



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

TRABAJO DE FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA
MENCION INGENIERÍA DE SOFTWARE

Aplicación móvil para la gestión y monitorización de la
salud física y mental

Autor:
Pedro Rodríguez Vegas

Tutor:
Jesús María Vegas Hernández



Agradecimientos

En primer lugar, me gustaría agradecerles a mis padres y a mi hermana por todo el apoyo y toda la ayuda que siempre me han dado.

A mis amigos, Abel, Christopher y Héctor por haber hecho de este grado de ingeniería informática una experiencia muy divertida e increíble.

Y a mi tutor de TFG, Jesús muchas gracias por todo el trabajo y por haberme ayudado en todo este proyecto.

Resumen

En este proyecto se expone el proceso de desarrollo de una aplicación móvil orientada al cuidado y concienciación de la salud física y mental de sus usuarios. Para ello se hará uso de un Smartwatch, el cual permita obtener los datos de salud física de los usuarios en las distintas métricas de salud contempladas, y de una serie de cuestionarios, para que el usuario realice una autoevaluación de su actual estado de salud mental. Este objetivo se ha conseguido realizando una API Rest, una aplicación móvil y una persistencia de los datos suministrados.

Este proyecto se ha llevado a cabo utilizando en la parte del frontend los frameworks Ionic y React, junto con el lenguaje Typescript, y en la parte del backend el framework Spring y los lenguajes Java y SQL.

Abstract

This project presents the development process of a mobile application aiming at caring for and raising awareness about physical and mental health of its users. In order to do that, a Smartwatch will be used, which will allow users to record their physical health data in the different health metrics considered, and a series of questionnaires, which will allow users to perform a self evaluation of their current mental health. This target has been achieved by creating an API Rest, a mobile application, and a persistence of the supplied data.

This project has been carried out using the frameworks Ionic and React, along with the Typescript language in the frontend side, and the framework Spring along with Java and SQL languages for the backend side.

Índice general

Agradecimientos	III
Resumen	V
Abstract	VII
Lista de figuras	XI
Lista de tablas	XIII
1. Introducción	1
1.1. Contexto	1
1.2. Motivación	2
1.3. Objetivos	2
1.3.1. Objetivos del sistema	2
1.3.2. Objetivos de formación	3
2. Estado del arte	5
2.1. Dispositivos y aplicaciones	5
2.2. Historia de los Smartwatches	6
2.3. Aplicaciones similares	9
3. Planificación y presupuesto	11
3.1. Planificación	11
3.2. Historia de usuario	13
3.3. Product Backlog	13
3.4. Gestión y análisis de riesgos	14
3.5. Análisis de costes	18
4. Herramientas utilizadas	21
4.1. GitHub	21
4.2. Visual Studio Code	21
4.3. Android Studio	22
4.4. PostgreSQL Shell(PSQL)	22
4.5. Overleaf	23
4.6. Astah	23

4.7. Balsamiq	24
4.8. nRFCConnect	24
4.9. Postman	24
4.10. PostgreSQL	25
4.11. Mockaroo	25
4.12. Spring	26
4.13. Ionic	26
4.14. React	27
4.15. Node	27
4.16. Npm	28
4.17. Android	28
4.18. Gmail	28
5. Análisis	31
5.1. Roles de usuario	31
5.2. Requisitos de usuario	32
5.2.1. Requisitos funcionales	32
5.2.2. Requisitos no funcionales	33
5.3. Casos de uso	33
6. Diseño	35
6.1. Diseño de la base de datos	35
6.2. Diseño de la API	37
6.3. Diseño de la aplicación de usuario	43
6.4. Interfaz de usuario	44
6.4.1. Plantillas de la interfaz de usuario	45
6.4.2. Interfaz de usuario	49
7. Desarrollo del proyecto	57
7.1. Sprint 1	57
7.2. Sprint 2	59
7.3. Sprint 3	61
7.4. Sprint 4	63
7.5. Sprint 5	64
7.6. Sprint 6	66
8. Pruebas	67
9. Conclusiones	79
9.1. Trabajos futuros	80
Bibliografía	80

Índice de figuras

2.1. Primer Smartwatch, Pulsar, 1972.	6
2.2. IBM WatchPad, 2000.	7
2.3. Microsoft SPOT, 2004.	7
2.4. Pebble Smartwatch, 2012.	8
2.5. Apple Watch, 2015.	8
2.6. App salud física, Google Health.	9
2.7. App salud física, VeryFitPro.	9
2.8. App salud mental, TherapyChat.	10
3.1. Diagrama de Gantt.	12
3.2. Categorización de riesgos basada en Lyytinen's sociotechnical model of risk.	15
4.1. Logo de la herramienta GitHub.	21
4.2. Logo de la herramienta Visual Studio Code.	22
4.3. Logo de la herramienta Android Studio.	22
4.4. Logo de la herramienta Overleaf.	23
4.5. Logo de la herramienta Astah.	23
4.6. Logo de la herramienta Balsamiq.	24
4.7. Logo de la herramienta nRFConnect.	24
4.8. Logo de la herramienta Postman.	25
4.9. Logo de la herramienta Postgres.	25
4.10. Logo de la herramienta Mockaroo.	26
4.11. Logo de la herramienta Spring.	26
4.12. Logo de la herramienta Ionic.	27
4.13. Logo de la herramienta React.	27
4.14. Logo de la herramienta Node.	28
4.15. Logo de la herramienta Android.	28
4.16. Logo de la herramienta Gmail.	29
5.1. Diagrama de casos de uso.	34
6.1. Diagrama de despliegue.	35
6.2. Diagrama relacional de la base de datos.	36
6.3. Diagrama de la estructura de la API.	39
6.4. Diagrama de los controladores de la API.	39

6.5. Diagrama de los servicios de la API.	40
6.6. Diagrama de los repositorios de la API.	40
6.7. Diagrama de los modelos de la API.	41
6.8. Diagrama de la estructura de peticiones de la API.	42
6.9. Diagrama de la estructura del frontend.	43
6.10. Plantilla de la vista de login.	44
6.11. Plantilla de la vista de login.	45
6.12. Plantillas de las vistas de salud física.	45
6.13. Plantilla de la vista del menú de cuestionarios.	46
6.14. Plantillas de las vistas del formulario de cuestionarios.	46
6.15. Plantilla de la vista del historial de cuestionarios.	47
6.16. Plantilla de la vista de ajustes.	47
6.17. Plantillas de las vistas de menús de ajustes.	48
6.18. Plantilla de la vista de registro de usuario.	48
6.19. Plantillas de las vistas de recuperación de contraseña.	49
6.20. Vista del menú de login.	49
6.21. Vista del menú de salud física.	50
6.22. Vista del menú de estadísticas de salud física.	51
6.23. Vista de la leyenda del menú de estadísticas de salud física	51
6.24. Vista del menú de cuestionarios.	52
6.25. Vistas del formulario de cuestionarios.	52
6.26. Vista del historial de cuestionarios.	53
6.27. Vista del menú de ajustes.	53
6.28. Vistas de los menús de ajustes.	54
6.29. Vista del registro de usuarios.	54
6.30. Vistas de la recuperación de contraseña.	55
7.1. Vista de los dispositivos y servicios de una Mi Band 6 obtenidos en nrfConnect. .	60
7.2. Vista del proceso de obtención de un token privado con el software Huami-token.	61

Índice de tablas

3.1. Tabla de historias de usuario.	13
3.2. Product backlog	14
3.3. Tabla de Riesgos, tarea demasiado compleja.	15
3.4. Tabla de Riesgos, miembro del equipo enfermo.	16
3.5. Tabla de Riesgos, modificación requisitos.	16
3.6. Tabla de Riesgos, falta de formación en tecnologías utilizadas.	17
3.7. Tabla de Riesgos, implementación de nuevas tecnologías.	17
3.8. Tabla de Riesgos, pérdida o corrupción de tareas.	17
3.9. Tabla de Riesgos, aparición de estrés.	18
3.10. Tabla de costes.	19
7.1. Product backlog Sprint 1	58
7.2. Product backlog Sprint 2	59
7.3. Product backlog Sprint 3	62
7.4. Product backlog Sprint 4	63
7.5. Product backlog Sprint 5	65
7.6. Product backlog Sprint 6	66
8.1. Prueba registro	68
8.2. Prueba recuperación de contraseña	70
8.3. Prueba inicio sesión	70
8.4. Prueba cerrar sesión	71
8.5. Prueba visualización datos de métricas de salud	72
8.6. Prueba realizar cuestionario salud mental	73
8.7. Prueba visualizar resultados y explicación de cuestionarios de autoevaluación de salud mental.	74
8.8. Prueba visualizar histórico de cuestionarios de cuestionarios de autoevaluación de salud mental.	75
8.9. Prueba editar los datos personales del usuario.	76
8.10. Prueba editar los sistemas de medida de las métricas de salud.	77
8.11. Prueba editar los objetivos de las métricas de salud.	78

Capítulo 1

Introducción

1.1. Contexto

Actualmente la humanidad está viviendo uno de los periodos de la historia con más concienciación en torno a la salud de las personas, tanto física como mentalmente. Esta concienciación se debe en gran medida a una mayor educación y difusión de estos mensajes a través de escuelas, medios de comunicación, redes sociales, etc.

Sin embargo, esto no quita que la incidencia de enfermedades relacionadas con estos ámbitos haya crecido a un ritmo alarmante en los últimos años. Cabe destacar aquellas relacionadas con la salud mental (depresión, ansiedad, etc) ya que según la OMS[2] la mayor aparición de estas enfermedades ha desembocado en un aumento de las muertes de forma prematura, debido a enfermedades físicas prevenibles, y de los suicidios.

Y aunque muchos de estos trastornos son evitables actualmente de forma eficaz y con un coste relativamente reducido, un gran porcentaje de estas personas no son capaces de acceder a ellos, ya sea por falta de medios o estigmatización de los mismos.

A lo largo de la historia, y en algunos casos sobre todo en los últimos años, se ha intentado combatir este tipo de problemas de numerosas formas, actualmente una de las más eficientes ha sido la aparición del *Smartwatch* y las aplicaciones móviles de monitorización de la salud física, las cuales permiten a sus usuarios tener un mayor control y conciencia de su estado de salud actual.

Aunque este tipo de dispositivos y aplicaciones ha aumentado en gran medida la concienciación y la elusión de enfermedades relacionadas con la salud física, en especial la gran adopción, aceptación y asequibilidad de los dispositivos *Smartwatch*, no ha sido así en lo referente a las relacionadas con la salud mental, aunque en la actualidad se presente una mayor tendencia hacia ellas.

Por ello este proyecto de fin de grado se ha concebido para intentar trabajar sobre estas deficiencias desarrollando una aplicación de monitorización de la salud del individuo. A través de esta aplicación el usuario podrá obtener y gestionar sus datos de salud tanto física como mental.

De entre las opciones de monitorización de salud física, el usuario puede acceder a la información del número de calorías gastadas, el número de pasos realizados, el ritmo cardiaco que se está produciendo, el estrés que presenta y su peso actual. Cada una de estas métricas presenta una serie de variables tales como mínimo y máximo número de calorías gastadas, IMC actual (en Kg o Lb), distancia de pasos (en Km o Millas), entre otras. Además, el usuario puede acceder a un registro gráfico de la última semana de cada una de las métricas de salud disponibles, junto con una pequeña explicación de cada métrica y que representan los posibles resultados para la salud del individuo.

En cuanto a la salud mental, el usuario tiene a su disposición una serie de cuestionarios realizados por profesionales de la salud y que son de la propiedad del Ministerio de Sanidad[6]. Los cuales, han podido utilizarse en este proyecto, debido a que en el portal en el que se ofrecen se permite su reproducción total o parcial, modificación, distribución y comunicación para usos comerciales y no comerciales.

1.2. Motivación

Este proyecto tiene como motivación promover en mayor medida la difusión y concienciación de la sociedad en la lucha contra las enfermedades relacionadas con la salud física (tales como obesidad, anorexia, sedentarismo, etc)[1] y aquellas relacionadas con la salud mental (tales como la ansiedad, depresión, etc), las cuales hoy en día representan uno de los mayores problemas que afronta nuestra población debido a un gran incremento de las mismas. Aunque en la actualidad se está poniendo un mayor foco en la concienciación y en el tratamiento de estas, aún no es suficiente ya que existe cierto rechazo y estigmatización de las mismas (en muchos casos consciente o irracional).

Con el objetivo de combatir contra todas estas barreras y promover una mayor calidad de vida se ha decidido llevar a cabo esta aplicación, intentando aportar una perspectiva diferente y más cercana a cualquier tipo de usuario.

1.3. Objetivos

El objetivo principal de este proyecto es el desarrollo de una aplicación móvil la cual permita al usuario poder monitorizar y visualizar los datos relacionados con su salud física y mental. Además se perseguirán también los siguientes objetivos secundarios.

1.3.1. Objetivos del sistema

Al término del proyecto se pretende tener completados los siguientes objetivos del sistema, los cuales, aunque tienen una menor prioridad que el objetivo principal del proyecto, son de gran

importancia a la hora de obtener un producto de gran calidad:

- Permitir que el usuario pueda conectarse con su dispositivo smartwach.
- Permitir que el usuario pueda completar cuestionarios de evaluación de su actual estado mental.
- Implementar un histórico de los datos relacionados con la salud física del usuario.
- Implementar un histórico de los datos relacionados con la salud mental del usuario.
- Permitir que el usuario pueda cambiar de sistema de medida de las distintas unidades.

1.3.2. Objetivos de formación

En este proyecto se hará uso de ciertas herramientas y tecnologías que no han podido llegar utilizarse durante la carrera, o se profundizará en otras con las que ya se ha trabajado, por lo que los principales objetivos en el desarrollo profesional serán:

- Aprendizaje del framework Spring Boot para la creación de una API REST.
- Aprendizaje de los frameworks Ionic y React para la creación de aplicaciones híbridas.
- Aprendizaje del lenguaje Typescript para la creación de aplicaciones web.

Capítulo 2

Estado del arte

2.1. Dispositivos y aplicaciones

Antes de poder hablar de la historia de los siguientes elementos (*Smartwatches* y apps de salud), es necesario realizar una pequeña introducción para definir adecuadamente estos conceptos.

En primer lugar, un *Smartwatch* o una *Smartband* [3], a día de hoy, es básicamente un ordenador portátil el cual tiene forma de reloj y posee una pantalla táctil a través de la cual el usuario puede interactuar con el mismo. Las principales diferencias entre ambas son principalmente que los *Smartbands* están enfocadas en aquellas actividades relacionadas con la salud física y el ejercicio, mientras que los *Smartwatches* suelen incluir más funcionalidades. Por conveniencia en esta memoria se procederá a utilizar *Smartwatch* para englobar ambos dispositivos.

Mientras que los primeros modelos únicamente permitían realizar funciones básicas tales como mostrar la hora o utilizar una calculadora los más modernos permiten realizar una gran cantidad de funciones tales como llamar, mostrar mensajes, conectarse a teléfonos móviles, monitorizar constantes de salud, reproducir música ..., existiendo una gran diversidad de ellos en el mercado tanto en funcionalidades como en sistemas operativos como en métodos de conectividad.

Por otro lado, tenemos las aplicaciones de gestión y monitorización de constantes de salud, las cuales son aplicaciones generalmente utilizadas en sistemas móviles que ofrecen la posibilidad al usuario de tener un mayor conocimiento y control sobre diversas constantes vitales o categorías de salud. Estas funciones son realizadas a través de los datos obtenidos a través de dispositivos como *Smartwatches* o *Smartbands*. De entre las más comunes destacan:

- Monitorización de deportes y actividades físicas.
- Monitorización de la nutrición.
- Monitorización del peso.
- Registro de los ciclos de sueño

- Registro del estrés y la relajación.
- Registro de diversas métricas de salud (pasos, calorías, ritmo cardiaco, ciclo de menstruación, etc.).

2.2. Historia de los Smartwatches

La idea del primer reloj inteligente o *Smartwatch*, [5] aparece por primera vez en la década de 1970, cuando se empezó a comercializar el primero de ellos, el Pulsar, construido por Hamilton Watch Company. Estos dispositivos fueron mejorándose y añadiendo más funcionalidades tales como calculadora, alarmas y en algunos casos incluso permitiendo jugar videojuegos.



Figura 2.1: Primer Smartwatch, Pulsar, 1972.

En los siguientes años siguieron lanzándose nuevos dispositivos que iban añadiendo nuevas tecnologías, aunque no fue hasta la década de los 2000 que se empezaron añadir muchas de las funcionalidades que utilizamos a día de hoy. Entre todos ellos cabe destacar el SPOT (Smart Personal Objects Technology) de Microsoft y el WatchPad de IBM. El modelo de IBM [3] fue un prototipo de un WatchPad de muñeca que funcionaba con Linux destacando en gran medida por la duración de su batería que se consiguió que alcanzase las 12 horas, la inclusión de un calendario y la comunicación por Bluetooth. El SPOT [3] fue lanzado en el 2004 e incluía, en contraste con sus predecesores, información sobre el clima, noticias, resultados deportivos y se comunicaba a través de ondas de radio FM.



Figura 2.2: IBM WatchPad, 2000.



Figura 2.3: Microsoft SPOT, 2004.

Por último en la década de los 2010 estos dispositivos acabaron consolidándose y extendiendo su uso al público general. De entre los más importantes cabe destacar el Pebble y el Apple Watch.

El Pebble [4] fue uno de los *Smartwatch* más innovadores de la época, cabe destacar que este proyecto se llevó a cabo a través de Kickstarter el cual recaudó más de diez millones de dólares. Este dispositivo incluía una pantalla LCD con tinta electrónica, conexión con smartphones, notificaciones y control de reproductor de música.

Por otro lado, el Apple Watch[5] fue el *Smartwatch* más vendido del mundo. Este dispositivo dio a conocer al público general los *Smartwatch* y poco a poco extendió su uso entre la población. Este *Smartwatch* contaba con funcionalidades como registrar datos de deportes realizados, enviar y recibir mensajes, y realizar llamadas telefónicas.



Figura 2.4: Pebble Smartwatch, 2012.



Figura 2.5: Apple Watch, 2015.

En la actualidad, una de las áreas donde más atención se está poniendo es en la recolección de datos biométricos a través de estos dispositivos, facilitando que profesionales de la salud puedan acceder a ellos para realizar diagnósticos más completos y acertados, pudiendo incluso prevenir ciertas enfermedades (infartos, obesidad, etc).

Aunque esta recolección de datos provoca que exista un alto riesgo de ciberseguridad, ya que muchos de estos dispositivos no utilizan las herramientas necesarias para proteger la información de los usuarios.

A pesar del declive e impacto económico presentado en la actualidad debido a la pandemia mundial del COVID-19, uno de los campos que se mostraron menos afectados en la demanda fue el de la tecnología, incluyendo los *Smartwatches*. Existen diversos estudios que detallan que la demanda de estos dispositivos en 2020 se incrementó en un 20% en el primer trimestre del 2020, lo que significó un incremento de 13,7 millones de dispositivos nuevos en funcionamiento[7]. En la actualidad, fuera de esta pandemia, estos datos no han decrecido si no todo lo aumentando

en un 13% el número de ventas con respecto a trimestres anteriores del mismo año. Esto nos indica que estos dispositivos se encuentran en un mercado completamente en auge y crecimiento, con una base de clientes muy elevada y con perspectivas de ser todavía mayor[8].

2.3. Aplicaciones similares

Tras un exhaustivo proceso de búsqueda, tanto en portales web como portales de descarga de aplicaciones móviles, no se ha encontrado ninguna aplicación que mezcle los conceptos de salud física y salud mental, razón que me llevó a realizar este proyecto, por lo que no podemos realizar una comparación como tal por lo que se procederá a comparar las aplicaciones más conocidas con cada ámbito.

En cuanto a la salud físico tenemos Google Fit la cual es la aplicación de salud física de Google la cual permite monitorizar la actividad física de sus usuarios. Esta aplicación aunque incluye información sobre ritmo cardiaco, pasos, calorías, sueño, entre otros, incluyendo además representación gráfica en cada una de ellas. Además, incluye un diario de actividad en el cual el usuario puede registrar de forma manual datos sobre su presión arterial, peso, actividades y entrenamientos realizados.



Figura 2.6: App salud física, Google Health.

Otra aplicación relacionada con la salud física es VeryFitPro, esta aplicación permite seguir un registro de las calorías gastadas, la distancia recorrida, el ritmo cardiaco y el sueño. A diferencia de la aplicación anterior incluye únicamente representación gráfica en las métricas de pasos realizados, horas de sueño y ritmo cardiaco. Esta aplicación incluye también un mayor control de las funciones del *Smartwatch*, tales como alertas, control del reproductor de música, encontrar mi dispositivo, etc.



Figura 2.7: App salud física, VeryFitPro.

Por último, en cuanto a la salud mental, no he encontrado ninguna aplicación que permita a sus usuarios realizar autoevaluaciones psicológicas de su estado anímico y mental. La más similar en este ámbito sería TherapyChat. Esta aplicación permite a sus usuarios realizar llamadas o utilizar chats de apoyo con profesionales de la salud.



Figura 2.8: App salud mental, TherapyChat.

Capítulo 3

Planificación y presupuesto

3.1. Planificación

Este proyecto se ha realizado siguiendo una planificación con la metodología ágil SCRUM adaptada a un equipo de una sola persona, de forma que un único individuo asumirá todos los roles de la misma (SCRUM Master, Product Owner, etc).

La metodología SCRUM[9] es un marco de gestión de proyectos de metodología ágil que ayuda a equipos de desarrollo a estructurar y gestionar el trabajo mediante un conjunto de valores, principios y prácticas. Se ha decidido utilizar esta metodología ya que con ella se prioriza la velocidad de entrega sobre la documentación y la producción de prototipos de software para su posterior evaluación por parte de los usuarios.

Al utilizar la metodología ágil SCRUM el proyecto se ha dividido en una serie de periodos de tiempo, con una duración cada uno de dos semanas, los cuales reciben el nombre de sprints.

La fecha de inicio del proyecto se ha establecido el día 12 de Octubre del 2022 y su fecha de finalización el 06 de Febrero de 2023. Se han establecido 6 sprints, cada uno con una duración de 14 días, los cuales establecen una duración de 12 semanas. Ya que el TFG consta de 12 créditos y cada crédito equivale a unas 25 o 30 horas de trabajo, este proyecto debe contar con una duración de al menos 300 horas de trabajo. Al tener que realizar un aprendizaje de un conjunto de tecnologías desconocidas para el desarrollador se han superado este número de horas establecidas.

Para gestionar la planificación de este proyecto se ha hecho uso de varias herramientas como Microsoft Excel y Microsoft Project. La herramienta Microsoft Project se ha utilizado para realizar un diagrama de Gantt del proceso de desarrollo.

3.1. PLANIFICACIÓN

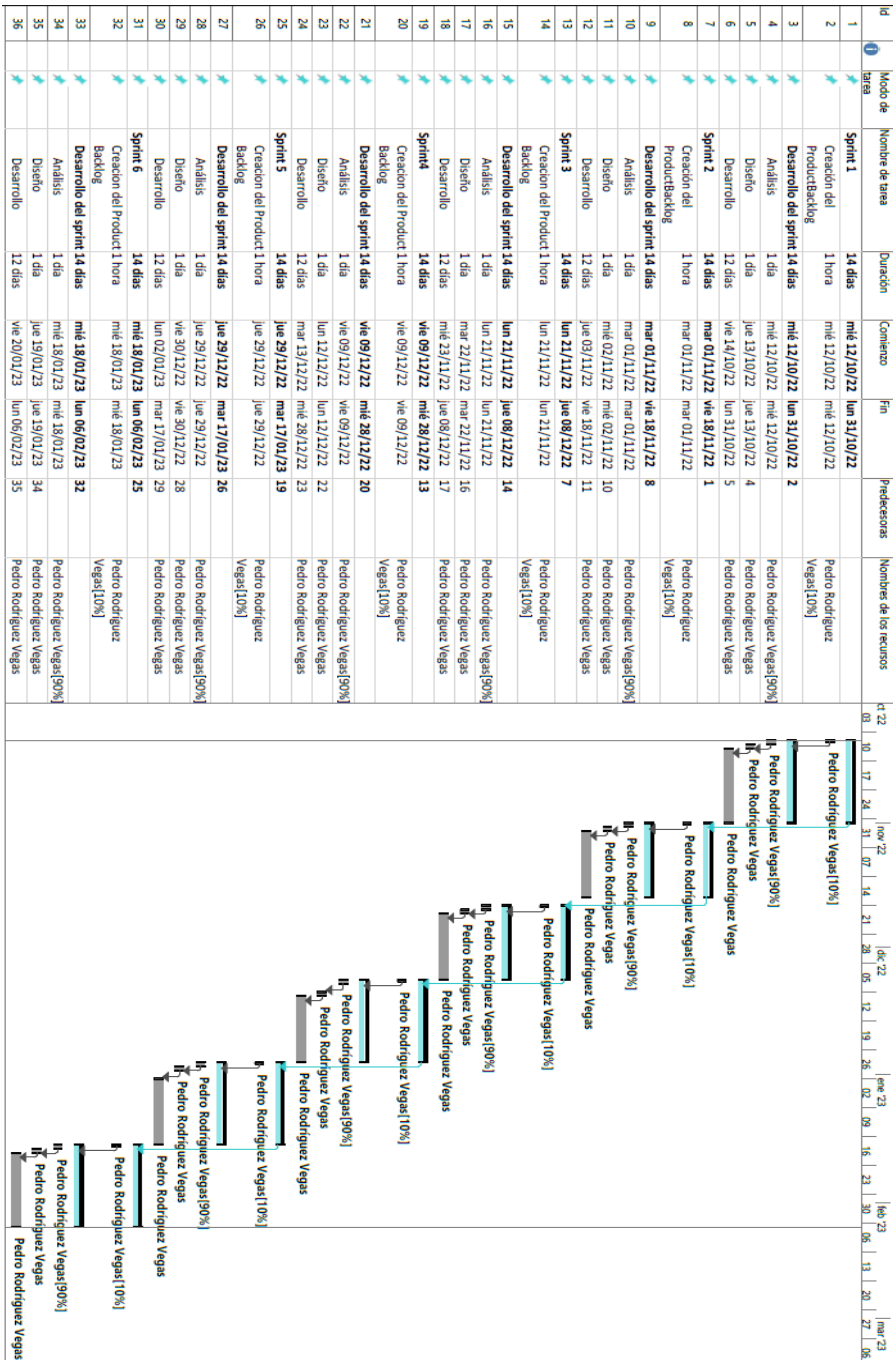


Figura 3.1: Diagrama de Gantt.

3.2. Historia de usuario

Una historia de usuario[10] es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final. Su función es determinar qué valor aportará a un cliente una función de software. En un marco ágil es la unidad de trabajo más pequeña posible que representa un objetivo concreto expresado desde el punto de vista de grupo de usuarios concreto.

Estas historias de usuario han sido extraídas y recogidos al inicio del proyecto en la siguiente tabla:

id	Nombre de las historias de usuario
1	Como usuario debo poder registrarme en la aplicación.
2	Como usuario debo poder iniciar sesión en la aplicación.
3	Como usuario debo poder cerrar sesión de la aplicación.
4	Como usuario debo poder vincular mi cuenta con mi correo electrónico.
5	Como usuario debo poder recuperar mi contraseña en caso de olvidarla.
6	Como usuario debo poder conectarme a mi Smartwatch.
7	Como usuario debo poder monitorizar los datos de varias métricas de salud física.
8	Como usuario debo poder ver los detalles de cada métrica de salud.
9	Como usuario debo poder realizar cuestionarios sobre la salud mental.
10	Como usuario debo poder ver una explicación sobre los resultados obtenidos en un cuestionario realizado.
11	Como usuario debo poder ver el histórico de cuestionarios que he realizado.
12	Como usuario debo poder editar mis datos de usuario.
13	Como usuario debo poder cambiar el sistema de medida de cada unidad.
14	Como usuario debo poder cambiar las marcas objetivo de las métricas de salud.

Tabla 3.1: Tabla de historias de usuario.

Este proyecto al tener una complejidad bastante alta y ser bastante extenso, existe la posibilidad de que la lista de estas historias deba modificarse, añadiendo nuevas, modificando alguna existente o dejando algunas sin resolverse en la versión presentada de este proyecto.

3.3. Product Backlog

El Product Backlog es una lista de tareas que contiene el trabajo ordenado por prioridades según el equipo de desarrolladores que se obtiene a partir de los requisitos y las expectativas del cliente sobre el proyecto. Además, a estas tareas se les asigna una estimación del esfuerzo que supondrá desarrollarlas, en este caso y he utilizado una escala del 1 al 8, teniendo un uno aquellas historias con menor complejidad y un ocho las que requieren mayor esfuerzo en su desarrollo.

En otras asignaturas de la carrera como Planificación y Gestión de Proyectos (PGP), para realizar este trabajo se han utilizado herramientas como Jira o Excel. En este caso se ha decidido

utilizar Excel ya que Jira supondría añadir una mayor complejidad y en este caso no añadiría ningún punto de relevancia en comparación, ya que al tener un equipo de un solo individuo no es necesario mostrar el avance de cada tarea al resto de integrantes.

id	Nombre de las historias de usuario	Esfuerzo
1	Como usuario debo poder registrarme en la aplicación.	2
2	Como usuario debo poder iniciar sesión en la aplicación.	2
3	Como usuario debo poder cerrar sesión de la aplicación.	4
4	Como usuario debo poder vincular mi cuenta con mi correo electrónico.	4
5	Como usuario debo poder recuperar mi contraseña en caso de olvidarla.	4
6	Como usuario debo poder conectarme a mi Smartwatch.	8
7	Como usuario debo poder monitorizar los datos de varias métricas de salud física.	6
8	Como usuario debo poder ver los detalles de cada métrica de salud.	6
9	Como usuario debo poder realizar cuestionarios sobre la salud mental.	5
10	Como usuario debo poder ver una explicación sobre los resultados obtenidos en un cuestionario realizado.	3
11	Como usuario debo poder ver el histórico de cuestionarios que he realizado.	3
12	Como usuario debo poder editar mis datos de usuario.	3
13	Como usuario debo poder cambiar el sistema de medida de cada unidad.	6
14	Como usuario debo poder cambiar las marcas objetivo de las métricas de salud.	3

Tabla 3.2: Product backlog

3.4. Gestión y análisis de riesgos

En un proyecto, un riesgo[11] es un evento o condición incierto el cual si ocurre puede llevar a producir un efecto positivo o negativo en los objetivos de un proyecto. El punto fundamental de una buena planificación de riesgos consiste en considerar y discernir la inexactitud del plan de trabajo elaborado, planear y controlar las situaciones que pueden llegar a resultar incorrectas.

Los riesgos que pueden llegar a presentarse en un proyecto se pueden categorizar de muchas formas[12], una de las principales los divide en:

- Riesgos de los Actores, se refieren a la gente involucrada en el desarrollo de la aplicación. Se identifican con la rotación del personal y la pérdida de conocimientos técnicos en el proyecto.
- Riesgos de Tecnología, se refieren a la tecnología involucrada en implementar la aplicación. Se identifican con la apropiación de las tecnologías y posibles fallos de las mismas.
- Riesgo de Estructura, se refieren a las relacionadas con las estructuras y el sistema de gestión, planificación y control. Se identifican con la participación de los usuarios.
- Riesgo de Tareas, se refieren a las relacionadas con el trabajo planificado. Se identifican con la complejidad y los retrasos del trabajo realizado.

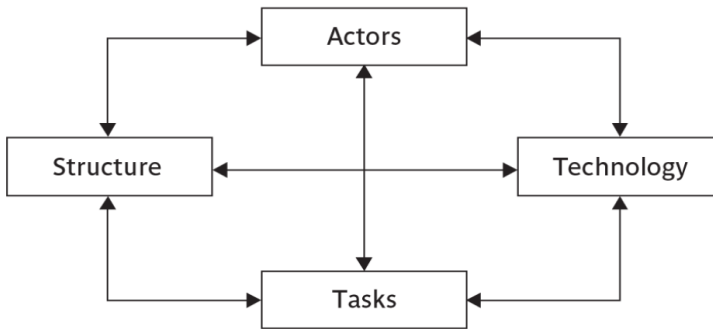


Figura 3.2: Categorización de riesgos basada en Lyytinen’s sociotechnical model of risk.

Al realizar este proyecto con un solo desarrollador este apartado adquiere especial importancia, ya que cualquier riesgo inesperado puede llegar a generar un elevado impacto sobre el transcurso de este. Los siguientes riesgos, han sido identificados utilizando la técnica de brainstorming con la experiencia personal del individuo.

Id de Riesgo	R001
Categoría	Tarea
Descripción	La tarea actual es demasiado compleja para poder completarla a tiempo.
Probabilidad	Alta
Impacto	Medio
Acciones de mitigación	Buscar más información sobre el problema y sobre la tecnología utilizada para encontrar otro enfoque.
Acciones de contingencia	Añadir un nuevo sprint y asignarle las tareas que no han podido completarse o añadirlo en una futura implementación.
Alcance	Sprint

Tabla 3.3: Tabla de Riesgos, tarea demasiado compleja.

Id de Riesgo	R002
Categoría	Actor
Descripción	Un miembro del equipo de trabajo enferma, provocando retrasos en el desarrollo.
Probabilidad	Media
Impacto	Medio
Acciones de mitigación	Establecer el inicio del proyecto antes de lo previsto de forma que siempre exista un número de días reservado para estas situaciones.
Acciones de contingencia	Analizar que tareas no han podido llevarse a cabo durante el transcurso de este sprint y asignárselas al siguiente.
Alcance	Todo el proyecto

Tabla 3.4: Tabla de Riesgos, miembro del equipo enfermo.

Id de Riesgo	R003
Categoría	Tarea
Descripción	Se necesita realizar modificaciones de distintos requisitos que conducen a retrasos en la planificación y por tanto se genera una pérdida de productividad del equipo.
Probabilidad	Media
Impacto	Medio
Acciones de mitigación	Adelantar el inicio del proyecto de forma que existan varios días reservados para estas situaciones.
Acciones de contingencia	Intentar aumentar el nivel de trabajo para compensar el tiempo de desarrollo necesario en caso de ser necesario añadir un nuevo sprint o añadir dichas modificaciones en una futura implementación.
Alcance	Todo el proyecto

Tabla 3.5: Tabla de Riesgos, modificación requisitos.

Id de Riesgo	R004
Categoría	Tecnología
Descripción	El equipo de trabajo no tiene formación en el uso de las tecnologías utilizadas de forma previa.
Probabilidad	Alta
Impacto	Medio
Acciones de mitigación	Formarse en dichas tecnologías mientras se realiza el proyecto.
Acciones de contingencia	Utilizar herramientas de consulta como foros, tutoriales o consultar con personas que poseen experiencia con las mismas.
Alcance	Todo el proyecto

Tabla 3.6: Tabla de Riesgos, falta de formación en tecnologías utilizadas.

Id de Riesgo	R005
Categoría	Estructura
Descripción	Se necesita implementar un nuevo tipo de tecnología que no estaba previsto.
Probabilidad	Baja
Impacto	Medio
Acciones de mitigación	Consultar la documentación de esa tecnología e intentar formarse mientras se realiza el proyecto.
Acciones de contingencia	Intentar adaptarse al uso de esa nueva tecnología y en caso de ser necesario añadir un nuevo sprint o incluir dicha funcionalidad en futuras implementaciones.
Alcance	Todo el proyecto

Tabla 3.7: Tabla de Riesgos, implementación de nuevas tecnologías.

Id de Riesgo	R006
Categoría	Estructura
Descripción	Se produce una pérdida o corrupción de parte del proyecto elaborado.
Probabilidad	Baja
Impacto	Alto
Acciones de mitigación	Utilizar herramientas de respaldo del trabajo realizado como por ejemplo GitHub.
Acciones de contingencia	Intentar recuperar las tareas perdidas y en caso de que se necesite de más tiempo o de rehacer dichas partes, añadir un nuevo sprint.
Alcance	Todo el proyecto

Tabla 3.8: Tabla de Riesgos, pérdida o corrupción de tareas.

Id de Riesgo	R007
Categoría	Actor
Descripción	Algún miembro del equipo de trabajo puede sufrir un aumento del estrés, reduciendo su productividad en el proyecto.
Probabilidad	Media
Impacto	Alto
Acciones de mitigación	Estar pendiente de la motivación del equipo antes de que llegue a afectar a la planificación.
Acciones de contingencia	Retrasar la planificación del proyecto y en caso de ser necesario añadir un nuevo sprint o implementar las tareas en futuras implementaciones.
Alcance	Todo el proyecto

Tabla 3.9: Tabla de Riesgos, aparición de estrés.

3.5. Análisis de costes

A la hora de calcular el presupuesto de este proyecto se ha procedido a buscar en varios portales el sueldo medio de un programador FullStack junior en España. Tras consultar en el sitio GlassDoor[13] se obtuvo una media de 30.000 € al año, mientras que el portal Talent[14] se obtuvo una de 35.000 € al año. Por ello se tomará en cuenta como sueldo medio la media aritmética de ambos portales, es decir 32.500 €/año. Este sueldo equivale a 2.708 € al mes y a 8,46 €/hora.

Según la planificación, quitando días que han sido tomados como festivos da lugar a 72 días de trabajo, trabajando unas 7 horas/día da lugar a unas 504 horas de trabajo en este proyecto, es decir 4.263,84 €.

Este proyecto ha sido desarrollado en un ordenador con un precio de 1200 €, del cual debemos calcular su amortización[15], es decir la depreciación efectiva que sufren los distintos elementos afectados por su propio funcionamiento, obtenemos que en equipos, útiles y herramientas para el tratamiento de la información, sistemas y programas informáticos el coeficiente lineal máximo es del 40% con un periodo máximo de 5 años. En este caso la amortización de este equipo sería de $1200 * 0,40 * (4/12) = 160$ €.

A todo lo anterior habría que sumarle 400 € en los periféricos utilizados (pantalla, ratón, teclado, Smartwatch, Smartphone ...), de los cuales al calcular su depreciación efectiva obtenemos una amortización de $400 * 0,40 * (4/12) = 53,33$ €.

Por último, tal y como hemos mencionado en el apartado de riesgos puede darse la situación en la que sea necesario añadir un nuevo sprint para poder hacer frente a los cambios que han aparecido y que no estaban previstos en la planificación inicial. Por ello se añadirá el presupuesto equivalente a dos sprints (es decir tendríamos unos 24 días laborables) para hacer frente a estas situaciones, por lo tanto, habría que añadir $24 * 7 * 8,46$ € = 1421,28 €.

Concepto	Precio
Presupuesto base	4263,84 €
Amortización del equipo	160,00 €
Amortización de los periféricos	53,33 €
Presupuesto prevención de riesgos	1421,28 €
Importe total	5898,45 €

Tabla 3.10: Tabla de costes.

Capítulo 4

Herramientas utilizadas

4.1. GitHub

GitHub[16] es un servicio de gestión de código que está basado en repositorios Git alojados en la nube, el cual ayuda a los desarrolladores a almacenar, gestionar y controlar cualquier cambio realizado en un proyecto. Esta aplicación al estar basada en Git incorpora un sistema de control de versiones, el cual permite a los desarrolladores poder acceder y recuperar versiones anteriores del proyecto, y un sistema de ramas, en el que los desarrolladores al trabajar en una nueva funcionalidad duplican la rama principal del proyecto, trabajan sobre ella y una vez han terminado y comprobado que todas las funcionalidades y test siguen funcionando lo fusionan con la línea principal. Al ser una empresa se ofrecen diversos planes de suscripción en función de su uso, en este proyecto se ha decidido utilizar el plan gratuito. Además, GitHub incluye una interfaz de usuario que es amigable para desarrolladores principiantes, ya que en caso de no querer utilizarla es necesario poseer ciertos conocimientos de Git y de sistemas de gestión y control de versiones.

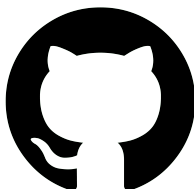


Figura 4.1: Logo de la herramienta GitHub.

4.2. Visual Studio Code

Visual Studio Code[17], también conocido como Visual Studio, es un entorno de desarrollo integrado (IDE) gratuito desarrollado por Microsoft, el cual permite editar, sugerir, depurar y compilar código en una gran cantidad de lenguajes y sistemas gracias a sus extensiones. Visual Studio incorpora un marketplace de extensiones bastante amplio, las cuales pueden ser

desarrolladas tanto por el propio Microsoft, como por otras compañías propietarias de ciertos lenguajes o tecnologías, o por la comunidad. Otro punto a favor en el uso de esta tecnología es que incluye de forma nativa integración con Git, permitiendo a sus usuarios poder ejecutar comandos, gestionar y controlar versiones del repositorio del proyecto en el que estamos trabajando.

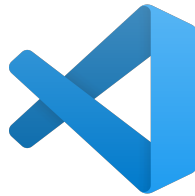


Figura 4.2: Logo de la herramienta Visual Studio Code.

4.3. Android Studio

Android Studio[18] es el entorno de desarrollo integrado (IDE) oficial de construcción de aplicaciones Android y fue desarrollado por Google. Esta herramienta está basada en uno de los editores de código más importantes y potentes actualmente, que es IntelliJ IDEA y en sus herramientas para desarrolladores. Uno de los puntos principales de Android Studio es la incorporación de emulación de dispositivos, el cual permite a los desarrolladores realizar un proceso de validación y depuración de una forma más rápida y sencilla. Su sistema de compilación está basado en Gradle, el cual ofrece al desarrollador la posibilidad de personalizar la construcción de proyectos. Además, Android Studio ofrece integración con Sistemas de versiones basados en Git, lo que permite poder a los desarrolladores tener un mayor control y seguridad sobre sus proyectos.



Figura 4.3: Logo de la herramienta Android Studio.

4.4. PostgreSQL Shell(PSQL)

PSQL[19] es un terminal interactivo que utiliza el sistema de gestión de bases de datos Postgres, el cual permite al desarrollador gestionar y realizar consultas SQL a un servidor que implemente Postgres. Esta terminal además de comandos también acepta ficheros los cuales contengan consultas SQL, e incorpora ciertas herramientas de personalización que faciliten su uso al desarrollador. Se ha decidido utilizar esta consola ya que se trabajó con ella en la asignatura de Análisis y Diseño de Bases de Datos.

4.5. Overleaf

Overleaf[20] es un editor desarrollado para el sistema de composición de textos Latex basado en un servicio de nube. Overleaf permite escribir, editar y publicar elementos de alta calidad tipográfica. Una de sus principales características es que al ser un servicio alojado en la nube permite que en un mismo documento se puedan realizar modificaciones por más de una persona de forma simultánea. Destaca también la integración que tiene este servicio con herramientas de respaldo de archivos, también en nube, como Dropbox o cualquier sistema de gestión de código basado en repositorios Git, como GitHub entre otros. Esta herramienta incluye también el uso de plantillas ya generadas por otros usuarios, permitiendo una mayor rapidez en la elaboración de documentos, incluye también un compilador propio el cual permite una visualización inmediata de las modificaciones realizadas y la descarga del documento completo en formato PDF.



Figura 4.4: Logo de la herramienta Overleaf.

4.6. Astah

Astah[21], también conocido como JUDE, es una herramienta de modelado de aplicaciones software UML, el cual es el lenguaje de modelado estándar utilizado en la ingeniería de software. Sus principales características son la simplicidad y la calidad en sus herramientas de desarrollo, ya que esta herramienta ofrece la posibilidad de realizar una gran variedad de diagramas UML en un corto periodo de tiempo. Entre los diagramas que permite desarrollar esta aplicación, destacan el diagrama de clases, el de interacción, el de casos de uso, el de despliegue... Astah no es una herramienta gratuita, por lo que ofrece varios planes de uso remunerados, en este proyecto se ha hecho uso de la licencia que tiene la universidad de Valladolid, la cual es gratuita para los alumnos.



Figura 4.5: Logo de la herramienta Astah.

4.7. Balsamiq

Balsamiq[22], también conocido como Balsamiq Wireframes, es una herramienta de diseño de interfaces de usuario, mockups o prototipos con una estética de pizarra o cuaderno de trabajo. Esta herramienta permite acelerar el proceso de desarrollo de las vistas de una aplicación permitiendo que el usuario pueda hacerse una mejor idea de cómo podría llegar a verse el producto final. Esta herramienta no dispone de un plan de uso gratuito, por lo que en el desarrollo de este proyecto se ha utilizado el periodo de prueba gratuito de un mes que se obtiene al registrarse. Esta herramienta permite exportar los diseños realizados en una gran variedad de formatos, y además se ha añadido la posibilidad de trabajar en la nube con todas sus funcionalidades, eliminando la necesidad de tener que instalar otro software en el dispositivo de trabajo.



Figura 4.6: Logo de la herramienta Balsamiq.

4.8. nRFConnect

nRFConnect[23] es una aplicación que se encuentra disponible tanto en sistemas IOS, como Android e incluso Windows, la cual permite a sus usuarios tener un mayor control sobre sus dispositivos nRF. Esta herramienta permite tener un mayor control sobre dispositivos BLE (Bluetooth Low Energy), entre los que se encuentran entre otros pulseras y relojes de actividad (SmartBand y SmartWatch respectivamente), permitiendo a sus usuarios poder escanear, explorar, vincular y comunicarse con estos dispositivos.



Figura 4.7: Logo de la herramienta nRFConnect.

4.9. Postman

Postman[24] es un entorno creado para el desarrollo y uso de APIs. Esta herramienta permite a los desarrolladores simplificar y optimizar el proceso de desarrollo de una API de una manera intuitiva. Postman ofrece una serie de herramientas que permiten almacenar, organizar y trabajar con varias APIs de forma independiente a través de espacios de trabajo organizados en los que pueden participar varios desarrolladores tanto de una misma empresa como de otras partes del

mundo. Además, incluye herramientas que permiten mejorar la calidad de las APIs utilizando reglas que garanticen los estándares propuestos para el desarrollo. Postman incluye diversos planes de uso en función de la finalidad que se vaya a desempeñar, en este caso se ha hecho uso del plan gratuito.



Figura 4.8: Logo de la herramienta Postman.

4.10. PostgreSQL

PostgreSQL[19], también conocido como Postgres, es un sistema de gestión de bases de datos relacionales orientado a objetos de código abierto y gratuito que destaca en gran medida por su robustez, desempeño y alta reputación. Fue creado originalmente por la universidad de California, y en la actualidad su desarrollo es llevado a cabo a través de una gran comunidad de colaboradores de todo el mundo, gracias a su licencia open source, y gracias a ello se ha convertido en una de las opciones más sólidas y con mayor reputación. Su principal uso se encuentra en el trabajo con bases de datos relacionales, en las que su gestión y administración es realizada a través de consultas relacionales SQL, aunque Postgres también ofrece la posibilidad de trabajar con bases de datos no relacionales, para lo cual utiliza ficheros de texto en formato JSON que contienen las instrucciones y reglas de las mismas.

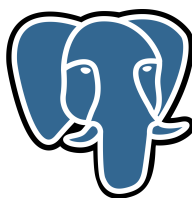


Figura 4.9: Logo de la herramienta Postgres.

4.11. Mockaroo

Mockaroo[25] es una aplicación web gratuita desarrollada por Mark Brocato que permite al usuario generar grandes cantidades de datos de forma automática y aleatoria. Esta herramienta está alojada en una página web, por lo que no es necesario instalar ningún tipo de software adicional. Entre sus principales peculiaridades se encuentra la gran variedad de tipos de datos ofrecidos que pueden ser generados de forma aleatoria, el uso de reglas o fórmulas que permiten controlar la generación de datos (las cuales pueden ser descritas por el desarrollador utilizando el lenguaje Ruby) y la gran variedad de formatos en los que permite exportar los datos generados.

Mockaroo ofrece varios planes de uso, la mayoría de ellos de pago, y un plan gratuito que permite al usuario generar hasta 100.000 líneas de datos de prueba en una sola ejecución.



Figura 4.10: Logo de la herramienta Mockaroo.

4.12. Spring

Spring[26] es un framework utilizado para el desarrollo de aplicaciones en la plataforma de Java, es de código abierto y fue desarrollado por Rod Johnson en 2003 bajo la licencia Apache 2.0. Spring ofrece un gran soporte de infraestructura a nivel de aplicación permitiendo a los desarrolladores implementar un gran abanico de modelos y configuraciones especialmente destinadas a aplicaciones para Java empresarial permitiendo el despliegue de las mismas en casi cualquier tipo de plataforma. Todo ello permite que los desarrolladores puedan abstraerse y trabajar directamente en la capa de negocio de la aplicación agilizando por tanto el proceso de desarrollo. Además, como importancia histórica, Spring puede considerarse como la base de los frameworks desarrollados para el entorno Java, ya que da soporte a muchos de ellos como Hibernate, EJB, Struts...



Figura 4.11: Logo de la herramienta Spring.

4.13. Ionic

Ionic[28] es un framework de código abierto SDK (kit de desarrollo software) destinado para el desarrollo de aplicaciones móviles híbridas o multiplataforma y fue creado por Drifty Co en 2013. La peculiaridad de Ionic es que utilizando código de aplicaciones web (CSS, HTML, Javascript ...) permite generar de forma nativa el equivalente a otras plataformas, en este caso Ionic ofrece la posibilidad de generar código tanto Android como IOS. Existen varios tipos de implementaciones que son aceptados por Ionic, las más importantes son Cordova, Capacitor y Native las cuales permiten ejecutar código Javascript en una vista web utilizando los distintos componentes y extensiones nativos de cada plataforma (acceso a memoria, peticiones http,

gestión de enlaces ...). Al poder utilizar código web como base de desarrollo para aplicaciones híbridas, Ionic permite la integración de distintos frameworks utilizados en desarrollo Frontend, entre los que se encuentran Angular, React y Vue. A diferencia de otros competidores como Flutter, que es desarrollado por Google, Ionic ofrece a sus usuarios un mayor control y una menor limitación con respecto a su competencia.



Figura 4.12: Logo de la herramienta Ionic.

4.14. React

React[29], también conocido como React.js, es un framework de código abierto creado en Javascript, es utilizado para el desarrollo de aplicaciones web basadas en componentes y fue desarrollado por Facebook. El principal objetivo de este framework es el desarrollo de aplicaciones web de forma rápida y eficiente utilizando la menor cantidad de código posible. Uno de los puntos y funcionalidades más importantes que ofrece React son los componentes, un componente es una porción de código independiente que puede ser reutilizado posteriormente. En este proyecto se decidió utilizar este framework, ya que es uno de los más utilizados en el mundo corporativo actual debido a su gran potencia y versatilidad. Además, aunque durante la carrera se ha hecho uso del framework Angular, el cual es similar a React y podría haber sido utilizado sin ningún problema ya que es compatible también con Ionic, se decidió utilizar este para adquirir nuevos conocimientos en el desarrollo de aplicaciones Frontend.

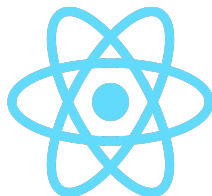


Figura 4.13: Logo de la herramienta React.

4.15. Node

Node[30] es un entorno en tiempo de ejecución basado en Javascript, de código abierto, asíncrono y destinado para la capa del servidor, aunque no está limitado a ella. Esta herramienta es especialmente utilizada en el diseño y desarrollo de aplicaciones que utilizan algún tipo de servidor, permitiendo a los desarrolladores ejecutar código Javascript de forma local o desplegar servidores sin aprender otro lenguaje. Fue creado con el objetivo de ser útil en la creación de aplicaciones web altamente escalables por Ryan Dahl en 2009.



Figura 4.14: Logo de la herramienta Node.

4.16. Npm

Npm[31], abreviatura de Node Package Manager, es un gestor de paquetes para el lenguaje de programación Javascript creado por npm Inc, la cual es una subsidiaria de Github y de Microsoft. Este gestor consiste en una línea de comandos que permite al desarrollador acceder a la base de datos del registro de paquetes del que está compuesto, tanto públicos como, en el caso de tener acceso a ellos, como privados. Además, este gestor de paquetes cuenta con una página web que permite al desarrollador buscar aquellos que necesite, consultar la documentación del mismo o implementaciones de sus funcionalidades.

4.17. Android

Android[32] es un sistema operativo móvil basado en una versión modificada del kernel del Linux y otros programas de código abierto, diseñado especialmente para pantallas táctiles y desarrollado por Google. Inicialmente el lenguaje de programación de este sistema operativo era Java, aunque actualmente se está imponiendo cada vez más el uso de Kotlin. En la actualidad más del 70 por ciento de los smartphones están basados en algún tipo de versión o distribución de Android y no solamente los smartphones si no cualquier tipo de dispositivo que utilice una cpu con arquitectura ARM, entre los que se encuentran por ejemplo dispositivos de TV (Fire TV, Chromecast ...) o un *Smartwatch*.



Figura 4.15: Logo de la herramienta Android.

4.18. Gmail

Gmail[33] es un servicio gratuito de correo electrónico desarrollado por Google en 2004 y que en la actualidad es el más utilizado en todo el mundo. Esta herramienta ofrece la posibilidad a su usuario de poder acceder a su cuenta de correo desde una gran variedad de dispositivos que tengan acceso a la red, siendo esta una aplicación multiplataforma. Además, una de las

características más interesantes de Gmail es que este servicio ofrece la posibilidad a sus usuarios de acceder a su correo electrónico utilizando los protocolos POP e IMAP.



Figura 4.16: Logo de la herramienta Gmail.

Capítulo 5

Análisis

Este proyecto consiste en crear una aplicación móvil que permita al usuario poder tener un mayor control e información sobre su salud física y mental. Estos objetivos se llevan a cabo utilizando un *Smartwatch* y una serie de cuestionarios sobre la autoevaluación de la salud mental que permiten obtener los datos e información de cada una de las métricas o puntos de interés. Además, se ha llevado a cabo el desarrollo de una API, la cual permite almacenar los datos obtenidos a través del usuario.

En este apartado se procederá a elaborar un análisis con el objetivo de poder recoger y modelar los conceptos principales de este proyecto de fin de carrera, utilizando para ello, entre otras cosas, diagramas UML y tablas de requisitos.

5.1. Roles de usuario

En UML (Unified Modeling Language) un actor representa un tipo de usuario el cual desempeña un determinado comportamiento o una serie de funcionalidades, las cuales son completamente diferenciadas entre los distintos tipos de sujetos existentes. Los actores pueden representar roles desempeñados tanto por grupos de usuarios como software externo.

En esta aplicación se ha decidido no utilizar un rol de administrador, ya que la aplicación no requiere de ningún tipo de tarea o funcionalidad que necesite verificación o aprobación de un determinado tipo de usuario.

Esta aplicación únicamente utiliza un tipo de rol de usuario, el cual será el de un usuario cualquiera el cual quiera tener un mayor conocimiento sobre su estado de salud actual. Al tener un solo tipo de usuario, este posee todas las funcionalidades presentes en la aplicación. El único requisito que se le pide a este usuario es el de tener una cuenta en la aplicación y la de tener un reloj inteligente para poder registrar sus métricas de salud física.

5.2. Requisitos de usuario

En UML, los requisitos de un sistema son una serie de puntos que describen los servicios y las restricciones que puede ofrecer o presentar un determinado sistema o aplicación al usuario para resolver un problema o conseguir un objetivo concreto.

Para obtener los requisitos de este sistema se ha llevado a cabo un procedimiento en cascada, en el cual se divide este proceso en varias etapas secuenciales, las cuales son:

- Descubrimiento de una necesidad o funcionalidad en el sistema.
- Análisis dicha necesidad y determinar su alcance.
- Documentación de dichas funcionalidades de forma detallada.
- Verificación de la nueva funcionalidad con los servicios y restricciones existentes del sistema.

5.2.1. Requisitos funcionales

Los requisitos funcionales, son aquellos que describen los servicios que el sistema debe proporcionar al usuario y de cómo reaccionar a una determinada situación. Los requisitos que van a intentar completarse en este proyecto son:

- **RF01:** El usuario debe poder registrarse en el sistema.
- **RF02:** El usuario debe poder iniciar sesión en el sistema utilizando su dirección de correo electrónico y su contraseña.
- **RF03:** El usuario debe poder cerrar sesión en el sistema.
- **RF04:** El usuario debe poder recuperar su cuenta de usuario en caso de haber olvidado su contraseña.
- **RF05:** El usuario debe poder verificar su identidad utilizando la dirección de correo electrónico asociada a su cuenta.
- **RF06:** El usuario debe de poder conectar su Smartwatch a la aplicación.
- **RF07:** El usuario debe poder monitorizar distintas métricas relacionadas con su salud física.
- **RF08:** El usuario debe poder realizar cuestionarios sobre la salud mental.
- **RF09:** El usuario debe de poder visualizar los resultados de los cuestionarios cuestionarios realizados.
- **RF10:** El usuario podrá acceder a un histórico de los cuestionarios que ha realizado.
- **RF11:** El usuario debe poder modificar sus datos dentro de la aplicación.
- **RF12:** El usuario debe poder modificar el tipo de sistema de medida de las distintas unidades utilizadas.
- **RF13:** El usuario debe poder modificar los objetivos de sus métricas de salud.

5.2.2. Requisitos no funcionales

Los requisitos no funcionales, son aquellos que describen una restricción sobre los servicios o funcionalidades que ofrece un sistema. Los requisitos que van a intentar completarse en este proyecto son:

- **RNF01:** El usuario deberá de validar su dirección de correo electrónico antes de poder utilizar la aplicación.
- **RNF02:** La aplicación deberá poder ser utilizada en los sistemas operativos móviles Android e IOS.
- **RNF03:** El sistema no deberá permitir a los usuarios modificar su dirección de correo electrónico.
- **RNF04:** La aplicación deberá permitir a sus usuarios conectarse a ella a través de Internet utilizando una conexión segura.
- **RNF05:** La aplicación deberá responder de modo fiable y con un tiempo de respuesta tolerable que no dificulte la interacción del usuario.
- **RNF06:** La aplicación debe de ser segura y almacenar los datos sensibles de los usuarios de forma encriptada.

5.3. Casos de uso

En UML, un caso de uso consiste en un conjunto de escenarios, los cuales pueden tener éxito o no, relacionados entre ellos a través del resultado que el actor espera obtener. Un escenario es una secuencia de acciones realizadas por el sistema, las cuales conducen a un resultado observable y valioso para un actor en particular. Un escenario no tiene por qué terminar de forma exitosa para cumplir un escenario, ya que se suelen crear escenarios alternativos que prevean situaciones alternativas.

Los casos de uso proporcionan una estructura que permite expresar los requisitos funcionales de un determinado sistema. Generalmente suelen representarse de forma visual utilizando un diagrama de caso de uso, el cual incluye los casos de uso y actores que interactúan con el sistema para llevar a cabo un determinado escenario. En este proyecto el diagrama de casos de uso que se ha utilizado es el siguiente (Figura 6.1).

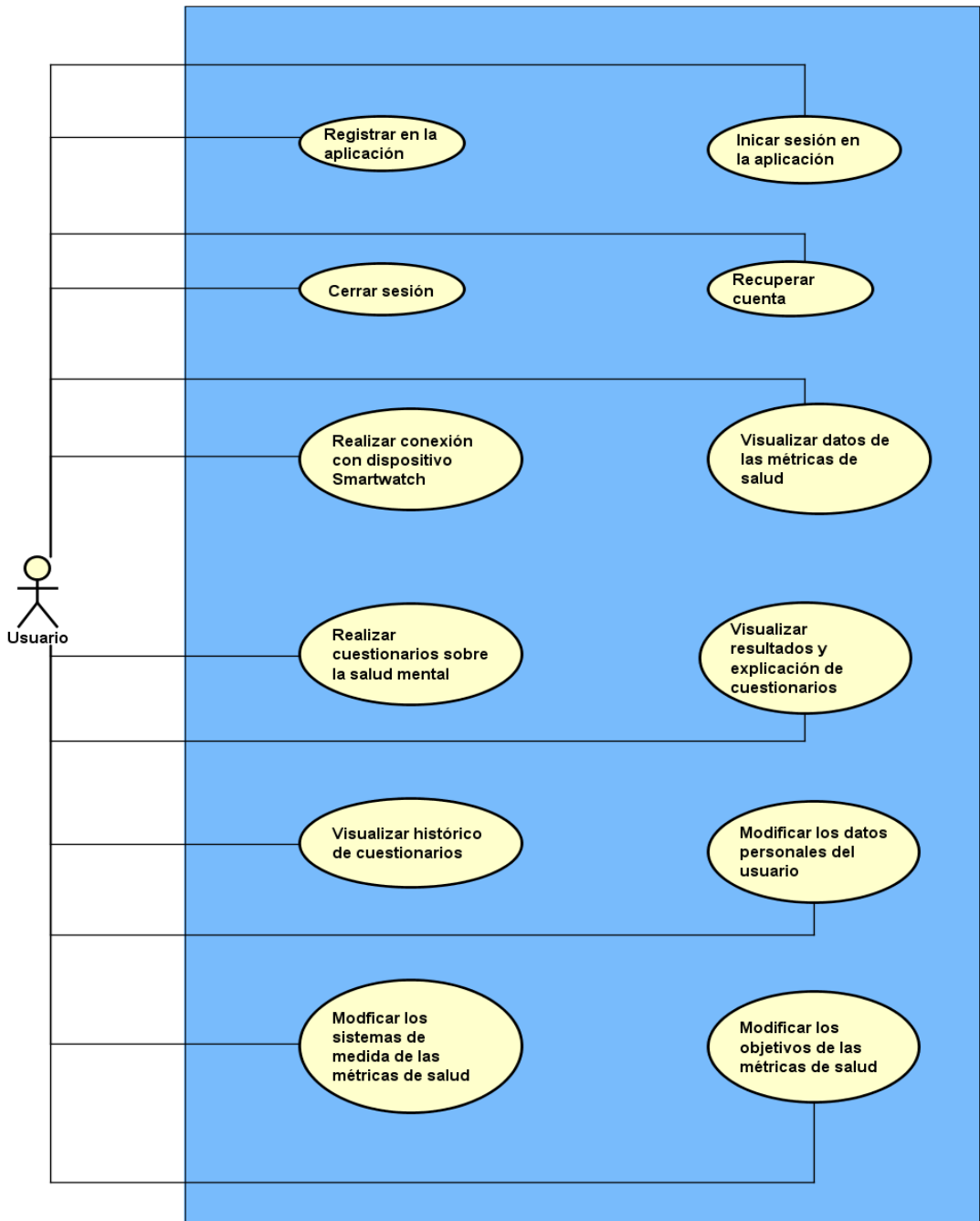


Figura 5.1: Diagrama de casos de uso.

Capítulo 6

Diseño

En este capítulo se hablará del diseño que se ha seguido al desarrollar la aplicación. Tal y como se llevó a cabo el análisis del sistema en el capítulo anterior, en los siguientes apartados se hablará de la estructura de la aplicación en todas sus partes, tanto en la aplicación móvil, como en API, como en la persistencia de la misma.

En cuanto a la interfaz de usuario se han incluido los bocetos originales, realizados en Balsamiq, que se utilizaron para desarrollar la interfaz final junto con su resultado final. Además, se ha incluido un diagrama de despliegue, el cual es un tipo de diagrama UML que permite representar la arquitectura del sistema en ejecución, incluyendo los entornos hardware o software y el middleware utilizado. La figura 7.1 muestra el diagrama de despliegue de este proyecto:

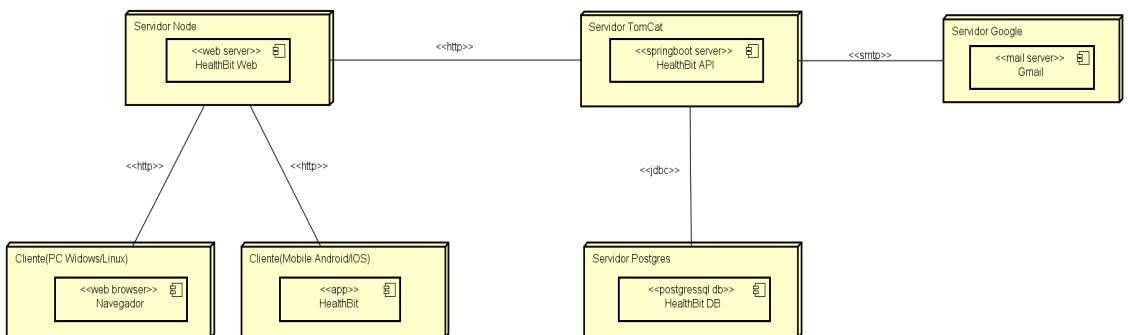


Figura 6.1: Diagrama de despliegue.

6.1. Diseño de la base de datos

En cuanto a la persistencia se ha realizado un diseño de la base de datos que permita almacenar los valores de las nuevas métricas de salud, junto con los cuestionarios que han sido realizados por el usuario. Cabe destacar la funcionalidad de dos tablas en concreto de todo el diseño, las

cuales son la tabla USER y la tabla TOKEN.

La tabla USER almacena la información del usuario, la cual es introducida en el registro o actualizada posteriormente por el mismo, pero cabe destacar que los campos de email y user_name son únicos, permitiendo de esta forma que no existan varios usuarios con el mismo nombre o dirección de correo. Este hecho es fundamental, ya que, aunque el usuario puede intentar editar su nombre en la aplicación, la cual comprobará que no se encuentre ya en uso, no podrá modificar su dirección de correo, ya que esta es usada para recuperar la cuenta del usuario y para crear los distintos tokens de verificación.

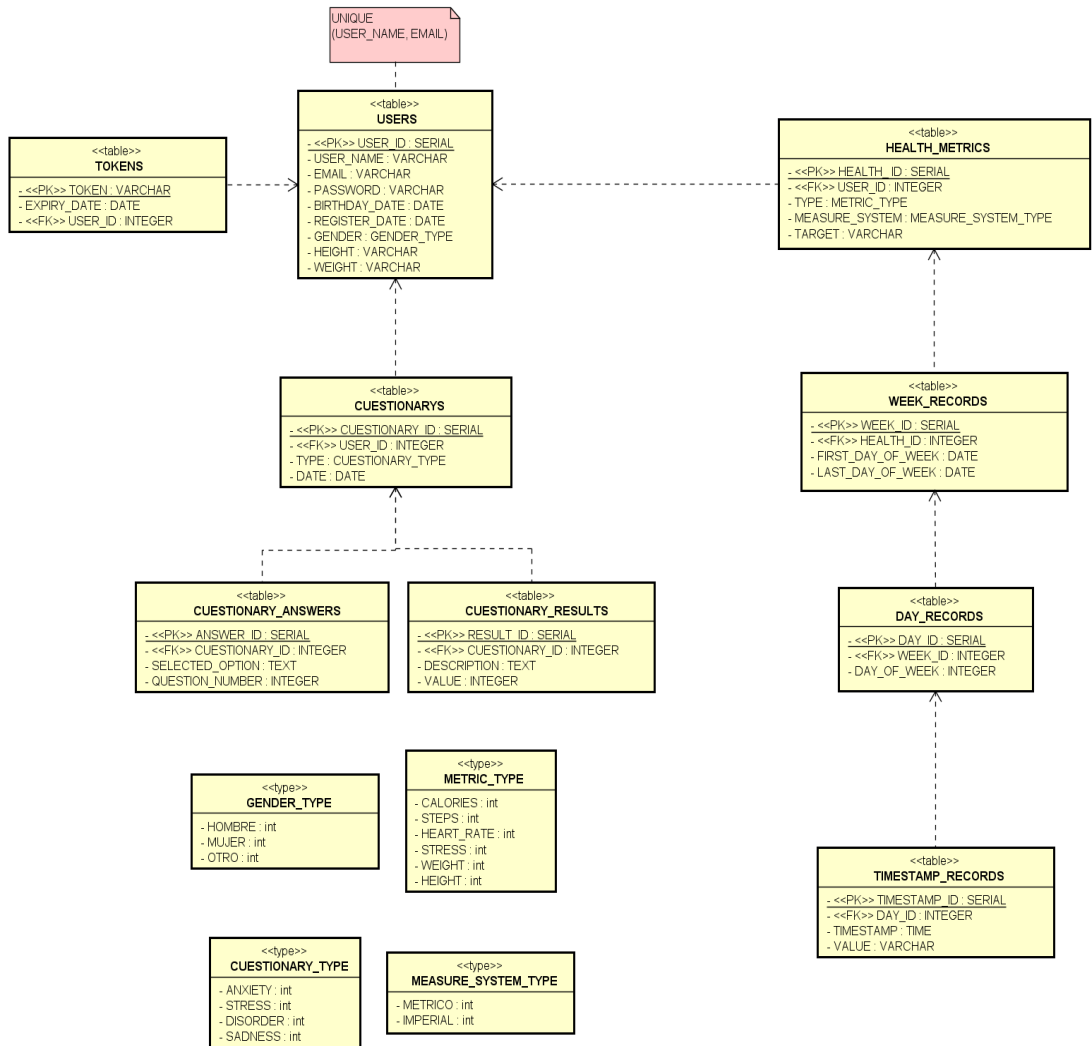


Figura 6.2: Diagrama relacional de la base de datos.

Por otro lado, la tabla TOKEN, tiene la función de almacenar todos los TOKENS que vayan a ser usados por los usuarios, aunque solamente aquellos que tengan larga duración. Esta aplicación utiliza cuatro tipos de tokens distintos dependiendo de la función que desempeñen. Los tokens que se utilizan en la aplicación son:

- **Bearer Token:** Este token se utiliza para validar la identidad del usuario a la hora de realizar una petición a la API. Se obtiene cuando el usuario inicia sesión correctamente en la aplicación, tiene una duración de *5 minutos* y no es almacenado en la base de datos. Este token tiene esta duración ya que en caso de que el usuario decida cerrar su sesión o se decida eliminarle el acceso actual a la misma, al utilizar tokens sin estado esta acción no sería posible por lo que se ha decidido establecer en este una duración muy reducida.
- **Refresh Token:** Este token es utilizado para permitir al usuario que siga utilizando la aplicación una vez su Bearer Token ha caducado, generando uno nuevo con la misma duración. Se obtiene cuando el usuario inicia sesión correctamente en la aplicación, tiene una duración de *1 semana* y es almacenado en la base de datos. Este token tiene esta duración con el objetivo de que el usuario no tenga que estar continuamente iniciando sesión para continuar utilizando los servicios de la aplicación.
- **Recover Token:** Este token se utiliza para verificar la identidad del usuario y que este pueda recuperar el acceso a su cuenta. Se obtiene cuando el usuario solicita a la API un cambio de contraseña, tras esto la aplicación envía este token junto con una URL al email asociado a la cuenta de usuario, antes de modificar la contraseña se comprobará que el token enviado es el mismo que el recibido. Este token tiene una duración de *30 minutos* ya que no es recomendable extender demasiado en el tiempo su vida útil y si que es almacenado en la base de datos.
- **Activate Token:** Este token se utiliza para activar la cuenta de usuario y verificar la dirección de correo electrónico. Se obtiene cuando el usuario se registra en el sistema, una vez registrado se envía este token junto con una URL al email introducido, y una vez que el usuario accede a este enlace la cuenta se activa en el sistema y puede empezar a utilizar la aplicación. Este token tiene una duración de *1 día* ya que puede que el usuario no tenga acceso a su email en ese momento y por tanto su vida útil debe ser algo mayor, además este token si que es almacenado en la base de datos.

6.2. Diseño de la API

Para el backend se ha desarrollado una API REST, utilizando el patrón MVC, la cual permite a los usuarios poder realizar peticiones sobre los datos de sus métricas de salud y cuestionarios. Entre las principales características del backend cabe destacar:

- Para acceder a la base de datos se ha hecho uso de JPA, Java Persistence API, la cual ofrece diversos tipos de consultas para realizar a la base de datos. En este proyecto se ha hecho uso de las native queries las cuales ofrecen al desarrollador la posibilidad de especificar el tipo de consulta que desea realizar utilizando código SQL, lo cual facilita enormemente el proceso de desarrollo y ofrece una gran variedad de posibilidades.
- A la hora de crear una nueva instancia en la base de datos se hace uso los generadores automáticos de spring boot los cuales a través de una serie de reglas creadas por el desarrollador permiten generar, en este proyecto, claves privadas siguiendo una determinada

secuencia. Para ello una vez se ejecuta la aplicación genera de forma automática secuencias SQL las cuales para cada nueva instancia asocia una clave primaria nueva y no repetida.

- El tipo de datos que devuelve la API son objetos de tipo DTO, los cuales son objetos utilizados para el transporte de información entre distintas capas a través de la capa de servicio.
- La creación de filtros de autenticación para el acceso a la API. A excepción del controlador de autenticación, salvo las peticiones de cierre de sesión que, sí requieren autenticación, el resto de peticiones de todos los controladores requieren de un proceso de verificación del usuario para permitir el acceso al recurso solicitado. Esta verificación se realiza comprobando el bearer token que envía el usuario en una de las cabeceras de la petición HTTP, comprobando su caducidad, integridad y que el resto de los campos por los que está compuesto sean válidos.
- El uso de cuatro controladores cuya función depende del tipo de recursos que son solicitados por el cliente en cada petición HTTP. Estos controladores y sus funciones son los siguientes:
 - El controlador de autenticación se encarga de gestionar la conexión, registro, recuperación de contraseña y activación de la cuenta en el sistema.
 - El controlador de cuestionarios se encarga de gestionar todo lo relacionado con la inclusión de un nuevo cuestionario o la recuperación de todos los realizados por un usuario.
 - El controlador de usuarios se encarga de gestionar todo lo relacionado con la recuperación o modificación de los datos personales del usuario.
 - El controlador de métricas de salud se encarga de gestionar todo lo relacionado con la recuperación de los datos ya incluidos en el sistema de las métricas de salud del usuario, la actualización del tipo de sistema de medida que desea utilizar el usuario, la modificación de los objetivos de las métricas de salud del usuario y la inclusión de nuevos valores para las distintas métricas de salud del usuario.
- Se ha añadido también una tarea periódica al sistema, la cual se ha configurado para que una vez al día, a las 0:00 con el objetivo de no entorpecer la usabilidad de los usuarios, el sistema sea capaz de eliminar aquellos tokens que se encuentran caducados en la base de datos. Esto se realiza recuperando todos aquellos tokens cuya fecha de expiración es posterior a la fecha actual, permitiendo de esta forma que tanto eliminar los tokens de sesiones de usuario que han permanecido mucho tiempo iniciadas o los tokens de cuentas que ya deberían haber sido activadas o los tokens de cuentas que ya deberían de haber modificado su contraseña.

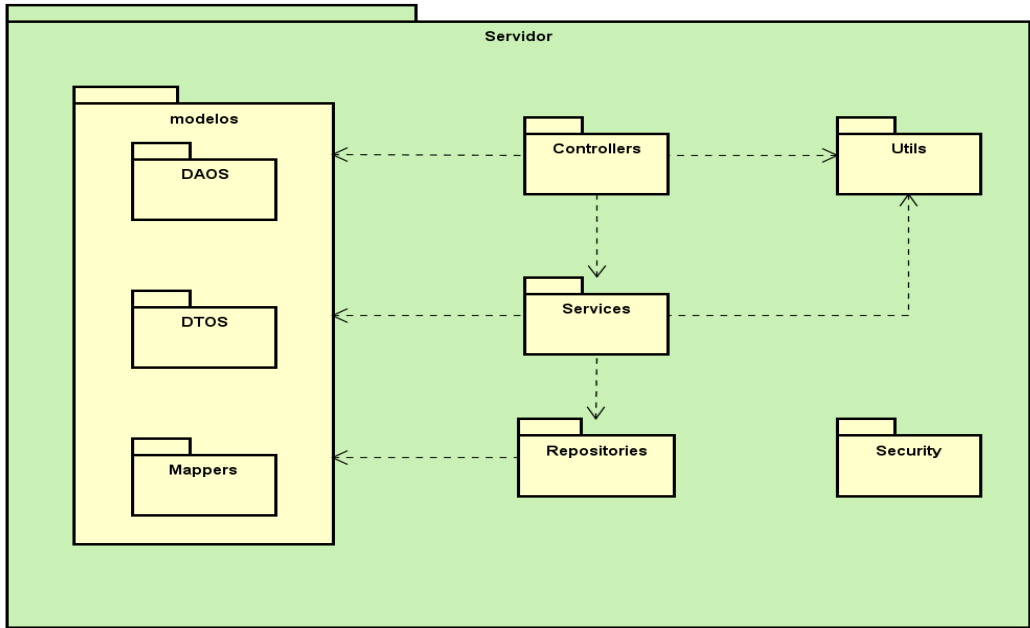


Figura 6.3: Diagrama de la estructura de la API.

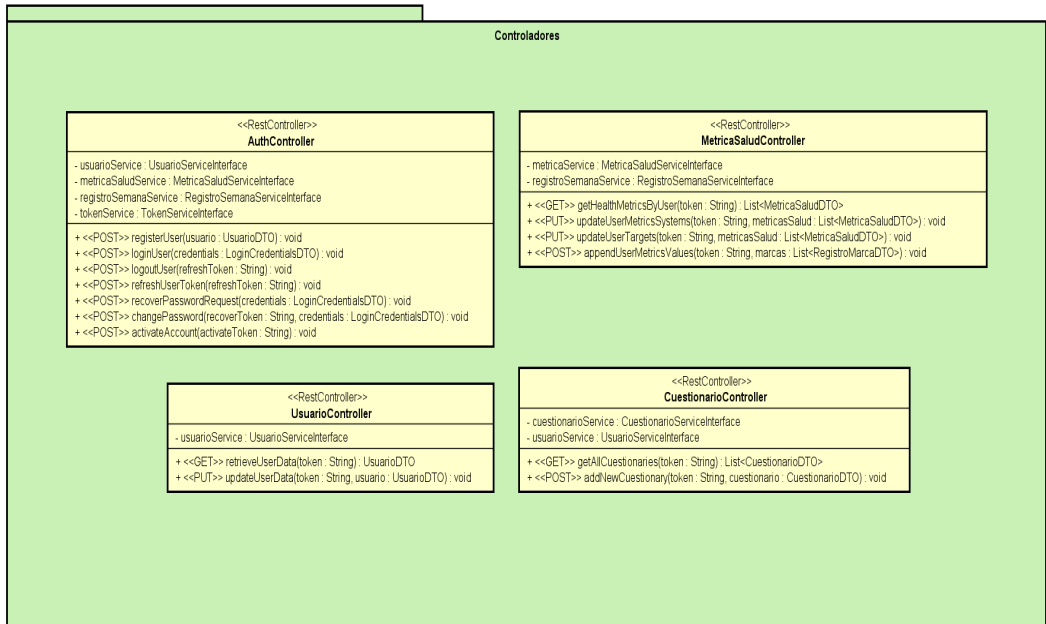


Figura 6.4: Diagrama de los controladores de la API.

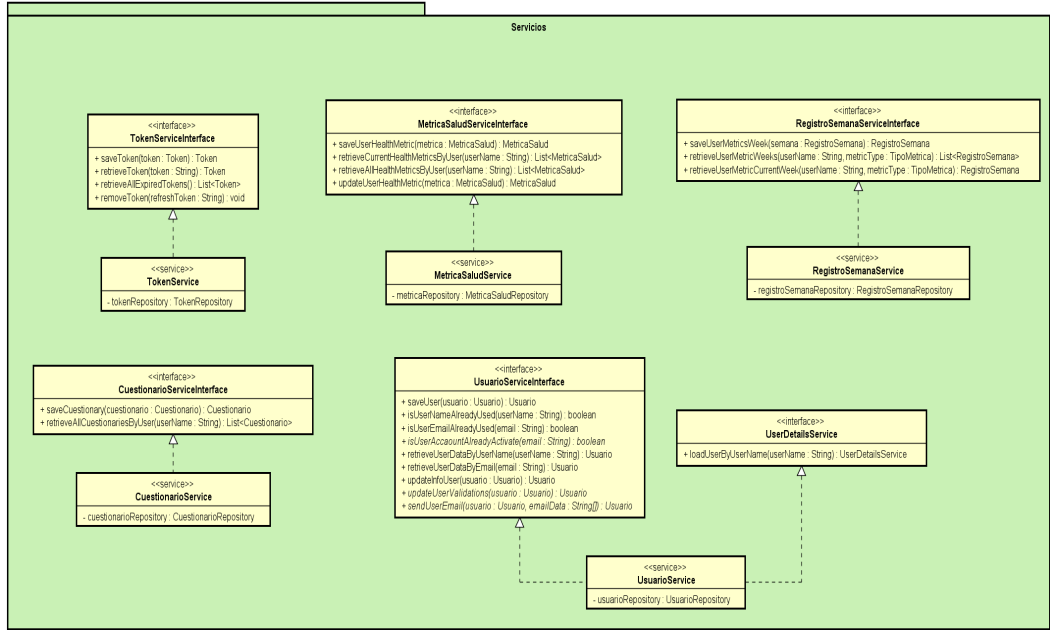


Figura 6.5: Diagrama de los servicios de la API.

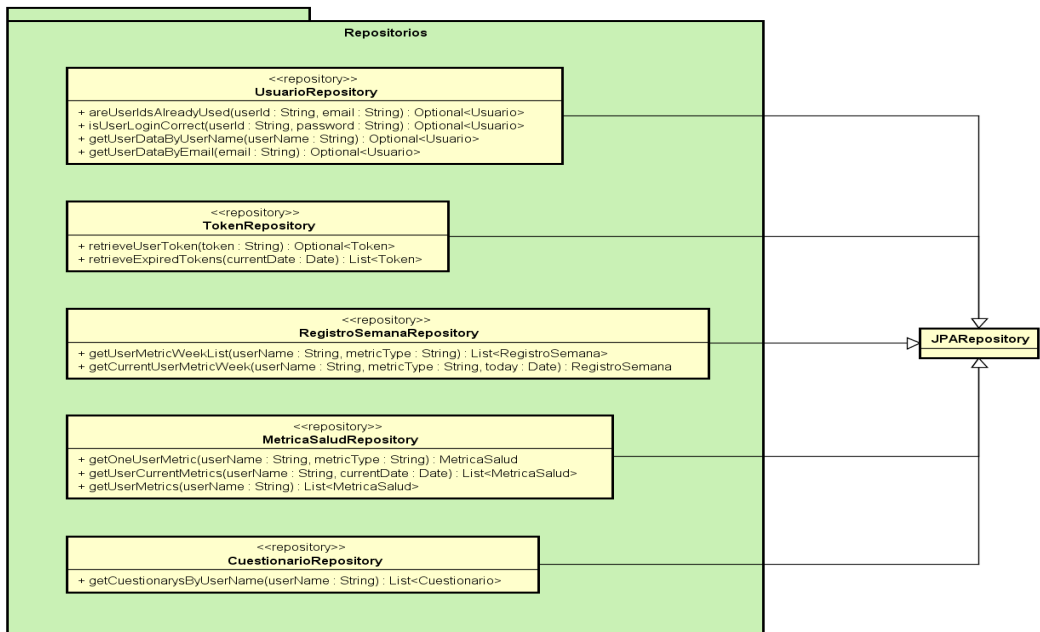


Figura 6.6: Diagrama de los repositorios de la API.

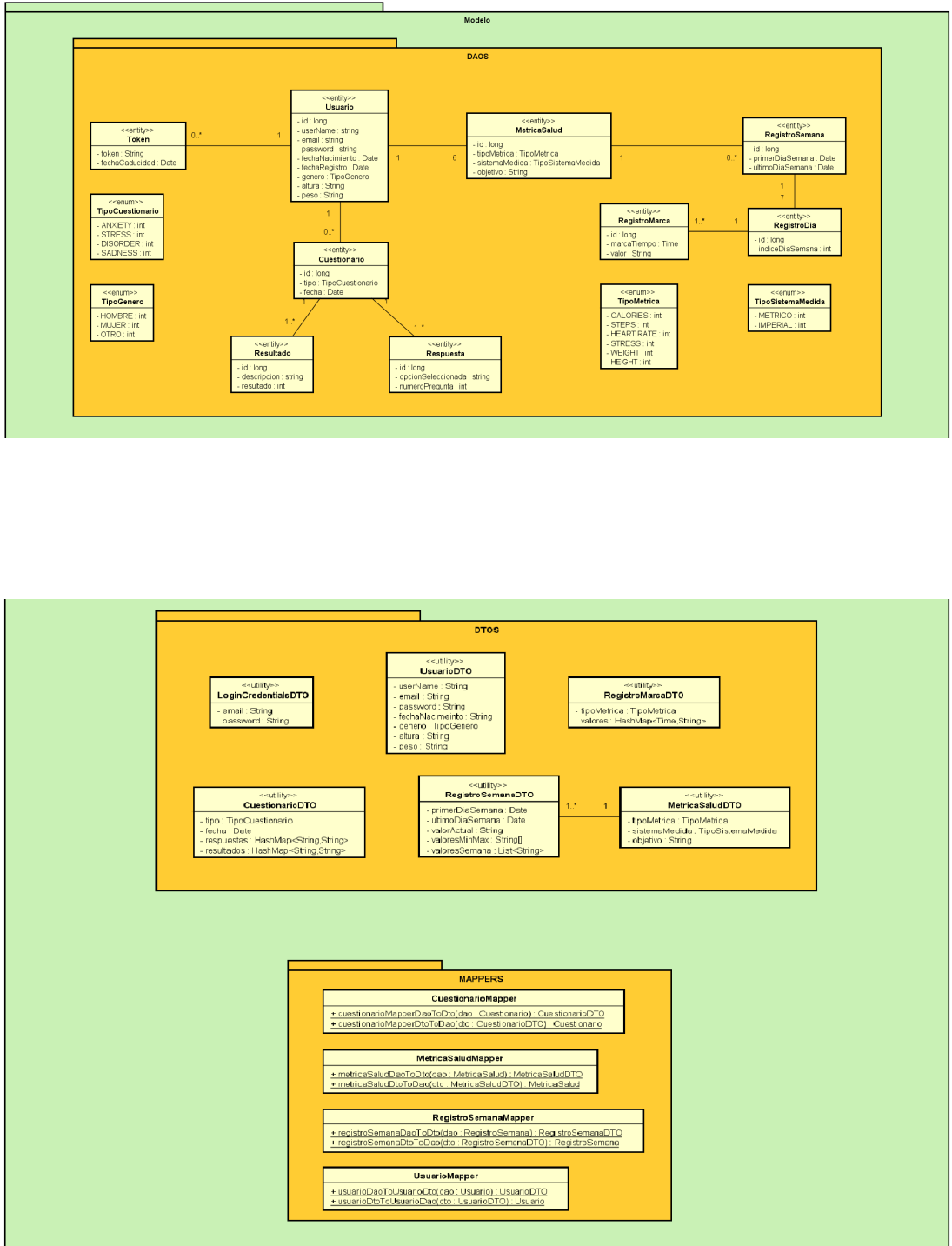


Figura 6.7: Diagrama de los modelos de la API.

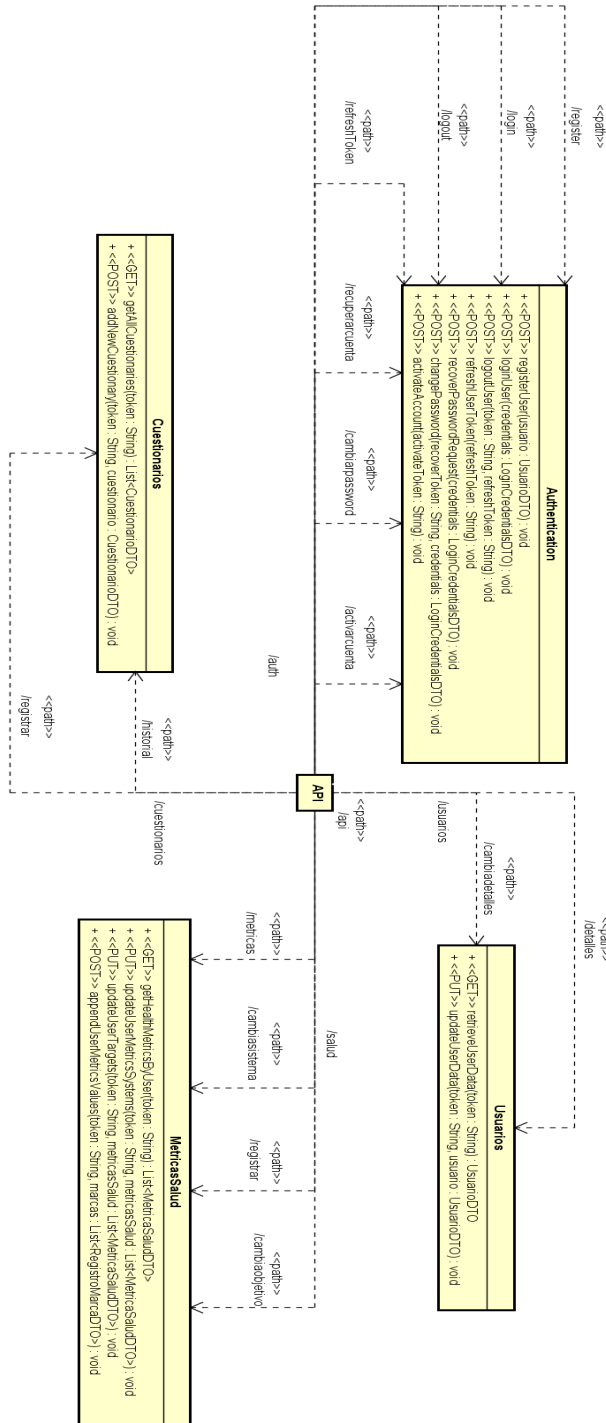


Figura 6.8: Diagrama de la estructura de peticiones de la API.

6.3. Diseño de la aplicación de usuario

En cuanto al desarrollo de la aplicación de usuario se han hecho uso de varios frontends de desarrollo de aplicaciones web, como React o Ionic. Ionic tiene la peculiaridad de que permite desarrollar aplicaciones híbridas utilizando código de aplicaciones web, en este caso de React, y generando con él aplicaciones nativas tanto en Android como en IOS. En cuanto a React, este framework permite al desarrollador construir sitios webs interactivos con los que el usuario pueda interactuar, poniendo especial atención a la eficiencia y velocidad de desarrollo del proyecto. Además se ha hecho uso en el parte de desarrollo de la herramienta Node.js para desplegar un servidor local.

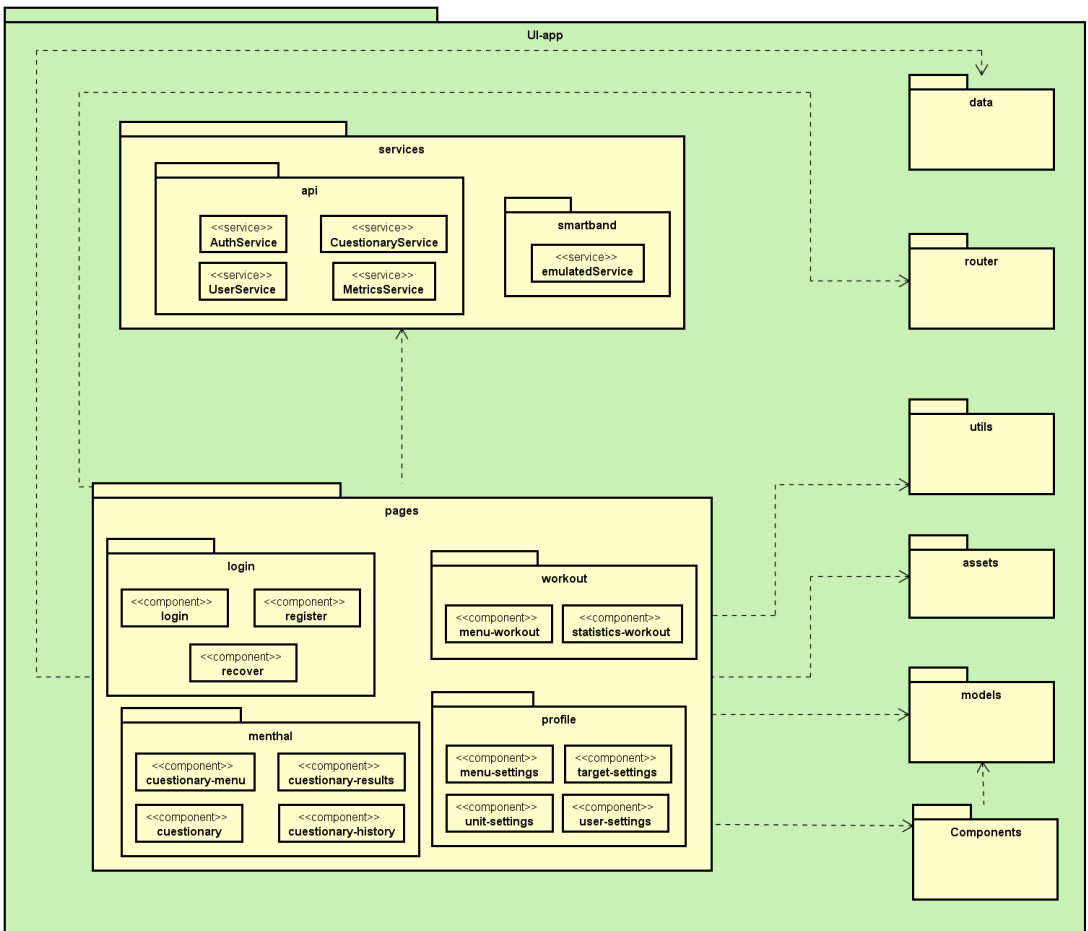


Figura 6.9: Diagrama de la estructura del frontend.

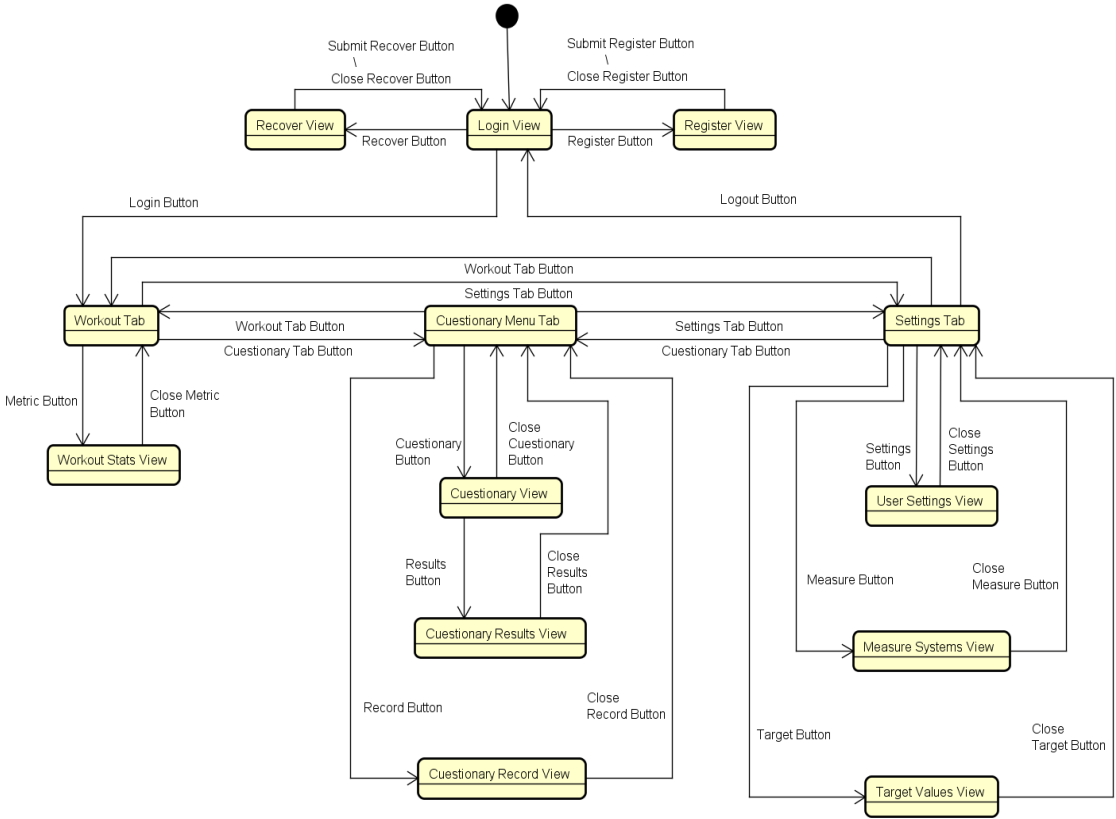


Figura 6.10: Plantilla de la vista de login.

6.4. Interfaz de usuario

Al tratarse de una aplicación móvil que tiene el objetivo de utilizarse en una gran variedad de dispositivos y por una gran variedad de usuarios, se ha puesto especial atención en la usabilidad realizando una interfaz sencilla y amigable. Además, para asegurar la consistencia y fiabilidad en cuanto a los tiempos de respuesta se han realizado numerosas modificaciones en la forma en la que se estructuraban o mostraban los datos en la aplicación, ya que las primeras aproximaciones utilizaban conjuntos de datos tan amplios que provocaban ralentizaciones en el uso y en algunos casos incluso congelaciones de la aplicación.

A continuación, se muestran tanto las primeras plantillas, realizados con la aplicación Balsamiq, como las vistas finales desarrolladas para la aplicación:

6.4.1. Plantillas de la interfaz de usuario

La siguiente figura, Figura 7.11, muestra la plantilla de login de la aplicación, en la que el usuario podrá iniciar su sesión.

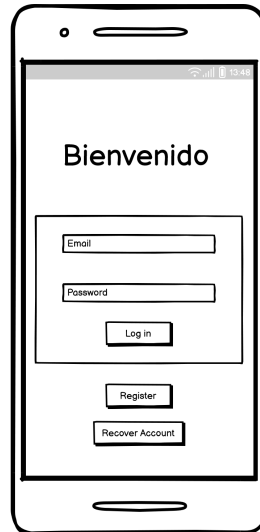


Figura 6.11: Plantilla de la vista de login.

La siguiente figura, Figura 7.12, muestra las plantillas de salud física de la aplicación, en la que el usuario monitoriza los datos relacionados con su salud física en las diferentes métricas de la aplicación.



Figura 6.12: Plantillas de las vistas de salud física.

La siguiente figura, Figura 7.13, muestra la plantilla del menú de los distintos cuestionarios de autoevaluación de la salud mental que el usuario puede realizar en la aplicación.

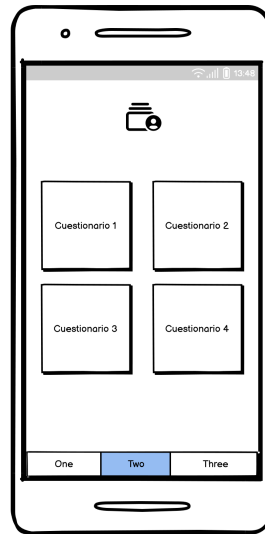


Figura 6.13: Plantilla de la vista del menú de cuestionarios.

La siguiente figura, Figura 7.14, muestra las plantillas del formulario de los cuestionarios de autoevaluación de la salud mental de la aplicación y de los resultados obtenidos al completarlos.



Figura 6.14: Plantillas de las vistas del formulario de cuestionarios.

La siguiente figura, Figura 7.15, muestra la plantilla del histórico de los cuestionarios de auto-evaluación de la salud mental que han sido desarrollados por el usuario en la aplicación, junto con una pequeña contextualización de cada uno.

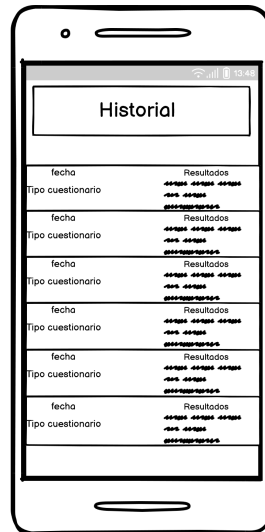


Figura 6.15: Plantilla de la vista del histórico de cuestionarios.

La siguiente figura, Figura 7.16, muestra la plantilla del menú de los ajustes de los datos de los usuarios en la aplicación.

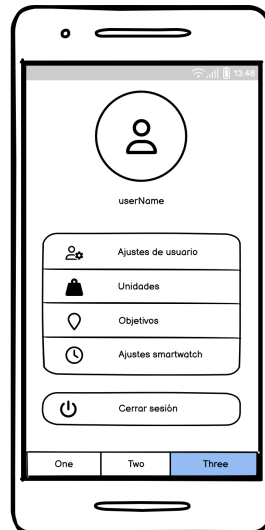


Figura 6.16: Plantilla de la vista de ajustes.

La siguiente figura, Figura 7.17, muestra las plantillas de los distintos menús de los ajustes de los datos que pueden ser modificados por parte de los usuarios en la aplicación.



Figura 6.17: Plantillas de las vistas de menús de ajustes.

La siguiente figura, Figura 7.18, muestra la plantilla del menú de registro con los campos que debe de completar un nuevo usuario para poder utilizar la aplicación.

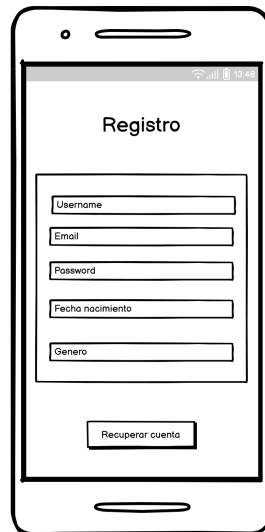


Figura 6.18: Plantilla de la vista de registro de usuario.

La siguiente figura, Figura 7.19, muestra las plantillas del menú de recuperación de la contraseña que deberá utilizar un usuario en caso de que no recuerde su contraseña actual en la aplicación.

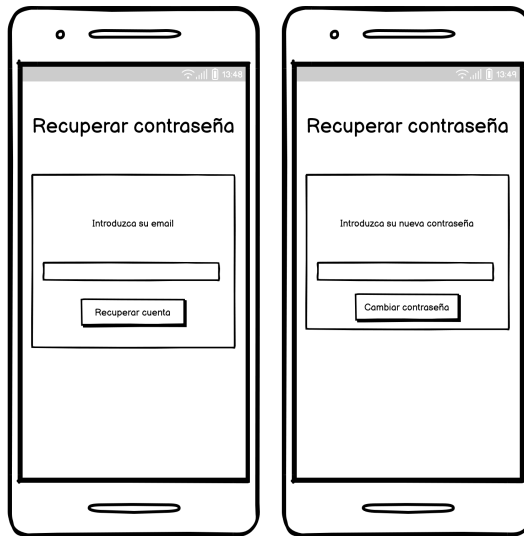


Figura 6.19: Plantillas de las vistas de recuperación de contraseña.

6.4.2. Interfaz de usuario

La siguiente figura, Figura 7.20, muestra el modelo de la vista de login en la que el usuario debe de introducir sus datos correctamente para acceder a la aplicación.

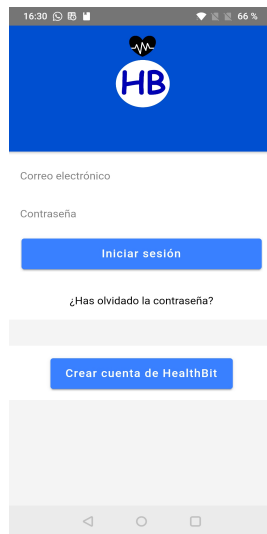


Figura 6.20: Vista del menú de login.

La siguiente figura, Figura 7.21, muestra el modelo de la vista del menú de salud física en la que el usuario podrá monitorizar los datos relacionados con su salud física en las distintas métricas que ofrece la aplicación.

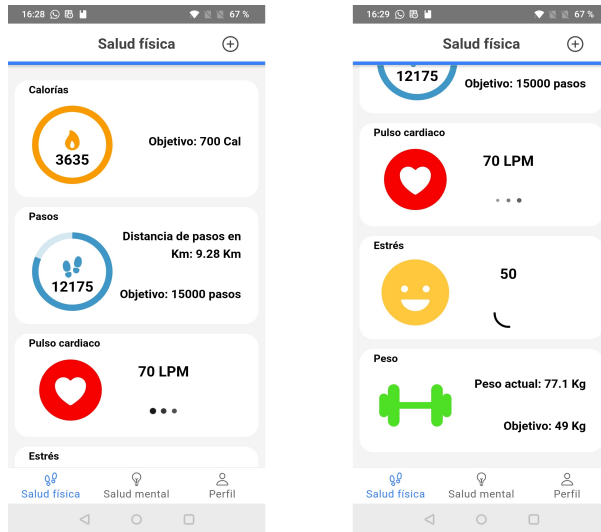


Figura 6.21: Vista del menú de salud física.

La siguiente figura, Figura 7.22, muestra el modelo de la vista del menú de estadísticas de salud física en la que el usuario podrá consultar un histórico de los datos de salud registrados, ciertos datos de interés y una pequeña descripción y explicación de cada métrica de salud consultada en la aplicación.



Figura 6.22: Vista del menú de estadísticas de salud física.

La siguiente figura, Figura 7.23, muestra el modelo de la vista de la leyenda del menú de estadísticas de salud física en la que el usuario podrá consultar, en caso de que la métrica disponga de una barra de rangos, una pequeña leyenda con la descripción de cada una de las secciones.



Figura 6.23: Vista de la leyenda del menú de estadísticas de salud física

La siguiente figura, Figura 7.24, muestra el modelo de la vista del menú de los distintos cuestionarios de autoevaluación de la salud mental que pueden ser realizados por el usuario haciendo

uso de la aplicación.

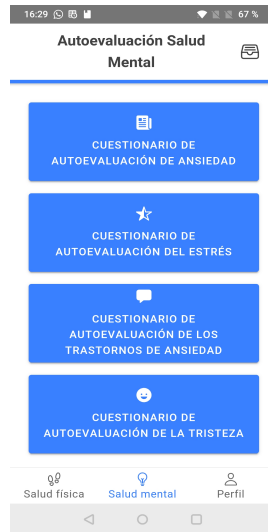


Figura 6.24: Vista del menú de cuestionarios.

La siguiente figura, Figura 7.25, muestra el modelo de las vistas del formulario de los cuestionarios de autoevaluación de la salud mental existentes en la aplicación y de los resultados obtenidos al completarlos.

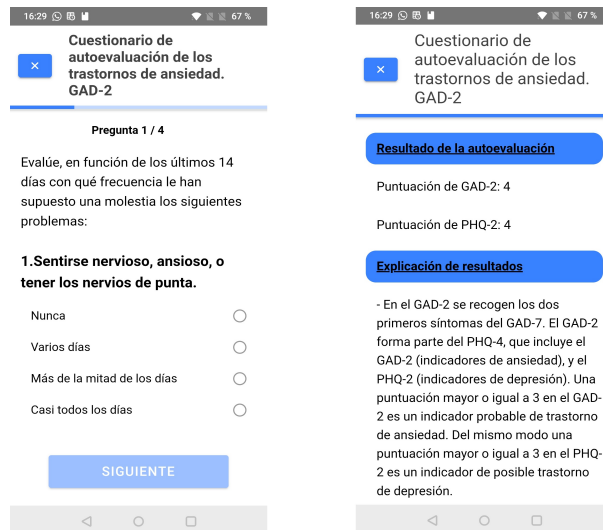


Figura 6.25: Vistas del formulario de cuestionarios.

La siguiente figura, Figura 7.26, muestra el modelo de la vista del histórico de los cuestionarios

de autoevaluación de la salud mental que han sido desarrollados por el usuario en la aplicación, además se ha incluido un pequeño pop up con una pequeña contextualización de cada uno de ellos.



Figura 6.26: Vista del historial de cuestionarios.

La siguiente figura, Figura 7.27, muestra el modelo de la vista del menú de los ajustes de los datos de los usuarios en la aplicación.

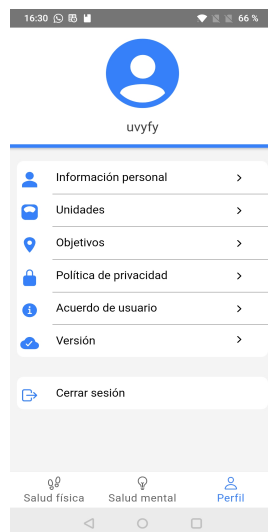


Figura 6.27: Vista del menú de ajustes.

La siguiente figura, Figura 7.28, muestra el modelo de las vistas de los distintos menús de los ajustes de los datos que pueden ser modificados por parte de los usuarios en la aplicación.

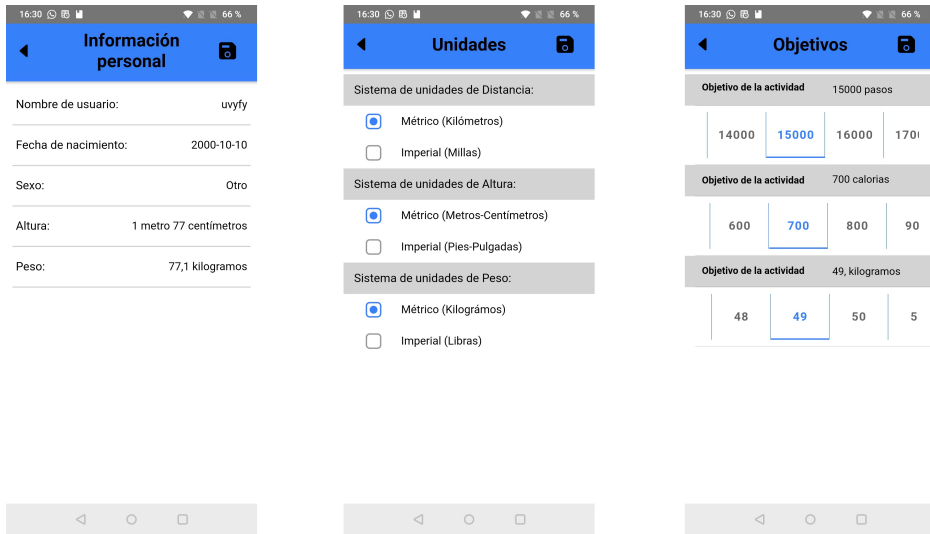


Figura 6.28: Vistas de los menús de ajustes.

La siguiente figura, Figura 7.29, muestra el modelo de la vista del menú de registro de un nuevo usuario en la aplicación, además incluye la imagen del correo electrónico recibido por el usuario al correo electrónico introducido para activar su cuenta en la aplicación.

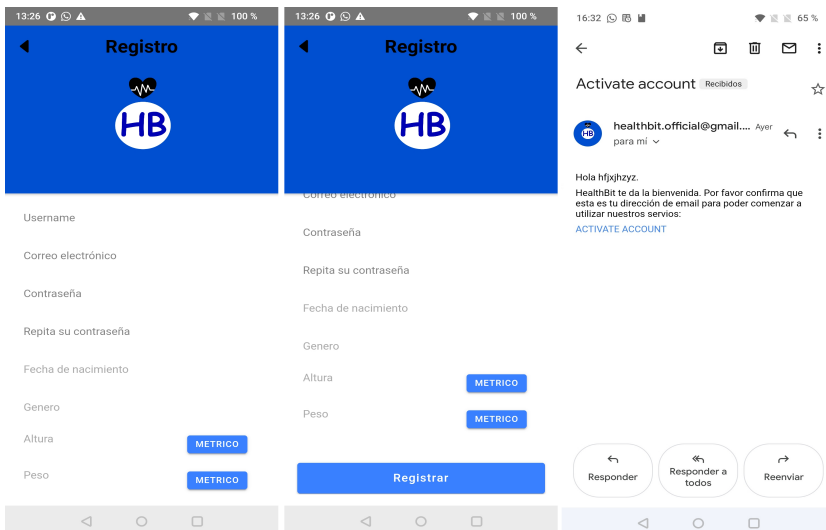


Figura 6.29: Vista del registro de usuarios.

La siguiente figura, Figura 7.30, muestra el modelo de las vistas del menú de recuperación de contraseñas que deberá utilizar un usuario en caso de que no recuerde su contraseña actual en la aplicación, además incluye la imagen del correo electrónico recibido por el usuario al correo electrónico introducido para verificar su identidad antes de poder modificar su contraseña.

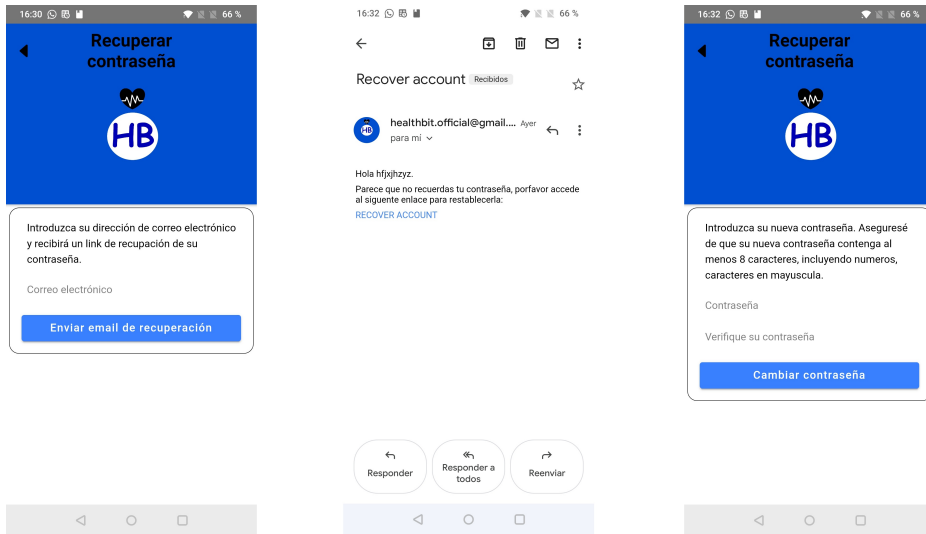


Figura 6.30: Vistas de la recuperación de contraseña.

Capítulo 7

Desarrollo del proyecto

En este apartado se procederá a describir el proceso de desarrollo del proyecto, profundizando en cómo se llevó a cabo y poniéndolo en contraste con la planificación inicial, hablando de los inconvenientes surgidos y de las pruebas realizadas en cada uno de los sprints.

7.1. Sprint 1

En este primer sprint se comenzó realizando la planificación del proyecto, recabando las historias de usuario y realizando el análisis de los posibles riesgos que podrían llegar a presentarse durante el transcurso del proyecto. Para ello se hizo uso de los conocimientos obtenidos en la asignatura de Planificación y Gestión de Proyectos realizada en el cuarto curso. Una vez realizados se procedió a realizar el product backlog del primer sprint (Tabla 5.1). En este sprint se ha decidido crear las bases de la aplicación y las historias referentes al login de usuario y la conexión con el *Smartwatch*, las cuales corresponden las historias de usuario 2 y 6 respectivamente.

Tras esto se procedió a comenzar el aprendizaje de cada una de las tecnologías que iban a ser necesarias para esta aplicación, ya que, aunque durante la carrera se han utilizado algunas de ellas, como SQL, Angular, Java..., hay otras de las que no se tenían conocimientos hasta ahora.

Cabe destacar el proceso de formación que se ha seguido para el framework Ionic, ya que este ha sido el que ha supuesto una mayor dificultad para el desarrollo y la finalización del proyecto. Ionic es un framework utilizado para el desarrollo de aplicaciones híbridas que utiliza código de aplicaciones web realizadas con otros frameworks como React, Angular o Vue. Aunque esta herramienta suele ser actualizada con bastante frecuencia añadiendo y modificando la documentación del mismo, esta ha resultado ser confusa, escasa y desigual. En primer lugar, aunque se promociona e integra su uso el uso de React, Ionic tiene una documentación que resulta escasa al trabajar con este framework. La mayoría de la documentación existente se encuentra desarrollada para Angular, es escasa en el caso React, y menor en Vue, y aunque en algunos casos pueden encontrarse en foros algún tipo de explicación o ayuda en diversas funcionalidades que la documentación no contiene existen otros que ni siquiera usando foros se pueden solventar.

Id	Id de la historia de usuario	Descripción	Estimación
1		Determinar la extensión del proyecto	4
2		Determinar las historias de usuario	4
3		Planificar el sprint	2
4		Formarse en Typescript	12
5		Formarse en React	12
6		Formarse en Ionic	12
7		Formarse en Spring	12
8		Configurar Visual Studio Code	2
9		Configurar proyecto Ionic	2
10	2	Realizar plantilla de la vista de login	2
11	2	Realizar vista de login	2
12	2	Identificar al usuario	4
13	2	Crear base de datos y tabla de usuarios	4
14	2	Crear en la API el servicio y el repositorio de usuarios	8
15	2	Crear el controlador de usuarios	8
16	6	Buscar información BLE (Bluetooth Low Energy)	12
17	6	Conectar SmartWatch con aplicación	8

Tabla 7.1: Product backlog Sprint 1

Una vez realizada la primera toma de contacto con estas herramientas se procedió a comenzar el desarrollo de la aplicación con las historias asignadas al primer sprint.

El resto de las historias de usuario y su desarrollo transcurrió según lo esperado. Aunque no se consiguieron terminar las tareas 16 y 17 del product Backlog, ya que no se encontró ninguna metodología clara para conectarse y recibir datos de un *Smartwatch*, ya que existen numerosas formas de comunicación entre ellas y no todas funcionan con el mismo tipo de tecnología, por lo que se procedió a centrar en foco de atención en aquellos *Smartwatch* de Xiaomi, las ya conocidas como MiBand en este caso se decidió trabajar con el modelo 6, ya que añadía un gran número de funcionalidades en contraste con los anteriores.

Buscando información sobre el Bluetooth se encontró en la página oficial del mismo que existe un estándar de códigos[34] en el que cada servicio tiene asignado un id concreto, y aunque no todos los *Smartwatch* disponibles en el mercado lo utilizan sí que lo hacen las MiBand. Por lo que se procedió a adquirir una MiBand de este modelo y a escanear los servicios que ofrecía para poder trabajar con ellos.

Al no poder completar estas tareas se ha procedido a añadirlas al product backlog del siguiente

sprint.

7.2. Sprint 2

Este sprint estará centrado en trabajar sobre las funcionalidades asociadas a la salud física y en completar la conexión con el *Smartwatch*, esto abarca las historias 7 y 8 y la historia 6 respectivamente. En este segundo sprint se comenzó realizando el product backlog del sprint (Tabla 5.2), añadiendo las tareas que no pudieron completarse durante el anterior sprint, y continuando el proceso de formación en las tecnologías utilizadas.

Id	Id de la historia de usuario	Descripción	Estimación
1		Planificar el sprint	2
2		Formarse en Ionic	4
3		Formarse en Typescript	4
4	6	Buscar información BLE (Bluetooth Low Energy)	12
5	6	Conectar SmartWatch con aplicación	20
6	7	Realizar plantilla de la vista de salud física	2
7	7	Realizar vista de salud física	2
8	7	Monitorizar los datos de las métricas de salud física	8
9	7	Crear tablas de salud física en la base de datos	12
10	7	Crear en la API los servicios y los repositorios de las métricas de salud física	12
11	7	Crear el controlador de las métricas de salud física	12
12	8	Realizar plantilla de estadísticas de salud física	2
13	8	Realizar vista de estadísticas de salud física	2
14	8	Añadir datos de interés en función de la estadística	8
15	8	Añadir rango de estrés para la métrica de estrés e IMC para la métrica de peso	4

Tabla 7.2: Product backlog Sprint 2

Se continuó buscando información que permitiese la conexión de dispositivos Bluetooth con aplicaciones Ionic. Se hizo uso de la herramienta nrfConnect, que actuaba como un sniffer de

dispositivos Bluetooth permitiendo conocer que servicios estaban expuestos y comunicarse con ellos (Figura 5.1). Buscando en numerosos foros se encontraron dos posibles implementaciones de este servicio:

- La primera era una aplicación llamada GadgetBridge[35] la cual es una aplicación de código abierto que permite conectar numerosos tipos de dispositivos *Smartwatch*. Se intentó realizar un proceso de ingeniería inversa para determinar como realizaban el proceso de conexión con dispositivos, pero el código era tan extenso y su licencia de uso no permitía la modificación del proyecto, por lo que no pudo llevarse a cabo.
- La segunda era un foro que contenía una descripción del proceso de autenticación del dispositivo *Smartwatch* con una aplicación. Este proceso consistía en utilizar la clave privada del dispositivo para realizar un proceso de autenticación en tres pasos[36], el cual consistía en modificar el atributo de notificaciones del servicio GATT (Generic Attribute Profile) con el id 0x0009. Para obtener esta clave privada se encontró un software llamado Huami-token[37] el cual consistía en realizar un man in the middle del proceso de autenticación del dispositivo, devolviendo así el token privado (Figura 5.2).

Tras probar y utilizar estas implementaciones, no se consiguió realizar el proceso de autenticación y vinculación de este, por lo que esta funcionalidad será implementada en una futura actualización de la aplicación.

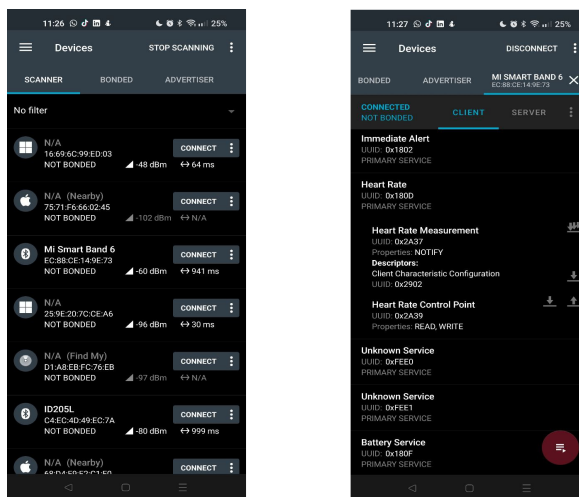


Figura 7.1: Vista de los dispositivos y servicios de una Mi Band 6 obtenidos en nrfConnect.

```

pedro@pedro-VirtualBox:~/huami-token-master(1)/huami-token-master$ python3 huami_token.py --method xiaomi --bt_keys
Getting access token with xiaomi login method...
Copy this URL to web-browser

https://account.xiaomi.com/oauth2/authorize?skip_confirm=false&client_id=600359586377620400&gt;0&scope=1+5000+13001+21000&redirect_uri=https%3A%2F%2Fm.xiaomi.com%2Fwatch.do%3Flocale=en_US&response_type=code

and login to your Mi account.

Paste URL after redirection here.
https://m.xiaomi.com/watch.do?code=A2AM5SRV_5CC7806F4BFF6DAEFCF88C49C8484BD1
Token: ['A2AM5SRV_5CC7806F4BFF6DAEFCF88C49C8484BD1']
Logging in...
Logged in! User id: 8178237079
Getting linked wearables...
-----
| ID | ACT | MAC | auth_key |
|-----|-----|-----|-----|
| 0 | 1 | D1:A5:2C:BD:58:4E | 0xe7bf49270254113d2fb7a3d4a86088cd |
-----
Logged out.
    
```

Figura 7.2: Vista del proceso de obtención de un token privado con el software Huami-token.

En cuanto al resto de tareas enumeradas en el product backlog de este sprint, transcurrieron sin casi ningún problema. Cabe destacar la implementación del diagrama de barras para la vista de estadísticas de salud física, la cual generó algún problema ya que al generarlas se crea un lienzo, el cual si no se cierra o elimina correctamente puede llegar a generar errores al abrir otra métrica de salud.

7.3. Sprint 3

En este sprint se trabajará sobre las funcionalidades relacionadas con los cuestionarios, desde su realización hasta el histórico de aquellos que han sido realizados, esto abarca las historias de usuario 9, 10 y 11. Este sprint se comenzó realizando el product backlog del sprint (Tabla 5.3).

En el desarrollo de este sprint se presentaron dos problemas que ocuparon gran parte del tiempo disponible.

El primer problema que surgió fue la búsqueda de cuestionarios que pudiesen implementarse en la aplicación ya que la mayoría de ellos se encontraban en otras aplicaciones móviles o webs, las cuales no permitían su uso a desarrolladores terceros.

Esto fue solventado al encontrar unos cuestionarios sobre salud mental publicados por el ministerio de sanidad, los cuales tenían una licencia de libre uso a terceros con la única condición de citar la fuente (tal y como se ha realizado en la documentación de este documento [5]). Estos adjuntaban una pequeña descripción de cada cuestionario y el resultado de la evaluación. Tras esto se procedió a determinar como se evaluaban los cuestionarios, al no encontrar dicho proceso en el código fuente, se procedió a realizar ingeniería inversa de los mismos utilizando el proceso de fuerza bruta para probar todas las posibles combinaciones de cada pregunta.

El siguiente problema que surgió fue durante el desarrollo de los repositorios de la API, ya que al hacer varias pruebas con Postman se descubrió que las consultas SQL no estaban realizándose correctamente.

Este problema resultó ser muy confuso ya que aunque utilizando la consola de Postgres (PSQL) y ejecutando los comandos consultas SQL si que funcionaban, al trasladarlas a java Spring uti-

lizaba datos que no se correspondían con lo obtenido en la consola. Este error resultó deberse a que al utilizar la operación SQL natural join en Java Spring, no distinguía las columnas según las tablas sino por sus nombres, y al haber utilizado el mismo nombre en varios campos de distintas tablas SQL mezclaba los datos y realizaba correctamente la consulta. Para solucionar este error, se procedió a reestructurar y modificar las tablas SQL que presentaban este problema.

El resto de las tareas planificadas en el product backlog de este sprint se desarrollaron de acuerdo a lo previsto. Aunque al haberse presentado estos problemas, no se pudo terminar de realizar la funcionalidad del historial de cuestionarios, por lo que esta historia de usuario será añadida al siguiente sprint.

Id	Id de la historia de usuario	Descripción	Estimación
1		Planificar el sprint	2
2		Formarse en Ionic	4
3		Formarse en Typescript	4
4	9	Realizar plantilla del menú de cuestionarios	2
5	9	Realizar vista del menú de cuestionarios	2
6	9	Búsqueda de cuestionarios sobre salud mental	12
7	9	Realizar plantilla del formulario de cuestionarios	2
8	9	Realizar vista del formulario de cuestionarios	2
9	9	Implementar cuestionario	4
10	10	Realizar plantilla de la vista de resultados de cuestionarios	2
11	10	Realizar vista de resultados de cuestionarios	2
12	10	Crear tablas de cuestionarios en la base de datos	8
13	10	Crear en la API los servicios y los repositorios de los cuestionarios	8
14	10	Crear el controlador de los cuestionarios	8
15	11	Realizar plantilla del historial de cuestionarios	2
16	11	Realizar vista del historial de cuestionarios	2
17	11	Implementar historial de cuestionarios	12

Tabla 7.3: Product backlog Sprint 3

7.4. Sprint 4

En este cuarto sprint se trabajará en las funcionalidades relacionadas con los ajustes de usuarios, las cuales se corresponden con las historias de usuario 12, 13 y 14. Además se ha añadido la tarea de implementación del histórico de cuestionarios realizados por el usuario, correspondiente a la historia 11, la cual no pudo ser finalizada en el sprint anterior por falta de tiempo en el mismo. El sprint se comenzó realizando el product backlog del sprint (Tabla 5.4).

En cuanto a la funcionalidad que no puedo completarse en el sprint anterior, relacionada con el histórico de cuestionarios realizados se finalizó sin ningún tipo de inconveniente. En el resto de tareas asociadas a los ajustes de la aplicación se encontraron con varias dificultades no previstas, todas ellas asociadas al despliegue de la aplicación en Android.

Id	Id de la historia de usuario	Descripción	Estimación
1		Planificar el sprint	2
2		Formarse en Ionic	4
3		Formarse en Typescript	4
4	11	Implementar historial de cuestionarios	4
5	12	Realizar plantilla de la vista de ajustes	2
6	12	Realizar vista de ajustes	2
7	12	Realizar plantilla de la vista de ajustes de usuario	2
8	12	Realizar vista de ajustes de usuario	2
9	12	Implementar modificación de los datos del usuario	8
10	13	Realizar plantilla de la vista de cambio de sistema de medida	2
11	13	Realizar vista de cambio de sistema de medida	2
12	13	Implementar modificación del sistema de medida	8
13	14	Realizar plantilla de la vista de cambio de objetivos de las métricas de salud	2
14	14	Realizar vista de cambio de objetivos de las métricas de salud	2
15	14	Implementar modificación de los objetivos de las métricas de salud	8

Tabla 7.4: Product backlog Sprint 4

La primera de ellas fue con el cambio de sistema de medida ya que, aunque en la persistencia y en el servidor se había previsto este cambio de sistemas de unidades y se habían añadido campos en las tablas y clases correspondientes para almacenar el tipo de sistema que deseaba utilizar el usuario, los problemas surgieron en la aplicación del usuario. Estos problemas se daban a la hora de convertir los datos entre los distintos sistemas y en generar los posibles valores de cada campo, ya que esto generaba ralentizaciones en el uso de la aplicación, por lo que se tuvieron que descartar varias opciones de valores y cambiar varias veces el tipo de sistema de generación de los mismos.

La segunda de ellas también se debe al despliegue de la aplicación en Android ya que en las vistas de usuario a la hora de modificar algún dato, las estructuras que se utilizaban ocupaban demasiada memoria del sistema y provocaban que se ralentizara la aplicación y en algunos casos la cantidad de datos que tenía que cargar era tan grande que provocaba un *crasheo* en la aplicación.

Este problema no pudo solventarse en este sprint por falta de tiempo ya que la mayoría del tiempo se dedicó al primer problema de generación y conversión de valores en otros sistemas de medida.

7.5. Sprint 5

En este sprint se trabajará sobre las funcionalidades referentes al registro y al cierre de sesión de la aplicación, referentes a las historias de usuario 1 y 3. Aunque en un primer momento se tenía pensado añadir más tareas a este sprint, al tener que implementar el apartado de seguridad se redujo la cantidad de estas. Además, se han incorporado las historias referentes a los ajustes de usuario que no pudieron completarse en el sprint anterior, referente a las historias de usuario 12 y 14. Este sprint, se comenzó realizando el *product backlog* de este sprint (Tabla 5.5).

En este sprint se consiguieron terminar las historias iniciadas en el primer sprint en poco tiempo, lo que permitió empezar pronto con las nuevas historias de este sprint.

En esta aplicación se ha hecho uso de JSON Web Tokens (JWT) para validar y autorizar las solicitudes de los usuarios en el lado del servidor, de tal forma que cuando un usuario realiza una petición a la API debe de enviar su token de autorización. Como este token no es recomendable que sea muy longevo, se implementó también un token de refresco el cual permite al usuario cada 5 min, poder renovar su token y de esta forma continuar su uso en la aplicación.

Id	Id de la historia de usuario	Descripción	Estimación
1		Planificar el sprint	2
2		Formarse en Ionic	4
3		Formarse en Spring	4
4	12	Implementar modificación de los datos del usuario	8
5	14	Implementar modificación de los objetivos de las métricas de salud	8
6	1	Realizar plantilla de la vista de registro de usuario	2
7	1	Realizar vista de registro de usuario	2
8	1	Implementar funcionalidad de registro de usuario	4
9	3	Crear tabla de tokens en la base datos	4
10	3	Crear repositorio y servicios de tokens	8
11	3	Crear controlador de autenticación	8
12		Configurar CORS del servidor	8
13		Gestionar la validez de los tokens	12
14		Implementar protocolo Https	4

Tabla 7.5: Product backlog Sprint 5

Cuando se empezó a realizar la funcionalidad de cierre de sesión se encontró un problema y es que no se este tipo de tokens al no tener estado no es posible de negarles el acceso e invalidar su uso. Este problema podía ser resuelto utilizando cookies o sesiones de usuario, las cuales fuesen almacenadas en el lado del servidor y de esta forma se pudiese almacenar el estado actual de la sesión del usuario, aunque esto supondría tener que rehacer la parte entera de peticiones a la API, tanto del lado del cliente como del servidor, por lo que decidió utilizarse otro enfoque.

Finalmente se decidió a crear una tabla que almacena tanto el token de refresco del usuario como la fecha en la que caduca, de tal forma que cuando un cliente desea cerrar su sesión en la aplicación se le solicita que envíe el token de refresco que está utilizando, se comprueba si se encuentra en la base de datos y en el caso de que exista lo elimina, permitiendo así que una vez el token de autorización caduque este no pueda ser renovado, ya que el token de refresco dejará de ser válido en la aplicación. Además, se ha implementado una tarea periódica diaria, la cual compruebe las fechas de todos los tokens almacenados y en caso de que exista alguno caducado, sea eliminado de forma automática.

En cuanto a las tareas relacionadas con la seguridad surgió un problema durante el desarrollo. Aunque la utilización de https en la API funcionaba correctamente en el navegador web durante la fase de pruebas, no pudo implementarse en la aplicación de Android. Por lo que esta tarea no pudo llegar a completarse en este sprint por falta de tiempo, por lo que esta tarea se incluyó en

el siguiente sprint.

7.6. Sprint 6

En este último sprint se tratarán de completar las historias de usuario restantes, estas son las referentes a la integración del correo electrónico en procesos de verificación del usuario, correspondientes a las historias 1, 4 y 5, además de las tareas que no pudieron completarse en el sprint anterior, como la implementación de conexiones seguras en la aplicación. Este sprint se comenzó realizando el product backlog, con las tareas mencionadas anteriormente (Tabla 5.6).

En cuanto a la implementación del protocolo Https no pudo llevarse a cabo debido a que al utilizar la aplicación, Android consideraba que la API, al estar firmada con un certificado propio y no estar validado por una entidad reconocida, era un sitio web peligroso y bloqueaba su uso al usuario. Para ello se intentó crear una whitelist de sitios a los que se tenía permitido el acceso, incluyendo el certificado de la API, pero no pudo conseguirse que Android realizase una solicitud al servidor de forma satisfactoria. Por este motivo se decidió finalmente a utilizar el protocolo Http e implementar el uso de su versión segura en una futura actualización.

Por otro lado, en lo referente a la implementación del uso del correo electrónico para verificar la identidad del usuario, se decidió ampliar la generalización de la clase token ya utilizada en los tokens de autorización y refresco, de tal forma que no hubiese que modificar demasiado la estructura de la misma.

Id	Id de la historia de usuario	Descripción	Estimación
1		Planificar el sprint	2
2		Formarse en el uso de Gmail con Spring	8
3		Implementar protocolo Https	12
4		Modificar y extender el uso de los tokens	12
5	1 y 4	Implementar verificación de email en el registro de un usuario en la API	4
6	5	Realizar la plantilla de la vista de recuperación de contraseña	2
7	5	Realizar la vista de recuperación de contraseña	2
8	4 y 5	Implementar la recuperación de contraseña	8
9		Buscar y reparar bugs	8
10		Limpiar y revisar el código de la aplicación	8

Tabla 7.6: Product backlog Sprint 6

Capítulo 8

Pruebas

Las pruebas o el proceso de testing en el desarrollo de software consiste en una etapa o proceso de evaluación y verificación de un componente o funcionalidad desarrollada en una aplicación software para determinar si su funcionamiento es el esperado. Al haber utilizado la metodología de desarrollo ágil SCRUM, antes de terminar y dar por finalizada cada historia de usuario se ha realizado un proceso de pruebas que permitiera garantizar la correcta implementación de la funcionalidad correspondiente a la historia usuario implementada.

En el contexto de testing en el desarrollo de software existe una gran variedad de pruebas que pueden realizarse en una aplicación convencional[39]. En este proyecto se han llevado a cabo las siguientes:

- **Pruebas unitarias.** Las pruebas unitarias son unas pruebas de bajo nivel que consisten en comprobar el funcionamiento de métodos y funciones utilizadas en clases, componentes o módulos de la aplicación. Una de sus mayores ventajas es la velocidad de ejecución que presentan.
- **Pruebas de integración.** Las pruebas de integración verifican que exista una correcta comunicación e interacción con distintos módulos o servicios utilizados en una aplicación.
- **Pruebas funcionales.** Las pruebas funcionales tienen el objetivo de verificar que los requisitos empresariales planteados en una aplicación sean alcanzados. Para ello únicamente comprueba el resultado de una determinada interacción obviando el resto de pasos intermedios.
- **Pruebas de extremo a extremo.** Las pruebas de extremo a extremo, también conocidas como pruebas integrales permiten al desarrollador verificar el funcionamiento del sistema intentando replicar el comportamiento de un usuario al utilizar el software desarrollado. Se incluye también la comprobación de flujos alternativos y excepciones que pueden no haber sido previstas durante el desarrollo.

A continuación se presentan una pequeña lista de las pruebas funcionales realizadas para cada uno de los requisitos funcionales completados que fueron planteados en el proceso de análisis.

Tipo de Prueba	Registro en la aplicación	
Dependencias		
Precondición	La dirección de correo electrónico y el nombre de usuario no deben estar ya registrados en el sistema.	
Descripción	El sistema permitirá al usuario registrarse en la aplicación completando todos los datos referentes a su perfil personal (nombre de usuario, correo electrónico, contraseña, fecha de nacimiento, género, altura y peso).	
Secuencia normal	<ul style="list-style-type: none"> •El usuario introduce los datos de su nueva cuenta en la aplicación. •El sistema verifica que tanto el correo electrónico como el nombre de usuario introducidos se encuentran disponibles. •El sistema comprueba la validez del resto de datos y envía un correo de verificación al nuevo usuario. •El usuario accede al enlace recibido en el correo de verificación. •El sistema verifica la identidad del individuo y activa su cuenta de usuario. 	
Postcondición	El usuario se encuentra registrado en el sistema.	
Excepciones	Caso	Acción
	El formato de los datos no es el correcto.	Mostrar al usuario mensaje de error indicando que campo no tiene el formato adecuado.
	Los identificadores introducidos por el usuario no se encuentran disponibles.	Mostrar al usuario un mensaje de error indicando que la dirección de correo electrónico o el nombre de usuario introducido no se encuentra disponible.
	El usuario no verifica su dirección de correo electrónico accediendo al enlace suministrado	Una vez caducado el token de activación, en un periodo de un día, tanto el token de activación como la información del usuario serán eliminados del sistema.

Tabla 8.1: Prueba registro

Tipo de Prueba	Recuperar contraseña olvidada	
Dependencias	<ul style="list-style-type: none"> • Registro en la aplicación. 	
Precondición	El usuario debe de estar registrado en la aplicación con su cuenta activada.	
Descripción	El sistema permitirá al usuario recuperar su contraseña teniendo que utilizar la dirección de correo electrónico asociada a su cuenta con el objetivo de verificar su identidad del mismo. Para ello se enviará a esta dirección un token de recuperación que permitirá al usuario editar su contraseña satisfactoriamente.	
Secuencia normal	<ul style="list-style-type: none"> • El usuario introduce la dirección de correo electrónico asociada a su cuenta y solicita recuperar su contraseña. • El sistema comprobará que la dirección de correo electrónico introducida corresponde a una cuenta existente en la aplicación. • El sistema enviará un email junto con un enlace, el cual contiene un token de recuperación, a la dirección de correo electrónico introducida por el usuario. • El usuario accederá a este enlace e introducirá su nueva contraseña. • El sistema verificará la identidad del usuario con los tokens suministrados. • El sistema comprobará que la contraseña introducida por el usuario cumple el formato establecido. • El sistema actualiza la nueva contraseña introducida por el usuario. 	
Postcondición	La contraseña asociada a la cuenta del usuario ha sido modificada.	
Excepciones	Caso	Acción
	La dirección de correo electrónico introducido no existe en el sistema.	Mostrar al usuario un mensaje de error indicando que la dirección de correo electrónico introducida no existe en el sistema.
	El formato de la nueva contraseña no es válido.	Mostrar al usuario un mensaje de error indicando que la nueva contraseña introducida no es válida y solicitarle que introduzca otra utilizando el formato requerido.

	El usuario no responde accede al enlace adjunto al correo electrónico de recuperación.	Una vez caducado el token de recuperación, en un periodo de 30 minutos, el enlace carecerá de validez, el usuario no podrá modificar su contraseña utilizando ese enlace y transcurrido un día el enlace caducado será eliminado del sistema.
--	--	---

Tabla 8.2: Prueba recuperación de contraseña

Tipo de Prueba	Inicio de sesión en la aplicación	
Dependencias	<ul style="list-style-type: none"> • Registro en la aplicación. 	
Precondición	El usuario debe de estar registrado en la aplicación con su cuenta activada.	
Descripción	El sistema permitirá al usuario iniciar sesión en la aplicación utilizando para ello la dirección de correo electrónico y la contraseña asociada con su cuenta.	
Secuencia normal	<ul style="list-style-type: none"> • El usuario introduce su dirección de correo electrónico y su contraseña. • El sistema comprueba que la información es correcta y permite al usuario iniciar sesión. 	
Postcondición	El usuario se encuentra con una sesión activa en la aplicación.	
Excepciones	Caso	Acción
	El formato de los datos no es el correcto.	Mostrar al usuario mensaje de error indicando que campo no tiene el formato adecuado.
	Los datos introducidos no corresponden a ningún usuario del sistema.	Mostrar al usuario un mensaje de error indicando que la dirección de correo electrónico o contraseña introducida no son válidos.

Tabla 8.3: Prueba inicio sesión

Tipo de Prueba	Cerrar sesión en la aplicación	
Dependencias	<ul style="list-style-type: none"> • Registro en la aplicación. • Sesión iniciada. 	
Precondición	El usuario debe tener una sesión iniciada en el sistema.	
Descripción	El sistema permitirá al usuario cerrar su sesión en la aplicación eliminando todos los tokens y datos almacenados de forma local pertenecientes a la sesión actual.	
Secuencia normal	<ul style="list-style-type: none"> • El usuario selecciona la opción de cerrar sesión. • El sistema verificará la identidad del usuario con los tokens suministrados. • El sistema eliminará los tokens almacenados en el sistema. • El sistema eliminará los datos almacenados de forma local referentes a la sesión actual, tras lo cual cerrará la sesión del usuario. 	
Postcondición	La sesión actual del usuario estará cerrada.	
Excepciones	Caso	Acción
	El formato de los datos no es el correcto.	Mostrar al usuario mensaje de error indicando que campo no tiene el formato adecuado.
	Los tokens suministrados no son válidos, por lo que no puede verificarse la identidad del usuario.	Bloquear la petición del usuario y no le permitirle cerrar la sesión referida. En caso de que se encuentre caducado expedir un nuevo token de verificación.

Tabla 8.4: Prueba cerrar sesión

Tipo de Prueba	Visualizar datos de las métricas de salud	
Dependencias	<ul style="list-style-type: none"> • Registro en la aplicación. • Sesión iniciada 	
Precondición	El usuario debe tener una sesión iniciada en el sistema.	
Descripción	El sistema permitirá al usuario visualizar información sobre los datos de sus métricas de salud que ofrece la aplicación.	
Secuencia normal	<ul style="list-style-type: none"> • El usuario decidirá consultar los datos relacionados con una de las métricas de salud disponibles en la aplicación. • El sistema verificará la identidad del usuario con los tokens suministrados. • El sistema mostrará al usuario los datos relacionados con la métrica de salud solicitada por el usuario junto con un conjunto de datos que contextualicen su actividad en el tiempo. 	
Postcondición	El usuario a podido visualizar los datos relacionados con su actividad en la métrica de salud solicitada.	
Excepciones	Caso	Acción
	La dirección de correo electrónico introducido no existe en el sistema.	Mostrar al usuario un mensaje de error indicando que la dirección de correo electrónico introducida no existe en el sistema.
	Los tokens suministrados no son válidos, por lo que no puede verificarse la identidad del usuario.	Bloquear la petición del usuario y no le permitirle cerrar la sesión referida. En caso de que se encuentre caducado expedir un nuevo token de verificación y volver a realizar la petición.
	El usuario no ha registrado ninguna marca la semana actual en la métrica física seleccionada.	El sistema mostrará al usuario que la semana actual no ha registrado ningún tipo de actividad física.

Tabla 8.5: Prueba visualización datos de métricas de salud

Tipo de Prueba	Realizar cuestionario salud mental	
Dependencias	<ul style="list-style-type: none"> • Registro en la aplicación. • Sesión iniciada 	
Precondición	El usuario debe tener una sesión iniciada en el sistema.	
Descripción	El sistema permitirá al usuario realizar diversos cuestionarios sobre la autoevaluación de su salud mental.	
Secuencia normal	<ul style="list-style-type: none"> • El usuario decidirá realizar un cuestionario sobre su salud mental. • El sistema solicitará al usuario que seleccione el tipo de cuestionario que desea realizar. • El usuario seleccionará el tipo de cuestionario que desea realizar. • El sistema mostrará al usuario el formulario del cuestionario y le permitirá completarlo. • El usuario completará el cuestionario. • El sistema verificará la identidad del usuario con los tokens suministrados. • El sistema almacenará y procesará los datos extraídos del cuestionario realizado. 	
Postcondición	El usuario debe haber realizado un cuestionario de autoevaluación de la salud mental.	
Excepciones	Caso	Acción
	Los tokens suministrados no son válidos, por lo que no puede verificarse la identidad del usuario.	Bloquear la petición del usuario y no le permitirle cerrar la sesión referida. En caso de que se encuentre caducado expedir un nuevo token de verificación y volver a realizar la petición.

Tabla 8.6: Prueba realizar cuestionario salud mental

Tipo de Prueba	Visualizar resultados y explicación de cuestionarios de autoevaluación de salud mental.	
Dependencias	<ul style="list-style-type: none"> • Registro en la aplicación. • Sesión iniciada • Realizar cuestionario de autoevaluación sobre la salud mental. 	
Precondición	El usuario debe de acabar de realizar un cuestionario.	
Descripción	El sistema permitirá al usuario visualizar los resultados obtenidos en el cuestionario de autoevaluación de salud mental realizado, junto con una pequeña explicación de los mismos.	
Secuencia normal	<ul style="list-style-type: none"> • El usuario finaliza un cuestionario de autoevaluación de la salud mental. • El sistema calcula los resultados obtenidos por el usuario. • El sistema muestra al usuario los resultados obtenidos y una explicación sobre ellos. 	
Postcondición	El usuario debe haber visualizado los resultados obtenidos en su cuestionario de autoevaluación de salud mental.	
Excepciones	Caso	Acción
	Los tokens suministrados no son válidos, por lo que no puede verificarse la identidad del usuario.	Bloquear la petición del usuario y no le permitirle cerrar la sesión referida. En caso de que se encuentre caducado expedir un nuevo token de verificación y volver a realizar la petición.

Tabla 8.7: Prueba visualizar resultados y explicación de cuestionarios de autoevaluación de salud mental.

Tipo de Prueba	Visualizar histórico de cuestionarios de cuestionarios de autoevaluación de salud mental.	
Dependencias	<ul style="list-style-type: none"> • Registro en la aplicación. • Sesión iniciada 	
Precondición	El usuario debe tener una sesión iniciada en el sistema.	
Descripción	El sistema permitirá al usuario visualizar un histórico de todos los cuestionarios de autoevaluación de la salud mental realizados junto con una pequeña contextualización de cada uno.	
Secuencia normal	<ul style="list-style-type: none"> • El usuario decidirá consultar el histórico de los cuestionarios de autoevaluación de la salud mental realizados previamente. • El sistema verificará la identidad del usuario con los tokens suministrados. • El sistema mostrará al usuario el histórico de todos los cuestionarios que ha realizado junto con una pequeña contextualización de cada uno. 	
Postcondición	El usuario debe haber podido consultar el histórico de todos los cuestionarios de autoevaluación de salud mental que ha realizado en su cuenta de usuario.	
Excepciones	Caso	Acción
	Los tokens suministrados no son válidos, por lo que no puede verificarse la identidad del usuario.	Bloquear la petición del usuario y no le permitirle cerrar la sesión referida. En caso de que se encuentre caducado expedir un nuevo token de verificación y volver a realizar la petición.
	El usuario no ha realizado ningún cuestionario en el sistema.	Mostrar al usuario un mensaje de error indicando que no ha realizado aún ningún cuestionario en la aplicación y devolverá al usuario al menú de cuestionarios.

Tabla 8.8: Prueba visualizar histórico de cuestionarios de cuestionarios de autoevaluación de salud mental.

Tipo de Prueba	Editar los datos personales del usuario.	
Dependencias	<ul style="list-style-type: none"> • Registro en la aplicación. • Sesión iniciada 	
Precondición	El usuario debe tener una sesión iniciada en el sistema.	
Descripción	El sistema permitirá al usuario modificar los datos asociados a su cuenta de usuario (nombre de usuario, fecha de nacimiento, sexo, altura y peso).	
Secuencia normal	<ul style="list-style-type: none"> • El usuario modifica los campos deseados de los datos de su perfil de usuario. • El sistema verificará la identidad del usuario con los tokens suministrados. • El sistema comprueba si el nombre de usuario ha sido modificado y si se encuentra disponible. • El sistema actualiza la nueva información introducida en el perfil del usuario de la aplicación. 	
Postcondición	El usuario debe de haber podido editar sus datos personales asociados a su cuenta de usuario.	
Excepciones	Caso	Acción
	El nuevo nombre de usuario introducido por el usuario ya se encuentra en uso.	Mostrar al usuario un mensaje de error indicando que el nuevo nombre de usuario introducido no se encuentra disponible en el sistema.
	Los tokens suministrados no son válidos, por lo que no puede verificarse la identidad del usuario.	Bloquear la petición del usuario y no le permitirle cerrar la sesión referida. En caso de que se encuentre caducado expedir un nuevo token de verificación y volver a realizar la petición.

Tabla 8.9: Prueba editar los datos personales del usuario.

Tipo de Prueba	Editar los sistemas de medida de las métricas de salud.	
Dependencias	<ul style="list-style-type: none"> • Registro en la aplicación. • Sesión iniciada 	
Precondición	El usuario debe tener una sesión iniciada en el sistema.	
Descripción	El sistema permitirá al usuario cambiar los distintos sistemas de medida para cada una de las unidades disponibles, distancia, altura y peso.	
Secuencia normal	<ul style="list-style-type: none"> • El usuario modifica los distintos sistemas de medida para cada una de las unidades disponibles. • El sistema verificará la identidad del usuario con los tokens suministrados. • El sistema actualiza la nueva información introducida en el perfil del usuario de la aplicación. 	
Postcondición	El usuario habrá modificado el tipo de sistema de medida que prefiere utilizar en cada una de las unidades disponibles.	
Excepciones	Caso	Acción
	Los tokens suministrados no son válidos, por lo que no puede verificarse la identidad del usuario.	Bloquear la petición del usuario y no le permitirle cerrar la sesión referida. En caso de que se encuentre caducado expedir un nuevo token de verificación y volver a realizar la petición.

Tabla 8.10: Prueba editar los sistemas de medida de las métricas de salud.

Tipo de Prueba	Editar los objetivos de las métricas de salud.	
Dependencias	<ul style="list-style-type: none"> • Registro en la aplicación. • Sesión iniciada 	
Precondición	El usuario debe tener una sesión iniciada en el sistema.	
Descripción	El sistema permitirá al usuario cambiar los objetivos para cada una de las métricas de salud disponibles (pasos, calorías y peso).	
Secuencia normal	<ul style="list-style-type: none"> • El usuario modifica los distintos objetivos de cada una de las métricas de salud disponibles. • El sistema verificará la identidad del usuario con los tokens suministrados. • El sistema actualiza la nueva información introducida en el perfil del usuario de la aplicación. 	
Postcondición	El usuario habrá modificado la marca objetivo de las métricas de salud que ha decidido editar en la aplicación.	
Excepciones	Caso	Acción
	Los tokens suministrados no son válidos, por lo que no puede verificarse la identidad del usuario.	Bloquear la petición del usuario y no le permitirle cerrar la sesión referida. En caso de que se encuentre caducado expedir un nuevo token de verificación y volver a realizar la petición.

Tabla 8.11: Prueba editar los objetivos de las métricas de salud.

Capítulo 9

Conclusiones

Este último capítulo aportará un pequeño resumen sobre aquellos puntos que no han podido completarse en este proyecto, la experiencia y las conclusiones resultantes de todo el desarrollo de este Trabajo de Fin de Grado.

De entre los objetivos que han sido completados en este proyecto se encuentran el desarrollo de una aplicación que permita a los usuarios poder monitorizar los datos relacionados con su salud física y mental, la implementación de cuestionarios que permitan a los usuarios realizar una evaluación de su estado mental actual, el desarrollo de históricos de los datos referentes a la salud física y mental de los usuarios y la implementación de distintos sistemas de medida para las distintas métricas de salud.

En cuanto a los objetivos que no se han conseguido implementar en esta primera versión de la aplicación cabe destacar la conexión de la aplicación con el *Smartwatch*. Este punto en concreto ha supuesto uno de los mayores desafíos presentados durante el desarrollo del proyecto, esto es debido a varios puntos en general.

El primero de ellos es que no existe mucha información sobre este proceso y la que se puede llegar a encontrar tiene, en la mayoría de los casos, unos cuantos años de antigüedad, lo que provoca que mucha de esta información haya quedado obsoleta y, aunque puede dar cierta idea sobre el proceso que debería de seguirse, no ha podido ser aprovechada. Otro punto de gran importancia es que, aunque la mayoría de estos dispositivos utilizan BLE (Bluetooth Low Energy) como protocolo de comunicación con otros dispositivos no existe ningún estándar de uso por lo que cada dispositivo realiza el proceso de conexión y vinculación de forma distinta complicando aún más la conexión con una gran variedad de *Smartwatch*.

Desde el punto de vista técnico se ha podido trabajar con una gran variedad de tecnologías, algunas de las cuales no se habían podido llegar a utilizar en durante el grado de ingeniería informática, por lo que ha resultado muy interesante poder profundizar en ellas, aunque en

algunos casos hayan supuesto una carga de trabajo que no se esperaba en la planificación original del proyecto.

Y desde el punto de vista personal, este proyecto ha supuesto un gran reto ya que, aunque durante el transcurso del grado se han realizado numerosos proyectos de desarrollo de software este es sin ninguna duda el más completo y el más exigente de todos ellos, llegando a ser en algunas ocasiones frustrante, aunque en su gran mayoría ha resultado ser muy interesante y gratificante a medida que se iba avanzando en el mismo.

9.1. Trabajos futuros

Aunque esta primera versión del proyecto se considera completada existen una serie de características o funcionalidades que no han podido ser implementadas debido a diversos factores, tales como falta de formación, falta de información y falta de tiempo a la hora de finalizar el proyecto. Por lo que a continuación se enumeran estas funcionalidades, las cuales deberían ser implementadas en futuras versiones con el objetivo de conseguir un proyecto más completo y pulido.

- Conexión con dispositivos *Smartwatch*. Este punto permitirá a los usuarios poder vincular sus dispositivos con la aplicación permitiendo obtener sus datos para las métricas disponibles en la misma.
- Implementación del protocolo HTTPS. Este punto permitirá a los usuarios utilizar la aplicación con un protocolo de conexión segura encriptando y protegiendo su información durante el proceso de transmisión.
- Hosting de la aplicación. Este punto permitirá a los usuarios utilizar la aplicación desde la red, sin necesidad de que el servidor y la aplicación se encuentren en la misma red. Una de las posibles candidatas para esta futura implementación podría ser AWS, Amazon Web Service, la cual ofrece un sistema de hosting muy popular actualmente debido a su gran variedad de servicios.
- Incluir más cuestionarios de autoevaluación de la salud mental. Este punto permitirá a un usuario que ha permanecido durante un largo periodo de tiempo utilizando la aplicación completar nuevos cuestionarios con cierta frecuencia, lo que puede llegar a incentivar aún más su permanencia y su uso de la aplicación.

Bibliografía

- [1] Salud física
Consultada por última vez el 11 de Mayo del 2023
<https://www.who.int/news-room/fact-sheets/detail/physical-activity>
- [2] Salud mental
Consultada por última vez el 11 de Mayo del 2023
<https://www.who.int/es/news-room/fact-sheets/detail/mental-health-strengthening-our-response>
- [3] SmartWatch
Consultada por última vez el 11 de Mayo del 2023
<https://en.wikipedia.org/wiki/Smartwatch>
- [4] SmartWatch Technology
Consultada por última vez el 11 de Mayo del 2023
<https://www.britannica.com/technology/smartwatch>
- [5] Tecnología SmartWatch
Consultada por última vez el 11 de Mayo del 2023
<https://www.techtarget.com/iotagenda/definition/smartwatch>
- [6] Tecnología Cuestionarios
Consultada por última vez el 11 de Mayo del 2023
<https://web.archive.org/web/20200916071844/https://bemocion.sanidad.gob.es/avisoLegal/home.htm>
- [7] El mercado de smartwatches sigue creciendo durante la crisis del COVID-19
Consultada por última vez el 11 de Mayo del 2023
<https://www.muycanal.com/2020/05/08/mercado-smartwatches-crece-durante-covid-19>
- [8] El mercado smartwatch crece a pesar de todo pronóstico económico
Consultada por última vez el 11 de Mayo del 2023
<https://www.unosantafe.com.ar/informacion-general/el-mercado-smartwatch-crece-pesar-todo-pronostico-economico-n2746760.html>
- [9] Guía de la metodología SCRUM: qué es, cómo funciona y cómo empezar
Consultada por última vez el 11 de Mayo del 2023
<https://www.atlassian.com/es/agile/scrum>

- [10] Historias de usuario con ejemplos y plantilla
Consultada por última vez el 11 de Mayo del 2023
<https://www.atlassian.com/es/agile/project-management/user-stories>
- [11] Programa de gestión de riesgos
Consultada por última vez el 11 de Mayo del 2023
<https://www.atlassian.com/es/trust/compliance/risk-management-program>
- [12] Apuntes Planificación y Gestión de Proyectos.
Consultada por última vez el 11 de Mayo del 2023
- [13] Sueldos para el puesto de Full Stack Developer en España
Consultada por última vez el 11 de Mayo del 2023
https://www.glassdoor.es/Sueldos/full-stack-developer-sueldo-SRCH_K00,20.htm
- [14] Salario medio para Full Stack en España, 2023
Consultada por última vez el 11 de Mayo del 2023
<https://es.talent.com/salary?job=full+stack>
- [15] Tabla de amortización simplificada, Agencia Tributaria
Consultada por última vez el 11 de Mayo del 2023
https://sede.agenciatributaria.gob.es/Sede/ayuda/manuales-videos-folletos/manuales-practicos/folleto-actividades-economicas/3-impuesto-sobre-renta-personas-fisicas/3_5-estimacion-directa-simplificada/3_5_4-tabla-amortizacion-simplificada.html
- [16] GitHub
Consultada por última vez el 11 de Mayo del 2023
<https://github.com/>
- [17] Visual Studio Code
Consultada por última vez el 11 de Mayo del 2023
<https://code.visualstudio.com/>
- [18] Introducción a Android Studio
Consultada por última vez el 11 de Mayo del 2023
<https://developer.android.com/studio/intro?hl=es-419>
- [19] PostgreSQL: Documentation
Consultada por última vez el 11 de Mayo del 2023
<https://www.postgresql.org/docs/current/app-psql.html>
- [20] Overleaf
Consultada por última vez el 11 de Mayo del 2023
<https://www.overleaf.com>
- [21] Astah
Consultada por última vez el 11 de Mayo del 2023
<https://astah.net/>
- [22] Balsamiq Wireframes
Consultada por última vez el 11 de Mayo del 2023
<https://balsamiq.com/>

- [23] nRF Connect
Consultada por última vez el 11 de Mayo del 2023
<https://play.google.com/store/apps/details?id=no.nordicsemi.android.mcp&pli=1>
- [24] Postman
Consultada por última vez el 11 de Mayo del 2023
<https://www.postman.com/>
- [25] Mockaroo
Consultada por última vez el 11 de Mayo del 2023
<https://mockaroo.com/>
- [26] Spring framework
Consultada por última vez el 11 de Mayo del 2023
<https://spring.io/projects/spring-framework>
- [27] Qué es Spring Framework y por qué usarlo
Consultada por última vez el 11 de Mayo del 2023
<https://openwebinars.net/blog/conoce-que-es-spring-framework-y-por-que-usarlo/>
- [28] Ionic framework
Consultada por última vez el 11 de Mayo del 2023
<https://ionicframework.com/>
- [29] React framework
Consultada por última vez el 11 de Mayo del 2023
<https://react.dev/>
- [30] Node.js
Consultada por última vez el 11 de Mayo del 2023
<https://nodejs.org/en>
- [31] Npm
Consultada por última vez el 11 de Mayo del 2023
<https://www.npmjs.com/>
- [32] Android
Consultada por última vez el 11 de Mayo del 2023
<https://www.android.com/>
- [33] Gmail
Consultada por última vez el 11 de Mayo del 2023
<https://mail.google.com/mail>
- [34] Bluetooth Assigned Numbers Document
Consultada por última vez el 11 de Mayo del 2023
<https://www.bluetooth.com/specifications/assigned-numbers/>
- [35] Gadgetbridge
Consultada por última vez el 11 de Mayo del 2023
<https://github.com/Freeyourgadget/Gadgetbridge/tree/1ddea9268d08edaf4b71c587e6db37017ea55c8b>

- [36] MiBand Authentication Procedure
Consultada por última vez el 11 de Mayo del 2023
<https://leojrfs.github.io/writing/miband2-part1-auth/>
- [37] Huami token
Consultada por última vez el 11 de Mayo del 2023
<https://github.com/argrento/huami-token>
- [38] Obtain "Auth key" for MiBand to allow you to use it with Notify or GadgetBridge, without also needing MiFit app installed
Consultada por última vez el 11 de Mayo del 2023
https://www.reddit.com/r/miband/comments/nn8s3v/obtain_auth_key_for_miband_to_allow_you_to_use_it/
- [39] Los distintos tipos de pruebas de software
Consultada por última vez el 11 de Mayo del 2023
<https://www.atlassian.com/es/continuous-delivery/software-testing/types-of-software-testing>