.

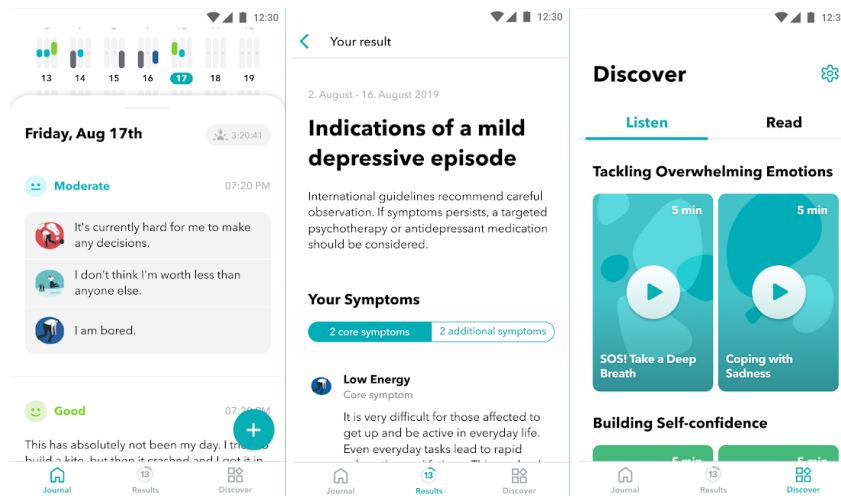


Figura 4.2: Capturas de pantalla de Moodpath

La aplicación también aporta información sobre la depresión y tiene una sección con recursos multimedia de autoayuda.

4.1.3. Youper

El gran aporte de esta aplicación es la interacción con una IA que, a través de preguntas, nos invita a describir cómo nos sentimos y qué relación guarda con nuestro entorno. El objetivo es llevar un historial de los estados de ánimo de una persona y qué los provoca, para así poder aconsejar sobre cómo sobrellevarlos de la mejor manera. También dispone de diferentes pruebas para detectar posibles trastornos y pruebas sobre la personalidad.

Tiene una versión gratuita pero son pocas las funcionalidades disponibles. Un aspecto a destacar es que (según comentarios de Play Store) a la gente que utiliza la aplicación le ayuda mantener una conversación donde poder expresar sus sentimientos, aunque sea con una IA.

4.1.4. Mind Diagnostics

Mind Diagnostics es una aplicación que dispone de una gran cantidad de tests sobre diferentes trastornos. Dispone de un total de 25 tests entre los cuales se encuentra el EPDS. Se puede visualizar un historial con los resultados y consultar información sobre cada trastorno. Adicionalmente, dispone de un chat para hablar con personal cualificado.

Un punto a destacar es que la aplicación permite acceder a todas sus funcionalidades de forma gratuita.

4.1.5. MoodTools

MoodTools es una aplicación enfocada a detectar la depresión a través del cuestionario PHQ-9. Las funcionalidades que encontramos son las siguientes:

- Realizar test y consultar un historial.
- Información sobre la depresión.
- Contenido multimedia de ayuda.
- Un diario donde apuntar los pensamientos que tenemos a lo largo del día.
- Un historial donde apuntar actividades que hemos realizado detallando cómo nos sentíamos antes y después de realizarla.
- Crear un plan de autoayuda en caso de padecer un episodio de crisis.
- Contactar con profesionales para solicitar ayuda o realizar terapia online.

Todas estas funcionalidades son gratuitas. La versión de pago añade funcionalidades estéticas y garantiza la privacidad y seguridad de los datos personales.

4.2. Análisis de las características

A continuación, realizaremos el análisis cuantitativo (características objetivas) y cualitativo (características subjetivas) de las aplicaciones expuestas y concluiremos con un análisis sobre el entorno actual.

La tabla 4.1 muestra un análisis de las características medibles. De las múltiples características se ha decidido analizar las siguientes:

- **Registro:** opción de autenticarse en la aplicación mediante credenciales.
- **Seguimiento:** posibilidad de realizar un seguimiento sobre la persona a lo largo del tiempo y así poder evaluar su evolución.
- **Interacción con personal experto:** opción de comunicarse con personal experto en la depresión postparto que preste apoyo.

- **Informes personalizados:** posibilidad de crear informes a medida donde se refleje la información que deseamos recabar.
- **Coste mensual:** coste en euros de la aplicación por mes.

Para estas características se ha contemplado medir si contienen o no dichos valores (Sí/No). El análisis cuantitativo es el siguiente:

Análisis cuantitativo					
Aplicación	Registro	Seguimiento	Interacción con personal experto	Informes personalizados	Coste mensual
eMoods	No	Sí	Sí	Sí	1,09€
Moodpath	No	Sí	Sí	No	4,49€
Youper	Sí	Sí	No	Sí	9,99€
Mind Diagnostics	No	Sí	Sí	No	0€
MoodTools	No	Sí	Sí	No	2,09€

Tabla 4.1: Análisis cuantitativo

Como podemos observar solamente una aplicación permite registro. Un registro mediante credenciales aporta seguridad en el acceso a los datos de la aplicación. También permite acceder mediante diferentes dispositivos, cargando la información correspondiente al usuario registrado.

Todas permiten llevar un seguimiento, característica fundamental para el entorno de DPP ya que de otra manera no se podría observar la evolución de la paciente. Además, menos una de ellas, permiten la interacción con personal experto, característica bastante interesante dentro del entorno de DPP.

Por último, solo dos permiten crear informes personalizados. En la aplicación a desarrollar, esta característica aportaría un gran valor, ya que aportaría flexibilidad pudiendo evaluar el estado de una paciente mediante cuestionarios personalizados.

La siguiente tabla (4.2) muestra el análisis de las características subjetivas. De las múltiples características se ha decidido analizar las siguientes:

- **Apariencia profesional:** percepción que se obtiene de la aplicación, teniendo en cuenta que haya sido desarrollada por un equipo cualificado. Niveles de medición: baja, media y alta.

- **Contenido:** se considera como el total de funcionalidades que ofrece la aplicación considerando su coste mensual. Niveles de medición: muy malo, malo, aceptable, bueno y muy bueno.
- **Utilidad:** se contempla como la eficacia del uso de la aplicación en la detección de la DPP. Niveles de medición: inútil, poco útil, útil y muy útil.
- **Facilidad de uso:** nivel de usabilidad de la aplicación. Niveles de medición: fácil, media y alta.

El análisis cualitativo es el siguiente:

Análisis cualitativo				
Aplicación	Apariencia profesional	Contenido	Utilidad	Facilidad de uso
eMoods	Alta	Muy bueno	Útil	Medio
Moodpath	Alta	Bueno	Poco útil	Fácil
Youper	Alta	Aceptable	Poco útil	Fácil
Mind Diagnostics	Alta	Bueno	Útil	Fácil
MoodTools	Media	Aceptable	Poco útil	Medio

Tabla 4.2: Análisis cualitativo

En cuanto a utilidad, solo encontramos a eMoods y Mind Diagnostics como destacables que sirvan para la detección de la DPP. Las funcionalidades que aportan las diferencian de las demás convirtiéndolas en las más útiles.

Observando las demás características, todas ellas presentan un nivel alto a excepción de una o dos. Un aplicación con un nivel alto de usabilidad será más satisfactoria en su uso. La apariencia profesional aporta más confianza al usuario a la hora de hacer uso de la aplicación y, también atrae más que la proporción de contenido por coste sea lo más alta posible.

4.3. Conclusión

No encontramos en el mercado ninguna aplicación que pueda suplir el problema expuesto en el apartado 3 «Antecedentes». Ninguna de las aplicaciones está adaptada para ser usada por personal sanitario, todas ellas están dirigidas a un usuario paciente, siendo él mismo el encargado de realizar el seguimiento y contactar con profesionales en caso de padecer algún trastorno.

5. Normas y referencias

5.1. Disposiciones legales y normas aplicadas

En este apartado listaremos las disposiciones legales y normas más relevantes que se han aplicado al proyecto. Han sido de aplicación aquellas relacionadas con la protección de datos, como también normas y estándares de la propia memoria y de archivos electrónicos del sistema.

- **Ley Orgánica 3/2018**, de 5 de diciembre, de protección de Datos Personales y garantía de los derechos digitales.
- **Real Decreto 994/1999**, de 11 de junio, por el que se aprueba el Reglamento de medidas de seguridad de los ficheros automatizados que contengan datos de carácter personal. (BOE 26.06.1999)
- **Norma CCII-N2016-02** Norma Técnica para la realización de la Documentación de Proyectos de Ingeniería Informática.
- **Guía para el desarrollo de documentos CDA.**

5.2. Métodos, Herramientas, Modelos, Métricas y Prototipos

5.2.1. Métodos y Herramientas

A continuación, enumeraremos las herramientas utilizadas para llevar a cabo este trabajo:

- **Java**: lenguaje de programación de gran extensión. Orientado a objetos e independiente de la plataforma son algunas de sus características. En este trabajo ha sido el lenguaje utilizado para el desarrollo de la aplicación.
- **ExpressJS**: es un framework para Node.js utilizado para la creación de webs y APIs. En nuestro sistema se utiliza para la creación de una API Rest que permita comunicar la aplicación móvil con la BD a través de peticiones https.
- **Android Studio**: entorno de desarrollo integrado (IDE) utilizado para el desarrollo de la aplicación. Es el IDE oficial para el desarrollo de aplicaciones Android, basado en IntelliJ IDEA. Contiene características de IntelliJ además de otras concretas en su contexto de uso, como por ejemplo un emulador de aplicaciones.
- **Visual Studio Code**: es el editor de código fuente utilizado para, junto con el framework ExpressJS, desarrollar el backend del sistema. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable.

- **MySQL:** sistema de gestión de base de datos relacional utilizado para asegurar la persistencia de los datos del sistema. Está considerada como una de las bases de datos de código abierto más populares del mundo para entornos de desarrollo web. Para su desarrollo y administración se ha utilizado la aplicación gráfica MySQL Workbench.
- **Lucidchart:** herramienta basada en la web que permite crear distintos tipos de diagramas y prototipos. Contiene una gran cantidad de plantillas y figuras y permite compartir fácilmente los proyectos con otros usuarios. En nuestro caso se ha utilizado para diseñar la BD y la aplicación.
- **GanttProject:** se trata de un programa de código abierto con licencia GPL escrito en Java. Es una herramienta fácil de usar y con una interfaz amigable. En este trabajo ha sido la herramienta utilizada para el diseño del diagrama de Gantt.
- **Git:** software utilizado para llevar a cabo un control de versiones del proyecto Android con la posibilidad de recuperar el código fuente en caso de pérdida. Su propósito es llevar un registro de los cambios en archivos y coordinar el trabajo que varias personas realizan sobre archivos compartidos.
- **Postman:** herramienta que permite el envío de peticiones HTTP rest sobre APIs de forma sencilla y con múltiples opciones de configuración. Con esta herramienta hemos realizado pruebas sobre la API para validar su correcto funcionamiento.

5.2.2. Modelos, Métricas y Prototipos

En este apartado se detallan los métodos utilizados para desarrollar los cálculos y estimaciones del proyecto. En nuestro caso se ha utilizado el modelo de análisis de puntos de función (FPA). Para realizar los cálculos y establecer los valores de los puntos de función y de las horas por punto de función se ha seguido las recomendaciones de la asociación IFPUG. En la siguiente tabla se muestran los detalles de este cálculo.

Funcionalidad	Tipo de componente	Puntos de función
RF01	Consulta externa	3
RF02	Salida externa	4
RF03	Salida externa	4
RF04	Salida externa	4
RF05	Salida externa	4
RF06	Consulta externa	3
RF08	Salida externa Entrada externa	7
RF09	Consulta externa	4
RF10	Salida externa Entrada externa	7
RF11	Entrada externa	3

Tabla 5.2: Análisis de puntos de función

Observando los detalles de la tabla establecemos que el total de puntos de función sin ajustar (PFSA) es 43. Después de obtener el factor de reajuste, con valor de 25, el resultado final es de 39 PFA.

La IFPUG considera que 1 punto de función equivale a 8 horas por persona desarrollando en un lenguaje de 4ª generación. Aplicando esta recomendación el coste en horas del proyecto sería de 312 horas. Obtenemos un resultado que se adapta al tiempo enmarcado en un TFG, sin embargo, este cálculo puede resultar impreciso al ser la primera vez que se desarrolla utilizando las herramientas especificadas en el apartado 5.2.1 «Métodos y Herramientas».

5.3. Mecanismos de control de calidad aplicados durante la redacción del proyecto

Esta memoria está redactada según la norma «*CCII-N2016-02 Norma Técnica para la realización de la Documentación de Proyectos en Ingeniería Informática*». La norma hace énfasis especial en la integridad de la documentación del proyecto y en su redacción clara, concisa y entendible por todos los participantes e interesados en el proyecto. Otro aspecto que incide en la calidad de la documentación es su completitud con el fin de evitar omisiones.

6. Definiciones y abreviaturas

A continuación, se describirán definiciones y abreviaturas que se encuentran en este documento. Se omitirán aquellos términos que se asuman como conocidos.

Definiciones:

- **Cuestionario:** herramienta de investigación que consiste en una serie de preguntas cerradas para extraer determinada información de un grupo de personas.
- **Cuestión:** pregunta cerrada que forma parte de un cuestionario. Está compuesta por múltiples respuestas en la que cada una tiene asociada una puntuación.
- **Depresión postparto:** depresión de moderada a intensa en una mujer después de dar a luz. Se puede presentar poco después del parto o hasta un año más tarde.
- **Baby blues:** tristeza puerperal o postparto, que afecta a un 80-85 por ciento de las madres durante los días que siguen al parto, probablemente debido a los cambios hormonales que están sufriendo en este momento crucial de sus vidas, aunque también al cansancio, los nervios y la falta de sueño que acarrea la nueva situación.
- **Expediente clínico:** en nuestro contexto, hacemos referencia a este término como el documento electrónico basado en el estándar CDA el cual contiene información clínica sobre la paciente.
- **Súper-usuario:** usuario médico con permisos adicionales. Este usuario puede acceder a funcionalidades adicionales relacionadas con la gestión de los elementos del sistema.

Abreviaturas:

- **DPP:** depresión post-parto.
- **EPDS:** Edinburgh Postnatal Depression Scale. Cuestionario que consta de 10 preguntas tipo test, creado para identificar mujeres que sufren depresión postparto.
- **CDA HL-7:** Clinical Document Architecture. Estándar ANSI que propone una estructura de documentos en formato XML. El CDA convierte a los documentos clínicos en objetos interpretables por multitud de aplicaciones y transferibles a través de cualquier medio electrónico.

- **HL-7:** Health Level Seven. Conjunto de estándares para facilitar el intercambio electrónico de información clínica que utiliza una notación formal de UML y XML.
- **PHQ-9:** Patient Health Questionnaire. Cuestionario compuesto por 9 preguntas destinado a la detección de depresión severa.

7. Requisitos iniciales

En este apartado pasaremos a describir los requisitos funcionales y no funcionales del sistema. Los requisitos funcionales se expresarán con el código RFxx, siendo xx el número de requisito, para los no funcionales la nomenclatura será similar: RNFxx.

7.1. Requisitos funcionales

A continuación, se muestran listados los requisitos funcionales, agrupados según el actor implicado:

Comunes

- **RF01:** Acceso identificativo a la aplicación.

Médico

- **RF02:** La aplicación mostrará un listado de las pacientes asignadas al usuario médico.
- **RF03:** El médico podrá seleccionar una paciente y acceder a una pantalla donde se muestren los tratamientos asignados y cuestionarios realizados.
- **RF04:** El médico podrá asignar a una paciente cuestionarios para que los cumplimente, siempre que no lo tenga ya asignado.
- **RF05:** El médico podrá asignar a una paciente un tratamiento. Se guardará la fecha de asignación.
- **RF06:** El médico podrá visualizar el expediente clínico de la paciente.

Administrador

- **RF07:** Si el médico que inicia sesión tiene permisos de súper-usuario, podrá acceder a una pantalla donde se encuentren las funcionalidades adicionales.
- **RF08:** Gestión de usuarios: el súper-usuario podrá listar los usuarios que hay en el sistema, consultar información sobre estos, editarla y crear nuevos.
- **RF09:** Desde el listado de pacientes del súper-usuario, este podrá asignar a una paciente los médicos encargados de realizar un seguimiento.
- **RF10:** Gestión de cuestionarios: el súper-usuario podrá listar cuestionarios, cuestiones y respuestas. También crear respuestas, cuestiones (a partir de las respuestas disponibles en el sistema) y cuestionarios (a partir de las cuestiones disponibles en el sistema).

- **RF11:** El súper-usuario podrá generar resultados ficticios de pacientes ficticias para su estudio.

7.2. Requisitos no funcionales

- **RNF01 Usabilidad:** la interfaz de la aplicación deberá ser diseñada con el propósito de resultar sencilla e intuitiva para el usuario.
- **RNF02 Persistencia:** los datos mostrados en la aplicación deben ser consistentes con los almacenados en la BD.
- **RNF03 Interoperabilidad:** la aplicación intercambiará la información con el backend en formato json, de esta manera otros sistemas pueden interactuar con este, como por ejemplo, la aplicación desde la cual las pacientes realizan los cuestionarios.
- **RNF04 Compatibilidad:** la aplicación deberá ser compatible con el dispositivo Android objetivo, tanto en su sistema operativo como en su visualización.
- **RNF05 Seguridad:** las comunicaciones entre frontend y backend se realizarán mediante peticiones *https*, además, solamente los usuarios registrados tendrán acceso a la aplicación.

8. Alcance

8.1. Alcance de la aplicación móvil

Para el presente proyecto, se ha optado por organizarlo por paquetes de trabajo (WP), donde cada uno está compuesto por tareas, y el resultado de cada uno consta de un entregable del cual algunos paquetes dependerán para comenzar. En total encontramos seis paquetes compuestos por sus respectivas tareas. En la siguiente imagen (8.1) de la estructura de descomposición del trabajo (EDT) podemos observar gráficamente su composición.

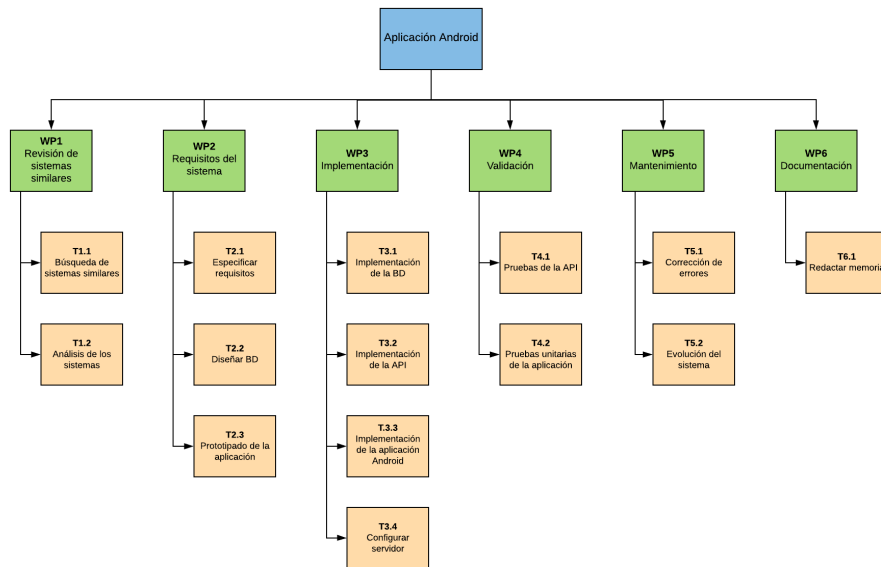


Figura 8.1: Diagrama EDT

En el primer paquete (WP1) encontramos dos tareas. Primeramente se realizó una búsqueda sobre sistemas similares al propuesto para la solución, después analizamos sus características comparando estos sistemas entre ellos y qué aportaban. El resultado de este análisis lo encontramos en el capítulo 4 «Descripción de los situación actual».

El segundo paquete (WP2) está compuesto por tres tareas. En la primera de ellas se especificó los requisitos que debería tener el sistema a desarrollar (este paquete dependía del anterior). Con los requisitos especificados se pasó al diseño de la BD y el prototipado de la aplicación.

En el tercer paquete (WP3) comenzó a implementarse el sistema a desarrollar, empezando primero por la BD, después pasando por la API y finalmente

la aplicación. Una vez la API y la aplicación estuvieron preparadas para su despliegue comenzó la cuarta tarea, relacionada con la configuración del servidor.

En el cuarto paquete (WP4) encontramos tareas relacionadas con el testeo de la aplicación, para la parte de la API y de la aplicación Android. Este paquete se realiza en paralelo con el tercero.

El quinto paquete (WP5) está compuesto por dos tareas. La primera de ellas se realiza en paralelo junto con el cuarto paquete, mientras se aplicaban pruebas al sistema se corregían los fallos que pudieran surgir. La segunda tarea está relacionada con el aporte de nuevas funcionalidades o de mejoras en existentes.

Finalmente en el sexto paquete (WP6) se encuentra la tarea relacionada con la redacción de esta memoria. Esta tarea se realiza en paralelo con los demás paquetes, ya que desde el inicio comenzó a generarse documentación que se aportaría a la memoria.

8.2. Alcance del sistema

Tal como se ha descrito en la introducción, el objetivo de este proyecto es el de ofrecer a los profesionales sanitarios una herramienta mediante la cual puedan realizar un seguimiento más exhaustivo y sencillo sobre las pacientes, con la posibilidad de acceder fácilmente desde sus tabletas a información sobre ellas.

Este trabajo de fin de grado está relacionado con otros tres, pero debido a ciertas circunstancias, no ha sido posible concertar reuniones y conseguir llevar un trabajo conjunto. Sin embargo, durante el desarrollo se ha tenido en cuenta la relación que existe con estos otros trabajos y se ha optado por implementar soluciones que les permitan acoplarse cuando sea necesario.

La aplicación, junto con los trabajos de los otros compañeros ofrecería, aparte de lo descrito anteriormente:

- La posibilidad de realizar los cuestionarios a través de una web.
- Una interfaz de usuario diseñada teniendo en cuenta el contexto de uso.
- Módulos de inteligencia los cuales ayudan en la tarea del médico en la detección de la DPP.

El conjunto formado por estos tres trabajos más el presentado en esta memoria, resultaría en un sistema totalmente funcional.

El sistema que forman quedaría descrito en la siguiente imagen:

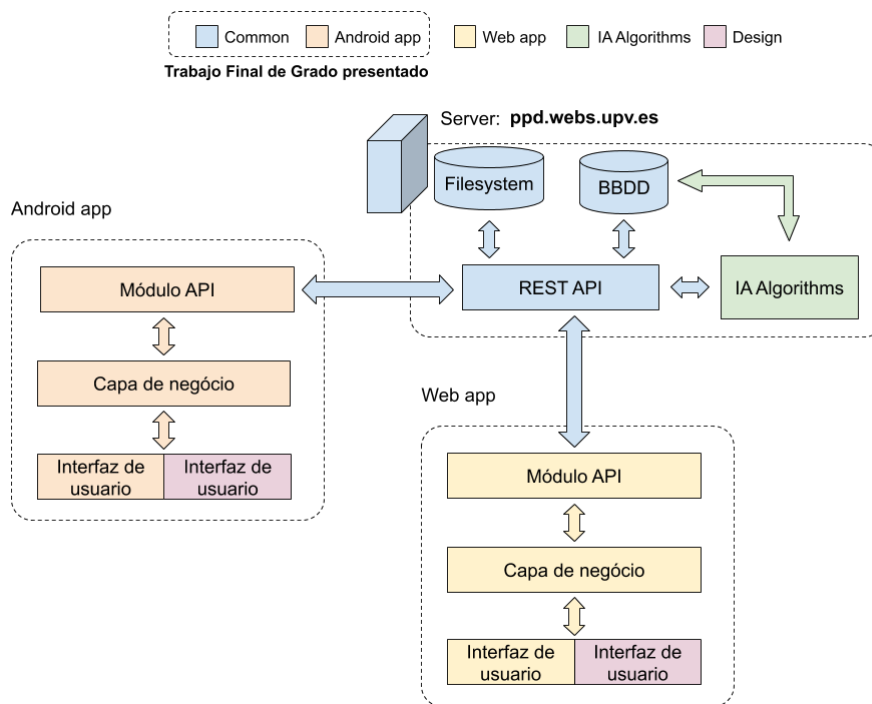


Figura 8.2: Esquema conceptual de sistema completo

La aplicación Android y la aplicación web se comunican con la API a través de peticiones *https*. La API se encuentra desplegada en el servidor **ppd.webs.upv.es**, donde también se encuentra la BD. Ambas implementan una interfaz diseñada inicialmente por sus programadores con el objetivo de validar la aplicación. Posteriormente, se implementaría la interfaz diseñada en uno de los otros trabajos. La forma de conectarse del módulo de inteligencia a la BD se deja abierta, o bien a través de la API, o bien contenida en el mismo servidor y conectándose directamente.

Pensando en el módulo de inteligencia artificial, se ha optado por un diseño para la BD donde la información se encuentra desgranada, de manera que permite recopilar datos sobre distintos elementos sin depender de otros. Esto permite, por ejemplo, calcular la respuesta más contestada a una cuestión o qué cuestiones son más relevantes.

Además, la aplicación permite crear cuestionarios a medida, de manera que bajo el criterio del personal sanitario puedan crear aquellos que se adapten a sus necesidades.

9. Hipótesis y restricciones

Antes de comenzar con la implementación, se estudió qué funcionalidades debía tener la aplicación. De ese estudio inicial hasta el resultado final, la aplicación ha variado mínimamente. Las principales características, la gestión de pacientes y la gestión de los elementos del sistema, se han mantenido aunque añadiendo alguna funcionalidad más y modificando alguna otra.

En el caso de los tratamientos, al principio se pensó en que fuera el propio médico el que dictaminara en qué consistía un tratamiento en el momento de la asignación. Más tarde, se optó por que los tratamientos estuvieran previamente definidos y el médico tan solo tuviera que especificar la fecha de inicio y la fecha final del tratamiento.

Lo más destacable es una funcionalidad que se planteó posteriormente al estudio inicial. Se trata de la posibilidad de disponer de una gráfica en la que el médico puede observar la evolución de la paciente a lo largo del tiempo. También es posible visualizar en qué periodo de tiempo ha estado vigente un tratamiento así como umbrales que indican cuando una paciente está en riesgo.

De esta nueva funcionalidad también derivan dos más: crear riesgos y asignar riesgos a un cuestionario.

10. Estudio de alternativas y viabilidad

Después de describir y analizar en el apartado 4 «Descripción de la situación actual» varios sistemas similares, nos encontramos con alternativas que no se ciñen a las necesidades expuestas en el apartado 3 «Antecedentes». Estos sistemas comparten la característica de servir a los usuarios como una herramienta de autoayuda, pero no están adaptadas para su uso por personal sanitario. Así pues, las posibles alternativas quedan descartadas, siendo necesario el desarrollo de este proyecto. En este apartado se estudiará la viabilidad del proyecto, a nivel económico, técnico y legal.

10.1. Viabilidad económica

En el apartado 15 «Resumen del presupuesto» se han detallado los costes económicos del desarrollo a nivel de software, hardware y mano de obra. Aunque se hayan calculado los costes de mano de obra, estos no suponen un gasto real, ya que se trata de un proyecto enmarcado dentro de un trabajo de fin de grado (TFG). Es por eso que el proyecto resulta viable desde el punto de vista económico.

10.2. Viabilidad técnica

Para el desarrollo de los requisitos especificados en el apartado 7 «Requisitos iniciales», existen herramientas informáticas que permiten lo especificado. Asimismo, el personal encargado de llevarlo a cabo se trata de un estudiante de ingeniería informática. Es por eso que el proyecto resulta viable desde el punto de vista técnico.

10.3. Viabilidad legal

En el apartado 5.1 «Disposiciones legales y normas aplicadas», se han descrito las normativas legales que ha de cumplir el proyecto. Adicionalmente, el sistema cumple con el requisito no funcional RFN05, especificado en el apartado 7.2 «Requisitos no funcionales». Es por eso que el proyecto resulta viable desde el punto de vista legal.

11. Descripción de la solución propuesta

A continuación, describiremos la solución propuesta. Para facilitar su comprensión, lo haremos a partir de un esquema conceptual sobre el sistema donde se puede apreciar las partes que lo componen.

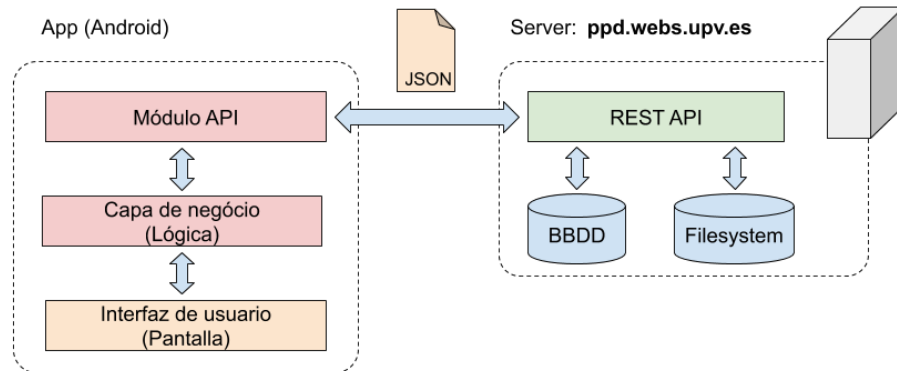


Figura 11.1: Esquema conceptual

Cada pantalla tiene asociada su lógica correspondiente, la cual se encarga de efectuar las llamadas al módulo API. Este módulo se encarga de enviar y gestionar las peticiones al servicio API, y también de proporcionar a la lógica la información solicitada. El módulo API envía peticiones https al servicio API, y este interactúa con la BD para obtener los datos demandados. Finalmente, el servicio API devuelve los datos en formato JSON, que a través del módulo API, pasarán a ser tratados por la lógica correspondiente para su visualización.

12. Análisis de Riesgos

Como en todo proyecto dentro de la ingeniería informática existe una serie de riesgos los cuales pueden afectar al desarrollo o al funcionamiento del sistema. Debido a la gran cantidad de riesgos que existen se ha optado por detallar aquellos más relevantes, los cuales son:

- **Fallo del servidor:** un fallo del servidor dejaría a la aplicación totalmente inoperativa. Esto impide a los usuarios continuar con sus funciones o les obliga a buscar alternativas.
- **Lentitud de respuesta:** un tiempo excesivo para poder visualizar la información o tratarla impediría utilizar la aplicación de forma óptima y cómoda.
- **Incompatibilidad:** si el sistema Android del dispositivo donde se va a utilizar la aplicación tiene una versión menor que la versión mínima en la que se ha desarrollado será imposible utilizarla.
- **Problemas de visualización:** debido a la gran variedad de pantallas que existen en el mercado, es posible que en algunos dispositivos los elementos que se muestran en pantalla se visualicen de manera incorrecta.
- **Inconsistencia de la BD:** la presencia de inconsistencias entre los datos de la BD y los de la aplicación no permitiría usar correctamente el sistema. Esto podría invalidar alguna de sus funcionalidades.
- **Mal uso de los usuarios:** una mala gestión del sistema por parte de los usuarios produciría comportamientos no deseables dentro del propio sistema. Esto podría invalidar alguna de sus funcionalidades.
- **Cambios en la legislación de protección de datos:** cambios en la legislación que condicionarán la forma de tratar los datos personales contenidos en el sistema.
- **La planificación inicial no se ajusta:** que ocurran cambios en la planificación provoca que tareas posteriores se vean aplazadas o incluso canceladas.
- **Curva de aprendizaje lenta:** para poder desarrollar el sistema de forma eficiente, es necesario que la curva de aprendizaje alcance un nivel idóneo para las tareas a realizar.

Los riesgos analizados se representan en la tabla 12.1.

El modelo a seguir es el modelo de PMBOK, este modelo se basa en determinar los riesgos y caracterizarlos por probabilidad e impacto.

Amenaza	Categoría	Probabilidad	Impacto		
			Alcance	Plazos	Coste
Fallo del servidor	Comunicación	Baja	Alto	Bajo	Bajo
Lentitud de respuesta	Comunicación	Alta	Medio	Medio	Medio
Incompatibilidad	Software	Baja	Bajo	Medio	Medio
Problemas de visualización	Hardware	Baja	Medio	Medio	Bajo
Inconsistencia de la BD	Persistencia	Baja	Alto	Alto	Medio
Mal uso de los usuarios	Persistencia	Medio	Medio	Bajo	Bajo
Cambios en la legislación de protección de datos	Jurídica	Baja	Bajo	Medio	Medio
La planificación inicial no se ajusta	Gestión	Baja	Medio	Bajo	Medio
Curva de aprendizaje	Gestión	Media	Medio	Bajo	Bajo

Tabla 12.1: Análisis de riesgos

El nombre indica el tipo de riesgo que se está evaluando. La categoría determina en qué grupo se puede englobar. La probabilidad es la capacidad de que se produzca. El impacto es cómo se afrontaría el riesgo en el sistema y se divide en tres secciones: el alcance es cuánto puede abarcar, los plazos, es el tiempo que durará confrontarlo, y el coste, es el precio que supone.

13. Organización y gestión del proyecto

13.1. Organización

En los siguientes subapartados se describirán los diferentes aspectos derivados de la organización del proyecto. Se describirán los actores que participan, la estructura interna y las interfaces externas.

13.1.1. Actores del proyecto y relaciones entre los mismos

En las siguientes tablas (13.1) se puede observar la información referente a las personas involucradas en el trabajo, detallándose su nombre, contacto y su responsabilidad dentro de este.

Nombre	Germán Molina Francés
Contacto	germofra@inf.upv.es
Rol	Programador

Nombre	José Luis Poza Luján
Contacto	jopolu@disca.upv.es
Rol	Tutor

Nombre	José Alberto Conejero Casares
Contacto	aconejero@upv.es
Rol	Cotutor

Tabla 13.1: Integrantes del proyecto

13.1.2. Estructura interna

En la siguiente imagen (13.1) podemos observar la estructura del proyecto y las relaciones que existen entre los diferentes miembros.

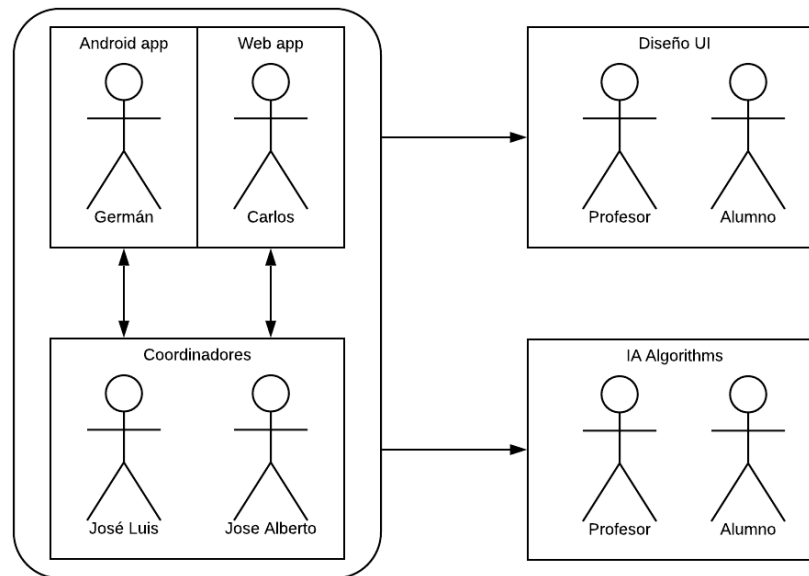


Figura 13.1: Estructura interna

En la parte de la izquierda observamos los miembros encargados de desarrollar las aplicaciones Android y web, este grupo está compuesto por Germán como desarrollador de la parte móvil, Carlos de la parte web y José Luis y José Alberto como coordinadores. Este grupo está relacionado con otros dos, uno encargado del diseño de la interfaz y otro del desarrollo de un módulo de inteligencia.

13.1.3. Interfaces externas

Como se ha mencionado en el apartado 8 «Alcance», este proyecto está compuesto principalmente por dos partes o sistemas, la parte móvil y la parte web. El alumno Carlos, citado en el subapartado 13.1.1 «Actores», es el encargado de desarrollar la parte web. Esta parte se enmarca dentro de su trabajo de fin de grado.

A grandes rasgos, el trabajo de este alumno ofrece la posibilidad de realizar los cuestionarios vía web. Como hemos explicado en el apartado 8 «Alcance», los trabajos relacionados no se encuentran acoplados. En el caso de la aplicación

web, esta utiliza su propia BD, donde almacena los resultados de los cuestionarios.

Otras interfaces externas que encontramos son el módulo de inteligencia artificial y el diseño de la interfaz de usuario. En el caso del módulo de inteligencia, se deja la puerta abierta a cómo obtiene los datos de la BD pudiendo optar por encontrarse dentro del servidor y acceder a esta directamente o acceder a los datos a través de la API.

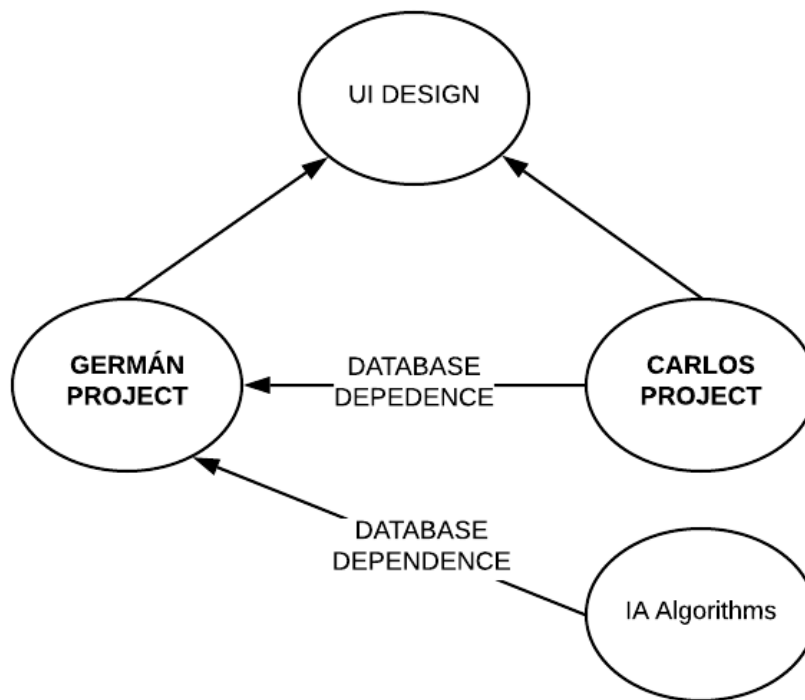


Figura 13.2: Interfaz externa

En el caso de que los distintos trabajos formaran un mismo sistema, en la figura 13.2 se muestra el diagrama donde se muestran las dependencias que existirían entre ellos.

13.2. Gestión del proyecto

En el desarrollo del proyecto se ha llevado a cabo una metodología ágil siguiendo la estrategia MVP. Dentro del proyecto encontramos dos partes, backend y frontend. Al principio se enfocó la actividad en conseguir un backend que respondiera a las necesidades mínimas de la aplicación. Una vez consolidada esta parte, se pasó al desarrollo de la aplicación y, a partir de ahí, se siguió un desarrollo incremental en ambas partes.

Adicionalmente, cada semana se concertaba una reunión en la que se comentaba el avance de las tareas, el estado del proyecto y el establecimiento de nuevas tareas.

14. Planificación temporal

En el siguiente diagrama de Gantt (14.1) podemos observar cómo se han planificado las entregas de los diferentes paquetes y las dependencias que existen entre ellos.

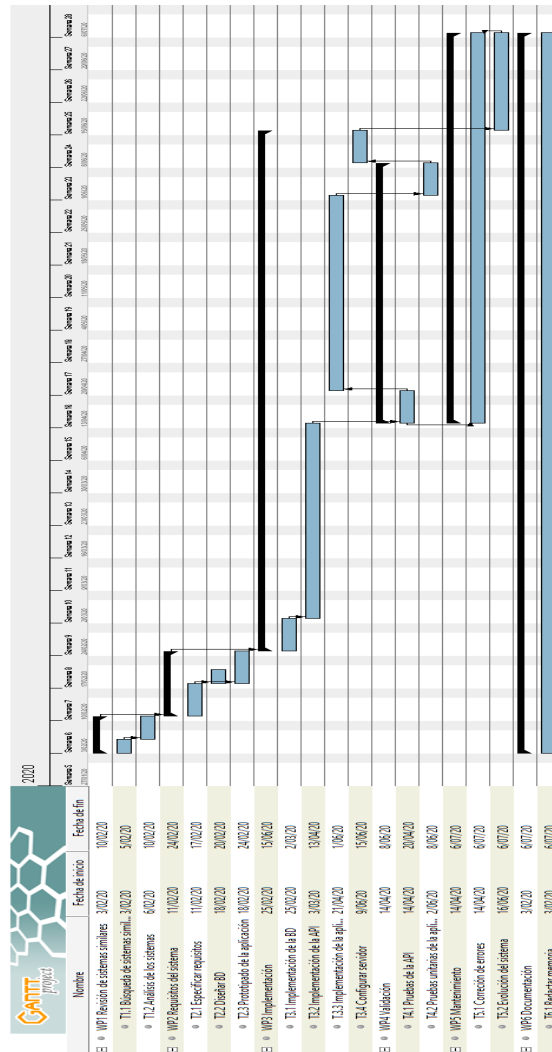


Figura 14.1: Diagrama de Gantt

En el diagrama observamos los paquetes descritos en capítulo 8 «Alcance». Algunas de las tareas que los conforman tienen dependencia entre ellas: en la imagen esto se indica gráficamente mediante las flechas. Sobre estas dependencias también hemos hablado en el capítulo 8 «Alcance». La duración de las tareas y las fechas previstas para estas ha sido prácticamente igual al que se muestra en el diagrama 14.1.

15. Resumen del presupuesto

15.1. Costes hardware

Para el desarrollo han sido necesario un ordenador de sobremesa y un dispositivo Android. A continuación se detallan sus especificaciones y el coste.

Componente	Ordenador sobremesa	Xiaomi Redmi 4A
Capacidad disco duro	1TB	32GB
Memoria RAM	8GB	2GB
Procesador	AMD FX-8350 4.00GHz	Quad-core 1.40GHz
Tarjeta gráfica/GPU	GeForce GTX 970	Adreno 308
Sistema operativo	Windows 10	Android 7.1
Precio	900€	110€

Tabla 15.1: Costes hardware

15.2. Costes software

Todo el software y las herramientas de diseño utilizadas en el desarrollo son totalmente gratuitas. El coste del sistema operativo Windows no se incluye aquí, ya que se considera que va incluido dentro del coste del ordenador de sobremesa.

15.3. Costes mano de obra

Para el desarrollo de este proyecto se han invertido un total de 5 meses, empezando en febrero de 2020 y acabando en julio del mismo año, con la entrega del proyecto. Durante todo ese tiempo se ha trabajado uniformemente 15 horas semanales aproximadamente, invirtiendo un total de 315 horas correspondientes a un trabajo de fin de grado.

Considerando un coste de 25€ por hora los costes de mano de obra serían de un total de 7.500€.

15.4. Costes totales

Una vez detallados los costes en los apartados anteriores, procedemos a calcular el coste total del proyecto.

	Componente	Precio
Hardware	Sobremesa	900€
	Móvil	110€
Software	-	0€
	Horas	
Mano de obra	300	7.500€
	TOTAL	8.510€

Tabla 15.2: Coste total

La suma total de los costes hardware, software y de mano de obra supone un coste total de 8.510€.

16. Conclusiones y trabajo futuro

16.1. Dificultades encontradas

La primera dificultad que apareció fue la de encontrar un documento electrónico que reflejara un expediente clínico. Existen varios tipos de documentos que implementan el estándar CDA HL7, desde recetas médicas hasta registros cardiovasculares. Finalmente se encontró el C-CDA (Continuity of Care Document), el cual permite incorporar otros documentos que se rigen por el CDA, formando así un expediente clínico.

Una vez obtenido había que encontrar la manera de visualizarlo. En Android es posible visualizar un documento html como si se tratara de una página web, por lo que la solución que se planteó era la de utilizar un archivo xsl para la transformación. Fue costoso encontrarlo, pero al final apareció uno el cual cumplía con los estándares establecidos.

Los otros problemas que aparecieron durante la implementación del proyecto derivan del hecho de ser la primera vez que se trabajaba con este tipo de tecnología. El aprender a programar una aplicación Android o crear un backend sin prácticamente conocimientos previos, ha resultado en más beneficio que posible dificultad. Mientras se desarrollaba, se encontraban dificultades que era necesario afrontar ampliando conocimientos sobre las herramientas.

Todas las dificultades encontradas han conseguido ser resueltas, es por ello que no se pueden considerar problemas. Además, el mundo de la informática está en permanente evolución, y por ello, es necesario estar constantemente formándose y aprendiendo sobre nuevas tecnologías y herramientas, la cual cosa supone un enriquecimiento para el saber de la persona.

16.2. Aportaciones del proyecto

Atendiendo a lo expuesto en el capítulo 2 «Objeto», el objetivo de este trabajo ha sido alcanzado. Se ha conseguido aportar una herramienta que permite monitorizar el estado de pacientes y gestionar sus elementos más relevantes, gracias a la cual los profesionales sanitarios podrán gestionar de manera más eficaz y eficiente el proceso de detección de DPP.

A nivel personal ha sido muy gratificante, ya que como se ha comentado en el apartado anterior, se partía de una base muy pequeña y al finalizar este proyecto se han consolidado conocimientos relacionados con el desarrollo de backend y sobre todo con el desarrollo de aplicaciones Android.

16.3. Ampliaciones futuras

Como ampliaciones futuras, sería interesante añadir nuevas funcionalidades. Por ejemplo, un soporte de documentos, permitiendo a los médicos asociar documentos a distintos elementos, como pudiera ser a una paciente, un tratamiento o un cuestionario.

También sería interesante la posibilidad de crear simulaciones que mediante la especificación de una serie de parámetros permita obtener diferentes situaciones hipotéticas.

Por último, se planteó la opción de recibir notificaciones o alertas dentro de la aplicación con el objetivo de informar al usuario sobre determinados eventos.

Parte II

PROYECTO TÉCNICO

17. ANEXOS

17.1. Anexo - Análisis del sistema

17.1.1. Actores

En este apartado detallaremos los actores que van a interactuar con el sistema. Encontramos dos actores:

- **Médico:** será la persona que tenga acceso a la aplicación para poder visualizar información y realizar un seguimiento sobre pacientes que tiene asignadas.
- **Administrador:** este actor representa a un médico con permisos adicionales. Aparte de todas las acciones que puede realizar un médico, un administrador puede gestionar los elementos del sistema.

17.1.2. Diagramas de casos de uso

En esta sección se detallarán y describirán los casos de uso asociados a los actores definidos en el apartado 17.1.1 «Actores».

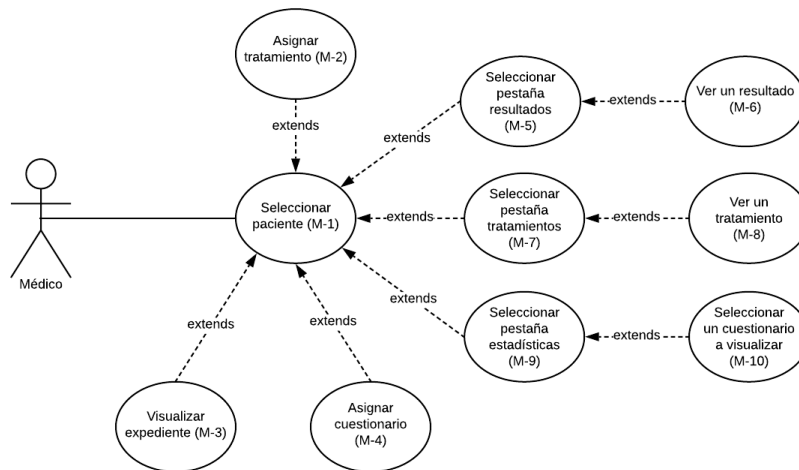


Figura 17.1: Diagrama de casos de uso de médico

M-1	
Descripción	Seleccionar paciente.
Entradas	Pulsar sobre una de las tarjetas listadas.
Procesos	Navega a la pantalla general de pacientes.
Salidas	Muestra la información de la paciente (resultados, tratamientos y estadísticas).

Tabla 17.1: Caso de uso M-1

M-2	
Descripción	Asignar tratamiento.
Entradas	Desplegar el menú superior y pulsar sobre la opción correspondiente. Pulsa sobre un tratamiento para cargar la información. Introduce fechas.
Procesos	Navega a la pantalla «Asignar tratamiento».
Salidas	Muestra un listado de tratamientos.

Tabla 17.2: Caso de uso M-2

M-3	
Descripción	Visualizar expediente.
Entradas	Desplegar el menú superior y pulsar sobre la opción correspondiente.
Procesos	Navega a la pantalla «Expediente».
Salidas	Muestra el expediente clínico.

Tabla 17.3: Caso de uso M-3

M-4	
Descripción	Asignar cuestionario.
Entradas	Desplegar el menú superior y pulsar sobre la opción correspondiente. Seleccionar el cuestionario deseado.
Procesos	Navega a la pantalla «Asignar cuestionario».
Salidas	Muestra un listado de los cuestionarios, indicando si ya está asignado.

Tabla 17.4: Caso de uso M-4

M-5	
Descripción	Seleccionar pestaña resultados.
Entradas	Pulsar sobre la opción «Resultados» en el menú inferior.
Procesos	Cargar pestaña «Resultados».
Salidas	Muestra un listado de los resultados de la paciente.

Tabla 17.5: Caso de uso M-5

M-6	
Descripción	Ver resultado.
Entradas	Pulsar sobre uno de los resultados listados.
Procesos	Navega a la pantalla «Ver resultado».
Salidas	Muestra las cuestiones contestadas indicando la respuesta elegida.

Tabla 17.6: Caso de uso M-6

M-7	
Descripción	Seleccionar pestaña tratamientos.
Entradas	Pulsar sobre la opción «Tratamientos» en el menú inferior.
Procesos	Cargar pestaña «Tratamientos».
Salidas	Muestra un listado de los tratamientos asignados a la paciente.

Tabla 17.7: Caso de uso M-7

M-8	
Descripción	Ver tratamiento.
Entradas	Pulsar sobre uno de los tratamientos listados.
Procesos	Navega a la pantalla «Tratamiento».
Salidas	Muestra la información del tratamiento junto con las fechas de principio y fin.

Tabla 17.8: Caso de uso M-8

M-9	
Descripción	Seleccionar pestaña estadísticas.
Entradas	Pulsar sobre la opción «Estadísticas» en el menú inferior.
Procesos	Cargar pestaña «Estadísticas».
Salidas	Muestra una gráfica donde se muestra los resultados obtenidos a lo largo del tiempo entre otros datos.

Tabla 17.9: Caso de uso M-9

M-10	
Descripción	Seleccionar un cuestionario a visualizar.
Entradas	Seleccionar un cuestionario.
Procesos	Carga los resultados correspondientes al cuestionario seleccionado.
Salidas	Muestra en la gráfica los resultados correspondientes al cuestionario seleccionado entre otros datos.

Tabla 17.10: Caso de uso M-10

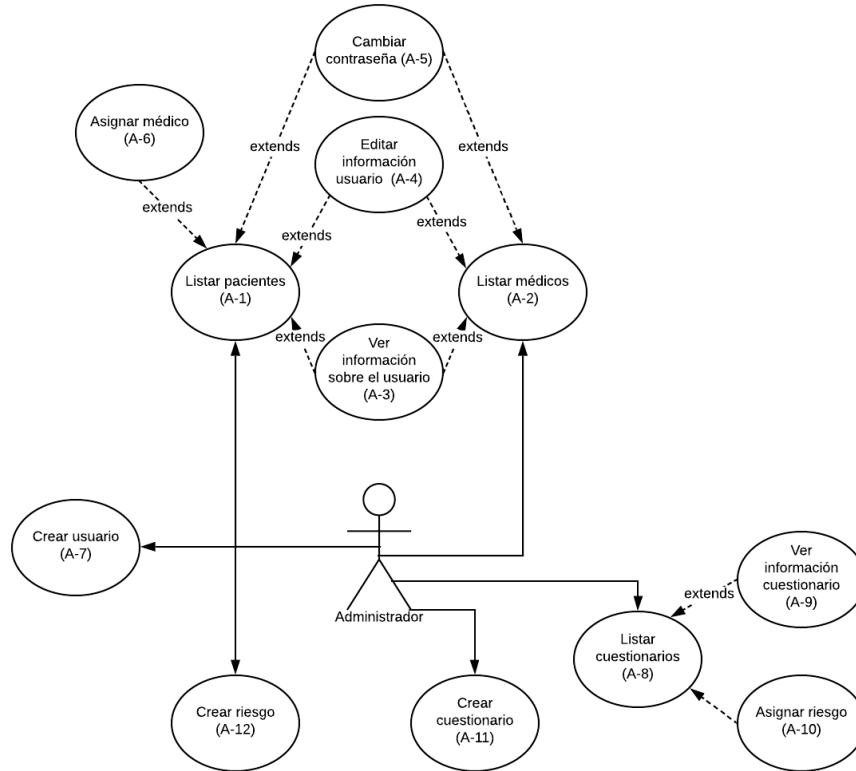


Figura 17.2: Diagrama de casos de uso de administrador

A-1	
Descripción	Listar pacientes.
Entradas	Pulsar sobre la opción «Listar pacientes».
Procesos	Carga un listado.
Salidas	Muestra un listado de las pacientes registradas en el sistema

Tabla 17.11: Caso de uso A-1

A-2	
Descripción	Listar médicos.
Entradas	Pulsar sobre la opción «Listar médicos».
Procesos	Carga un listado.
Salidas	Muestra un listado de los médicos registrados en el sistema.

Tabla 17.12: Caso de uso A-2

A-3	
Descripción	Ver información sobre el usuario.
Entradas	Pulsar sobre una de las tarjetas listadas.
Procesos	Navega a la pantalla «Info. usuario».
Salidas	Muestra la información del usuario.

Tabla 17.13: Caso de uso A-3

A-4	
Descripción	Editar información de un usuario.
Entradas	Despliega el menú de opciones de una de las tarjetas. Selecciona la opción correspondiente. Editar los campos deseados y guardar.
Procesos	Navega a la pantalla «Editar usuario».
Salidas	Muestra la información del usuario y permite su edición.

Tabla 17.14: Caso de uso A-4

A-5	
Descripción	Cambiar contraseña.
Entradas	Despliega el menú de opciones de una de las tarjetas. Selecciona la opción correspondiente. Introduce una nueva contraseña.
Procesos	Navega a la pantalla «Cambiar contraseña».
Salidas	Nombre y DNI de la paciente.

Tabla 17.15: Caso de uso A-5

A-6	
Descripción	Asignar médico.
Entradas	Despliega el menú de opciones de una de las tarjetas. Selecciona la opción correspondiente. Selecciona un médico.
Procesos	Navega a la pantalla «Asignar médico».
Salidas	Muestra un listado de médicos, indicando si ya está asignado.

Tabla 17.16: Caso de uso A-6

A-7	
Descripción	Crear usuario.
Entradas	Pulsar sobre la opción «Crear usuario». Rellenar los campos y guardar.
Procesos	Navega a la pantalla «Crear usuario».
Salidas	(No hay salida).

Tabla 17.17: Caso de uso A-7

A-8	
Descripción	Listar cuestionarios.
Entradas	Pulsar sobre la opción «Listar cuestionarios».
Procesos	Carga un listado.
Salidas	Muestra un listado de cuestionarios.

Tabla 17.18: Caso de uso A-8

A-9	
Descripción	Ver información cuestionario.
Entradas	Pulsar sobre una de las tarjetas listadas.
Procesos	Navega a la pantalla «Ver cuestionario».
Salidas	Muestra las cuestiones que componen el cuestionario.

Tabla 17.19: Caso de uso A-9

A-10	
Descripción	Asignar riesgo.
Entradas	Despliega el menú de opciones de una de las tarjetas. Selecciona la opción correspondiente. Selecciona un riesgo.
Procesos	Carga un listado.
Salidas	Muestra un listado de riesgos, indicando si ya está asignado.

Tabla 17.20: Caso de uso A-10

A-11	
Descripción	Crear cuestionario.
Entradas	Pulsar sobre la opción «Crear cuestionario». Escribir un nombre y seleccionar las cuestiones que lo van a componer.
Procesos	Navega a la pantalla «Crear cuestionario».
Salidas	Muestra un listado de cuestiones.

Tabla 17.21: Caso de uso A-11

A-12	
Descripción	Crear riesgo.
Entradas	Pulsar sobre la opción «Crear riesgo». Rellenar los campos y guardar.
Procesos	Navega a la pantalla «Crear riesgo».
Salidas	(No hay salida).

Tabla 17.22: Caso de uso A-12

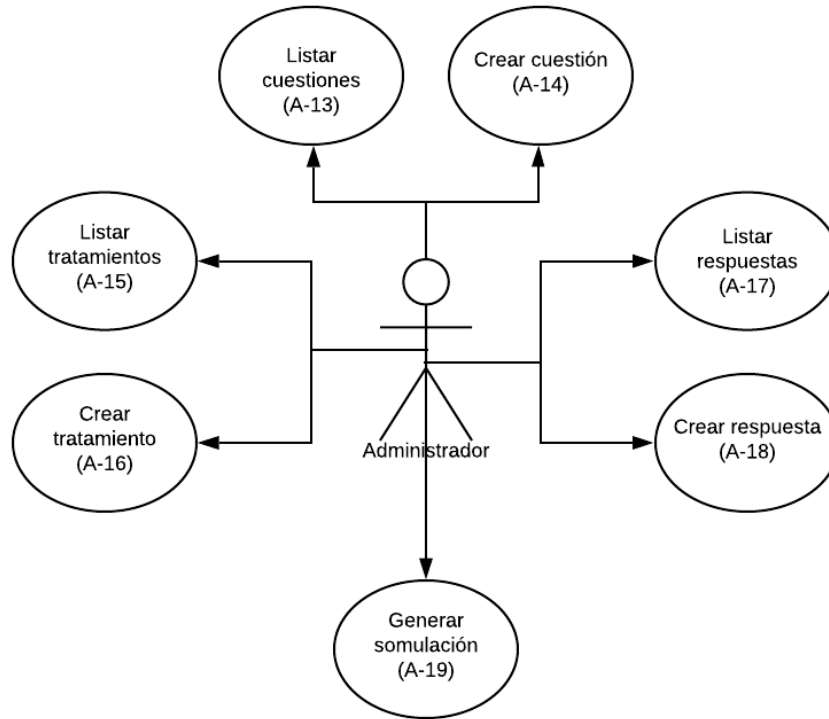


Figura 17.3: Diagrama de casos de uso de administrador

A-13	
Descripción	Listar cuestiones.
Entradas	Pulsar sobre la opción «Listar cuestiones».
Procesos	Carga un listado.
Salidas	Muestra un listado de cuestiones.

Tabla 17.23: Caso de uso A-13

A-14	
Descripción	Crear cuestión.
Entradas	Pulsar sobre la opción «Crear cuestión». Escribir la pregunta y seleccionar las respuestas que la van a componer.
Procesos	Navega a la pantalla «Crear cuestión».
Salidas	Muestra un listado de respuestas.

Tabla 17.24: Caso de uso A-14

A-15	
Descripción	Listar tratamientos.
Entradas	Pulsar sobre la opción «Listar tratamientos».
Procesos	Carga un listado.
Salidas	Muestra un listado de tratamientos.

Tabla 17.25: Caso de uso A-15

A-16	
Descripción	Crear tratamiento.
Entradas	Pulsar sobre la opción «Crear tratamiento». Rellenar los campos y guardar.
Procesos	Navega a la pantalla «Crear tratamiento».
Salidas	(No hay salida).

Tabla 17.26: Caso de uso A-16

A-17	
Descripción	Listar respuestas.
Entradas	Pulsar sobre la opción «Listar respuestas».
Procesos	Carga un listado.
Salidas	Muestra un listado de respuestas.

Tabla 17.27: Caso de uso A-17

A-18	
Descripción	Crear respuesta.
Entradas	Pulsar sobre la opción «Crear respuesta». Rellenar los campos y guardar.
Procesos	Navega a la pantalla «Crear respuesta».
Salidas	(No hay salida).

Tabla 17.28: Caso de uso A-18

A-19	
Descripción	Generar simulación.
Entradas	Pulsar sobre la opción «Generar simulación». Seleccionar el cuestionario sobre el que crear una simulación y crear.
Procesos	Navega a la pantalla «Generar simulación».
Salidas	Muestra el nombre de la paciente de prueba creada.

Tabla 17.29: Caso de uso A-19

17.2. Anexo - Diseño del sistema

En este anexo se detallarán los aspectos referentes al diseño del sistema. Tanto el diseño como la implementación se han basado en el patrón MVC (Model, View, Controller) con tal de conseguir una arquitectura típica de un sistema Cliente-Servidor, donde las capas de presentación, negocio y de datos se encuentran separadas. A continuación se describen las tres capas.

17.2.1. Capa de presentación

La capa de presentación se encarga de mostrar al usuario la información y de recoger datos que introduce el usuario, además debe ser lo más amigable posible.

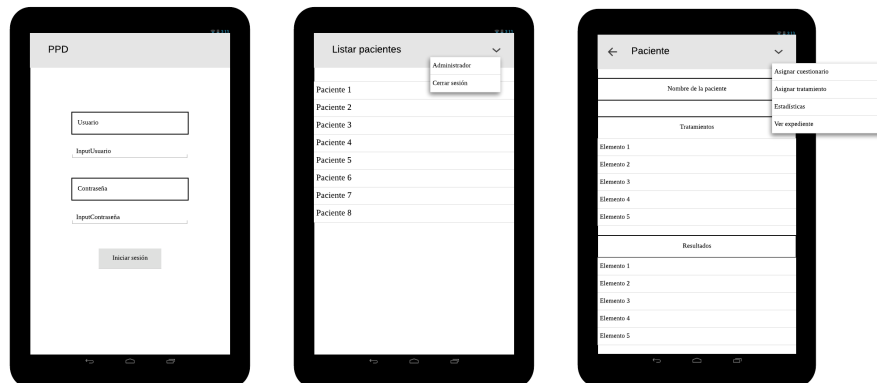


Figura 17.4: Mockups de la aplicación

Todas las interfaces gráficas tienen en común la cabecera situada en la parte superior de la pantalla. Aparte de otorgar más consistencia a la interfaz, en esta se indica la parte de la aplicación en la que se encuentra el usuario y también dispone de un botón que permite retroceder hacia una pantalla anterior.

Además, los diferentes elementos visuales compartirán el mismo estilo.

En algunas pantallas existirán funcionalidades adicionales que serán accesibles desde el menú desplegable de la cabecera, de manera que se ahorre espacio en pantalla y sean fácilmente accesibles. En el caso de la pantalla principal del administrador será un menú lateral debido a la cantidad de funcionalidades de las que dispone, haciendo un mejor uso del espacio en pantalla y consiguiendo una mejor usabilidad.

Al ser una aplicación donde su principal objetivo es el de consultar información, la aplicación se compone de múltiples listados. En estos se podrá visualizar información relacionada con la parte de la aplicación donde se encuentre el usuario y permitirá acceder a otras con más detalle sobre el elemento seleccionado. En algunos casos existirán más opciones que la de seleccionar un elemento, por ejemplo, marcar varios o desplegar un menú el cual nos ofrece más opciones a realizar sobre un elemento.

17.2.2. Capa de negocio

La capa de negocio se encarga de ejecutar el código que implementa las funcionalidades del sistema. Todas ellas siguen un patrón parecido, el cual consiste en los siguientes pasos:

1. El usuario interactúa con la interfaz gráfica.
2. En respuesta al evento de la interfaz, se recaba la información requerida para realizar la petición y se llama al módulo API proporcionándosele.
3. El módulo API, envía una petición *https* al *endpoint* correspondiente de la API con la información requerida para que sea resuelta.
4. La API procesa la petición, se comunica con la BD, recaba la información según los parámetros proporcionados y realiza las acciones pertinentes.
5. Finalmente el usuario recibe la respuesta a la petición y la información recibida se visualiza en pantalla.

El patrón descrito se trata de un patrón general, en algunos casos serán necesarios pasos adicionales o incluso menos de los indicados.

En el siguiente diagrama (17.5) se muestra gráficamente el proceso descrito.

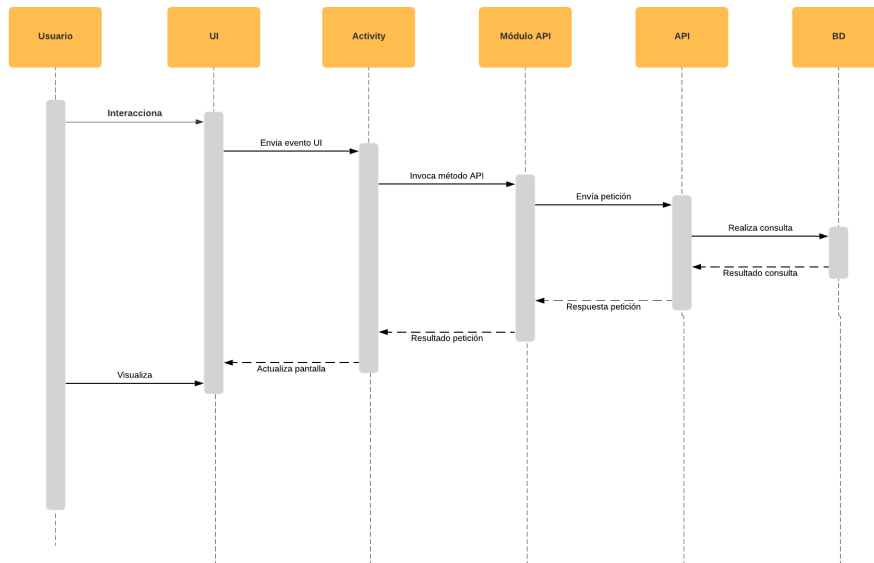


Figura 17.5: Diagrama de secuencia general

Con el fin de no ocupar la sección con información redundante y alargarla innecesariamente se ha optado por detallar el patrón general con un diagrama debido a la similitud que guardarían entre ellos.

17.2.3. Capa de persistencia

La capa de persistencia se encarga de alojar los datos del sistema y de acceder a ellos. Se ha optado por una BD relacional, utilizando el sistema MySQL.

Para el diseño de la BD, se optó por una estructura flexible que permita el análisis de sus datos, teniendo en cuenta la necesidad del módulo de inteligencia para tratarlos. También que permita la creación de nuevos cuestionarios y cuestiones a partir de elementos ya existentes.

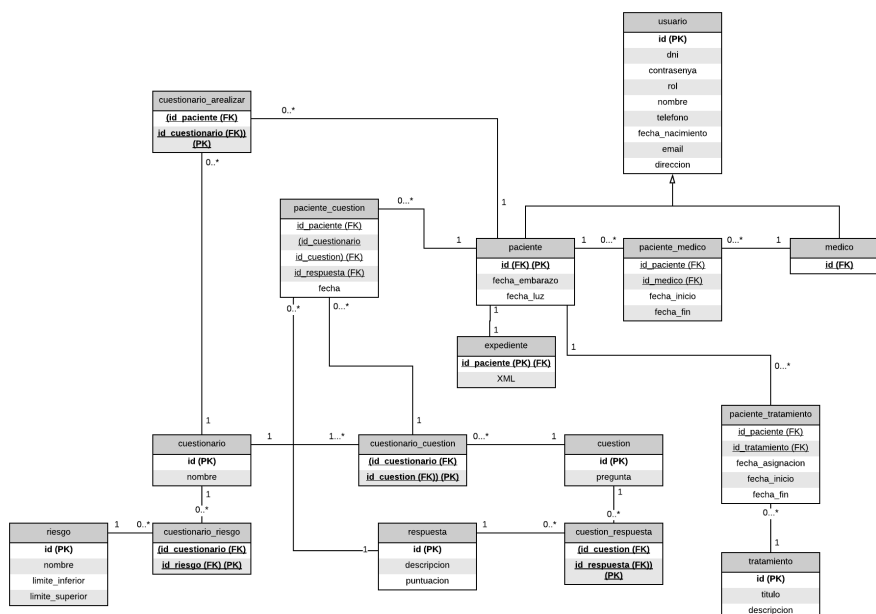


Figura 17.6: Esquema BD

A continuación, describiremos brevemente las tablas más relevantes del esquema:

- **usuario:** esta tabla contiene la información sobre los usuarios registrados en el sistema.
- **paciente:** tabla especializada de la tabla usuario que contiene los id de las pacientes junto con fechas relevantes.
- **paciente_medico:** en esta tabla se guardan las relaciones que indican qué pacientes están asignadas a qué médicos, junto con las fechas de principio y fin de este seguimiento.
- **expediente:** tabla que contiene el nombre del expediente clínico que se encuentra en el sistema de ficheros del servidor.
- **paciente_tratamiento:** tabla que especifica los tratamientos que tiene asignados una paciente, junto con la fecha del momento en que se asignó, la fecha en la que comienza el tratamiento y la fecha en la que finaliza.
- **paciente_cuestion:** en esta tabla es donde se encuentran los resultados de los cuestionarios que han realizado las pacientes. Cada entrada detalla la respuesta seleccionada, junto con la cuestión a la que está relacionada, el cuestionario que contiene esta cuestión y la fecha en la que se rellenó.

- **cuestionario_erealizar:** en esta tabla se especifica los cuestionarios que tiene asignada una paciente pendientes de rellenar.

17.3. Anexo - Implementación

En este anexo se detallarán los aspectos referentes a la implementación del sistema, siguiendo el esquema MVC mencionado en el anexo anterior.

17.3.1. Aplicación móvil

Como se ha mencionado en el apartado 5.2.1 «Métodos y herramientas», la aplicación se ha desarrollado en el lenguaje Java utilizando el IDE Android Studio. Generalmente, un proyecto de Android Studio está formado por actividades y las vistas que implementa cada una. En la siguiente imagen podemos observar la estructura de ficheros del proyecto.

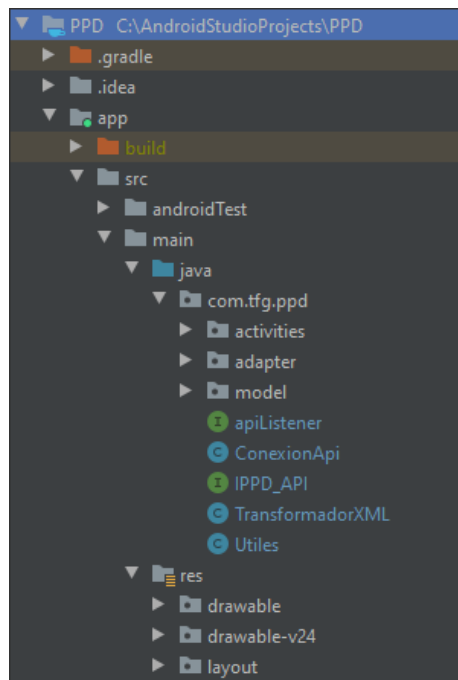


Figura 17.7: Estructura de la aplicación móvil

En esta imagen podemos observar la estructura del proyecto que comienza en la carpeta raíz llamada PPD. Las carpetas más destacables serían la carpeta *main/java* y la carpeta *res/layout*.

En la carpeta *java* es donde encontramos la lógica del proyecto, concretamente en *activities* se encuentran las clases encargadas de implementar las funcionalidades correspondientes a sus vistas. Estas vistas se encuentran en la carpeta *layout*, las cuales se tratan de un fichero *xml* en la que se especifican los elementos que componen la vista.

Volviendo a la carpeta *java*, encontramos la carpeta *adapter*, en la que se encuentran las clases encargadas de manejar la visualización de los listados de una forma óptima, utilizando para ello vistas de la carpeta *layout*.

Además, en la carpeta *model* encontramos los objetos *Java* que corresponden a los objetos *JSON* que se envían y reciben con la API. Por último, encontramos varios elementos comunes que son utilizados en el resto de clases, como por ejemplo *ConexionApi*, módulo encargado de realizar las peticiones y gestionar las respuestas con la API.

A continuación mostraremos el resultado de implementar el diseño descrito en el anexo anterior (17.2.1 «Capa de presentación»).

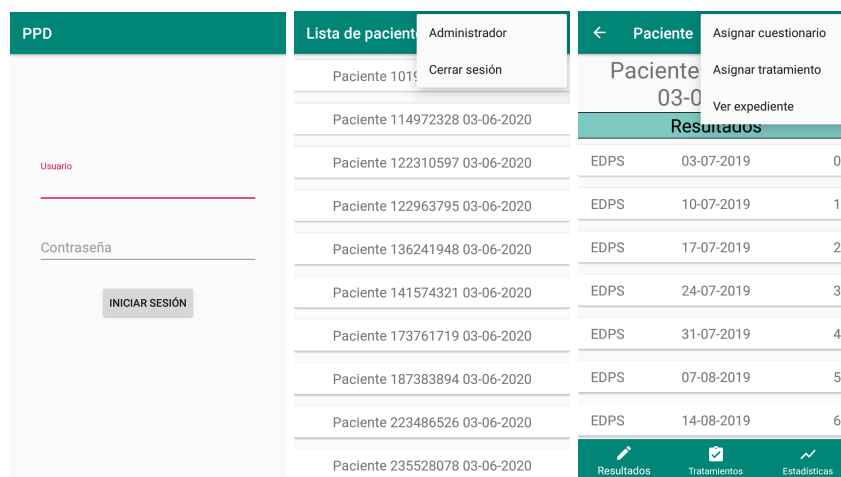


Figura 17.8: Implementación del diseño de la aplicación

Tomando las imágenes del anexo anterior y las correspondientes a la implementación, podemos realizar una comparativa con el objetivo de validar el diseño propuesto.

En lo que respecta a la implementación del código, mostraremos un ejemplo del patrón descrito en el anexo anterior (17.2.2 «Capa de negocio»). Se trata de la visualización de las pacientes asociadas a un médico. Esta funcionalidad se encuentra en la pantalla principal.

Primero de todo, la *activity* llamará al método correspondiente del módulo API para efectuar la petición.

```
1  conexionApi.listarPacientes(id_medico, token, this);
```

Código 1: Invocación del método listarPacientes()

```
1  public void listarPacientes(int id_medico, String token, apiListener listener)
2  {
3      Call<List<Usuario>> call = apiconexion.listarPacientes(id_medico, token);
4      call.enqueue(new Callback<List<Usuario>>() {
5          @Override
6          public void onResponse(Call<List<Usuario>> call,
7              Response<List<Usuario>> response) {
8              if(response.isSuccessful()) listener.setData(response.body(),
9                  Utiles.L_PACIENTES);
10                 else listener.setNotFound();
11             }
12         }
13         @Override
14         public void onFailure(Call<List<Usuario>> call, Throwable t) {
15             listener.setFailure(t);
16         }
17     });
18 }
```

Código 2: Método listarPacientes()

Este método realizará una petición *GET* a la *url*: base + ruta petición, siendo la *URL* base, `http://ppd.webs.upv.es`, y la ruta de esta petición, `/paciente/listar/{id_medico}`, definidas previamente en el objeto *apiconexion* y en la interfaz *IPPD_API* respectivamente. En la petición se envían como parámetros el identificador del médico y el token que permite autenticar y comprobar permisos. En caso de que la petición haya sido resuelta correctamente se devolverá a la *activity* invocadora una lista de usuarios, la cual será visualizada en pantalla.

17.3.2. API

Como se ha mencionado en el apartado 5.2.1 «Métodos y herramientas», la API se ha desarrollado con ExpressJS utilizando el editor de código fuente Visual Studio Code, la estructura es la siguiente:

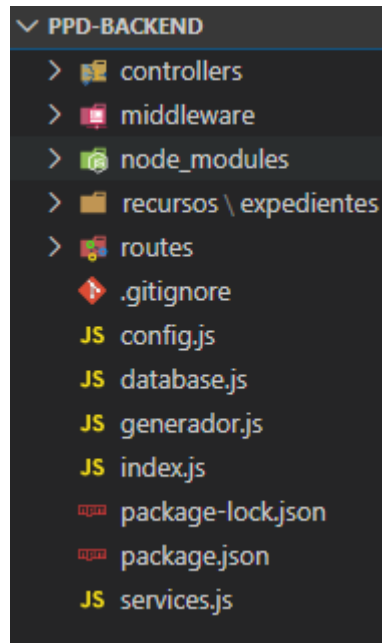


Figura 17.9: Estructura de la API

- **controllers**: carpeta donde se encuentran los ficheros que contienen las funciones que interactúan con la BD.
- **middleware**: carpeta donde se encuentran los ficheros que contienen las funciones a ejecutar antes de interactuar con la BD. En este caso solo contiene la función que comprueba si el usuario está autorizado para realizar la petición entrante.
- **recursos/expedientes**: carpeta donde se guardan los expedientes de los pacientes.
- **routes**: carpeta donde se encuentran los ficheros que contienen los *end-points* de la API y que llaman a las funciones requeridas.
- **config.js**: fichero que contiene la definición de las variables de entorno.
- **database.js**: fichero que se encarga de establecer la conexión a la BD mediante un usuario y contraseña.

- **generador.js**: fichero que contiene el código encargado de crear simulaciones.
- **index.js**: fichero encargado de iniciar el servidor en la *URL* y puerto especificados.
- **services.js**: fichero que contiene la función encargada de crear los tokens requeridos para la autenticación.

En lo que respecta a la implementación del código, continuaremos con el ejemplo del apartado anterior por la parte de la API, mostrando la gestión de la petición explicada en el apartado 17.2.2 «Capa de negocio».

Cuando se recibe la petición esta es dirigida hacia su *endpoint* correspondiente. En nuestro caso, este *endpoint* se encuentra en el fichero *paciente.js*.

```
1 router.get('/paciente/listar/:id_medico', autenticacion.permisoMedico,
  controller.paciente_listarPacientes);
```

Código 3: *Endpoint* de la API

El primer paso a realizar es comprobar que el usuario que realiza la petición tiene los permisos necesarios y está registrado en la BD. Esto se realiza a través de la función *permisoMedico*. En este caso son de nivel médico.

A continuación, la petición pasa a ser procesada por la función *paciente_listarPacientes*, la cual realiza una consulta sobre la BD para recuperar el listado de pacientes. El resultado de la consulta será la información que envíe como respuesta a la petición.

```
1 exports.paciente_listarPacientes = function(req, res) {
2   const { id_medico } = req.params;
3   const query = `SELECT id, nombre
4                   FROM usuario
5                   WHERE id IN (SELECT id_paciente FROM paciente_medico WHERE
6                               id_medico = ? AND fecha_fin IS NULL)
7                   ORDER BY nombre`;
8
9   mysqlConnection.query(query, [id_medico], (err, rows) => {
10    if(err) {
11      console.log(err);
12    } else {
13      if(rows.length == 0) res.status(404).send({ message: 'No se han
14        encontrado recursos con los parametros proporcionados'});
15      else res.json(rows);
16    }
17  }
```

```
16     });  
17     };
```

Código 4: Función paciente_listarPacientes()

Cifrado de contraseña

Cuando un nuevo usuario es registrado o una contraseña se modifica, en la BD se almacena la contraseña cifrada. De esta manera se oculta a toda persona que tenga acceso a la BD.

El cifrado de la contraseña se realiza en el servidor. Cuando se registra un nuevo usuario, al servidor llega la contraseña en texto plano. Antes de introducir en la BD todos los datos, se cifra la contraseña, almacenándose como contraseña el resultado. Cuando un usuario inicia sesión, se compara la contraseña proporcionada con la contraseña cifrada en la BD. Si la comparación es correcta, el usuario será autenticado.

Autenticación con *tokens*

Para acceder a las funcionalidades de la API es necesario incluir en estas un *token*. El proceso es el siguiente. El usuario se autentica en el sistema proporcionando su usuario y contraseña. Cuando el servidor responde satisfactoriamente, le proporciona el *token*. Este *token* se incluirá en la cabecera de todas las peticiones que realice, de manera que permita a la API identificar los permisos del usuario.

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.  
eyJzdWIiOiJEsImRuaSI6IjEyMzQ1Njc4OSIsInJvbCI6IkFETSJ9.  
fgzqN034C3Uw-MPODWSaeNFCv09yEc7BpNjXSQZnQWw
```

Código 5: Ejemplo de *token*

Un *token* está compuesto por 3 *strings* los cuales contienen la información necesaria para identificar al usuario y sus permisos.

Este elemento sirve como una firma cifrada y permite una autenticación sin estado, asegurando que solo los usuarios registrados obtengan la información a la que están autorizados.

17.3.3. Base de datos

Como se ha mencionado en el apartado 5.2.1 «Métodos y herramientas» la BD se ha desarrollado con MySQL utilizando la herramienta visual MySQL Workbench.

Para facilitar la recuperación de la BD, se ha creado un *script* que crea las tablas y las relaciones entre estas. De esta manera, se consigue mayor facilidad y automatización en el proceso de recuperación de la BD, ya que durante las fases de desarrollo y pruebas se llegaban a estados no deseados. Además, también se ha creado otro *script* que añade un súper-usuario y el test EPDS para probar todas sus funcionalidades, entre otras cosas.

```

1 CREATE TABLE IF NOT EXISTS usuario (
2   id INT AUTO_INCREMENT PRIMARY KEY,
3   dni VARCHAR(9) NOT NULL UNIQUE CHECK(length(dni)=9),
4   contrasena VARCHAR(100) NOT NULL,
5   rol VARCHAR(3) NOT NULL CHECK (rol IN ("ADM", "MED", "PAC")),
6   nombre VARCHAR(100) NOT NULL,
7   telefono VARCHAR(15) NOT NULL,
8   fecha_nacimiento DATE NOT NULL,
9   email VARCHAR(50) NOT NULL,
10  direccion VARCHAR(100) NOT NULL
11 );

```

Código 6: Ejemplo de tabla de BD

Después de ejecutar el *script* podemos observar en la siguiente imagen el conjunto de cuestiones que componen el test EPDS.

	id_cuestionario	nombre	id_cuestion	pregunta
▶	1	EDPS	1	He podido reír y ver el lado bueno de las cosas:
	1	EDPS	2	He mirado al futuro con placer para hacer cosas:
	1	EDPS	3	Me he culpado sin necesidad cuando las cosas marchaban mal:
	1	EDPS	4	He estado ansiosa y preocupada sin motivo alguno:
	1	EDPS	5	He sentido miedo o pánico sin motivo alguno:
	1	EDPS	6	Las cosas me oprimen o agobian:
	1	EDPS	7	Me he sentido tan infeliz, que he tenido dificultad para dormir:
	1	EDPS	8	Me he sentido triste y desgraciada:
	1	EDPS	9	Me he sentido tan infeliz que he estado llorando:
	1	EDPS	10	He pensado en hacerme daño:

Figura 17.10: Ejemplo de datos de la BD

17.4. Anexo - Fase de validación y pruebas

17.4.1. Pruebas con Postman

Para cada nuevo *endpoint* que se añadía a la API, se utilizaba Postman para comprobar su correcto funcionamiento. En este apartado no vamos a mostrar todas las pruebas efectuadas sobre la API por su extensión, centrándonos en las pruebas unitarias de la aplicación.

17.4.2. Pruebas unitarias

Conforme se implementaban las funcionalidades, se creaba un conjunto de pruebas unitarias para cada una con tal de validar el correcto funcionamiento de cada requisito. A continuación mostraremos estos conjuntos de pruebas. Cada prueba estará codificada de la siguiente manera: RFXXPUYY, siendo XX el número de requisitos funcional y YY el número de prueba unitaria de ese requisito.

Cada conjunto de pruebas unitarias se expresará en una tabla compuesta por los siguientes elementos:

- Código de la prueba unitaria
- Descripción
- Resultado esperado
- Resultado obtenido

Para que una prueba sea considerada válida, el resultado esperado y el resultado obtenido deben coincidir, en caso contrario la prueba no será superada y será considerada inválida.

Conjunto de pruebas unitarias RF01

RF01PU01	
Descripción	Iniciar sesión con usuario existente
Resultado esperado	Iniciar sesión
Resultado obtenido	Iniciar sesión

RF01PU02	
Descripción	Iniciar sesión con usuario inexistente
Resultado esperado	No iniciar sesión
Resultado obtenido	No iniciar sesión

RF01PU03	
Descripción	Iniciar sesión con una contraseña incorrecta
Resultado esperado	No iniciar sesión
Resultado obtenido	No iniciar sesión

Tabla 17.30: Pruebas unitarias RF01

Conjunto de pruebas unitarias RF02

RF02PU01	
Descripción	Mostrar el listado de las pacientes asignadas
Resultado esperado	Visualizar el listado
Resultado obtenido	Visualizar el listado

Tabla 17.32: Pruebas unitarias RF02

Conjunto de pruebas unitarias RF03

RF03PU01	
Descripción	Pulsar sobre una paciente del listado
Resultado esperado	Se visualiza una pantalla con el nombre de la paciente, los tratamientos asignados y los resultados de cuestionarios
Resultado obtenido	Se visualizan los datos esperados

Tabla 17.33: Pruebas unitarias RF03

Conjunto de pruebas unitarias RF04

RF04PU01	
Descripción	Asignar un cuestionario a una paciente
Resultado esperado	El cuestionario queda asignado y se visualiza
Resultado obtenido	El cuestionario queda asignado y se visualiza

RF04PU02	
Descripción	Desasignar un cuestionario a una paciente
Resultado esperado	El cuestionario queda desasignado y se visualiza
Resultado obtenido	El cuestionario queda desasignado y se visualiza

Tabla 17.34: Pruebas unitarias RF04

Conjunto de pruebas unitarias RF05

RF05PU01	
Descripción	Asignar un tratamiento
Resultado esperado	Se carga la información del tratamiento
Resultado obtenido	Se carga la información del tratamiento

RF05PU02	
Descripción	Se guarda un tratamiento sin una o las dos fechas
Resultado esperado	No se guarda el tratamiento
Resultado obtenido	No se guarda el tratamiento

RF05PU03	
Descripción	Se guarda siendo la fecha de principio posterior a la de final
Resultado esperado	No se guarda el tratamiento
Resultado obtenido	No se guarda el tratamiento

RF05PU04	
Descripción	Se seleccionan fechas consistentes
Resultado esperado	Se guarda el tratamiento
Resultado obtenido	Se guarda el tratamiento

Tabla 17.35: Pruebas unitarias RF05

Conjunto de pruebas unitarias RF06

RF06PU01	
Descripción	Seleccionar la opción de ver el expediente
Resultado esperado	Se visualiza el expediente de la paciente
Resultado obtenido	Se visualiza el expediente de la paciente
Observaciones	Se asume que existe. En caso de que no, se informa en pantalla

Tabla 17.36: Pruebas unitarias RF06

Conjunto de pruebas unitarias RF07

RF07PU01	
Descripción	Iniciar sesión con un súper-usuario
Resultado esperado	Opción de acceder a la parte de gestión habilitada
Resultado obtenido	Opción de acceder a la parte de gestión habilitada

RF07PU02	
Descripción	Iniciar sesión con un usuario normal
Resultado esperado	Opción de acceder a la parte de gestión deshabilitada
Resultado obtenido	Opción de acceder a la parte de gestión deshabilitada

Tabla 17.37: Pruebas unitarias RF07

Conjunto de pruebas unitarias RF08

RF08PU01	
Descripción	Listar pacientes
Resultado esperado	Visualizar un listado de las pacientes registradas en el sistema
Resultado obtenido	Visualizar un listado de las pacientes registradas en el sistema

RF08PU02	
Descripción	Listar médicos
Resultado esperado	Visualizar un listado de los médicos registrados en el sistema
Resultado obtenido	Visualizar un listado de los médicos registrados en el sistema

RF08PU03	
Descripción	Pulsar sobre un usuario
Resultado esperado	Visualizar la información sobre el usuario
Resultado obtenido	Visualizar la información sobre el usuario

RF08PU04	
Descripción	Editar la información sobre el usuario
Resultado esperado	Guardar la información editada
Resultado obtenido	Guardar la información editada

RF08PU05	
Descripción	Modificar la contraseña del usuario
Resultado esperado	La contraseña es modificada
Resultado obtenido	La contraseña es modificada

RF08PU06	
Descripción	Crear un usuario inexistente
Resultado esperado	Usuario creado
Resultado obtenido	Usuario creado

RF08PU07	
Descripción	Crear un usuario con DNI ya existente
Resultado esperado	Usuario no creado
Resultado obtenido	Usuario no creado

Tabla 17.38: Pruebas unitarias RF08

Conjunto de pruebas unitarias RF09

RF09PU01	
Descripción	Asignar un médico a una paciente
Resultado esperado	El médico queda asignado con la fecha de asignación
Resultado obtenido	El médico queda asignado con la fecha de asignación

RF09PU02	
Descripción	Designar un médico a una paciente
Resultado esperado	El médico queda desasignado con la fecha de desasignación
Resultado obtenido	El médico queda desasignado con la fecha de desasignación

Tabla 17.39: Pruebas unitarias RF09

Conjunto de pruebas unitarias RF10

RF10PU01	
Descripción	Listar tratamientos
Resultado esperado	Visualizar el listado de tratamientos
Resultado obtenido	Se visualiza el listado de tratamientos

RF10PU02	
Descripción	Listar cuestionarios
Resultado esperado	Visualizar el listado de tratamientos
Resultado obtenido	Se visualiza el listado de tratamientos

RF10PU03	
Descripción	Pulsar sobre un cuestionario
Resultado esperado	Visualizar un listado de sus cuestiones
Resultado obtenido	Se visualiza un listado de sus cuestiones

RF10PU04	
Descripción	Listar cuestiones
Resultado esperado	Visualizar el listado de cuestiones
Resultado obtenido	Se visualiza el listado de cuestiones

RF10PU05	
Descripción	Listar respuestas
Resultado esperado	Visualizar el listado de respuestas
Resultado obtenido	Se visualiza el listado de respuestas

RF10PU06	
Descripción	Crear tratamiento con título y descripción
Resultado esperado	Tratamiento guardado
Resultado obtenido	Tratamiento guardado

RF10PU07	
Descripción	Crear tratamiento sin título, descripción o ambos
Resultado esperado	Tratamiento no guardado
Resultado obtenido	Tratamiento no guardado

RF10PU08	
Descripción	Crear cuestionario con nombre y al menos una cuestión
Resultado esperado	Cuestionario creado
Resultado obtenido	Cuestionario creado

RF10PU09	
Descripción	Crear cuestionario sin nombre
Resultado esperado	Cuestionario no creado
Resultado obtenido	Cuestionario no creado

RF10PU10	
Descripción	Crear cuestionario sin seleccionar cuestiones
Resultado esperado	Cuestionario no creado
Resultado obtenido	Cuestionario no creado

RF10PU11	
Descripción	Crear riesgo con la falta de algún elemento
Resultado esperado	Riesgo no creado
Resultado obtenido	Riesgo no creado

RF10PU12	
Descripción	Crear riesgo con una puntuación mínima superior a la máxima
Resultado esperado	Riesgo no creado
Resultado obtenido	Riesgo no creado

RF10PU13	
Descripción	Crear riesgo con puntuaciones consistentes
Resultado esperado	Riesgo creado
Resultado obtenido	Riesgo creado

RF10PU14	
Descripción	Asignar un riesgo a un cuestionario
Resultado esperado	El riesgo queda asignado y se visualiza
Resultado obtenido	El riesgo queda asignado y se visualiza

RF10PU15	
Descripción	Desasignar un riesgo a un cuestionario
Resultado esperado	El riesgo queda desasignado y se visualiza
Resultado obtenido	El riesgo queda desasignado y se visualiza

RF10PU16	
Descripción	Crear cuestión con pregunta y al menos una respuesta
Resultado esperado	Cuestión creada
Resultado obtenido	Cuestión creada

RF10PU17	
Descripción	Crear cuestión sin pregunta
Resultado esperado	Cuestión no creada
Resultado obtenido	Cuestión no creada

RF10PU18	
Descripción	Crear cuestión sin seleccionar respuestas
Resultado esperado	Cuestión no creada
Resultado obtenido	Cuestión no creada

RF10PU19	
Descripción	Crear respuesta con la falta de algún elemento
Resultado esperado	Respuesta no creada
Resultado obtenido	Respuesta no creada

RF10PU20	
Descripción	Crear respuesta con descripción y puntuación
Resultado esperado	Respuesta creada
Resultado obtenido	Respuesta creada

Tabla 17.40: Pruebas unitarias RF10

Conjunto de pruebas unitarias RF11

RF11PU01	
Descripción	Generar una simulación de un test
Resultado esperado	La paciente creada aparece en el listado de pacientes asignadas y con resultados del cuestionario
Resultado obtenido	La paciente creada aparece en el listado de pacientes asignadas y con resultados del cuestionario

Tabla 17.41: Pruebas unitarias RF11

17.5. Anexo - Despliegue

17.5.1. Software necesario

Para que el sistema pueda ser ejecutado serán necesarios las siguientes versiones de software:

Para la implementación del servidor:

- **NodeJS** - Versión 12.16.1
- **Express** - Versión 4.17.1
- **MySQL** - Versión 8.0.19

Para la implementación de la aplicación móvil:

- **Android** - Versión 5.1

17.5.2. Puesta en marcha

En nuestro caso, la UPV nos ha facilitado un dominio donde alojar nuestro servidor, en la dirección <http://ppd.webs.upv.es>. La instalación ha resultado sencilla gracias a la herramienta Plesk.

Para el despliegue de la BD, mediante la herramienta phpMyAdmin importamos los *scripts* ya mencionados en el apartado 17.3.3 «Bases de datos». Para el despliegue de la API importamos la carpeta que la contiene y configuramos los

parámetros necesarios para su ejecución, como el fichero de arranque (*index.js*), las variables de entorno o la instalación de los paquetes *npm*. Para que la API se conecte a la BD será necesario actualizar los datos del archivo *database.js*.

A partir de este momento tan solo falta arrancar la aplicación NodeJS instalada (API) y ya tendremos en marcha el servidor.

Bibliografía

- [1] Consejo de colegios de ingeniería informática (CCII). *CCII-N2016-02*. 2019.
- [2] S. Asensio-Cuesta y col. “Robustness and findings of a web-based system for depression assessment in a university work context”. En: *International Journal of Environmental Research and Public Health* 16.4 (2019), pág. 644.
- [3] S. Asensio-Cuesta y col. “Smartphone sensors for monitoring cancer-related quality of life: app design, EORTC QLQ-C30 mapping and feasibility study in healthy subjects”. En: *International Journal of Environmental Research and Public Health* 16.3 (2019), pág. 461.
- [4] J.L. Cox y col. “Validation of the Edinburgh Postnatal Depression Scale (EPDS) in non-postnatal women”. En: *Journal of Affective Disorders* 39.3 (1996), págs. 185-189.
- [5] Android Developers. URL: <https://developer.android.com>.
- [6] MDN web docs. *Recursos para desarrolladores, creados por desarrolladores*. URL: <https://developer.mozilla.org/es>.
- [7] L. Garcia-Esteve y col. “Validation of the Edinburgh postnatal depression scale (EPDS) in Spanish mothers”. En: *Journal of Affective Disorders* 75.1 (2003), págs. 71-76.
- [8] E. Jadresic y R. Araya. “Prevalencia de depresión posparto y factores asociados en Santiago, Chile”. En: *Rev Med Chile* 123.6 (1995), págs. 694-9.
- [9] Julie S. Moldenhauer. “Depresión posparto”. En: *MSD* (2020).
- [10] Stack Overflow. *Where Developers Learn, Share and Build Careers*. URL: <https://stackoverflow.com>.
- [11] Retrofit. *A type-safe HTTP client for Android and Java*. URL: <https://square.github.io/retrofit>.
- [12] Wikipedia contributors. *Edinburgh Postnatal Depression Scale* — *Wikipedia, The Free Encyclopedia*. 2019.