

UNIVERSIDAD POLITÉCNICA DE MADRID

E.T.S. DE INGENIERÍA DE SISTEMAS INFORMÁTICOS

PROYECTO FIN DE MÁSTER

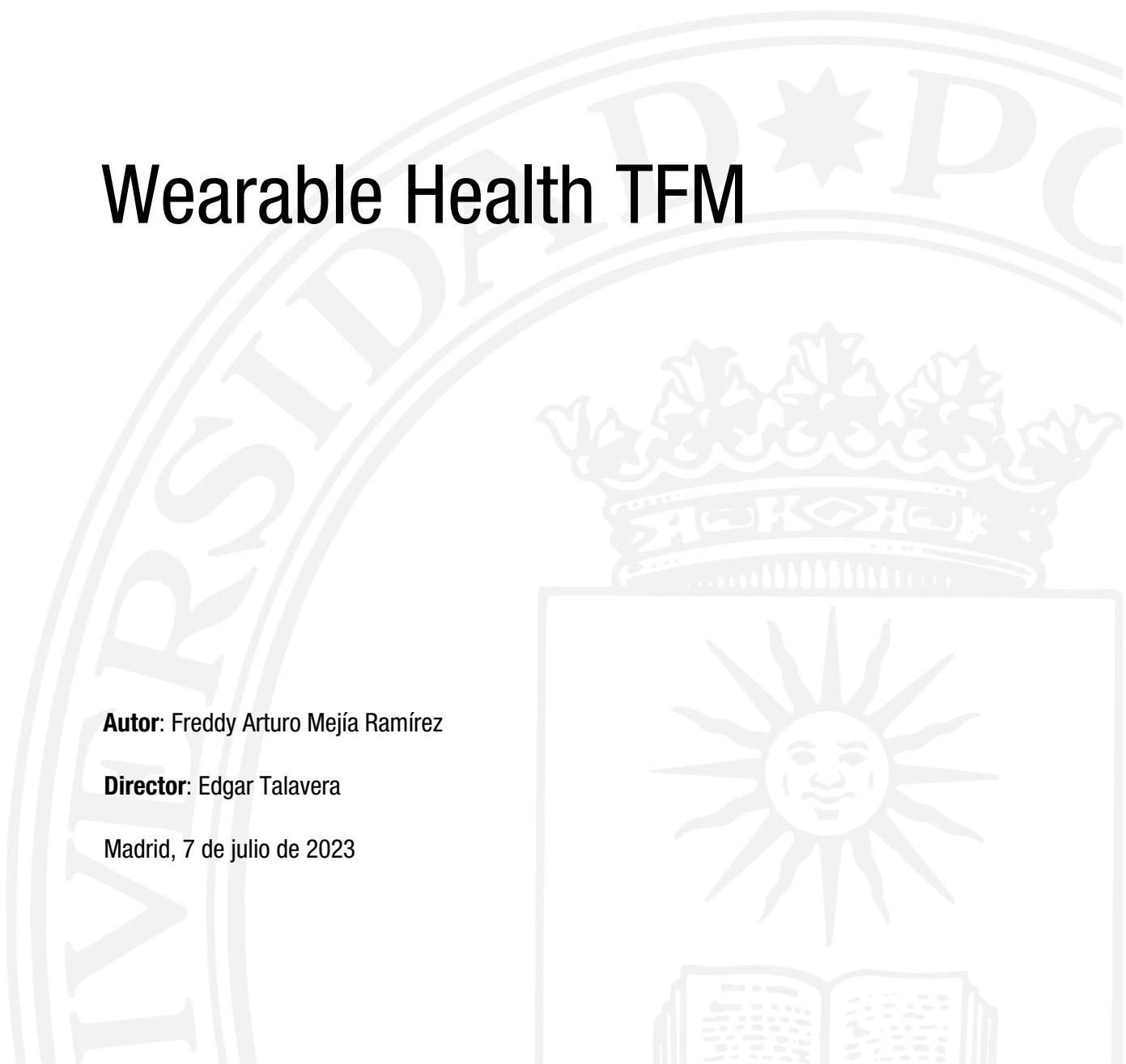
**MÁSTER UNIVERSITARIO EN DESARROLLO DE APLICACIONES Y SERVICIOS
PARA DISPOSITIVOS MÓVILES**

Wearable Health TFM

Autor: Freddy Arturo Mejía Ramírez

Director: Edgar Talavera

Madrid, 7 de julio de 2023



Freddy Arturo Mejía Ramírez

Wearable Health TFM

Proyecto Fin de Máster, 7 de julio de 2023

Director: Edgar Talavera

E.T.S. de Ingeniería de Sistemas Informáticos

Campus Sur UPM, Carretera de Valencia (A-3), km. 7

28031, Madrid, España

Esta obra está bajo una licencia [Creative Commons «Atribución-NoComercial-CompartirIgual 4.0 Internacional»](#).



Agradecimientos

Mi agradecimiento a Dios por la vida que me ha dado y la oportunidad de estar donde estoy. Agradezco especialmente a mi madre Mirian Ramírez Valarezo por ayudarme en cada paso que doy, sin ti todo esto no sería posible, tus consejos, juegos y largas charlas que me agradan siempre. Gracias a mi padre Jaime Mejía Ochoa por tu apoyo y consejería. Gracias a mi hermana Andrea Mejía Ramírez por tus consejos y sabiduría. Asimismo mi profundo agradecimiento a mi esposa Katherine Alvarado Terreros por sus buenos consejos y apoyo incondicional. Agradezco a toda mi familia y amigos que me guían y apoyan siempre.

Me gustaría agradecer a mi tutor Edgar Talavera por su consejería, dedicación, conocimientos y paciencia a lo largo de este trabajo.

Agradezco infinitamente a los docentes de esta universidad, sus conocimientos han sido una guía muy valiosa permitiéndome ampliar mis habilidades y destrezas hacia un nivel superior.

Resumen

El desarrollo tecnológico ha influido enormemente en la vida de los seres humanos y en los aspectos sociales y laborales, de tal manera que se dispone de una gran cantidad de información, utilizada adecuadamente es de gran utilidad en las diversas actividades y en la productividad que realizan las personas. El presente trabajo de fin de máster tiene el propósito de convertirse en una herramienta muy utilitaria para las personas que tienen limitaciones en acceder de manera inmediata a un centro de salud, éste dispositivo obtiene datos médicos de su salud, mediante sensores que monitorizan sus signos vitales en cualquier circunstancia, permitiendo al usuario utilizarlo de acuerdo a su inmediatez, e inclusive puede transferir estos datos a un familiar o amigo de confianza a fin de que realicen lo más conveniente para el usuario. Los wearables de muñeca con sus aplicaciones existentes: Fitbit, Garmin Connect, Apple Health, por su fácil acceso, utilidad y desarrollo permiten al usuario ampliar el conocimiento de su salud y también la administración de la medicación en su tiempo oportuno. La metodología en el desarrollo de este proyecto es la metodología Scrum que permite optimizar el producto en función de las necesidades. Para una funcionalidad firme se ha implementado cuatro Sprints de gran utilidad siendo éstos: Sprint Login y Registro; Sprint Perfil y Medicina; Sprint Amigos; y, Sprint wearables. El resultado de la aplicación de la metodología Scrum permitirá realizar cambios y establecer prioridades, optimizando frecuentemente en función de las necesidades del usuario. Palabras claves: Desarrollo, Salud, Medicina, Tecnología, Signos vitales.

Palabras clave: Desarrollo, Salud, Medicina, Tecnología, Signos vitales

Abstract

Technological development has greatly influenced the lives of human beings and in social and labor aspects, so that a large amount of information is available, properly used is very useful in the various activities and productivity performed by people. This master's thesis has the purpose of becoming a very useful tool for people who have limitations in accessing immediately to a health center, this device obtains medical data of their health, through sensors that monitor their vital signs in any circumstance, allowing the user to use it according to their immediacy, and can even transfer this data to a family member or trusted friend to do what is most convenient for the user. Wrist wearables with their existing applications: Fitbit, Garmin Connect , Apple Health, for its easy access, utility and development allow the user to expand the knowledge of their health and also the administration of medication in a timely manner. The methodology in the development of this project is the Scrum methodology that allows to optimize the product according to the needs. For a firm functionality four useful Sprints have been implemented being these: Sprint Login and Registration; Sprint Profile and Medicine; Sprint Friends; and, Sprint wearables. The result of the application of the Scrum methodology will allow changes to be made and priorities to be established, frequently optimizing according to the user's needs. Keywords: Development, Health, Medicine, Technology, Vital signs.

Keywords: Development, Health, Medicine, Technology, Vital signs.

Índice general

| | | |
|----------|------------------------------------|-----------|
| 1 | Introducción | 1 |
| 1.1 | Objetivos | 1 |
| 1.2 | Motivación | 2 |
| 1.3 | Justificación | 2 |
| 1.4 | Estructura de la memoria | 4 |
| 2 | Estado del arte | 5 |
| 2.1 | Signos vitales, medicina | 5 |
| 2.2 | Wearables | 6 |
| 3 | Metodología y diseño | 9 |
| 3.1 | Sprint Login y registro | 10 |
| 3.2 | Sprint perfil y medicina | 11 |
| 3.3 | Sprint amigos | 12 |
| 3.4 | Sprint Wearables | 13 |
| 3.5 | Versionamiento Github | 14 |
| 3.6 | Diseño | 17 |
| 4 | Fitbit | 29 |

| | | |
|-----|--------------------------------------|----|
| 5 | Desarrollo e implementación | 32 |
| 5.1 | Tecnologías | 32 |
| 5.2 | Autenticación fitbit | 45 |
| 5.3 | Obtención datos fitbit | 46 |
| 6 | Pruebas | 47 |
| 7 | Análisis del impacto socio-ambiental | 48 |
| 8 | Trabajos futuros | 50 |
| 9 | Conclusiones | 52 |

Índice de figuras

| | | |
|-----|---|----|
| 3.1 | sprint login y registro | 11 |
| 3.2 | sprint perfil y medicina | 12 |
| 3.3 | sprint perfil y medicina | 13 |
| 3.4 | sprint perfil y medicina | 14 |
| 3.5 | Repositorio GitHub de la app Android | 15 |
| 3.6 | Repositorio github web y backend api rest | 16 |
| 3.7 | Página web | 16 |
| 3.8 | Onboarding, login y registro app | 18 |
| 3.9 | Perfil y salud app | 19 |

| | | |
|------|---|----|
| 3.10 | Medicina app | 20 |
| 3.11 | Wearables app | 21 |
| 3.12 | Amigos y mis datos wearables app | 22 |
| 3.13 | Modelo entidad relación apiHealth | 23 |
| 4.1 | Fitbit inspire 3 [16] | 29 |
| 5.1 | Arquitectura del proyecto | 37 |
| 5.2 | Estructura de la aplicación | 39 |
| 5.3 | Estructura Adapter | 40 |
| 5.4 | Estructura Interface | 40 |
| 5.5 | Estructura Model | 41 |
| 5.6 | Estructura Interface api rest | 41 |
| 5.7 | Estructura datasource | 42 |
| 5.8 | Estructura repositorio | 42 |
| 5.9 | Estructura View Model | 43 |
| 5.10 | Estructura Fragment | 43 |
| 5.11 | Estructura Actividades | 44 |
| 5.12 | Estructura utilitaria | 44 |
| 5.13 | Formulario registro aplicación | 45 |
| 6.1 | Análisis Sonarqube | 47 |

Índice de tablas

| | | |
|------|--|----|
| 3.1 | Estructura tabla notificación | 24 |
| 3.2 | Estructura tabla health_users | 24 |
| 3.3 | Estructura tabla devices | 25 |
| 3.4 | Estructura tabla fitbit_keys | 25 |
| 3.5 | Estructura tabla forgot_pass_users | 26 |
| 3.6 | Estructura tabla metrics | 26 |
| 3.7 | Estructura tabla detail_metrics | 26 |
| 3.8 | Estructura tabla friends | 27 |
| 3.9 | Estructura tabla medicine | 27 |
| 3.10 | Estructura tabla medicine_detail | 28 |
| 3.11 | Estructura tabla users | 28 |

1.

Introducción

El presente trabajo tiene su origen en la necesidad de ayudar a las personas a realizar un monitoreo permanente de sus signos vitales y dar a conocer mediante esta aplicación el estado de su salud de tal manera que le permita al usuario realizar los correctivos pertinentes e incluso una información veras al médico de confianza a fin de que se pueda realizar las recomendaciones inmediatos.

En tal virtud el presente trabajo es muy importante porque es una herramienta que aporta datos precisos y cuyos resultados pueden ser útiles oportunamente e inclusive el impacto en el nivel social al cual va destinado para conocimiento de la calidad de vida que tienen.

Quién desearía tener un instrumento que realice lecturas de su estado de salud? Indudablemente que todos quisieran tener a la mano una aplicación que les permita una lectura inmediata de su estado de salud, consecuentemente es un gran beneficio para el contexto social.

En el Capítulo 1 se exponen los objetivos que persigue el presente trabajo, motivados hacia la ayuda inmediata y oportuna que se brinda al usuario, en éste sentido se justifica su implementación porque se atiende una necesidad social. El Capítulo 2 contiene la configuración de la memoria mediante la cual se describe la aplicación móvil android. Los componentes de la plantilla constan en el Capítulo 3, mediante el cual se indican los componentes existentes. En el Capítulo 4 consta sobre la marca fitibit. En el Capítulo 5 se mencionan sobre el desarrollo e implementación de la aplicación. En el capítulo 6 se mencionan las pruebas. En el Capítulo 7 se habla sobre el impacto social-ambiental. El capítulo 8 consta de trabajos futuros, Las conclusiones se plasman en el Capítulo 9. Las referencias, glosarios e índices se plasman en a continuación.

1.1. Objetivos

Describir la aplicación wearable health tfm permitirá evidenciar la informacion de signos vitales y administracion de medicinas a fin de que el usuario utilice oportunamente esta información y a la vez compartir con familiares y amigos, para que estén informados, puedan recordar al beneficiario del estado de su salud, por tal razón de detallan los siguientes objetivos específicos:

- **Obtener** y mostrar información de wearable fitbit para conocer el estado de salud de la persona.
- **Notificar** sobre la hora de tomar medicinas para que el usuario mantenga un adecuado control de sus medicamentos..
- **Agregar** amigos para que puedan ver la información recolectada y conocer el estado de salud de la persona.

1.2. Motivación

Lo que me ha motivado a la realización del presente trabajo de tesis ha sido la observación y sobre todo ver la necesidad de las personas que con mucha frecuencia descuidan o se olvidan de la administración de sus medicinas y también de un control rápido y verás de sus signos vitales como son frecuencia cardiaca, pasos, temperatura, que influye en la decisión de acudir oportunamente al médico. Consecuentemente éste aspecto de la motivación ha sido muy importante para tomar la decisión de brindar la ayuda y servicio humanitario a las personas en la limitación que a veces se evidencian de las buenas costumbres.

Utilizar tecnología innovadora para monitorear signos vitales, recordar tomar la medicina, logrando fomentar la prevención y el cuidado de la salud para contribuir con una sociedad saludable.

Generar un impacto social y ayuda a los demás, se experimenta una motivación intrínseca al saber que se está haciendo una diferencia en la vida de los demás, y ser apoyo en su estado de salud.

1.3. Justificación

La justificación para el estudio, la realización y ejecución del presente trabajo ha sido la observación y evidencia de la necesidad de las personas a tener acceso a un dispositivo que informe permanentemente del estado de su salud, pues con frecuencia se olvidan de hacerse atender oportunamente con el médico, como también de la administración adecuada de la medicación que deben tomar, en algunos casos por efectos de la edad y otras por descuido, consecuentemente esta aplicación permitirá a las personas un acceso oportuno y monitoreo de su estado de salud física permanentemente.

Los antecedentes expuestos son las razones predominantes para que haya escogido el desarrollo del presente proyecto DE APLICACIONES Y SERVICIOS PARA DISPOSITIVOS MÓVILES, permitiendo que los usuarios tengan acceso fácil e inmediato de su salud, consecuentemente la importancia y factibilidad del presente trabajo permitirá una mejor calidad de vida del usuario.

La salud es un aspecto fundamental de la vida humana, y el monitoreo de los signos vitales desempeña un papel crucial en la prevención, detección temprana y gestión de diversas condiciones médicas. El desarrollo de una aplicación para mostrar signos vitales, agregar medicamentos y observar los signos vitales de amigos puede tener un impacto significativo en el bienestar y la calidad de vida de las personas, para que los familiares y amigos puedan acceder a esta información es el usuario del dispositivo el único que puede abrir y generar una vinculación. A continuación, se presentan los puntos clave de la justificación:

Necesidad de monitoreo constante: El monitoreo regular de los signos vitales, como la frecuencia cardíaca, la presión arterial y la saturación de oxígeno, es esencial para detectar cualquier cambio o anomalía en la salud de una persona. El desarrollo de una aplicación que permita a las personas monitorear y registrar sus signos vitales de manera conveniente y precisa puede ayudar a identificar problemas de salud potenciales y tomar medidas preventivas a tiempo.

Mejora de la adherencia a la medicación: El seguimiento adecuado de los medicamentos es crucial para el manejo de enfermedades crónicas y el cumplimiento de los tratamientos. Una aplicación que permita a los usuarios agregar y recordar sus medicamentos, así como recibir recordatorios personalizados, puede ayudar a mejorar la adherencia a la medicación y garantizar una gestión adecuada de las condiciones médicas.

Promoción de la conciencia y educación en salud: El acceso fácil a la información sobre los signos vitales y su interpretación puede empoderar a las personas para que tomen decisiones informadas sobre su salud.

Fortalecimiento de las redes de apoyo: La posibilidad de observar los signos vitales de amigos y seres queridos en una aplicación puede fortalecer las redes de apoyo y el cuidado mutuo. Esto permite a las personas estar conectadas y preocuparse por la salud de los demás, lo que puede resultar especialmente beneficioso en situaciones en las que se requiere un cuidado a distancia o durante la atención de enfermedades crónicas.

1. **Relevancia del tema:** Muchas personas tienen una comprensión limitada de la importancia de los signos vitales y cómo interpretarlos. Una aplicación que muestre los signos vitales y proporcione información educativa relacionada puede promover la conciencia en salud y empoderar a los usuarios para que comprendan mejor su estado de salud. Esto puede conducir

a una toma de decisiones más oportuna e informada sobre el cuidado personal y la búsqueda de atención médica cuando sea necesario.

2. **Justificación teórica:** Generalmente la vamos a encontrar en la literatura médica que a veces es limitada para el usuario, pero que en el dispositivo lo encuentran de manera práctica, indicadores que solamente ofrecen una información mas no son curar o dar tratamiento a un malestar o enfermedad.
3. **Brecha en el conocimiento:** Es importante explorar en mayor medida cómo las aplicaciones pueden ser personalizadas y adaptadas a las necesidades específicas de los usuarios. El proyecto puede abordar esta brecha al desarrollar una aplicación que permita la personalización de las funcionalidades, recordatorios, brindando a los usuarios una experiencia adaptada a sus necesidades individuales, considerando aspectos como la edad, peso, estatura, entre otras que son importantes para la individualización de cada persona.
4. **Contribución práctica:** Indudablemente que el presente trabajo generará un gran apoyo a la conciencialización de la salud, el dispositivo por su fácil manejo y aplicabilidad puede ser utilizado de manera práctica por un gran sector social, niños, jóvenes, adultos, personas de la tercera edad, contribuyendo muy específicamente en el cuidado de su salud.

1.4. Estructura de la memoria

A continuación se detalla la estructura de este trabajo en los siguientes capítulos:

1. Estado del arte:
2. Metodología y diseño:
3. Fitbit:
4. Desarrollo e implementación:
5. Pruebas:
6. Análisis del impacto socio-ambiental:
7. Trabajos futuros:
8. Conclusiones:

2.

Estado del arte

Dentro de esta capítulo atenderemos el mundo de los signos vitales, la medicina y los wearables. Exploraremos los conceptos generales que subyacen a estas áreas para brindarte una comprensión sólida y amigable.

Los signos vitales son indicadores de nuestra salud para nuestro bienestar. Lo cual es importante prestar la atención necesaria a tales como la frecuencia cardíaca, la presión arterial, la temperatura corporal y la saturación de oxígeno. Comprender cómo estos signos vitales pueden fluctuar y qué significan esos cambios es fundamental para mantenernos saludables y detectar posibles problemas a tiempo.

Los wearables son dispositivos electrónicos que se llevan puestos en la muñeca y que están diseñados para monitorear nuestra salud y actividad física. Conocerás marcas conocidas, qué datos pueden recolectar y cómo pueden ayudarte a llevar un estilo de vida más saludable y activo.

2.1. Signos vitales, medicina

Los signos vitales son medidas objetivas y cuantificables que reflejan el funcionamiento básico del cuerpo humano. Estas mediciones son utilizadas para evaluar el estado de salud de una persona y monitorear sus funciones vitales. Los cuatro signos vitales principales son:

Frecuencia cardíaca: es el número de veces que el corazón late durante cierto periodo, por lo general un minuto. La frecuencia cardíaca se puede sentir en la muñeca, el lado del cuello, la parte de atrás de las rodillas, la parte de adelante de los pies, la ingle y otros lugares del cuerpo donde haya una arteria cerca de la piel. La frecuencia cardíaca en reposo normalmente oscila entre 60 y 100 latidos por minuto en un adulto sano en estado de reposo. La medición de la frecuencia cardíaca proporciona información importante sobre la salud de una persona. [\[1\]](#)

Presión arterial: Fuerza que ejerce contra la pared arterial la sangre que circula por las arterias. La presión arterial incluye dos mediciones: la presión sistólica, que se mide durante el latido del

corazón (momento de presión máxima), y la presión diastólica, que se mide durante el descanso entre dos latidos (momento de presión mínima). Primero se registra la presión sistólica y luego la presión diastólica, por ejemplo: 120/80. También se llama presión sanguínea arterial y tensión arterial. [2]

Frecuencia respiratoria: Es el número de respiraciones completas (inhalación y exhalación) por minuto. Se puede contar observando los movimientos del pecho o utilizando un dispositivo de medición específico.

Temperatura corporal: La temperatura corporal es una medida de la capacidad del organismo de generar y eliminar calor. El cuerpo es muy eficiente para mantener su temperatura dentro de límites seguros, incluso cuando la temperatura exterior cambia mucho. [3]

Estos signos vitales proporcionan información importante sobre el funcionamiento del sistema cardiovascular, respiratorio y otras funciones corporales esenciales. Son indicadores clave para evaluar la condición general de una persona y pueden ayudar a detectar posibles problemas de salud o cambios en el estado físico. El monitoreo regular de los signos vitales es fundamental en la atención médica, la gestión de enfermedades y la toma de decisiones clínicas.

Es importante cuidar nuestra salud para prevenir enfermedades. Probablemente una persona que realice ejercicio constantemente tendrá menos probabilidades de enfermarse de cualquier afección que una persona que lleve una vida sedentaria, es por tal razón que es beneficioso contar con un recordatorio de medicamentos para garantizar la correcta toma de las dosis necesarias.

2.2. Wearables

A lo largo de la evolución, el ser humano ha desarrollado diversas formas de obtener datos de signos vitales, desde métodos rudimentarios hasta tecnologías avanzadas. Estos avances han llevado al desarrollo de dispositivos portátiles, como los wearables de muñeca, que permiten obtener datos de signos vitales de manera conveniente y precisa.

En sus primeras etapas, el ser humano dependía de la observación directa y la palpación para evaluar los signos vitales, como el pulso y la respiración. Con el tiempo, se inventaron instrumentos como el estetoscopio y el esfigmomanómetro, que permitieron una medición más precisa de la frecuencia cardíaca y la presión arterial.

Sin embargo, el avance tecnológico ha llevado a la miniaturización de los dispositivos de monitoreo

de signos vitales. Esto ha dado lugar a los wearables de muñeca, como los smartwatches y las pulseras de actividad física, que ahora pueden registrar y mostrar datos de signos vitales en tiempo real.

Los wearables como tecnologías médicas se están convirtiendo en parte integrante de la analítica personal, midiendo el estado físico, registrando parámetros fisiológicos o informando de la programación de la medicación. Estas plataformas tecnológicas en continua evolución no sólo prometen ayudar a las personas a llevar un estilo de vida más saludable, sino que también proporcionan datos médicos continuos para el seguimiento activo del estado metabólico, el diagnóstico y el tratamiento. Los avances en la miniaturización de la electrónica flexible, los biosensores electroquímicos, la microfluídica y los algoritmos de inteligencia artificial han dado lugar a dispositivos vestibles que pueden generar datos médicos en tiempo real dentro del Internet de las cosas. Estos dispositivos flexibles pueden configurarse para entrar en contacto con interfaces epidérmicas, oculares, intracocleares y dentales para recoger señales bioquímicas o electrofisiológicas. [4]

Estos dispositivos utilizan tecnologías como los sensores ópticos y los acelerómetros para medir la frecuencia cardíaca, la variabilidad del ritmo cardíaco y la actividad física. Algunos modelos más avanzados también pueden medir la saturación de oxígeno en la sangre y la temperatura corporal.

La ventaja de los wearables de muñeca es su conveniencia y accesibilidad. Estos dispositivos son portátiles y de uso diario, lo que permite a las personas monitorear sus signos vitales en cualquier momento y en cualquier lugar. Además, la integración con aplicaciones móviles y servicios en la nube permite un seguimiento a largo plazo y el análisis de los datos recopilados.

En resumen, la evolución de la tecnología ha permitido al ser humano pasar de métodos rudimentarios de obtención de datos de signos vitales a la comodidad y precisión de los wearables de muñeca. Estos dispositivos han revolucionado el monitoreo de la salud al proporcionar una forma conveniente y precisa de obtener datos de signos vitales, lo que permite a las personas tomar un papel activo en el cuidado de su propia salud. Por consiguiente se detalla aplicaciones existentes en el mercado:

1. Fitbit: - Accede a datos únicos que te revelarán cómo duermes, disfruta de cientos de entrenamientos, meditaciones y mucho más para aprender a llevar una vida más sana. [5]
2. Garmin Connect: - La aplicación Garmin Connect para móvil y web es la herramienta perfecta para realizar el seguimiento, analizar y compartir las actividades deportivas registradas con tu dispositivo Garmin. Los usuarios pueden ver información como la distancia, el ritmo y la altitud, y analizar su rendimiento en tiempo real o después de la actividad. [6]
3. Apple Health: - Es una aplicación que recopila y centraliza datos de salud y actividad física de

varios dispositivos y aplicaciones compatibles. Permite a los usuarios tener una visión holística de su bienestar, al proporcionar datos sobre pasos, frecuencia cardíaca, sueño y más, todo en un solo lugar. [7]

4. Samsung Health: - Samsung Health registra y administra tus actividades y patrones de sueño para que puedas crear hábitos saludables. También puedes registrar y administrar la información de tu dieta y las condiciones de salud, como la frecuencia cardíaca y el estrés. [8]

5. Google Fit: - Con Google Fit, puedes medir, monitorizar y almacenar información sobre tu actividad física en tus dispositivos móviles y smartwatches. Medir automáticamente cuánto te queda para alcanzar tus objetivos, ver los tipos de actividades que realizas al día y su duración. [9]

3.

Metodología y diseño

Scrum es un conjunto sencillo de prácticas y reglas que engloba los requisitos de transparencia, inspección y adaptación inherentes al control empírico de procesos. [10]

Esto se traduce a la transparencia, la cual permite a todas las personas del equipo conocer como se maneja y trabaja en el proyecto, inspección de como evoluciona el trabajo y finalmente adaptación a los cambios haciendo que el equipo sea flexible y adaptable. [11]

Para el desarrollo de este proyecto, se ha utilizado la metodología Scrum, la cual es una metodología ágil ampliamente adoptada en el desarrollo de software debido a sus numerosos beneficios y ventajas.

Una de las razones por las que Scrum es una elección acertada para este proyecto es su enfoque iterativo e incremental. Scrum divide el desarrollo en ciclos de trabajo denominados "sprints", que son períodos de tiempo fijos y cortos en los que se realizan actividades de planificación, desarrollo, pruebas y revisión. Esto permite una entrega rápida y continua de funcionalidades, lo que es especialmente valioso en el contexto de una aplicación móvil, donde los ciclos de lanzamiento pueden ser más cortos.

Además, Scrum fomenta la colaboración estrecha entre los miembros del equipo de desarrollo, el cliente. Mediante reuniones regulares, como las reuniones diarias de seguimiento y las revisiones de sprint, todos los involucrados tienen la oportunidad de proporcionar retroalimentación y realizar ajustes a medida que avanza el proyecto. Esto promueve la transparencia, la comunicación efectiva y la resolución temprana de problemas, lo que resulta en un producto final más satisfactorio.

Scrum también se basa en la adaptabilidad y flexibilidad. A medida que se avanza en cada sprint, es posible realizar cambios en los requisitos o prioridades del proyecto. Esto permite ajustar y optimizar continuamente el producto en función de las necesidades cambiantes del mercado y del cliente.

Otro aspecto positivo de Scrum es su enfoque en la mejora continua. Al final de cada sprint, se lleva a cabo una retrospectiva para identificar áreas de mejora y oportunidades de optimización en el proceso de desarrollo. Esto fomenta un ambiente de aprendizaje y crecimiento constante, lo que

conduce a la evolución y refinamiento continuo de la aplicación.

Los sprints ayudan al equipo a lanzar mejor software con menos quebraderos de cabeza. [12] Permitiendo la incrementación del producto.

Existen cuatro etapas esenciales de los sprints, a continuación se relatan:

1. Planificación: en esta etapa se enfoca en las tareas que se va a realizar, así como la asignación de estas a cada miembro del equipo de desarrolladores.
2. Scrum diario: consiste en realizar reuniones de máximo quince minutos para conocer problemas, avances, mejoras que surjan en el desarrollo del sprint.
3. Revisión del Sprint: se lleva a cabo al finalizar el sprint para obtener retroalimentación de las tareas realizadas y así se pueda aprobar o rechazar el sprint.
4. Retrospectiva del Sprint: aquí se enfoca mayormente en el equipo de trabajo más que en la Iteración. Aquí se pretende mirar cómo han trabajado, los problemas que se presentaron, del mismo modo determinar la forma de corregir dichas falencias y poder dar inicio a un nuevo Sprint. [13]

Para este proyecto se ha decidido crear cuatro sprints, que a continuación se detallan:

3.1. Sprint Login y registro

Este sprint consiste en la desarrollar las actividades de registro, login y de olvidar contraseña para que el usuario logre acceder a la app, usando su correo y contraseña.

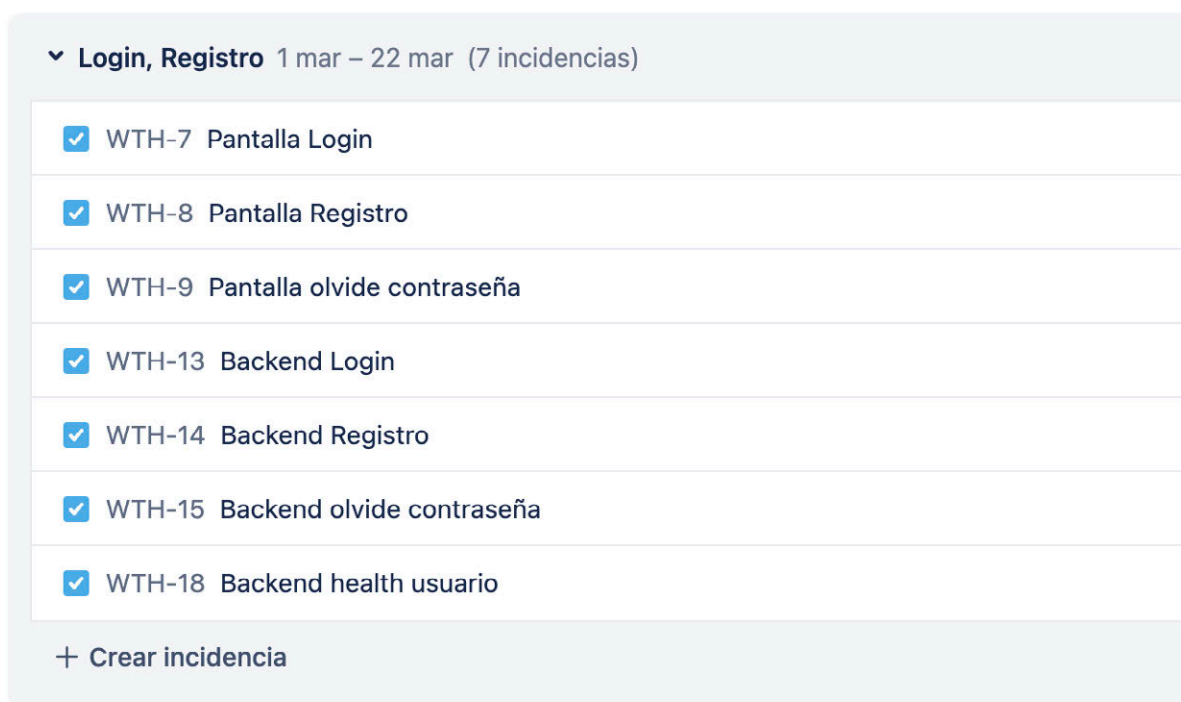


Figura 3.1. sprint login y registro

3.2. Sprint perfil y medicina

Este sprint consiste en desarrollar las actividades de perfil y medicina para que el usuario pueda modificar sus datos de salud, personales. En este sprint el usuario tambien puede agregar, modificar o eliminar medicinas para su respectivo recordatorio.



Figura 3.2. sprint perfil y medicina

3.3. Sprint amigos

Este sprint consiste en desarrollar las actividades de amigos y salud para que el usuario pueda agregar amigos , así tambien para que pueda observar sus datos de salud respectivos.



Figura 3.3. sprint perfil y medicina

3.4. Sprint Wearables

Este sprint consiste en la desarrollar las actividades de wearables para que el usuario pueda loguear su cuenta de fitbit y así pueda observar sus datos de salud respectivos.



Figura 3.4. sprint perfil y medicina

3.5. Versionamiento Github

GitHub es una plataforma de alojamiento de código para el control de versiones y la colaboración. Permite al equipo de desarrollo trabajar juntos en proyectos desde cualquier lugar. Gracias a esta plataforma se ha podido alojar y llevar un control del código realizando cambios y resolviendo bugs a lo largo del desarrollo de la app, a continuación se pueden observar los dos repositorios alojados en esta plataforma. [14]

App Android

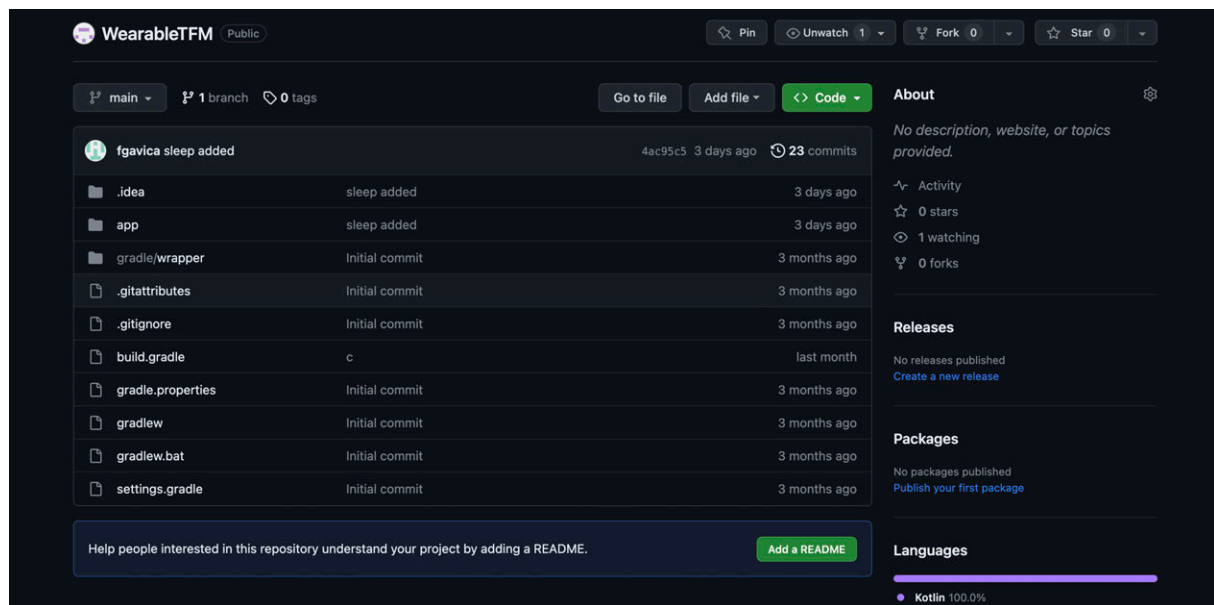


Figura 3.5. Repositorio GitHub de la app Android
Código fuente en el siguiente enlace: <https://github.com/freddmejia/WearableTFM>.

Web y backend api rest

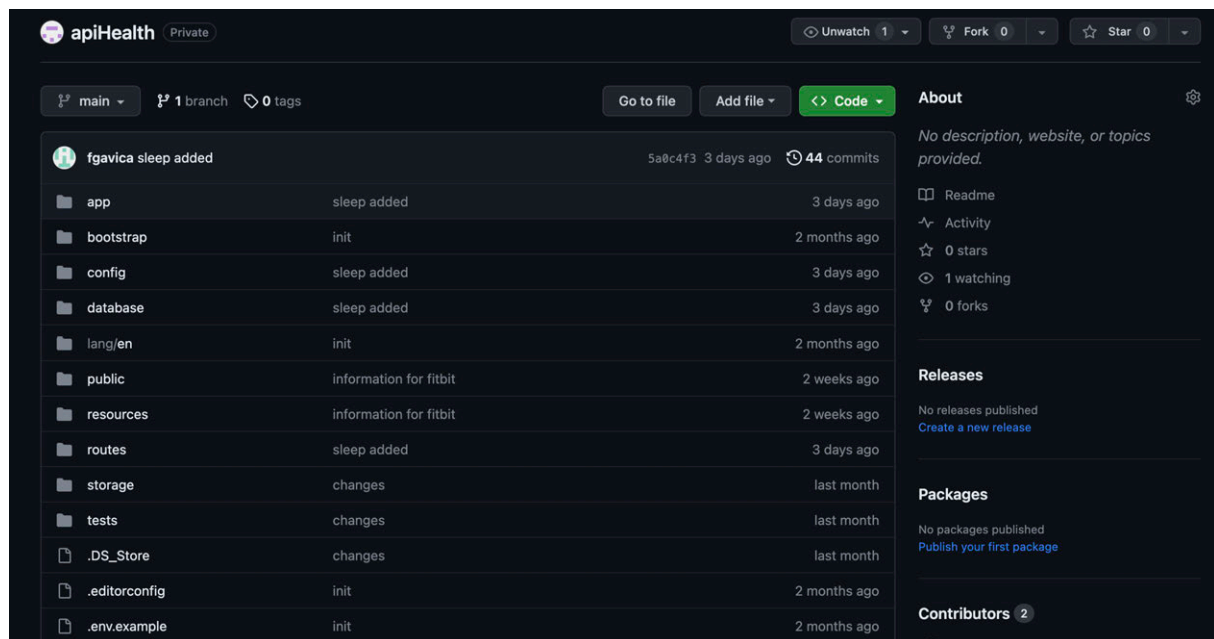
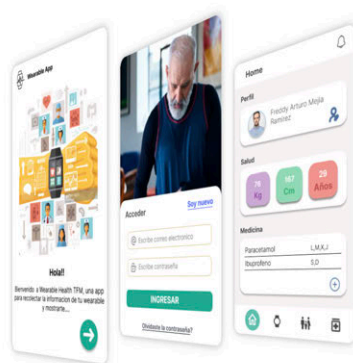


Figura 3.6. Repositorio github web y backend api rest

Código fuente en el siguiente enlace: <https://github.com/freddmejia/apiHealth>.

Wearable health TFM

[Términos del Servicio](#) [Política de Privacidad](#)



Descubre el poder de Wearable Health TFM, nuestra innovadora aplicación móvil diseñada exclusivamente para dispositivos Android. Obtén valiosos datos sobre tu ritmo cardíaco, calidad del sueño, pasos dados y calorías quemadas, todo al alcance de tu mano. Con Wearable Health TFM, podrás registrar y hacer un seguimiento de tu progreso de manera sencilla y efectiva. Además, podrás establecer recordatorios para tus medicamentos, asegurándote de tomarlos puntualmente y sin olvidos. Una característica única de nuestra app es la posibilidad de agregar amigos y compartir tu información de signos vitales. Invita a tus seres queridos a unirse y supervisar juntos sus metas de bienestar. ¡Celebren los logros juntos y brinden apoyo mutuo en su camino hacia un estilo de vida saludable! Descarga Wearable Health TFM en tu dispositivo Android y comienza a disfrutar de una experiencia completa para el cuidado de tu salud.

Figura 3.7. Página web

Sitio web: <http://wearablehealth-tfm.com/>.

Cabe recalcar que para alojar esta api rest, se lo ha hecho en los servidores de amazon web servi-

ces, por ello ha jugado un papel importante tener alojado el repositorio api rest en github, ya que mediante autenticación desde aws se pudo descargar el código fácilmente para su respectivo despliegue.

3.6. Diseño

Dentro de este capítulo se detalla el desarrollo de las pantallas de la aplicación móvil en Figma, además se utiliza una base de datos MySQL para la creación de la estructura donde se alojarán los datos. El diseño se enfocó en ser amigable para el usuario, mientras que la base de datos se diseñó con robustez para soportar cambios.

3.6.1. Figma

Se ha utilizado la plataforma web gratuita Figma para el diseño de la aplicación móvil. Figma sirve para crear, compartir, probar diseños de sitios web y aplicaciones móviles. Es una herramienta muy popular entre diseñadores, jefes de producto, redactores y desarrolladores, la cual permite ayudar a todos los que participan en el proceso de diseño a contribuir, dar su opinión y tomar mejores decisiones más rápidamente. [15]

Con Figma, se ha podido crear interfaces de usuario personalizadas y visualmente atractivas para nuestro público objetivo. La plataforma ofrece una amplia gama de funciones y recursos que nos han ayudado a plasmar nuestras ideas de manera concisa y precisa.

Además, Figma ofrece opciones de personalización flexibles, lo que nos ha permitido ajustar cada detalle del diseño de nuestra aplicación móvil. Desde la selección de colores y fuentes además nos ha brindado la libertad creativa necesaria para crear una experiencia de usuario única y agradable.

Diseño en el siguiente enlace: <https://www.figma.com/file/F2Ng4NI dv06myySgo6nCdP/wearable-app?type=design&node-id=0-1&mode=design&t=rHYpMwtBfFeMkKoJ-0>.

A continuación se muestra el diseño de las pantallas de la aplicación móvil.

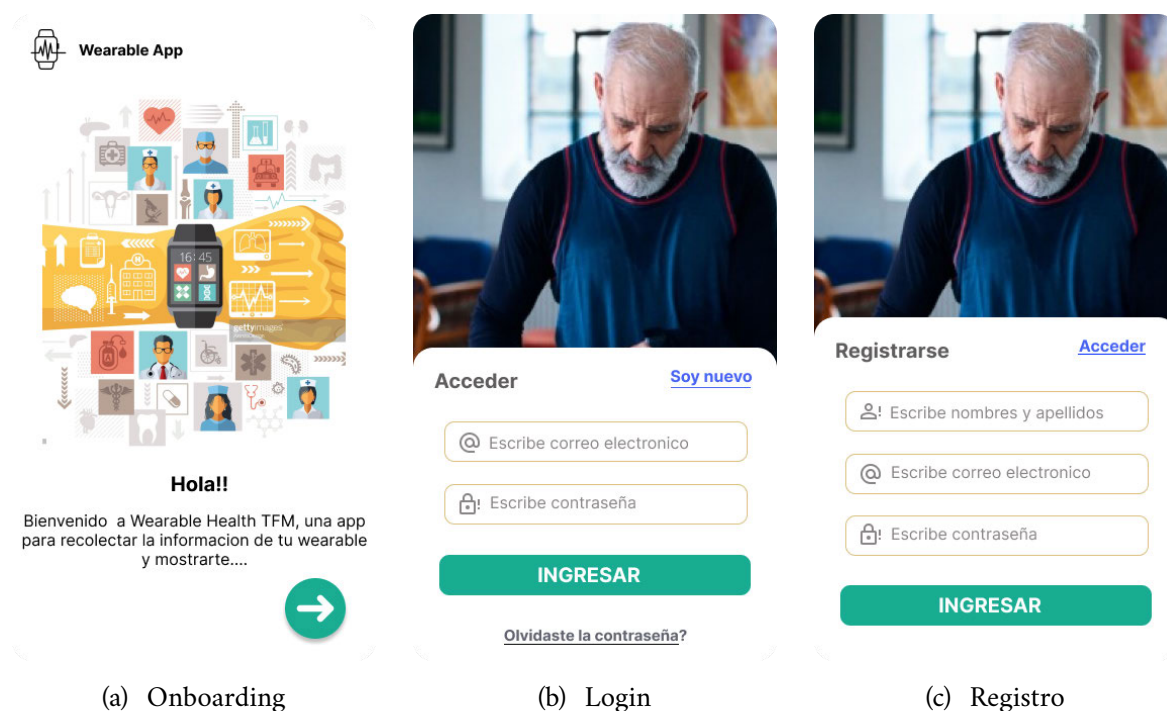


Figura 3.8. Onboarding, login y registro app

En esta sección, los usuarios tienen la posibilidad de registrarse en el sistema mediante un proceso sencillo y rápido. Simplemente necesitan proporcionar su dirección de correo electrónico, nombre y establecer una contraseña. En pocos pasos, podrán crear una cuenta personal y comenzar a disfrutar de todas las funcionalidades de la aplicación.

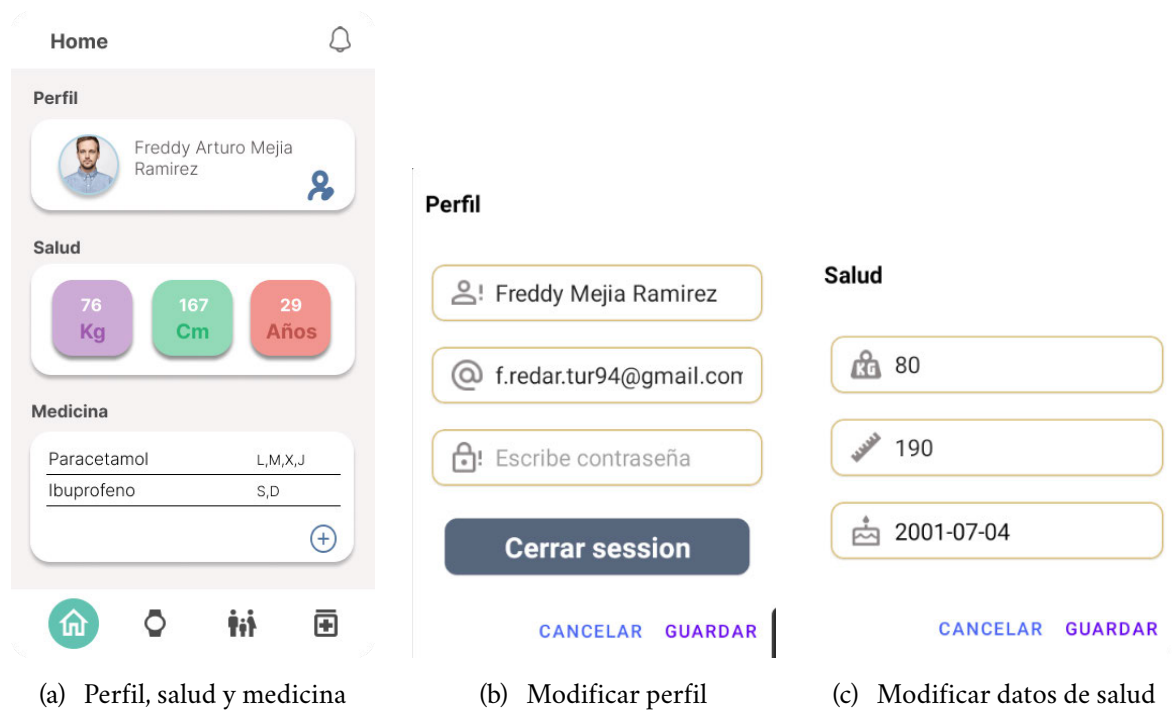


Figura 3.9. Perfil y salud app

Además, hemos incorporado funcionalidades relacionadas con la salud, donde los usuarios pueden ingresar y actualizar su peso corporal y estatura, así como también modificar datos provenientes de su cuenta personal.



(a) Medicinas

(b) Modificar medicina

Figura 3.10. Medicina app

Otra característica destacada es la capacidad de gestionar medicamentos. Los usuarios tienen la opción de agregar, editar o eliminar medicinas según sus necesidades. Esto les permite mantener un registro completo de su régimen de medicación y realizar cambios cuando sea necesario.

Para garantizar que los usuarios sigan su plan de medicación de manera efectiva, hemos implementado un sistema de notificaciones. Después de agregar un medicamento, se enviará una notificación para recordarles que deben ingerir la medicina en el momento adecuado. Esto ayuda a mejorar la adherencia al tratamiento y garantizar un cuidado óptimo de la salud.

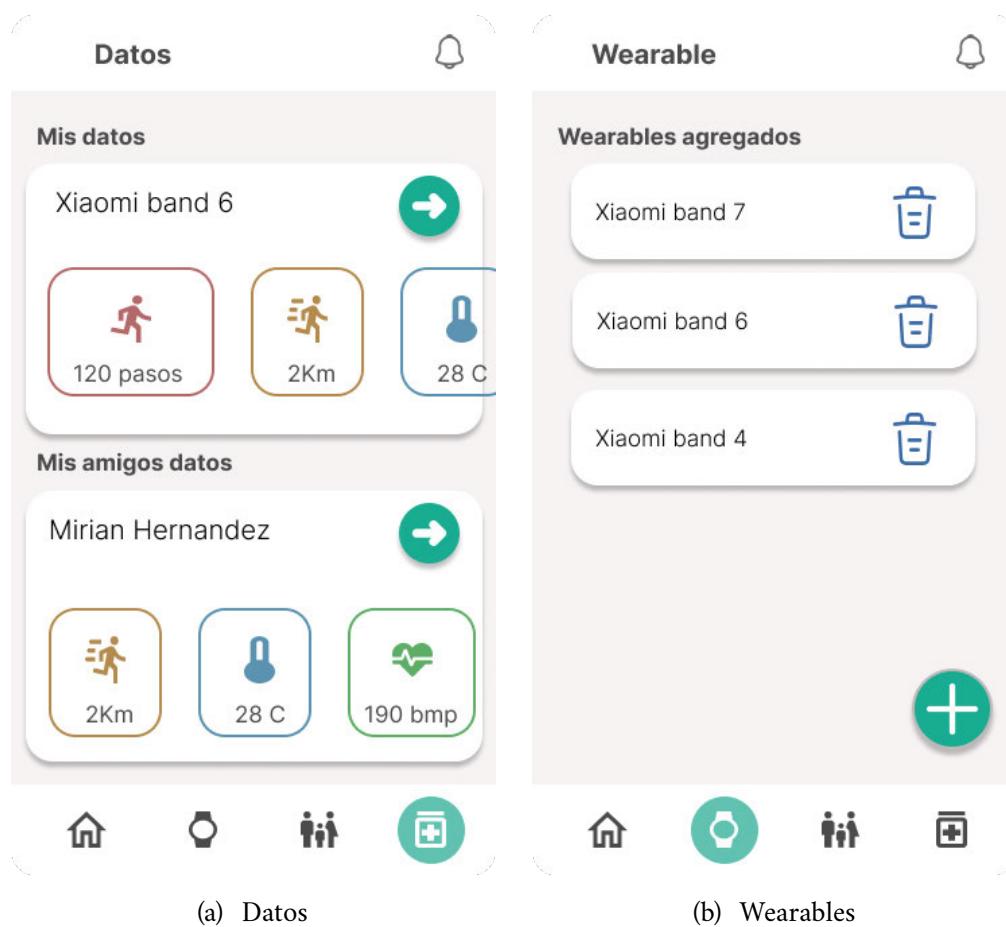


Figura 3.11. Wearables app

También se ha integrado la posibilidad de conectar el wearable Fitbit utilizando las credenciales de correo electrónico y contraseña del usuario. Una vez establecida la conexión, se obtienen datos valiosos provenientes de la sincronización entre la aplicación Fitbit y el wearable.

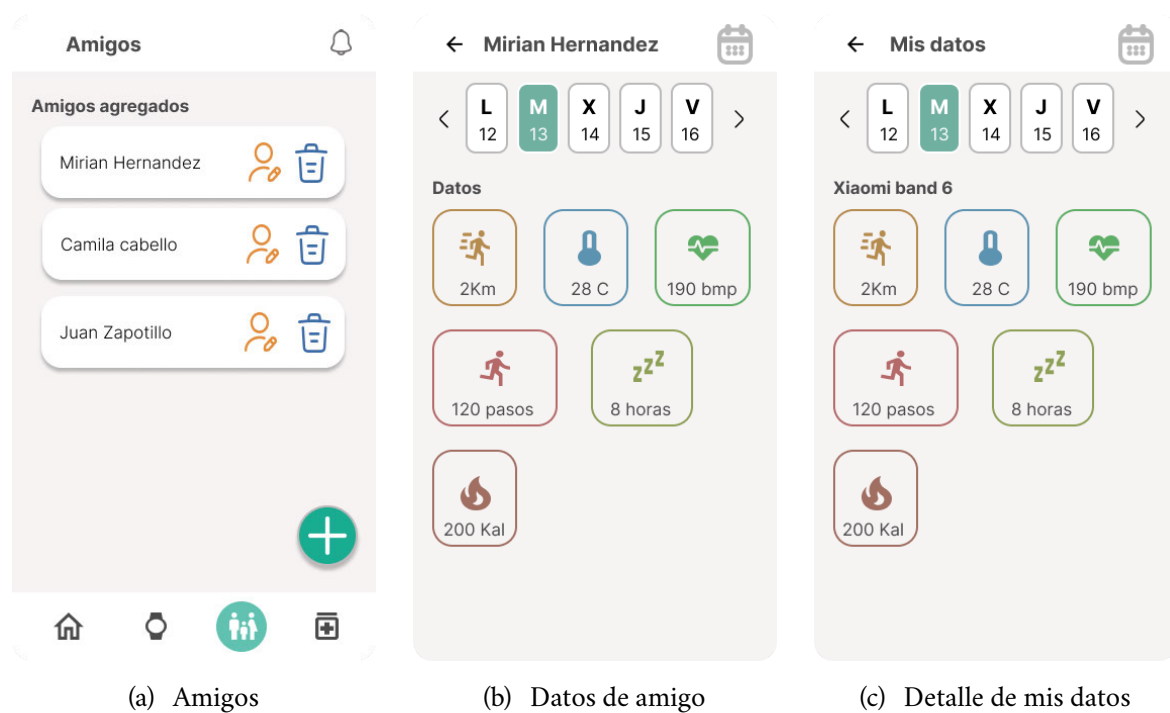


Figura 3.12. Amigos y mis datos wearables app

Además, se ha incluido la opción de agregar amigos en la aplicación. Esto permite a los usuarios compartir y visualizar el estado de los signos vitales de sus amigos. Es una forma de fomentar la comunidad y brindar apoyo en la mejora de la salud y el bienestar.

3.6.2. Base de datos

A continuación se detalla el modelo entidad-relación que describe la estructura y relaciones de las entidades en la base de datos.

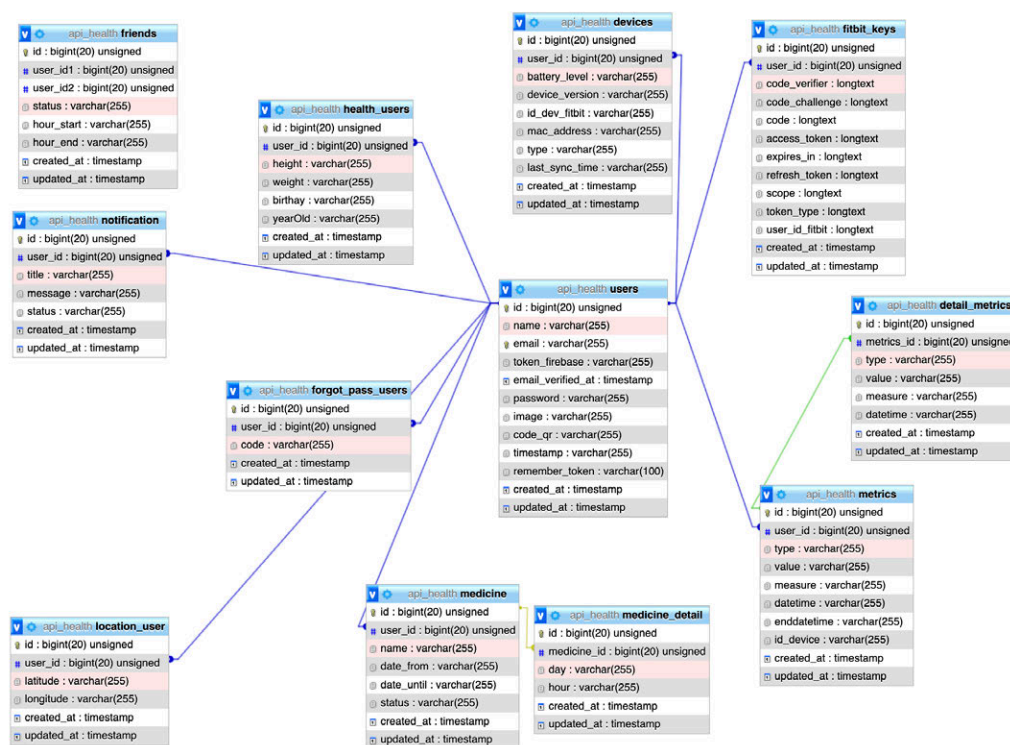


Figura 3.13. Modelo entidad relación apiHealth

Dentro del entorno de la base de datos, las tablas se conocen como entidades. Cada entidad representa un conjunto de datos relacionados y se utiliza para organizar y almacenar la información de manera estructurada. Cada entidad en la base de datos corresponde a una tabla en la que se definen las columnas y las filas que contienen los datos. A continuación se detallan las entidades que posee esta base de datos:

En esta estructura de datos se almacenan todas las notificaciones que son referentes al usuario. Las notificaciones son de tipo informativo: cuando se notifica al usuario que debe tomarse la medicina. (Ver tabla 3.1)

Tabla 3.1. Estructura tabla notificación

| Elemento | Descripción |
|------------|---|
| Id | Identificador único de la notificación |
| User_id | Identificador único referencia al usuario |
| Title | Título de la notificación |
| Message | Mensaje de la notificación |
| Status | Estado de la notificación |
| Created_at | Fecha de creación de la notificación |
| Updated_at | Fecha de modificación de la notificación |

En esta estructura de datos se almacenan los datos correspondientes a la salud del usuario, tales como: Peso, altura, edad. (Ver tabla 3.2)

Tabla 3.2. Estructura tabla health_users

| Elemento | Descripción |
|------------|---|
| Id | Identificador único de health_users |
| User_id | Identificador único referencia al usuario |
| Height | Tamaño del usuario en centímetros |
| Weight | Peso del usuario en centímetros |
| Birthday | Fecha de nacimiento |
| YearOld | Años de edad |
| Created_at | Fecha de creación de health_users |
| Updated_at | Fecha de modificación de health_users |

En esta estructura de datos se almacenan los wearables emparejados a la cuenta fitbit del usuario. (Ver tabla 3.3)

Tabla 3.3. Estructura tabla devices

| Elemento | Descripción |
|----------------|---|
| Id | Identificador único de devices |
| User_id | Identificador único referencia al usuario |
| Battery_level | Nivel de batería del wearable |
| Device_version | Version del wearable |
| Id_dev_fitbit | Identificador único del wearable |
| Mac_address | Identificador único del wearable |
| Type | Tipo de wearable |
| Last_sync_time | Última fecha sincronización wearable |
| Created_at | Fecha de creación de devices |
| Updated_at | Fecha de modificación de devices |

En esta estructura de datos se almacenan la configuración de fitbit para tener acceso a los datos del wearable del usuario previamente emparejado. (Ver tabla 3.4)

Tabla 3.4. Estructura tabla fitbit_keys

| Elemento | Descripción |
|----------------|---|
| Id | Identificador único de fitbit_keys |
| User_id | Identificador único referencia al usuario |
| Code_verifier | Identificador único pkce |
| Code_challenge | Identificador asociado a code_verifier |
| Code | Identificador asociado a code_verifier y code_challenge |
| Access_token | Identificador acceso a fitbit |
| Expires_in | Fecha de expiración de access_token |
| Refresh_token | Nuevo identificador acceso a fitbit |
| Scope | Lista de colecciones de datos solicitadas |
| Token_type | Texto bearer |
| User_id_fitbit | Identificador único del usuario en los servidores de fitbit |
| Created_at | Fecha de creación de fitbit_keys |
| Updated_at | Fecha de modificación fitbit_keys |

En esta estructura de datos se almacenan la configuración para poder reestablecer la cuenta del usuario, cambiando contraseña. (Ver tabla 3.5)

Tabla 3.5. Estructura tabla forgot_pass_users

| Elemento | Descripción |
|------------|---|
| Id | Identificador único de forgot_pass_users |
| User_id | Identificador único referencia al usuario |
| Code | Identificador único para lograr cambiar la contraseña |
| Created_at | Fecha de creación de forgot_pass_users |
| Updated_at | Fecha de modificación de forgot_pass_users |

En esta estructura de datos se almacenan los datos de signos vitales del usuario. (Ver tabla 3.6)

Tabla 3.6. Estructura tabla metrics

| Elemento | Descripción |
|-------------|---|
| Id | Identificador único de metrics |
| User_id | Identificador único referencia al usuario |
| Id_device | Referencia a device |
| Type | Tipo de metrica |
| Value | Valor de metrica |
| Measure | Medida de la metrica |
| DateTime | Fecha de creación de la metrica |
| EndDateTime | Fecha de finalización de la metrica |
| Created_at | Fecha de creación de metrics |
| Updated_at | Fecha de modificación de metrics |

En esta estructura de datos se almacenan los datos detalle de sleep (Ver tabla 3.7)

Tabla 3.7. Estructura tabla detail_metrics

| Elemento | Descripción |
|------------|--|
| Id | Identificador único de detail_metrics |
| Metrics_id | Identificador único referencia a metrics |
| Type | Tipo de metrica |
| Value | Valor de metrica |
| Measure | Medida de la metrica |
| DateTime | Fecha de creación del detail_metric |
| Created_at | Fecha de creación de detail_metrics |
| Updated_at | Fecha de modificación de detail_metrics |

En esta estructura de datos se almacenan las solicitudes de amistades entre usuarios (Ver tabla 3.8)

Tabla 3.8. Estructura tabla friends

| Elemento | Descripción |
|------------|---|
| Id | Identificador único de friends |
| User_id1 | Identificador único referencia al usuario amigo A |
| User_id2 | Identificador único referencia al usuario amigo B |
| Status | Estado solicitud amistad (1 solicitud, 2 amigos) |
| Hour_start | Hora inicio para visualizar datos de latitud y longitud |
| Hour_end | Hora fin para visualizar datos de latitud y longitud |
| Created_at | Fecha de creación de friends |
| Updated_at | Fecha de modificación de friends |

En esta estructura de datos se almacenan los datos de medicina que el usuario deba ingerir(Ver tabla 3.9)

Tabla 3.9. Estructura tabla medicine

| Elemento | Descripción |
|------------|---|
| Id | Identificador único de medicine |
| User_id | Identificador único referencia al usuario |
| Name | Nombre de la medicina |
| Date_from | Fecha de inicio de ingerir la medicina |
| Date_until | Fecha de fin de ingerir la medicina |
| Status | Estado de la medicina (0 eliminada, 1 creada) |
| Created_at | Fecha de creación de medicine |
| Updated_at | Fecha de modificación de medicine |

En esta estructura de datos del detalle de la medicina que el usuario deba ingerir(Ver tabla 3.10)

Tabla 3.10. Estructura tabla medicine_detail

| Elemento | Descripción |
|-------------|---|
| Id | Identificador único de medicine_detail |
| Medicine_id | Identificador único referencia a medicine |
| Day | Dia a ingerir la medicina |
| Hour | Hora a ingerir la medicina |
| Created_at | Fecha de creación de medicine_detail |
| Updated_at | Fecha de modificación de medicine_detail |

En esta estructura de datos se guardan los datos de usuarios registrados(Ver tabla 3.11)

Tabla 3.11. Estructura tabla users

| Elemento | Descripción |
|-------------------|---|
| Id | Identificador único de users |
| Name | Nombre del usuario |
| Email | Correo electrónico del usuario |
| Token_firebase | Identificador único por dispositivo móvil |
| Email_verified_at | Correo electrónico del usuario |
| Password | Contraseña del usuario |
| Image | Ruta de imagen del usuario |
| Code_qr | Código identificador unido del usuario |
| Timestamp | Fecha de creación de users |
| Remember_token | Recordar el token |
| Created_at | Fecha de creación de users |
| Updated_at | Fecha de modificación de users |



Figura 4.1. Fitbit inspire 3 [16]

Los wearables de la marca Fitbit posee sensores diseñados para recopilar de manera precisa los signos vitales. Estos sensores permiten recolectar información valiosa sobre tu actividad física, salud y bienestar en general. A continuación de detallan los signos vitales tales como ritmo cardíaco, sensor de sueño, actividad deportiva y temperatura:

1. Frecuencia cardíaca: permite medir el ritmo cardíaco de forma continua y en tiempo real, permitiéndote monitorear tu salud cardiovascular durante el ejercicio y en reposo. [16]
2. Pasos, distancia y calorías: permite la lectura precisa de tus pasos, distancia recorrida y calorías quemadas de todo el día. [16]
3. Temperatura: este sensor detecta cambios en tu temperatura corporal a lo largo del día, lo que conlleva a detectar posibles variaciones y tomar medidas preventivas. [16]

4. Sensor de sueño: mediante este sensor se obtiene el tiempo total de sueño y el tiempo que pasas en las fases de sueño ligero, profundo y REM. [16]

Fitbit proporciona datos provenientes de las siguientes fases del sueño:

1. Sueño profundo: es la fase en la que se desaceleran las funciones del cuerpo y las ondas cerebrales, la fase en la que se produce la regeneración celular y la síntesis de proteínas, tan importante para hacer crecer los músculos. El sueño profundo es la fase en la que ocurre la magia, y es el número que debe importarte cuando mires tu pulsera. [17]
2. Sueño ligero: durante esta fase la mente y el cuerpo son sensibles a los estímulos externos, lo cual se traduce que es más probable que nos despertemos con facilidad en respuesta a ruidos o estímulos ambientales.
3. Sueño Rem: estimula las regiones del cerebro que se utilizan para el aprendizaje. En esta fase se producen los sueños, por tal motivo dentro de esta investigación [18] enfatiza que es importante porque el cerebro capta información del mundo exterior de forma selectiva. Esa información se traduce en actividad neuronal que debe ser procesada y que contribuye así al desarrollo y la maduración del cerebro. De esta forma, el sueño REM favorece a aspectos tan importantes en el día a día como la memoria, el aprendizaje o la concentración. [19]

Inicialmente consideramos Xiaomi Band como una opción debido a su popularidad y su amplia gama de características. Sin embargo, al investigar más a fondo, descubrimos que el SDK oficial de Xiaomi Band no estaba disponible o no era fácilmente accesible. Esto nos presentó un desafío significativo para integrar y aprovechar las capacidades del dispositivo en nuestra aplicación.

Además, la documentación de la API REST de Xiaomi Band no estaba del todo clara y no proporcionaba la guía necesaria para una integración fluida. La falta de ejemplos prácticos y explicaciones detalladas dificultaba la comprensión completa de cómo utilizar plenamente las funcionalidades de Xiaomi Band.

A raíz de estas limitaciones, consideramos otras opciones y nos decidimos por Fitbit. La marca Fitbit ofrece una documentación fácil de entender, lo que nos permite integrar sus dispositivos en nuestra aplicación de manera eficiente. La claridad y la coherencia de la documentación de Fitbit nos proporcionan la confianza necesaria para aprovechar al máximo las capacidades de sus wearables y brindar una experiencia de usuario excepcional.

Adicionalmente la elección de Fitbit se basa en varias razones convincentes. En primer lugar, Fitbit proporciona una amplia gama de dispositivos portátiles que ofrecen un seguimiento preciso de ac-

tividades como pasos, distancia recorrida, frecuencia cardíaca y calidad del sueño. Estos datos son fundamentales para nuestra aplicación móvil, ya que nos permiten brindar a los usuarios información valiosa sobre su salud y bienestar.

Fitbit ofrece una sólida documentación y recursos de desarrollo que hacen que sea más fácil para los desarrolladores comprender y utilizar sus tecnologías. La documentación clara y bien estructurada nos permitió agilizar el proceso de integración de los dispositivos Fitbit en nuestra aplicación. Esto nos permitió ahorrar tiempo y recursos, y nos dio la confianza de que estábamos trabajando con una marca confiable y orientada al desarrollo.

Otro aspecto convincente de Fitbit es su sólida reputación y experiencia en el mercado de wearables. Como una marca establecida y confiable, Fitbit ha demostrado su compromiso con la calidad, la innovación y la satisfacción del cliente a lo largo de los años. Esto nos brinda confianza en la precisión y confiabilidad de los datos proporcionados por sus dispositivos, lo que es fundamental para la efectividad de nuestra aplicación móvil.

5. Desarrollo e implementación

Dentro de este apartado se detallan las tecnologías usadas en el desarrollo de la aplicación móvil, así como la estructura, buenas prácticas y obtención de datos provenientes del wearable Fitbit.

5.1. Tecnologías

A continuación se detallan las tecnologías usadas:

1. Backend api rest

Software

Es un conjunto de eventos que realiza el recurso humano mediante una interfaz gráfica donde se aplican reglas informáticas para ejecutar una acción específica, llámese acción específica en informática a poder guardar, modificar y eliminar datos provenientes de una entidad que almacena datos, ya sea una base de datos o un manejador de archivos. [20] En software se escribe código, dependiendo del lenguaje de programación usado, ejemplos de software son Microsoft Word, Excel, Power Point, Google Chrome, Photoshop, MySQL, etc. [21]

API REST

Una API, o interfaz de programación de aplicaciones, es un conjunto de reglas que definen cómo las aplicaciones o los dispositivos pueden conectarse y comunicarse entre sí. Una API REST es una API que cumple los principios de diseño del estilo de arquitectura REST o transferencia de estado representacional. [22]

Backend

El backend es la parte invisible pero esencial de un sitio, encargada de manejar la lógica y el procesamiento de datos necesarios para que todo funcione de manera correcta y segura. [23]

Servidor cloud

Es un servicio por suscripción que permite obtener espacio de almacenamiento en red y recursos informáticos. [24]

Laravel

Es un framework de aplicaciones web con una sintaxis expresiva y elegante, utiliza el lenguaje de programación PHP. Proporciona una estructura robusta y elegante para crear aplicaciones web de manera eficiente. Además ofrece características como enrutamiento, migraciones de base de datos, autenticación de usuarios y manejo de sesiones, entre otras, que agilizan el desarrollo y mejoran la seguridad de las aplicaciones web, además es ampliamente utilizado para desarrollar backend de API REST. Proporciona un conjunto de herramientas y características que facilitan la creación y el mantenimiento de API RESTful de alta calidad. [25]

Laravel ofrece enrutamiento flexible y controladores que permiten definir fácilmente las rutas y acciones para las diferentes operaciones de la API, como obtener, crear, actualizar y eliminar recursos. Además, Laravel proporciona una capa de abstracción de base de datos a través de su ORM (Object-Relational Mapping) llamado Eloquent, lo que simplifica la interacción con la base de datos y permite realizar operaciones CRUD de manera más eficiente. [26]

Php

Es un lenguaje de programación de código abierto ampliamente utilizado para el desarrollo de aplicaciones web. PHP se ejecuta en el lado del servidor y permite la creación de aplicaciones web dinámicas e interactivas. Proporciona una amplia gama de funciones y bibliotecas que facilitan tareas comunes, como el manejo de formularios, la manipulación de bases de datos y la generación de contenido dinámico. [27]

LAMP

Es un conjunto de tecnologías de código abierto se está abriendo camino en el desarrollo de software corporativo: la pila LAMP, que incluye el sistema operativo Linux, el servidor web Apache, la base de datos MySQL y los lenguajes de programación Perl, PHP y Python. LAMP ofrece costes más bajos, mayor flexibilidad y mejoras más rápidas en las tecnologías que lo componen. [28]

Con esta tecnología se puede alojar y ejecutar aplicaciones web, api rest, ya que proporciona un entorno de desarrollo sólido y escalable.

MVC

Es un patrón de diseño que considera dividir una aplicación en tres módulos claramente identificables y con funcionalidad bien definida: El Modelo, las Vistas y el Controlador. [29]

Amazon web services lightsail es un proveedor de servidores privados virtuales (VPS) y es la forma más sencilla de comenzar con AWS para desarrolladores, pequeñas empresas, estudiantes y otros usuarios que necesitan una solución para crear y alojar sus aplicaciones en la

nube. Lightsail proporciona a los desarrolladores capacidad de cómputo, almacenamiento y redes, así como capacidades para implementar y administrar sitios web y aplicaciones web en la nube. [30]

Haciendo uso de estas tecnologías ha sido desarrollado el api rest en el framework laravel, usando como lenguaje de programación php, como patron de diseño MVC, por la seguridad, fiabilidad y modularidad que ofrece el framework ademas por su rapidéz para desplegar aplicaciones tipo api rest y web.

MySQL

Es una base de datos es una colección ordenada de datos, que normalmente se almacena en uno o varios archivos asociados. Los datos se estructuran en forma de tablas, en las que es posible hacer referencias cruzadas entre tablas. entre ellas. La existencia de tales relaciones entre las tablas hace que la base de datos base de datos relacional. [31]

Tambien se considera una base de datos a un conjunto de datos almacenados en memoria externa que están organizados mediante una estructura de datos. Una base de datos se puede percibir como un gran almacén de datos que se define y se crea una sola vez, y que se utiliza al mismo tiempo por distintos usuarios. En una base de datos todos los datos se integran con una mínima cantidad de duplicidad. De este modo, la base de datos no pertenece a un solo departamento sino que se comparte por toda la organización [32]

Cronjob

Es una tarea que se ejecuta de manera automatizada en los sistemas operativos. Para la ejecución de este trabajo, el sistema emite órdenes de manera regular. Puede tratarse de una sola tarea, pero normalmente se trata de varias tareas complejas, es decir, de varios cronjobs. La gran ventaja es que los cronjobs presentan un gran potencial de individualización. [33]

2. Backend app

Android studio

Este entorno de desarrollo fue presentado en el año 2013 en el Google I/O y paso a tomar la batuta como la herramienta, este IDE fue desarrollado por Google, compañía propietaria de Android asi que es nombrado el IDE oficial de desarrollo para aplicaciones Android. Esta basado en IntelliJ asi que cuenta con diferencias notables en comparación con Eclipse, esto no hace ni mas dificil ni mas sencillo el desarrollo de una aplicación, en realidad cambian algunas cosas como la compilacion, use de librerias o paquetes externos, pero el codigo de una aplicaciOn en su mayoria funciona de la misma firma sin importar el IDE que se este utilizando. [34]

Kotlin

Es un lenguaje de programación creado en 2010 por JetBrains, Kotlin es una alternativa a Java, que suple varios de los problemas más habituales que los programadores nos encontramos en dicho lenguaje. Además, desde que tuviera lugar en el evento Google I/O de 2017, Kotlin es un lenguaje oficial para el desarrollo en Android, por lo que su uso se ha disparado y actualmente todas las empresas buscan desarrolladores en Kotlin. Varias de las características que hacen de Kotlin un lenguaje superior a Java son:

- a) Seguro contra nulos: uno de los mayores problemas de usar Java son los `NullPointerException`. Esto ocasiona una gran cantidad de problemas a la hora de desarrollar. Con Kotlin nos olvidaremos de esto, pues nos obliga a tener en cuenta los posibles nulos.
- b) Ahorra código: con Kotlin podemos evitar muchísimas líneas de código en comparación con otros lenguajes. Imagina tener un POJO (Plain Old Java Objects) en una sola línea en vez de 50-100.
- c) Características de programación funcional: Kotlin está desarrollado para que trabajemos tanto de modo orientado a objetos, como en modo funcional (e incluso mezclarlos), lo que nos da mucha más libertad y la posibilidad de usar características como higher-order functions, function types y lambdas.
- d) Fácil de usar: al estar inspirado en lenguajes ya existentes, como Java, C# o Scala, la curva de aprendizaje nos será bastante sencilla.

[35]

Android

Es un sistema operativo inicialmente pensado para teléfonos móviles y tablets, hasta que poco a poco fue dominando mercados y ahora lo podemos encontrar en televisores, relojes, e incluso coches. La gran ventaja de Android es que está basado en Linux, un sistema operativo libre y multiplataforma. Otra de sus ventajas es la facilidad para desarrollar aplicaciones y subirlas a Play Store, con la posibilidad de generar beneficios con ellas. Actualmente, Google ha ido actualizando el sistema operativo con mucha rapidez, arreglando problemas y agregando nuevas funcionalidades. [35]

Modelo-Vista-Vista-Modelo (MVVM)

Es un patrón de diseño popular en el desarrollo de aplicaciones móviles. El Modelo representa los datos y la lógica de negocio de la aplicación, la Vista se encarga de mostrar los datos al usuario y el ViewModel sirve de enlace entre el Modelo y la Vista. Además, el patrón MVVM incorpora la vinculación de datos para sincronizar automáticamente los datos entre el modelo y la vista. Esto permite que la vista se actualice automáticamente cuando cambia el modelo.

[36]

3. Patrón repositorio

El patrón de repositorio es un patrón de diseño que aísla la capa de datos del resto de la app. La capa de datos hace referencia a la parte de tu app, independiente de la IU, que controla los datos y la lógica empresarial de la app, lo que expone API coherentes de modo que el resto de la app acceda a esos datos. Mientras que la IU presenta información al usuario, la capa de datos incluye elementos como el código de red, las bases de datos de Room, el manejo de errores y cualquier código que lea o manipule datos. [37]

4. Inyección de dependencias

La inyección de dependencias es una de las técnicas de ingeniería de software que mejora el mantenimiento de una aplicación de software mediante la gestión de los componentes dependientes, con esto se consigue mejorar la refactorización, mantenibilidad, mejoras en el código. [38]

5. Retrofit

Es una librería que permite realizar peticiones HTTP a una api rest.

6. Firebase cloud messaging

Firebase Cloud Messaging (FCM) es una solución de mensajería multiplataforma que permite enviar mensajes de forma fiable y gratuita. [39]

7. Frontend

Frontend es la parte de un programa, sitio web o dispositivo a la que un usuario puede acceder directamente. En el contexto de diseño web y desarrollo web, se refiere a todas las tecnologías que corren en el navegador y que se encargan de la interactividad con los usuarios. [23]

8. Frontend web

HTML

Lenguaje de marcado de hipertexto (HTML) es el conjunto de símbolos o códigos de marcado que se insertan en un archivo destinado a mostrarse en Internet. El marcado indica a los navegadores cómo mostrar las palabras e imágenes de una página web.

Cada fragmento individual de código de marcado (que se situaría entre los caracteres '<' y '>') se denomina elemento, aunque mucha gente también se refiere a él como etiqueta. Algunos elementos vienen en pares que indican cuándo debe empezar y cuándo debe terminar un efecto de visualización. [40]

9. Frontend app android

XML

Significa lenguaje de marcación extensible, que es una forma de describir los datos mediante un documento basado en texto. Debido a que el formato XML es extensible y muy flexible, se utiliza para varias cosas diferentes, incluida la definición del diseño de la IU de las apps para Android. [41]

10. Seguridad

El api rest que esta alojado en la instancia de Amazon Lightsail la cual posee dos firewall(IPv4 - IPv6) que controla el tráfico permitido para lograr conectarse a esta usando la dirección IP pública. Cada firewall contiene un conjunto de reglas que filtran el tráfico que entra en la instancia. Ambos firewalls son independientes entre sí. [42]

11. Arquitectura del proyecto

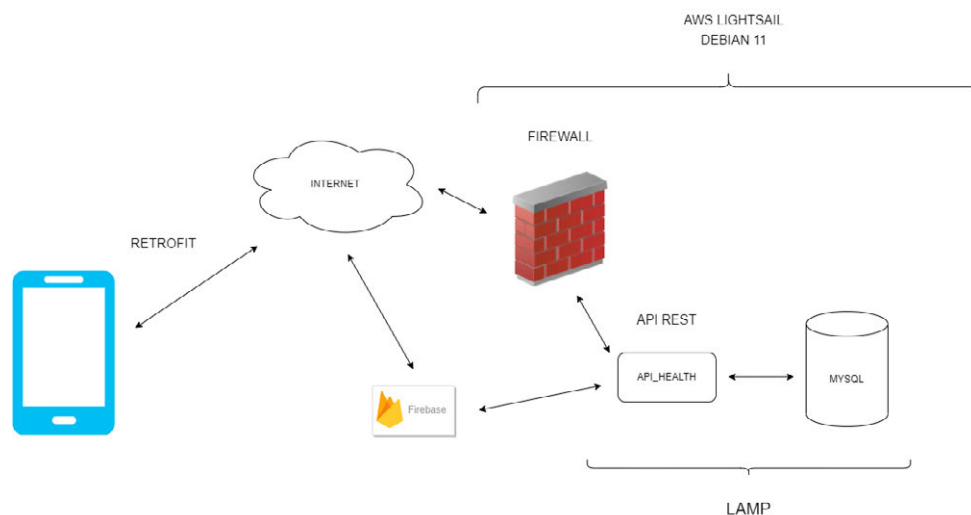


Figura 5.1. Arquitectura del proyecto

Se ha implementado un servidor LAMP en AWS Lightsail, específicamente utilizando Debian 11 Linux como sistema operativo base. Este servidor se ha configurado para alojar el API REST y el sitio web, que permite el intercambio de datos entre la aplicación móvil y el backend. Para facilitar esta comunicación, se ha utilizado la biblioteca Retrofit, que proporciona una interfaz sencilla para realizar peticiones HTTP y procesar las respuestas de manera eficiente, además, para mostrar ubicaciones dentro de un mapa en la aplicación.

Cuando se requiere enviar notificaciones push desde el servidor al dispositivo móvil, se ha utilizado la biblioteca Firebase Messaging. Para que esto sea posible, se ha obtenido un token

de Firebase, que identifica de manera única el dispositivo y permite enviar notificaciones personalizadas y en tiempo real.

Estas tecnologías y herramientas se combinan para proporcionar una aplicación móvil con una funcionalidad robusta y una experiencia de usuario enriquecedora.

12. Estructura de la aplicación

La aplicación móvil ha sido desarrollada con el patrón repositorio y modelo vista vista modelo a continuación se detalla la explicación de cada una de las carpetas de la estructura del proyecto.

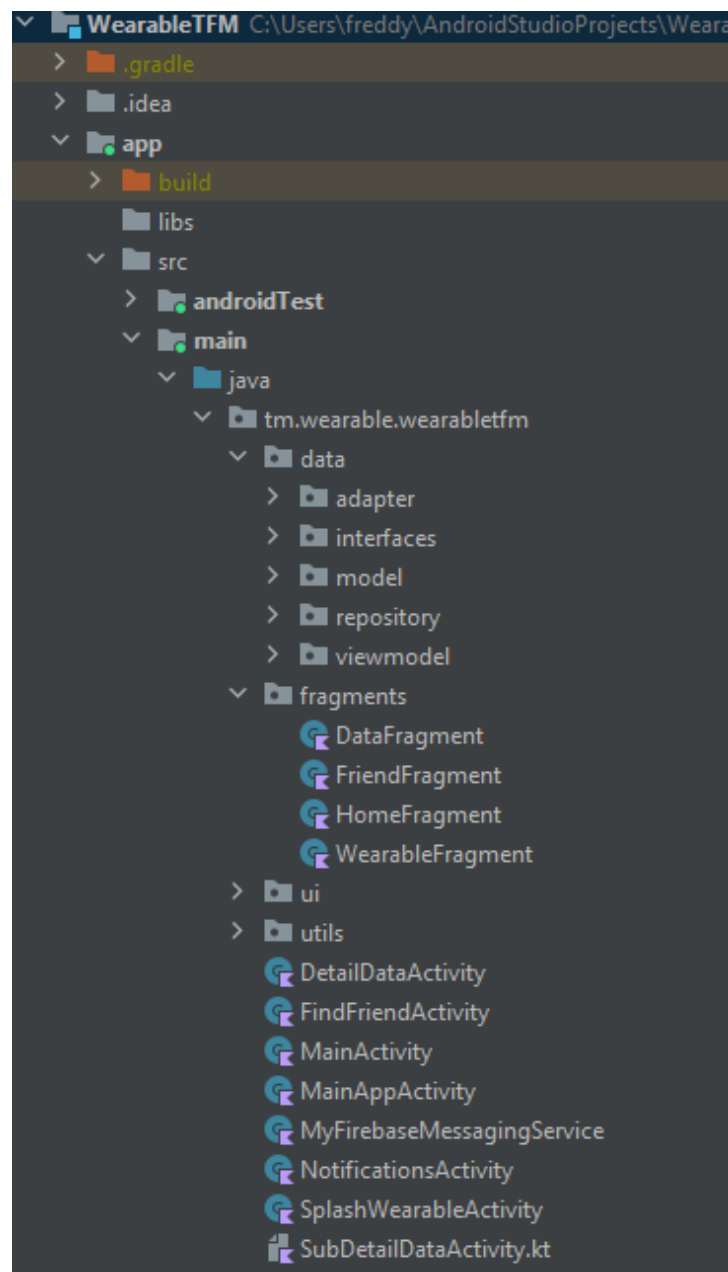


Figura 5.2. Estructura de la aplicación

Carpeta data

Dentro de esta carpeta se encuentran las siguientes sub carpetas que se detallan a continuación:

Carpeta adapter

Dentro de esta carpeta se encuentran las clases que actúan como puente entre la fuente de

datos y la vista. Sirve para mostrar los datos en una vista específica tal como un RecyclerView, además se encarga de gestionar los elementos de la vista y la detección de eventos clic de cada uno de sus elementos.

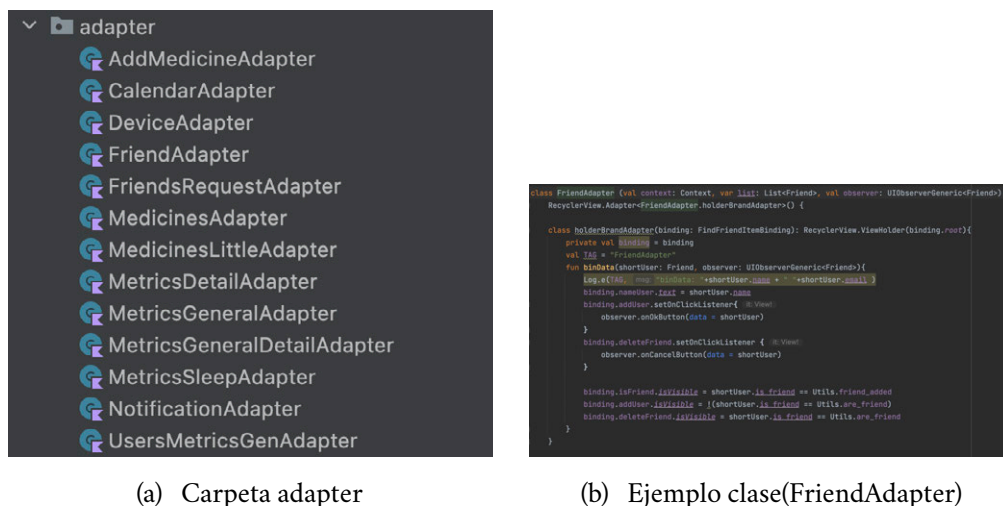


Figura 5.3. Estructura Adapter

Carpeta interfaces

Dentro de esta carpeta se encuentran interfaces las cuales son una declaración de un conjunto de métodos abstractos y que pueden ser implementados por una clase, lo cual hace fácil una comunicación entre la actividad o fragmento y los items del adapter.

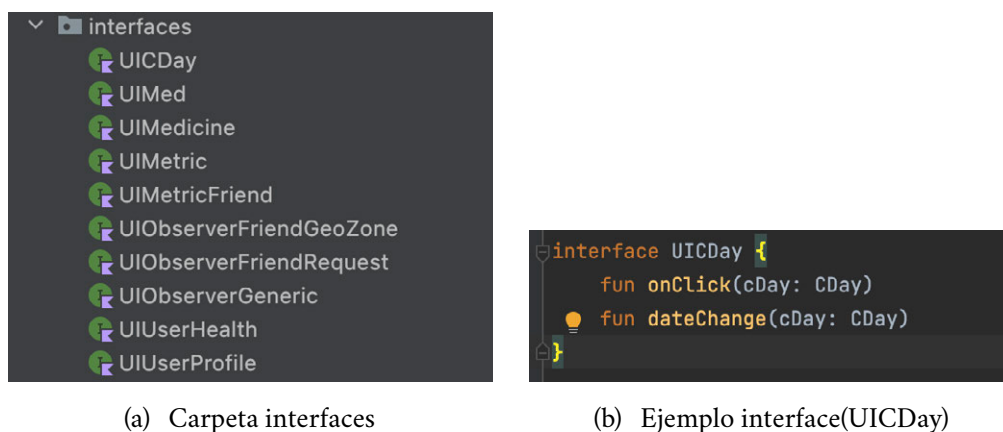
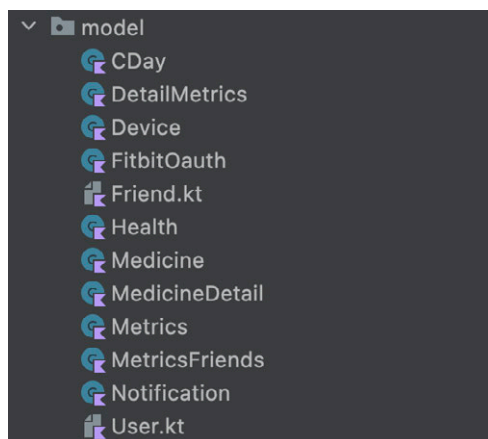


Figura 5.4. Estructura Interface

Carpeta model

Dentro de esta carpeta se encuentran las clases models lo cual se refiere a la representación de datos, estas clases contienen propiedades y características según su funcionalidad, así también como métodos para manejar y acceder a sus datos.



(a) Carpeta model

```
class User {
    var id: Int,
    var name: String,
    var email: String,
    var token_refresh: String? = null,
    var image: String? = null,
    var code_auth: String? = null,
    var token: String,
    var health: Health? = null,
    var password: String = ""
} {
    constructor() : this(0, name = "", email = "", token_refresh = "", image = "", code_auth = "", token = "")
    constructor(jsonObject: JSONObject): this {
        jsonObject.getString("id"),
        jsonObject.getString("name"),
        jsonObject.getString("email"),
        jsonObject.getString("token_refresh"),
        jsonObject.getString("image"),
        jsonObject.getString("code_auth"),
        jsonObject.getString("token")
    }
}
```

(b) Ejemplo clase(User)

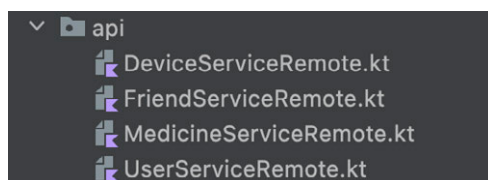
Figura 5.5. Estructura Model

Carpeta repository

Dentro de esta carpeta se encuentran las siguientes subcarpetas:

Carpeta api

Dentro de esta carpeta se encuentran las interfaces que definen un servicio remoto de comunicación hacia una api rest.



(a) Carpeta api

```
interface DeviceServiceRemote {
    @POST("utils.fetch_devices")
    suspend fun fetch_devices(@Body requestBody: Map<String, String>): retrofit12.Response<ListDevicesResponseApi>

    @POST("utils.fetch_metrics_by_user_data")
    suspend fun fetch_metrics_by_user_data(@Body requestBody: Map<String, String>): retrofit12.Response<ListMetrics>

    @POST("utils.fetch_last_metrics_by_user")
    suspend fun fetch_last_metrics_by_user(@Body requestBody: Map<String, String>): retrofit12.Response<ListMetrics>

    @POST("utils.fetch_metrics_by_user_type_data")
    suspend fun fetch_metrics_by_user_type_data(@Body requestBody: Map<String, String>): retrofit12.Response<ListMetrics>

    @POST("utils.fetch_metrics_friends_by_user")
    suspend fun fetch_metrics_friends_by_user(@Body requestBody: Map<String, String>): retrofit12.Response<ListMetrics>

    @POST("utils.fetch_metrics_friends_by_user")
    suspend fun fetch_metrics_friends_by_user(@Body requestBody: Map<String, String>): retrofit12.Response<ListMetrics>
}

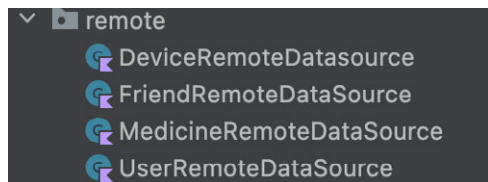
class ListDevicesResponseApi {
    @SerializedName("devices") var devices: ArrayList<Device> = ArrayList()
    @SerializedName("message") var message: String = ""
}
```

(b) Ejemplo interface(DeviceServiceRemote)

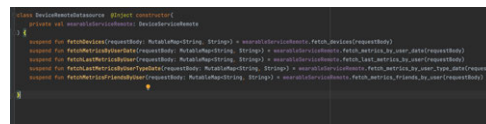
Figura 5.6. Estructura Interface api rest

Carpeta datasource

Dentro de esta carpeta se encuentra la subcarpeta remote donde se definen las funciones de un servicio remoto haciendo uso de la interfaz DeviceServiceRemote



(a) Carpeta remote

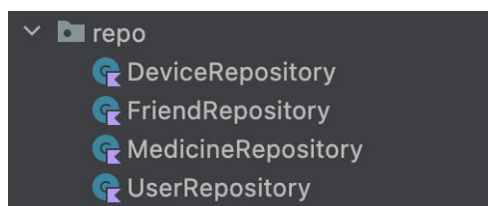


(b) Ejemplo clase se(DeviceRemoteDataSource)

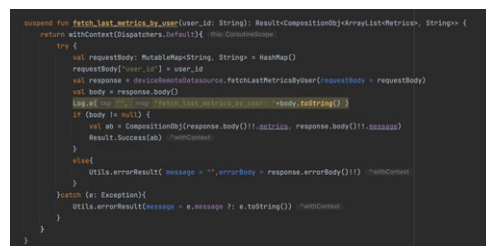
Figura 5.7. Estructura datasource

Carpeta repo

Dentro de esta carpeta se encuentran funciones las cuales utilizan la inyección de dependencias (@Inject) para obtener una instancia de DeviceRemoteDatasource y lograr realizar las implementaciones respectivas.



(a) Carpeta repo



(b) Ejemplo clase(DeviceRepository)

Figura 5.8. Estructura repositorio

Carpeta viewmodel

Dentro de esta carpeta se encuentran las clases que forman parte de la arquitectura de componentes las cuales se encargan de gestionar, persistir y comunicar datos entre la interfaz del usuario y el acceso a cada una de las interfaces de datos.



Figura 5.9. Estructura View Model

Carpeta fragments

Dentro de esta carpeta se encuentran las clases fragments, las cuales son componentes de interfaz de usuario reutilizable que representa una parte modular y autónoma de una actividad. Cada fragmento puede tener su propio diseño y comportamiento diferente.

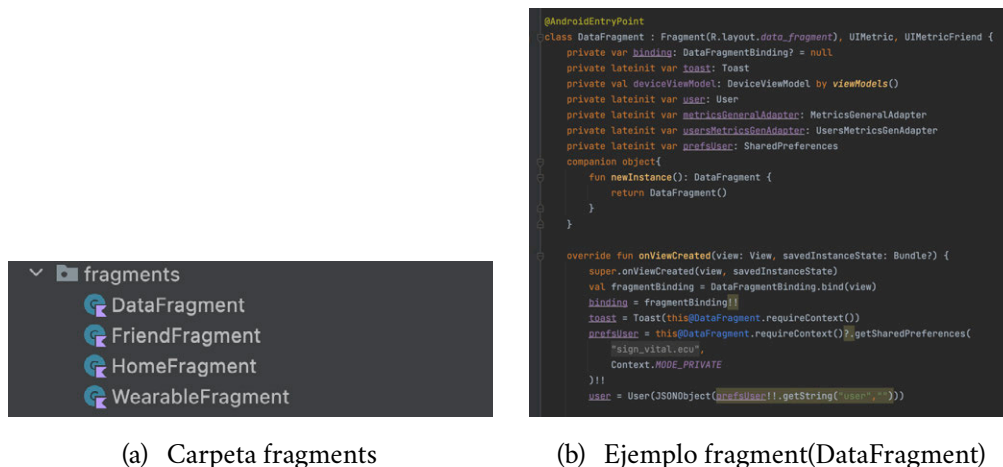


Figura 5.10. Estructura Fragment

Carpeta ui

Dentro de esta carpeta se encuentran las clases actividades, las cuales son componentes para administrar la interfaz de usuario.



Figura 5.11. Estructura Actividades

Carpeta utils

Dentro de esta carpeta de encuentran las funciones utilitarias, constantes tipo endpoint para ser usados en las implementaciones api rest, las cuales son encargadas de contener, cambio de formato de fechas, imágenes correspondientes dependiendo el tipo de dato, validaciones para mostrar resultado correcto o error, obtención de fechas futuras o cercanas.



Figura 5.12. Estructura utilitaria

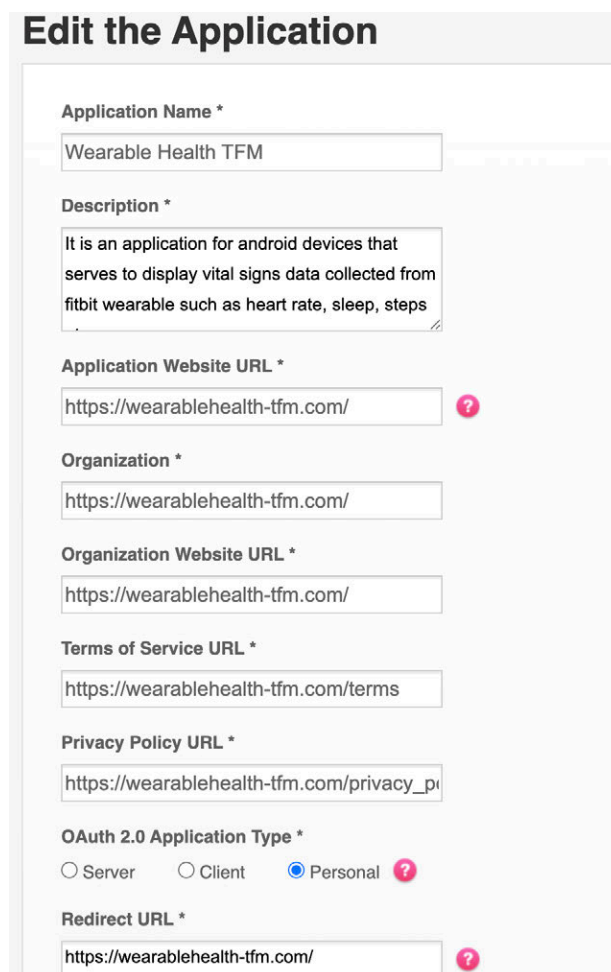
Archivo AppMoule

Dentro de este fichero se encuentran las instancias para crear componentes(inyección de dependencias), las cuales son usadas en todo el codigo de la aplicacion, con esto logramos modularidad, refactorizar, desacoplamiento , mantenibilidad y flexibilidad del codigo.

5.2. Autenticación fitbit

Para lograr la autenticación en Fitbit, se ha realizado lo siguiente:

1. Llenar el formulario de registro aplicación (ver la Figura 5.13).



Edit the Application

Application Name *
Wearable Health TFM

Description *
It is an application for android devices that serves to display vital signs data collected from fitbit wearable such as heart rate, sleep, steps

Application Website URL *
<https://wearablehealth-tfm.com/>

Organization *
<https://wearablehealth-tfm.com/>

Organization Website URL *
<https://wearablehealth-tfm.com/>

Terms of Service URL *
<https://wearablehealth-tfm.com/terms>

Privacy Policy URL *
https://wearablehealth-tfm.com/privacy_pi

OAuth 2.0 Application Type *
☐ Server ☐ Client ☒ Personal

Redirect URL *
<https://wearablehealth-tfm.com/>

Figura 5.13. Formulario registro aplicación

Registro de la aplicación en el siguiente enlace: <https://dev.fitbit.com/apps/new>.

Se creó un sitio web donde se detallan el uso de términos del servicio y política de privacidad para lograr registrar la app correctamente. Términos del servicio en el siguiente enlace: <http://wearablehealth-tfm.com/terms>.

2. Generar el verificador de código y el desafío de código. Un valor criptográfico aleatorio de

entre 43 y 128 caracteres denominado verificador de código. Un hash SHA-256 del verificador de código, codificado en Base64Url con relleno omitido, denominado desafío de código. La transformación del código verificador en el código de comprobación requiere el uso de bibliotecas que implementen las especificaciones de hash SHA-256 y codificación Base64Url. Los caracteres de relleno opcionales de Base64Url (=) deben omitirse. Por ejemplo, si el verificador de código es 01234567890123456789012345678901234567890123456789, el pseudocódigo `base64UrlEncode(sha256Hash(verificador_código))` debería producir un desafío de código de `-4cf-Mzo_qg9-ug0F4QwWhRh4AjcAqNx7SbYVsdmyQM`. [43]

3. Generar el enlace respectivo a mostrar en una pantalla al usuario para que pueda loguearse con su cuenta de Fitbit.
4. Obtener `access_token`, `refresh_token` y `user_id` para obtener datos de signos vitales.

5.3. Obtención datos fitbit

Para lograr obtener los datos de la plataforma Fitbit, se ha realizado lo siguiente:

1. Crear un cronjob que se ejecuta cada diez minutos en el lado del backend.
2. Crear algoritmo para que realice una petición GET correspondiente de cada signo vital a los diferentes endpoint de Fitbit.
3. Parseo de datos para obtener cada uno de sus valores.

6.

Pruebas

SonarQube es una plataforma de análisis estático de código ampliamente utilizada en el desarrollo de software. Proporciona una variedad de herramientas y métricas para evaluar la calidad del código y detectar posibles problemas y vulnerabilidades.

Se ha realizado un test de código con la herramienta de software sonarqube la cual no ha encontrado bugs, vulnerabilidades pero solo menciona que deben realizarse mantenimientos a las actividades, evitando variables que no se utilizan e importación de librerías. [44] A continuación se encuentra en imagen el resultado de análisis de SonarQube a la aplicación móvil:

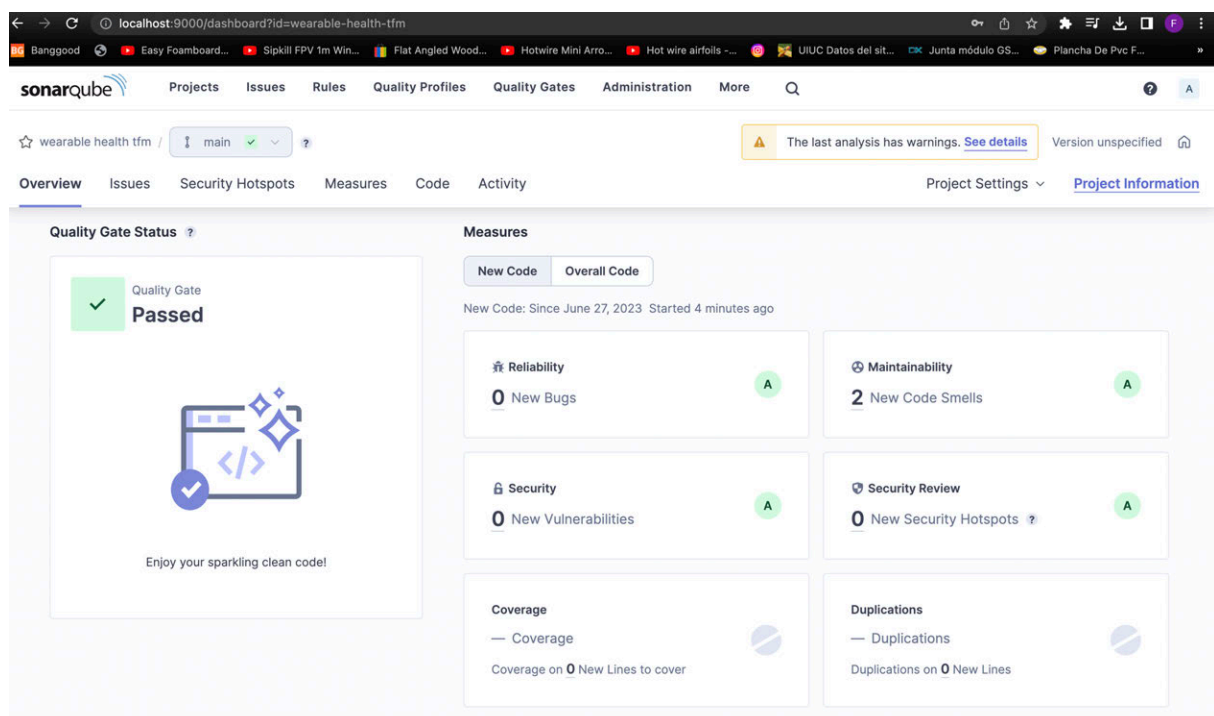


Figura 6.1. Análisis Sonarqube

7.

Análisis del impacto socio-ambiental

El análisis del impacto socio-ambiental de nuestra aplicación móvil es crucial para comprender cómo nuestras funciones y características pueden influir positivamente en la sociedad y el medio ambiente. A continuación, se presenta un resumen sobre el impacto socio-ambiental de nuestra aplicación:

1. **Mejora del bienestar y la salud:** Nuestra aplicación móvil brinda a los usuarios la capacidad de monitorear sus signos vitales y recibir recordatorios de medicamentos. Al promover la atención médica preventiva y el cumplimiento de las prescripciones, estamos contribuyendo a mejorar la salud y el bienestar de los usuarios, lo que a su vez tiene un impacto positivo en su calidad de vida y productividad. Recordemos que nuestro bienestar físico y mental tienen una enorme influencia a fin de desarrollar nuestras actividades eficientemente y hacer frente al estrés y depresión, permitiendo ser útiles a la sociedad.
2. **Fomento de la conciencia y la responsabilidad:** Al permitir que los amigos visualicen el estado de salud de los usuarios, nuestra aplicación promueve la transparencia y la comunicación en las relaciones cercanas. Esto puede fomentar un mayor apoyo social y una mayor conciencia sobre la importancia de cuidar la salud personal y de los demás, proporciona en cierto modo una mutua responsabilidad.
3. **Reducción del desperdicio de medicamentos:** Los recordatorios de medicamentos de nuestra aplicación ayudan a evitar la omisión de dosis y el desperdicio de medicamentos. Esto contribuye a una gestión más eficiente de los recursos médicos y reduce el impacto ambiental asociado con la producción y eliminación de medicamentos no utilizados, si se realiza de modo adecuado y oportuno según la prescripción médica se evitaría no solamente un ahorro económico si no también la contaminación ambiental, pues, generalmente las personas lo que hacen es botar junto con el resto de la basura, ocasionando un daño ambiental, consecuentemente la utilización práctica de este dispositivo es de gran ayuda para la conservación de la naturaleza.

-
4. **Potencial para la investigación médica:** Al recopilar datos de signos vitales de los usuarios, nuestra aplicación puede contribuir a la generación de información y conocimientos valiosos para la investigación médica. Estos datos pueden ayudar a identificar patrones, tendencias y factores de riesgo, lo que puede tener un impacto significativo en el avance de la medicina y el desarrollo de nuevos tratamientos, pueden ser focalizados con mayor exactitud según la investigación médica y posteriormente mediante cruce de información establecer patrones que ayudarían a darle un tratamiento muy eficaz y oportuno.
 5. **Acceso y equidad en la atención médica:** Al ofrecer una herramienta accesible y fácil de usar para monitorear la salud, nuestra aplicación tiene el potencial de cerrar la brecha en el acceso a la atención médica. Esto es especialmente relevante para comunidades remotas o desatendidas que pueden enfrentar dificultades para acceder a servicios médicos de calidad. Al empoderar a los usuarios para que sean conscientes de su estado de salud y tomen medidas preventivas, estamos contribuyendo a la equidad en la atención médica.

La equidad en la salud es un componente muy importante en la justicia social, establece la ausencia de diferencias evitables, injustas o remediabiles entre grupos de personas debido a circunstancias sociales, económicas, demográficas o geográficas, permitiendo la atención médica en la que el caso requiera sin obstáculos, acceso de mujeres, hombres y niños en iguales oportunidades a los servicios de salud, en virtud de la aplicación móvil orientada a éste servicio, es decir de un pronto acceso y atención médica requerida.

8.

Trabajos futuros

Durante el desarrollo de esta aplicación móvil no se han alcanzado a desarrollar puntos muy importantes por lo cual se ha planteado lo siguiente:

- Implementar cifrado de datos.
- Disponibilidad de ampliar el código, para tener una app para iOS.
- Permitir realizar el login usando redes sociales.
- Permitir realizar el login usando reconocimiento facial.
- Aplicar inteligencia artificial para obtener datos del giroscopio del móvil y mediante algún algoritmo de salud poder saber como se ha incrementado o ha disminuido la caminata del usuario, luego hacer recomendaciones para que se mantenga activo físicamente.
- Implementar una arquitectura robusta para manejar gran cantidad de usuarios, desplegar microservicios, desplegar varios cluster de servidores.
- Mantener una base de datos local dentro del móvil para lograr mayor fluidez y rapidez.
- Implementar nuevas marcas wearables, tales como Garmin y Samsung.
- Poder elegir el idioma.
- Generar identificador único (QR: código de barras) para cada usuario para lograr mayor rapidez al momento de enviar la solicitud a un amigo.
- Implementar geolocalización del amigo.
- Integrar plataforma web de fácil acceso a los datos para los médicos de cabecera.
- Integración sdk(kit de desarrollo de software) de Fitbit para no esperar la disponibilidad del acceso a los datos de signos vitales.
- Implementar video llamada permitiendo la comunicación entre el médico de cabecera y el paciente.

- Implementar un apartado de noticias, para que el usuario conozca sobre promociones de wearables, asesoramiento alimenticio y ejercicios para la salud física.

9.

Conclusiones

El presente trabajo permite realizar las siguientes conclusiones:

- Existen aplicaciones para cuantificar y monitorizar determinados ámbitos de la salud, sin embargo éste dispositivo es mucho más amplio porque establece con precisión qué caracteres personalizados requieren información a fin de lograr un control de salud en beneficio del usuario.
- Permite compartir la información con familiares y amigos que el usuario desee involucrarlos, que generalmente son personas de confianza, quienes estarán atentos a las notificaciones, permitiendo tres cosas importantes: Comunicación, Información e Interacción.
- Fomenta una cultura de ser atentos y meticulosos con su salud al tener una herramienta muy útil que le permita contar con la información pertinente y oportuna.
- Garantizar una información muy conveniente del usuario para el análisis y utilización en los medios públicos o privados de los servicios de salud, sean estos hospitales, clínica o dispensarios, ser atendido con prontitud.
- Los medicamentos de la índole que sea serán administrados en los tiempos requeridos y se evitará la contaminación del medio ambiente tanto del agua como del aire.
- El desarrollo de esta herramienta tecnológica es susceptible de ser mejorada con el transcurso del tiempo, en las posibilidades de aumentar los conocimientos, desarrollos y avances tecnológicos, y sobre manera, las necesidades humanas.
- Una herramienta recomendada por su fácil acceso y de gran utilidad en todo sector social sin discriminación.
- Es recomendable utilizar la App de Wearable Health TFM Porque su diseño y programación van de acuerdo con la necesidad personalizada del usuario.

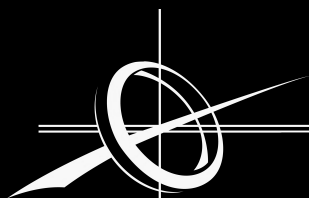
Bibliografía

- [1] D. de cáncer del NCI, «Frecuencia Cardíaca,» dirección: <https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/frecuencia-cardiaca>.
- [2] D. de cáncer del NCI, «Presión arterial,» dirección: <https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/presion-arterial>.
- [3] Cigna, «Temperatura corporal,» dirección: <https://www.cigna.com/es-us/knowledge-center/hw/pruebas-mdicas/temperatura-corporal-hw198785>.
- [4] Yetisen, «Wearables,» 2018. dirección: <https://onlinelibrary.wiley.com/doi/full/10.1002/adma.201706910>.
- [5] Fitbit, «Fitbit,» dirección: <https://www.fitbit.com/global/es/home>.
- [6] Garmin, «Que es garmin,» dirección: <https://www.garmin.com/es-ES/p/125677>.
- [7] Apple, «Que es Apple health,» dirección: <https://support.apple.com/es-es/HT203037>.
- [8] Samsung, «¿Qué es Samsung Health?,» dirección: <https://www.samsung.com/co/support/apps-services/what-is-samsung-health/>.
- [9] G. fit, «Cómo comenzar a usar Google Fit,» dirección: <https://support.google.com/fit/answer/6075067?hl=es-419>.
- [10] K. Schwaber, «Scrum: What Is Scrum,» 2014. dirección: http://www.volaroint.com/wp-content/uploads/dlm_uploads/2014/03/DC-VOLARO-Training-Scrum-What_Is_Scrum.pdf.
- [11] L. Martinez, «Metodología Scrum, ¿Qué es y cómo funciona?,» 2021. dirección: https://performanze.io/metodologia-scrum-que-es-como-funciona/?psafe_param=1&utm_term=&utm_campaign=Din%C3%A1mica+ES&utm_source=adwords&utm_medium=ppc&hsa_acc=5624581664&hsa_cam=12428335676&hsa_grp=117966659003&hsa_ad=501291292710&hsa_src=g&hsa_tgt=dsa-19959388920&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gclid=Cj0KCQjw756lBhDMARIsAEI0Agm3RNpmOgWZmx50KOlCoybTx_cHoDJD4X2z-MovsQ7-W1YDdLN0JIYaAoVuEALw_wcB.

- [12] D. M. REHKOPF, «Metodología Scrum, ¿Qué es y cómo funciona?,» dirección: <https://www.atlassian.com/es/agile/scrum/sprints#:~:text=son%20los%20sprints%3F-,Un%20sprint%20es%20un%20per%C3%ADodo%20breve%20de%20tiempo%20fijo%20en,con%20menos%20quebraderos%20de%20cabeza..>
- [13] A. Pérez, «Las 5 etapas en los “Sprints” de un desarrollo Scrum,» dirección: <https://www.globalbit.co/2020/05/22/que-son-y-como-funcionan-los-sprints-de-la-metodologia-scrum/>.
- [14] Github, «Github,» dirección: <https://docs.github.com/en/get-started/quickstart/hello-world>.
- [15] Figma, «What is Figma?,» dirección: <https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma->.
- [16] Fitbit, «Inspire 3,» dirección: <https://www.fitbit.com/global/us/products/trackers/inspire3?sku=424BKBK>.
- [17] E. diario, «La importancia crucial del sueño profundo,» dirección: https://www.eldiario.es/consumoclaro/tu-mejor-yo/importancia-crucial-sueno-profundo_1_6466331.html.
- [18] A. A. Putilov, O. G. Donskaya y E. G. Verevkin, «Sleep spindles, K-complexes, limb movements and sleep stage proportions may be associated with cognitive performance, brain morphometry, and psychological variables,» *Neuroscience letters*, 2020.
- [19] Sonodina, «¿Qué es la fase REM del sueño? Dirección: <https://sonodina.es/que-es-la-fase-rem-del-sueno/>.
- [20] G. Global, «¿Qué es hardware y software?,» dirección: <https://edu.gcfglobal.org/es/informatica-basica/que-es-hardware-y-software/1/>.
- [21] Sanjoy, «Difference between Hardware and Software,» dirección: <https://www.geeksforgeeks.org/difference-between-hardware-and-software/>.
- [22] RedHat, «¿Qué es una API de REST?,» dirección: <https://www.redhat.com/es/topics/api/what-is-a-rest-api>.
- [23] Maldeadora, «Qué es Frontend y Backend: características, diferencias y ejemplos,» 2018. dirección: <https://platzi.com/blog/que-es-frontend-y-backend/#:~:text=El%20frontend%20se%20refiere%20a,utilizado%20por%20el%20sitio%20web..>
- [24] A. Huth y J. Cebula, «The Basics of Cloud Computing,» dirección: <http://findnerd.s3.amazonaws.com/data/152759075583.pdf>.
- [25] Laravel, «The PHP Framework for Web Artisans,» dirección: <https://laravel.com/>.

- [26] V. A. Flores, «10 ventajas de Laravel que lo hacen el mejor framework PHP,» 2023. dirección: <https://codersfree.com/posts/ventajas-laravel-mejor-framework-php>.
- [27] Php, «What is PHP?,» dirección: <https://www.php.net/manual/en/intro-what-is.php>.
- [28] IEEE, «LAMP lights enterprise development efforts,» 2005. dirección: <https://ieeexplore.ieee.org/abstract/document/1510562>.
- [29] E. Bascón Pantoja, «Modelo Vista Controlador (MVC) en el desarrollo de aplicaciones de software,» 2004. dirección: http://www.scielo.org.bo/scielo.php?pid=S1683-07892004000100005&script=sci_arttext.
- [30] A. web services, «What is Amazon Lightsail?,» dirección: <https://aws.amazon.com/lightsail/faq/>.
- [31] M. Kofler, «MySQL,» 2009. dirección: https://link.springer.com/chapter/10.1007/978-1-4302-0669-9_1.
- [32] M. Marqués, «Bases de datos,» págs. 1-4, 2011.
- [33] IONOS, «Cronjob: ejecución automática de tareas cron,» 2021. dirección: <https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/cronjob/>.
- [34] J. D. L. Castillo, «Desarrollo de aplicaciones android con Android Studio,» págs. 14-16, 2019.
- [35] A. G. Orozco, «Iniciación a android en kotlin casos prácticos,» págs. 2-3, 2018.
- [36] R. Ferrer García, «MVVM: Model-View-ViewModel,» 2023. dirección: https://link.springer.com/chapter/10.1007/978-1-4842-9069-9_4.
- [37] Android, «Almacenamiento en caché y patrón de repositorio,» dirección: <https://developer.android.com/codelabs/basic-android-kotlin-training-repository-pattern?hl=es-419%5C#3>.
- [38] D. D. Baharestani, «Mastering Ninject for Dependency Injection,» 2013.
- [39] Google, «Firebase Cloud Messaging,» dirección: <https://firebase.google.com/docs/cloud-messaging>.
- [40] A. Hayes, «HyperText Markup Language (HTML): What It Is, How It Works,» 2022. dirección: <https://www.investopedia.com/terms/h/html.asp>.
- [41] Android, «Lee y comprende XML,» dirección: <https://developer.android.com/codelabs/basic-android-kotlin-training-xml-layouts?hl=es-419%5C#2>.
- [42] A. W. Services, «Understanding Firewall and Port Mappings in Amazon Lightsail,» 2022. dirección: https://lightsail.aws.amazon.com/ls/docs/es_es/articles/understanding-firewall-and-port-mappings-in-amazon-lightsail.

-
- [43] Fitbit, «Authorizacion,» dirección: <https://dev.fitbit.com/build/reference/web-api/developer-guide/authorization/>.
- [44] Sonarqube, «Sonarqube,» dirección: <https://www.sonarsource.com/products/sonarqube/>.



Universidad
Politécnica
de Madrid

**ETSI SISTEMAS
INFORMÁTICOS**