Desarrollo de una aplicación y una plataforma web con enfoque en telemedicina, encargadas de la captura, el monitoreo, la administración y el envío de información de pacientes.

Angie Loreina García Delgado y Andrés Felipe Cabeza Serrano

Trabajo de Grado para Optar al Título de Ingeniería de Sistemas

Director

Fernando Antonio Rojas Morales

Maestría en Ciencias Computacionales

Tutor

Francisco Javier González Silva

Ingeniería de Sistemas

Universidad Industrial de Santander

Facultad de Ingenierías Físico-mecánicas

Escuela de Ingeniería de Sistemas e Informática

Ingeniería de Sistemas

Bucaramanga

2022

Dedicatoria

A mis padres, Rodolfo y Sandra por ser el apoyo durante el curso de mi carrera profesional, su comprensión, amor, y empatía me ayudó a cumplir mis sueños.

A mis abuelos ya que fueron como unos segundos padres, y siempre estuvieron allí cuando requerí, a aumentar mi confianza, brindar apoyo y los demás integrantes de mi familia, ya que sin ellos nada hubiese sido posible.

Gracias a mi compañera Angie Loreina, ya que fue un apoyo para sacar este proyecto adelante y una gran compañera para el intenso trabajo en equipo que involucró el desarrollo del proyecto, una gran compañera, líder y resiliente en los momentos difíciles, que me ayudó a salir adelante cuando lo necesité.

A mis compañeros y docentes de la universidad porque gracias a ellos se fue forjando el carácter de profesional para salir al mundo laboral.

Andrés Felipe Cabeza Serrano

A Dios, sin su voluntad nada de esto sería posible.

A mi nonito Hernán García, no hay palabras que expresen lo mucho que lo extraño. Desde el cielo es mi inspiración y sé que estaría muy orgulloso de mis logros.

A mis padres Patricia Delgado y Cristóbal García, su amor y apoyo fueron incondicionales, confiaron siempre en mis capacidades y me llenaron de ganas de salir adelante, les debo todo lo que soy.

A mis abuelos, Carmen Villamizar, Eugenia Valbuena y Ricardo Delgado, todo su cariño y palabras de apoyo me llenaron de motivación en los momentos más difíciles.

A toda mi familia, mis hermanos, tíos y primos. Cada uno de ellos aportó de una forma u otra en mi proceso de formación, en especial mi tío Ángel Delgado y su familia. Tuvieron fe en mí y mis logros, comprendieron mis ausencias y siempre me tendieron la mano.

A Pablo Serrano por toda su comprensión, cariño, apoyo y por motivarme cada día a ser mejor.

A mis amigos de colegio y universidad, todas las experiencias que compartimos también fueron muy importantes para finalizar esta etapa.

A mi compañero Felipe Cabeza por ser parte del proceso de sacar adelante este proyecto final. Su apoyo, disciplina y resiliencia fueron fundamentales.

Angie Loreina García Delgado

Agradecimientos

Gracias a la Universidad Industrial de Santander, alma mater que nos acogió y fue como un segundo hogar todos estos años de carrera universitaria, dejando enseñanzas que trascienden de lo solo académico, formó profesionales que contribuyen a la sociedad.

Gracias a nuestro director de proyecto Fernando Antonio Rojas por el apoyo durante todo el proyecto, su orientación y experiencia fueron fundamentales.

Gracias a Juan Carlos Abaunza por la oportunidad de realizar nuestra practica en su empresa y por todas las enseñanzas adquiridas en el proceso, tanto profesionales como personales. Más que un jefe es un líder.

Gracias a todo el equipo de Whatoko Health, compañeros de trabajo, nuestro líder técnico y tutor Francisco González por toda su paciencia y orientación, a los demás colaboradores por ser promotores de aprendizaje, ser un apoyo y proporcionar los recursos para el desarrollo del proyecto.

Gracias a la empresa A&A Soluciones -TIC por darnos la oportunidad de pertenecer a su equipo y hacer parte del desarrollo de Whatoko Health.

Tabla de Contenido

Pág.

Lista de Tablas

Pág.

Lista de Figuras

Pág.

Resumen

Título: Desarrollo de una aplicación y una plataforma web con enfoque en telemedicina, encargadas de la captura, el monitoreo, la administración y el envío de datos e información de pacientes[*]

Autor: Andrés Felipe Cabeza Serrano, Angie Loreina García Delgado[**]

Palabras Clave: Telemedicina, Hipertensión, Diabetes, Whatoko, Desarrollo web, Aplicación.

Descripción: Según estadísticas de la organización panamericana de la salud y de la organización mundial de la salud, entre el 20 y el 40% de la población adulta en América latina sufre de hipertensión y solamente entre el 25 y el 30% lleva un control adecuado de este problema. Además, aproximadamente sesenta y dos millones de personas en América sufren diabetes, un total de 422 millones de personas en todo el mundo. La diabetes fuera de metas o mal controlada puede aumentar la probabilidad de complicaciones como daño ocular, enfermedad renal, enfermedades cardiovasculares, neuropatías, trombosis cerebrales, que pueden desencadenar en mortalidad prematura.

La empresa A&A Soluciones TIC en su labor colaborativa y con base en su experiencia en desarrollo de proyectos, da la oportunidad a practicantes de la Universidad Industrial de Santander de vincularse a problemáticas del mundo real, donde se puedan aplicar los conocimientos adquiridos, por tanto, permite la participación en el desarrollo de un producto llamado Whatoko Health basado en telemedicina, que permita atenuar la problemática mencionada anteriormente, facilitando la atención oportuna a los pacientes a través de un seguimiento continuo de su salud, por medio de conexiones a dispositivos wearables y terceros, procesando y generando valor en los datos suministrados, apoyado con otros módulos, para generar visualización de predicciones sobre un grupo de interés.

---

[*] Trabajo de Grado

[**] Facultad de Ingenierías Físico-mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Fernando Antonio Rojas Morales. Maestría en Ciencias Computacionales. Tutor: Francisco Javier González Silva. Ingeniería de Sistemas.

Abstract

Title: Development of an application and a web platform focused on telemedicine, to capture, track, manage and send patient data and information[*]

Author(s): Andrés Felipe Cabeza Serrano, Angie Loreina García Delgado[**]

Key Words: Telemedicine, Hypertension, Diabetes, Whatoko, Web Development, Application.

Description: According to statistics from the Pan American Health Organization and the World Health Organization, between 20 and 40% of the adult population in Latin America suffers from hypertension and only between 25 and 30% have adequate control of this condition. In addition, approximately sixty-two million people in America have diabetes, a total of 422 million people worldwide. Diabetes poorly controlled can increase the probability of complications such as eye injury, kidney disease, cardiovascular diseases, neuropathies, cerebral thrombosis, which can trigger on premature mortality.
The company A&A Soluciones TIC based on its collaborative work and on its experience in project development, gives students from the Universidad Industrial de Santander the opportunity to be linked to real-world problems, where the knowledge acquired can be applied, therefore, this company allows students to participate on a product called Whatoko Health based on telemedicine, which mitigates the problem mentioned above, facilitating timely care for patients through continuous monitoring of their health, through connections to wearable devices and third party providers, processing and adding value with the data provided, supported with other modules and generating a visualization of predictions on interest group.

[*] Degree Work

[**]Faculty of Physical-Mechanical Engineering. School of Systems Engineering and Informatics. Director: Fernando Antonio Rojas Morales. Master of Computer Science. Tutor: Francisco Javier González Silva. Systems Engineer.

Introducción

Las TIC (Tecnologías de información y comunicaciones) han incidido de manera contundente en el diario vivir, influyendo en las actividades del ser humano y facilitando la atención a ciertas necesidades, unido a esto, la democratización de smartphones y tablets, junto con la mejora de recursos de hardware ha conllevado a la proliferación de aplicaciones de distinta índole. Según (AppBrain, 2022) actualmente Google Play cuenta con un poco más de 2.5 millones de aplicaciones y sólo el 3.90% son de la categoría HealthCare lo que es desafortunado teniendo en cuenta la cantidad de problemas en el ámbito de salud que existen hoy en día como, por ejemplo, la diabetes y la hipertensión.

La organización panamericana de la salud informó que entre el 20 y el 40% de la población adulta en América latina sufre de hipertensión, principal causa de muerte por enfermedades cardiovasculares, además, un gran porcentaje de esta población no lleva un control adecuado (OPS/OMS, s.f.). “Aproximadamente 62 millones de personas en América sufren de diabetes, la diabetes sin un control apropiado puede aumentar las posibilidades de daño ocular, enfermedad renal, enfermedades cardiovasculares, neuropatías, trombosis cerebrales, que pueden desencadenar en mortalidad prematura” (OPS/OMS, s.f.).

En este proyecto se van a indagar diferentes maneras tecnológicas enfocadas en telemedicina para facilitar el monitoreo y atención de pacientes. Es importante tener una noción de algunas plataformas existentes hoy en día para monitorear a pacientes hipertensos y/o diabéticos, por ejemplo, aplicaciones como SocialDiabetes, GluQUO, Bant, son especiales para pacientes diabéticos con funcionalidades como conexión a Google Fit, ver el historial de los datos propiciados por los dispositivos wearables en gráficas, etc. (GaeaPeople, 2020). Por otro lado,

Instant Heart Rate es una aplicación que sirve como aliado para las personas hipertensas, ya que les facilita llevar un registro de sus datos, por ejemplo, el monitoreo de la frecuencia cardiaca (Apps para controlar la hipertensión arterial, 2019).

Entre el 40 y 60% de las personas con hipertensión tienen probabilidad de desarrollar diabetes, poseer ambas enfermedades multiplica la probabilidad de sufrir una insuficiencia cardiaca (Eusko Jaurlaritza, 2021). Por tanto, se quiere abordar una solución que consiste en una aplicación y una plataforma web, que fusionen el análisis de la problemática de la diabetes y la hipertensión, así mismo que se generen alertas en caso de encontrar anormalidades con el fin de facilitar una atención oportuna a los pacientes a través de un seguimiento continuo de su salud.

1. Planteamiento y justificación del problema

La hipertensión arterial (HTA) es la afección crónica más frecuente en la población adulta en el planeta; se comporta como factor de riesgo para padecer las enfermedades que se encuentran entre las más importantes causas de muerte en los países desarrollados y en la mayor parte de los países en vía de desarrollo, como son la cardiopatía isquémica, los accidentes cerebrovasculares, la insuficiencia cardiaca y la enfermedad renal crónica. La participación de la HTA en el desarrollo de estas afecciones se acrecienta notablemente cuando coexiste con otros factores de riesgo cardiovascular como la dislipidemia, el tabaquismo, la obesidad, el sedentarismo o la diabetes. El papel de la HTA en estas situaciones es de mayor trascendencia cuando no se lleva un control adecuado. En el XXXVI Congreso Argentino de Cardiología, FAC-2007, se señala que en el mundo se producen 5,1 millones de muertes al año por enfermedades cardiovasculares y que de ellas 62% están vinculadas al control subóptimo de la presión arterial y allí mismo se dio a llamar

la HTA como "una pandemia sin control". Mientras mejores resultados se obtengan en el control de la HTA menor será el riesgo cardiovascular del paciente y de la comunidad en su conjunto (Pérez y otros, 2011).

En el mundo hay aproximadamente 371 millones de casos de diabetes donde el 9% pertenecen a Latinoamérica, y el 40% de pacientes ignoran su condición. En Colombia existen 2,671,400 casos entre adultos entre 20-79 años (Carlos, 2019). Es una enfermedad que va en ascenso ya que en la actualidad hay más de 199 millones de mujeres viviendo con diabetes, y se calcula que este total aumentará hasta los 313 millones para el 2040. Además, la prevalencia mundial de la enfermedad en adultos mayores de 18 años se incrementó del 4,7 % en 1980 al 8,5 % en 2014 y se acrecentó más rápido en países de ingresos bajos y medianos en la última década, afirma el informe mundial sobre la diabetes de la OMS (Sáenz, 2017).

La HTA es una comorbilidad extremadamente frecuente en los diabéticos, afectando entre 20 y 60% de la población con diabetes. La prevalencia de HTA en la población diabética es de 1,5 a 3 veces superior que en no diabéticos. La HTA contribuye en el desarrollo y la progresión de las complicaciones crónicas de la diabetes, es decir hay una correlación entre estas dos enfermedades (Araya-Orozco, 2004).

En la diabetes y en la HTA, los pacientes no suelen tener un control adecuado de su salud, por ejemplo, “el jefe de Prestaciones Médicas del IMSS, refirió que sólo el 40% de los diabéticos mantiene la enfermedad controlada, es decir el 60% minimiza el padecimiento” (Hipertensión y diabetes: Descuido fatal, 2018).

Ahora del lado de la HTA, se estima que en el mundo hay 1280 millones de adultos de 30 a 79 años con HTA, el 46% de los adultos con HTA desconocen que padecen esta afección y

solamente se trata a menos de la mitad de los adultos que la presentan (Organización Mundial de la Salud, 2021).

La diabetes sin controlar puede abarcar pérdida de la vista, la capacidad renal, imputación de una extremidad, estado de incapacidad o la muerte (Hipertensión y diabetes: Descuido fatal, 2018). Y para "la hipertensión puede producir daños cardiacos graves. El exceso de presión puede endurecer las arterias, con lo que se reducirá el flujo de sangre y oxígeno que llega al corazón" (Organización Mundial de la Salud, 2021).

En resumen, la HTA y la diabetes son enfermedades frecuentes que están muy correlacionadas y de las que el autocuidado de los pacientes en un porcentaje bastante considerable no es el óptimo, y al no serlo, puede tener consecuencias muy graves.

La empresa A&A SOLUCIONES - TIC en su labor colaborativa quiere abordar esta problemática con un producto llamado Whatoko Health y brindar la oportunidad a practicantes de la escuela de ingeniería de sistemas de la Universidad Industrial de Santander de participar en el desarrollo de este producto. Aprovechando de una mejor manera los datos suministrados por dispositivos wearables y terceros, propendiendo apoyar la gestión y el control oportuno de pacientes hipertensos y/o diabéticos con la captura, monitoreo, administración y envío de datos e información.

Se va a realizar captura de datos en la aplicación móvil, conectando directamente un dispositivo wearable o por medio de una API, los datos capturados en la aplicación varían según las mediciones que se le entreguen, es decir, que por ejemplo la API solo proporcione datos de actividad física y no de composición corporal.

La plataforma web a su vez recibirá los datos enviados por la aplicación, en esta se administrarán los datos y se conectarán con otros módulos de Whatoko Health que realizarán el

respectivo formateo, procesamiento estadístico y de inteligencia artificial, proporcionando unas reglas para determinar las anomalías de la enfermedad (cabe aclarar que este procesamiento estadístico no se abordará en la presente práctica).

Por último, en la plataforma web tendrá una visualización de la información por medio de un dashboard orientado al profesional de la salud para encontrar apoyo en su diagnóstico y en la aplicación una visualización del historial de los datos.

2. Objetivos

2.1 Objetivo General

Desarrollar un aplicativo móvil y una plataforma web que apoyen la gestión y el control de pacientes hipertensos y/o diabéticos, a través de la captura de datos del paciente en tiempo real e incluyendo un procesamiento, con el fin de mostrar alertas y asistir de manera oportuna al paciente.

2.2 Objetivos Específicos

Desarrollar una aplicación para uso del paciente, que con su autorización capture las mediciones de los dispositivos wearables conectados o información registrada de forma manual tales como, calorías quemadas, número de pasos, actividad física y distancia recorrida.

Recibir información médica del paciente a través de un formulario.

Establecer la ubicación en tiempo real del paciente para una atención oportuna en caso de emergencia.

Realizar conexión de la aplicación móvil con la plataforma web para transferencias de métricas de pacientes en ambas vías.

Gestionar el procesamiento de los datos e información suministrados por el paciente, mediante la conexión y el consumo de otros módulos del producto (API Gateway y motor de inteligencia artificial) por medio de servicios web.

Generar alertas basadas en la información suministrada por terceros, (motor de inteligencia artificial) para notificar posibles anormalidades en la condición del paciente.

3. Marco referencial

3.1 Marco teórico

3.1.1 Hipertensión arterial (HTA)

La tensión arterial es la fuerza que ejerce la sangre contra las paredes de las arterias, que son grandes vasos por los que circula la sangre en el organismo. Se considera que la persona presenta hipertensión cuando su tensión arterial es demasiado elevada.

De la tensión arterial se dan dos valores: el primero es la tensión sistólica y corresponde al momento en que el corazón se contrae o late, mientras que el segundo, la tensión diastólica, representa la presión ejercida sobre los vasos cuando el corazón se relaja entre un latido y otro.

Para establecer el diagnóstico de hipertensión se han de tomar mediciones dos días distintos y en ambas lecturas la tensión sistólica ha de ser superior o igual a 140 mmHg y la diastólica superior o igual a 90 mmHg (Organización Mundial de la Salud, s.f.).

3.1.2 Diabetes

La diabetes mellitus es una enfermedad crónica en la cual el páncreas no secreta suficiente insulina (Diabetes tipo I), o cuando el organismo no utiliza eficazmente la insulina que produce (Diabetes tipo II). La insulina es una hormona que regula la concentración de glucosa en la sangre. El efecto de la diabetes no controlada es la hiperglucemia (es decir, la glucemia elevada), que, con el tiempo, daña gravemente muchos órganos y sistemas, sobre todo los nervios y los vasos sanguíneos.

La diabetes más común es la tipo II, además también existe la diabetes gestacional que es la diabetes que se contrae durante el embarazo y que puede abarcar a contraer diabetes tipo II en el futuro (Organización Mundial de la Salud, 2021).

3.1.3 Cloud Computing

La computación en la nube (Cloud Computing) es una tecnología que permite acceso remoto a softwares, almacenamiento de archivos y procesamiento de datos por medio de Internet, siendo así, una alternativa a la ejecución en una computadora personal o servidor local. En el modelo de nube, no hay necesidad de instalar aplicaciones localmente en computadoras (Salesforce, s.f.).

3.1.4 Docker

Docker es una tecnología de código abierto (y un formato de archivo de contenedor) que permite automatizar la implementación de aplicaciones como contenedores portátiles autosuficientes que pueden ejecutarse en la nube o en entornos locales (Microsoft Azure, s.f.).

3.1.5 Kubernetes

Kubernetes es un software de orquestación de código abierto que ofrece una API para controlar la forma de ejecución de los contenedores. Permite ejecutar contenedores y cargas de trabajo de Docker y dar solución a algunas de las complejidades de funcionamiento al escalar varios contenedores implementados en varios servidores.

Con Kubernetes, se puede organizar un clúster de máquinas virtuales y programar los contenedores para que se ejecuten en esas máquinas según los recursos de proceso disponibles y los requisitos de recursos de cada contenedor. Los contenedores se agrupan en pods, que es la unidad operativa básica de Kubernetes (Microsoft Azure, s.f.).

3.1.6 Balanceador de carga (NGINX)

Un balanceador de carga actúa como el "policía de tráfico" frente a sus servidores y enrutando las solicitudes de los clientes a través de todos los servidores capaces de cumplir con esas solicitudes para maximizar la velocidad y la utilización de la capacidad y asegura que ningún

servidor esté sobrecargado de trabajo, lo que podría degradar el rendimiento. Si un solo servidor deja de funcionar, el balanceador de carga redirige el tráfico a los servidores en línea restantes. Cuando se agrega un nuevo servidor al grupo de servidores, el balanceador de carga automáticamente comienza a enviarle solicitudes (NGINX, s.f.).

3.1.7 API REST

Una API es conjunto de reglas para determinar cómo las aplicaciones pueden conectarse o comunicarse entre sí; una API REST se ajusta a los principios REST que se componen de lo siguiente:

3.1.7.1 Interfaz uniforme. Todas las solicitudes de API para el mismo recurso deben ser iguales, independientemente de la procedencia de la solicitud. La API REST debe asegurarse de que el mismo dato, como el nombre o la dirección de e-mail de un usuario, pertenezca a un único identificador uniforme de recurso (URI).

3.1.7.2 Separación entre cliente y servidor. En el diseño de API REST, las aplicaciones de cliente y de servidor deben ser completamente independientes entre sí. La única información que la aplicación de cliente debe conocer es el URI del recurso solicitado.

3.1.7.3 Sin estado. Es decir, que cada solicitud debe incluir toda la información necesaria para procesarla. En otras palabras, las API REST no requieren ninguna sesión en el lado del servidor.

3.1.7.4 Capacidad de almacenamiento en memoria caché. Siempre que sea posible, los recursos deben poder almacenarse en la memoria caché en el lado del cliente o el servidor. El objetivo es mejorar el rendimiento en el lado del cliente, al mismo tiempo que aumenta la escalabilidad en el lado del servidor.

3.1.7.5 Arquitectura de sistema de capas. En las API REST, las llamadas y respuestas pasan por diferentes capas. Como regla general, no debe suponerse que las aplicaciones de cliente y de servidor se conectan directamente entre sí.

3.1.7.6 Código bajo demanda (opcional). Generalmente, las API REST envían recursos estáticos, pero en algunos casos, las respuestas también pueden contener un código ejecutable (como applets de Java). En estos casos, el código solo debería ejecutarse bajo demanda (IBM, s.f.).

3.1.8 Control de versiones

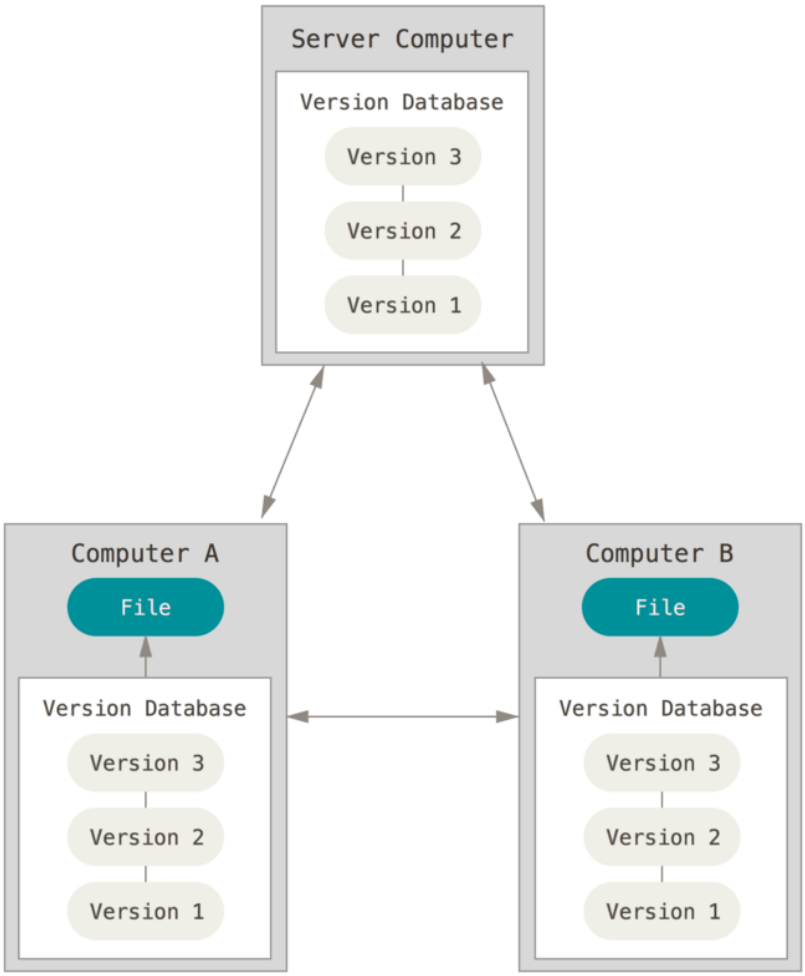
Un control de versiones es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que se pueda recuperar versiones específicas más adelante, es decir tener un historial del proyecto.

Los controles de versiones pueden ser distribuidos y centralizados, pero se va a centrar principalmente en los sistemas de control de versiones distribuidos.

Los sistemas de Control de Versiones Distribuidos (DVCS por sus siglas en inglés) ofrecen soluciones para los problemas que han sido mencionados. En un DVCS (como Git,Mercurial, Bazaar o Darcs), los clientes no solo descargan la última copia instantánea de los archivos, sino que se replica completamente el repositorio. De esta manera, si un servidor deja de funcionar y estos sistemas estaban colaborando a través de él, cualquiera de los repositorios disponibles en los clientes puede ser copiado al servidor con el fin de restaurarlo. Cada clon es realmente una copia completa de todos los datos (Git SCM, s.f.).

Figura 1

Sistema de control de versiones distribuido



Nota. Sistema de control de versiones distribuido. Tomado de *Git—Acerca del Control de Versiones*. (s. f.). Adaptado de 7 de mayo de 2022, de https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Acerca-del-Control-de-Versiones

Además, muchos de estos sistemas se encargan de manejar numerosos repositorios remotos con los cuales pueden trabajar, de tal forma que puedes colaborar simultáneamente con diferentes grupos de personas en distintas maneras dentro del mismo proyecto. Esto permite establecer varios

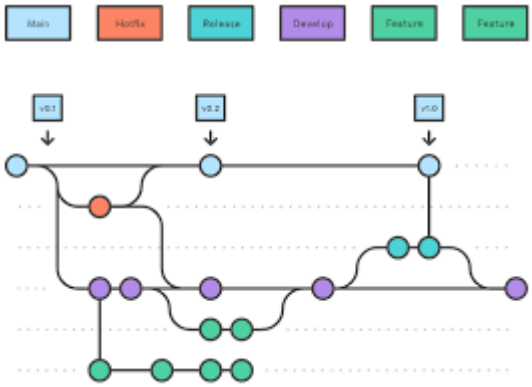
flujos de trabajo que no son posibles en sistemas centralizados, como pueden ser los modelos jerárquicos (Git SCM, s.f.).

3.1.8.1 Gitflow. Es un modelo alternativo de creación de ramas en Git en el que se utilizan ramas de función y varias ramas principales. Según este modelo, los desarrolladores crean una rama de función y retrasan su fusión con la rama principal del tronco hasta que la función está completa (Atlassian, s.f.).

Para este flujo de trabajo existe una rama principal donde se almacena la aplicación y unas ramas de publicación llamadas develop y feature que son ramas que no son muy estables y donde se van haciendo incrementos o cambios al producto, release que son lanzamientos, y hotfix son ramas para la solución de problemas en producción.

Figura 2

Estructura de ramas con GitFlow



Atlassian. (s. f.). *Flujo de trabajo de Gitflow | Atlassian Git Tutorial*. Atlassian. Recuperado 3 de septiembre de 2022, de https://www.atlassian.com/es/git/tutorials/comparing-workflows/gitflow-workflow

3.1.9 Dispositivos Wearables

Wearable traduce dispositivos que se pueden usar o llevar. La movilidad es su principal característica y el IoT la base de su planteamiento, ya que son los sensores los que permiten a los diferentes aparatos, accesorios y prendas recoger y emitir datos de forma constante, es decir, que son portables.

Si bien las compañías están constantemente creando nuevos tipos de dispositivos portátiles, la mayoría pueden encajar en estas categorías principales: aparatos de monitorización de ejercicio o relojes inteligentes (Universidad Internacional de Valencia, 2019).

Generalmente estos dispositivos envían sus datos recolectados por medio del protocolo de bluetooth de baja energía (BLE), similar al clásico protocolo de bluetooth, pero con menor gasto energético, y cuya función principal involucra el envío de datos e información entre dispositivos (Novidá, s.f.).

3.1.10 CI/CD (Integración continua/distribución continua)

La CI/CD es un método para distribuir las aplicaciones a los clientes con frecuencia mediante el uso de la automatización en las etapas del desarrollo de aplicaciones (Red Hat, 2018).

La implementación continua (la otra definición de "CD") hace referencia al lanzamiento automático de los cambios que implementa el desarrollador desde el repositorio hasta la producción, para ponerlos a disposición de los clientes y la CI se enfoca en el diseño, pruebas y combinación de cambios nuevos en el código de una aplicación con regularidad en un repositorio compartido (Red Hat, 2018).

3.1.11 DevOps

Bajo un modelo de DevOps, los equipos de desarrollo y operaciones ya no están “aislados”. A veces, los dos equipos se fusionan en uno solo, donde los ingenieros trabajan en todo el ciclo de

vida de la aplicación, desde el desarrollo y las pruebas hasta la implementación y las operaciones, y desarrollan una variedad de habilidades no limitadas a una única función (Amazon, s.f.).

3.1.12 Metodologías ágiles

El 48% de los proyectos de software no se terminan a tiempo, dado los problemas en tiempo y presupuesto a la hora de planificar proyectos de software se da origen a una serie de técnicas para agilizar el proceso de planificación y ejecución de proyectos de software dando rapidez y flexibilidad a los mismos, estas técnicas se conocen como metodologías ágiles (ADEN, 2021).

Para este proyecto se va a trabajar con la metodología ágil de scrum, la cual consiste en lo siguiente:

Scrum emplea un enfoque iterativo e incremental para fallar de manera rápida y aprender de estos fallos, de esta manera agilizar el proceso.

La unidad fundamental de Scrum es un pequeño equipo de personas (scrum team) consta de un scrum máster, un Product Owner y los developers. No hay jerarquías, es una unidad cohesionada de profesionales enfocados en un objetivo a la vez, el objetivo del producto (SCRUM GUIDES, 2020).

A continuación, se van a definir los conceptos más significativos de la metodología scrum:

3.1.12.1 Feedback. Es la retroalimentación del sprint, el objetivo de scrum es entregar rápido, para equivocarse rápido y corregir rápido.

3.1.12.2 Increment. “Es un peldaño concreto hacia el objetivo del producto. Cada Increment se suma a todos los Increments anteriores y se verifica minuciosamente” (SCRUM GUIDES, 2020).

3.1.12.3 Product Backlog. Es la lista de necesidades que requiere el producto, ordenada de manera priorizada (SCRUM GUIDES, 2020).

3.1.12.4 Sprint. “Son eventos de duración fija de un mes o menos para crear consistencia. Un nuevo Sprint comienza inmediatamente después de la conclusión del Sprint anterior” (SCRUM GUIDES, 2020).

3.1.12.5 Daily Meeting. “El objetivo de esta reunión es facilitar la transferencia de información y la colaboración entre los miembros del equipo para aumentar su productividad, al poner de manifiesto puntos en que se pueden ayudar unos a otros” (Maida & Pacienzia,2015)

3.1.12.6 Daily Sprint. Es un evento de 15 minutos para los Developers del Scrum Team, con el objetivo de monitorear los avances del proyecto (SCRUM GUIDES, 2020).

3.1.12.7 Sprint Planning. Abarca el trabajo que se realizará para el Sprint. Este plan resultante es creado por el trabajo colaborativo de todo el equipo (SCRUM GUIDES, 2020).

3.1.12.8 Sprint Review. Es una reunión informal donde el equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo, haciendo un recorrido por ellos lo más real y cercano posible al objetivo que se pretende cubrir (Gutierrez, 2014).

3.1.12.9 Sprint Goal. El sprint goal es el objetivo del sprint. El sprint goal también crea coherencia y enfoque, alentando al equipo scrum a trabajar en conjunto en lugar de en iniciativas separadas. El sprint goal se crea durante el evento Sprint Planning y luego se agrega al Sprint Backlog (SCRUM GUIDES, 2020).

3.1.12.10 Sprint Backlog. El Sprint Backlog se compone del Sprint Goal, el conjunto de elementos del Product Backlog seleccionados para el Sprint, así como un plan procesable para entregar el Increment (SCRUM GUIDES, 2020).

3.1.13 Web Service

Un servicio web es un sistema de software que admite la interacción interoperable de máquina a máquina a través de una red. Tiene una interfaz descrita en un formato procesable por máquina (específicamente, Lenguaje de definición de servicios web o WSDL).

Los servicios web cumplen una tarea específica o un conjunto de tareas. Un servicio web se describe utilizando una noción XML formal y estándar, denominada descripción del servicio, que proporciona todos los detalles necesarios para interactuar con el servicio, incluidos los formatos de mensaje (que detallan las operaciones), los protocolos de transporte y la ubicación (IBM, 2021).

3.1.14 Arquitectura basada en Microservicios

Una arquitectura de microservicios consta de una colección de servicios autónomos y pequeños. Cada uno de los servicios es independiente y debe implementar una funcionalidad de negocio individual dentro de un contexto delimitado. Un contexto delimitado es una división natural de una empresa y proporciona un límite explícito dentro del cual existe un modelo de dominio (Microsoft, s.f.).

3.1.15 Angular

Angular es un marco de diseño de aplicaciones y una plataforma de desarrollo para crear aplicaciones eficientes y sofisticadas de una sola página, está escrita en TypeScript. Como plataforma Angular incluye.

Un marco basado en componentes para crear aplicaciones web escalables.

Una colección de bibliotecas bien integradas que cubren una amplia variedad de funciones, incluido el enrutamiento, la gestión de formularios, la comunicación cliente-servidor y más.

Un conjunto de herramientas de desarrollo para ayudar a desarrollar, compilar, probar y actualizar el código (Angular, 2022).

3.1.15.1 Módulos. Una aplicación Angular tiene un módulo raíz, llamado **AppModule**, que proporciona el mecanismo de arranque para iniciar la aplicación (Gonçalves, 2021).

3.1.15.2 Componentes. Estos son piezas o fragmentos de código que define una clase que contiene la lógica y los datos de la aplicación. Un componente por lo general define una parte de la interfaz de usuario (UI) (Gonçalves, 2021).

3.1.15.3 Plantillas. Cada componente tiene una plantilla HTML que declara cómo se representa ese componente. Esta plantilla se define en línea o por ruta de archivo (Angular, 2022).

3.1.15.4 Metadatos. Los metadatos le dicen a Angular cómo procesar una clase. Se utiliza para decorar la clase para que pueda configurar el comportamiento esperado de una clase (Gonçalves, 2021).

3.1.15.5 Servicios. Los servicios sirven para compartir información entre componentes o incluso hacer peticiones http a APIs para obtener la información. Los servicios funcionan solo en ámbito de lógica o datos, no están asociados a la vista (Gonçalves, 2021).

3.1.15.6. Inyección de dependencia. Esta característica le permite mantener sus clases de componentes nítidas y eficientes. No obtiene datos de un servidor, no valida la entrada del usuario ni se registra directamente en la consola. En cambio, delega tales tareas a los servicios (Gonçalves, 2021).

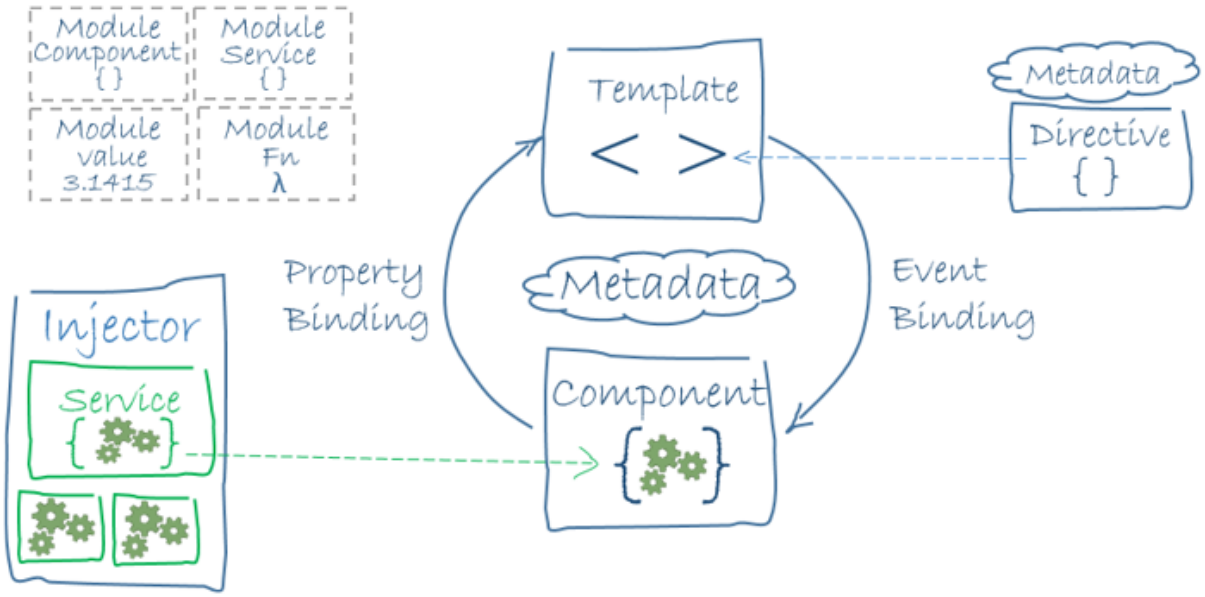
3.1.15.7. Directivas Angular. Las directivas amplían el HTML proporcionándole una nueva sintaxis (Gonçalves, 2021).

3.1.15.8. Ciclo de vida de Angular. Un componente Angular entra en su ciclo de vida desde el momento en que se crea hasta el momento en qué se destruye. Estos hooks brindan formas

de aprovechar estas fases y activar cambios en momentos específicos del ciclo de vida (Bastidas, 2020).

Figura 3

Relación de los elementos que componen Angular



Nota. Relación de los elementos que componen Angular. Tomado *Angular—Introducción a los conceptos de Angular*. (s. f.). Introducción a los conceptos de Angular. Recuperado 31 de julio de 2022, de https://docs.angular.lat/guide/architecture.

3.1.16 Flutter

SDK proporcionado por Google en el 2017 que permite el desarrollo de aplicaciones multiplataforma con el lenguaje de programación Dart con una baja curva de aprendizaje (Flutter, 2022).

Cuando recién comenzó el desarrollo de aplicaciones móviles, se usaba Java para el desarrollo de aplicaciones móviles en Android, a esto se le conoce también como desarrollo nativo, pero el problema es que solo era aplicable a dispositivos Android, si se quería implementar una

aplicación para dispositivos iOS, el código implementado en Android con Java era prácticamente inservible en iOS, ya que los dispositivos de Apple, suelen trabajar con Objetive-C, recientemente hay otros lenguajes de programación nativos como lo son Kotlin para Android y Swift para iOS pero en todo caso hay que implementar un código base para cada plataforma.

Varias empresas se percataron de este problema y con el objetivo de crear un marco de desarrollo para el desarrollo de aplicaciones multiplataforma empezaron a aparecer tecnologías como Xamarin, Ionic y Nativescript, pero no era muy robustas ya que la mayoría eran basadas en componentes web, así que para el desarrollo aplicaciones complejas no eran muy eficientes, ya que, por ejemplo, para acceder al hardware como del podómetro del dispositivo es difícil con una tecnología web (Maria & Mikhail, 2022).

Entonces apareció React Native un marco de desarrollo proporcionado por Facebook para el desarrollo de aplicaciones multiplataforma muy similar a React que es usado para el desarrollo de aplicaciones web, así fue rápidamente acaparando al mercado.

Google no se quedó cruzado de brazos y aquí fue donde lanzó Flutter un SDK que usa el lenguaje de programación Dart (Dart, s.f.), que es también otro proyecto de Google muy similar a Javascript con el enfoque de programación reactiva y asíncrona para el desarrollo de aplicaciones multiplataforma de manera rápida y con una baja curva de aprendizaje.

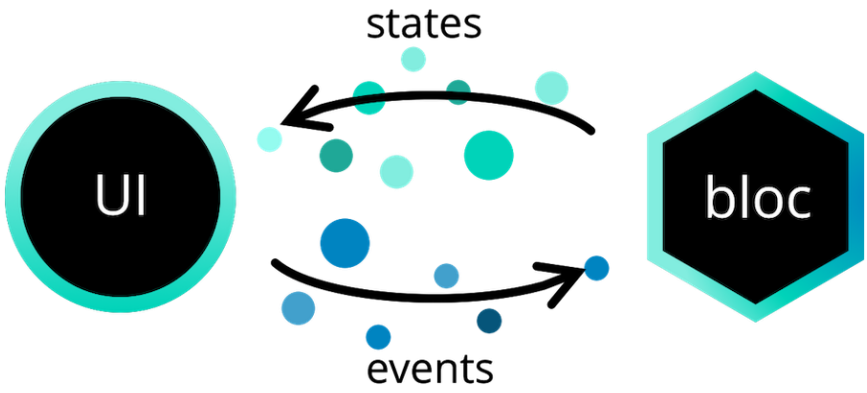
A pesar de que Flutter es un marco de desarrollo multiplataforma en la gran mayoría de casos hay que realizar unos cambios mínimos en el código para que la aplicación sea compatible para iOS, por ejemplo, hay plugins que requieren acceso a los permisos de Android y en la documentación de los mismos se requiere cambiar el fichero de AndroidManifest.xml, pero para iOS este fichero no existe y se maneja es con Info.plist, así que a pesar de que el código base de

Android no va a ser 100% funcional, la migración a iOS va a requerir días o quizás semanas y no un proyecto totalmente nuevo.

3.1.16.1 Patrón BLoC. Flutter posee varias opciones para el control de estado en el ciclo de vida de la aplicación. Algunas opciones son GetX, Provider, Redux y el patrón BLoC, para el presente proyecto se implementó el patrón BLoC, un patrón que ha ido ganando popularidad en los últimos años, básicamente es un patrón que maneja inyección de dependencias y lo que permite es que a un flujo de datos llamados streams en Flutter que se reciben como eventos, se haga un procesamiento o se conecte a un repositorio o servicio en caso de ser necesario y se envié un estado conforme al evento recibido, un ejemplo aplicado a un login con el patrón BLoC sería: Se recibe el evento que el usuario intente realizar una autenticación con correo y contraseña, el evento también debe enviar las credenciales, el BLoC recibe las credenciales y se conecta al repositorio (servidor backend para ver si las credenciales coinciden con algún usuario registrado) y en este caso se pueden emitir dos estados, que inicie sesión de manera exitosa o que falle, es decir la lógica estaría dentro del BLoC (Angelov, s.f.).

Figura 4

Patrón BLoC



Nota. Patrón BLoC. -. *Bloc State Management Library*. (s. f.). Bloc. Recuperado 31 de julio de 2022, de https://bloclibrary.dev/

3.1.16.2 Ciclo de vida de la app. En Flutter existen una serie presentes o a sobrescribir dependiendo si se trabaja con un StatefulWidget o un StatelessWidget. Además, manejan principalmente los siguientes métodos (StackOverflow, 2017):

3.1.16.2.1 createState(). Cuando el Framework recibe la orden de construir un StatefulWidget, llama inmediatamente a createState().

3.1.16.2.2 initState(). Este es el primer método llamado cuando se crea el Widget.

3.1.16.2.3 didChangeDependencies(). Este método se llama inmediatamente después de initState la primera vez que se construye el Widget.

3.1.16.2.4 build(). Método a sobrescribir cuando se crea un Widget. Es necesario, y debe devolver un Widget.

3.1.16.2.5 setState(). Este método se llama a menudo desde el propio framework y desde el desarrollador. Se utiliza para notificar al framework que los datos han cambiado y se reconstruya la pantalla.

3.1.16.2.6 dispose(). Es llamado cuando el objeto State es eliminado, lo cual es permanente. Este método es donde se debe dar de baja y cancelar todas las animaciones, flujos, etc.

3.1.16.3 Diseño responsivo. Diseño adaptable a varias pantallas, es decir que el usuario pueda interactuar con la aplicación independientemente de la resolución de la pantalla del dispositivo que esté manejando, aplica tanto para web como para móvil.

3.1.16.4 Firebase. Es un serverless proporcionado por Google que proporciona una serie de herramientas y características para las aplicaciones sin necesidad de implementar un backend como tal, algunas de sus características consisten en autenticación, manejo de archivos, bases de datos no relacionales y bases de datos en tiempo real, etc. (Firebase, s.f.).

3.1.16.5 Programación asíncrona. Paradigma de programación que permite que las tareas siguientes no se ejecuten hasta que la tarea se dé por terminada mejorando el rendimiento (Dart, s.f.).

3.1.16.6 Programación reactiva. Es un paradigma de programación que se enfoca en el flujo de datos asíncronos, observables y el uso de streams (Escoffier, 2017).

3.1.16.7 StatefulWidget. Es un tipo de widget que puede mutar su estado de acuerdo a ciertas características o eventos, por ejemplo, cuando por defecto se crea una aplicación en Flutter, la aplicación por defecto muestra un texto con un contador donde se muestra la cantidad de veces que se ha presionado un botón, cada vez que el botón es presionado se va actualizando el texto en pantalla de la cantidad de veces que el botón es presionado, esto es gracias a un StatefulWidget ya que se está cambiando su estado cada vez que se presiona el contador, la documentación de Flutter recomienda solo usarlos cuando sea necesario, porque su rendimiento es menos optimo que el de

un StatelessWidget y esto considerablemente afectará el rendimiento de la aplicación (Flutter, 2022).

3.1.16.8 StatelessWidget. Widgets que no son mutables, son más livianos que los StatefullWidgets, son los más usados en general, no poseen los métodos de initState y para cambiar su estado es necesario destruirlos y volverlos a crear (Flutter, 2022).

3.1.16.9 State Management. Es el enfoque o arquitectura que se provee en la aplicación para el manejo de su estado y comunicación entre widgets, es necesario para que las aplicaciones hoy en día sean escalables ya que Flutter al manejar un árbol de Widgets, puede miles de nodos hijos, para el flujo de información se necesita una manera de acceder a la información sin tener que estar pasando información de por cada uno de los nodos, para esto se utilizan los State management, sin este mismo, al querer comunicar un widget que está en la parte de arriba con una hoja del árbol tendría que pasarse widget por widget, lo que lo haría increíblemente ineficiente, sin embargo existen una serie de opciones para solucionar este problema en los que se encuentran opciones como provider, el patrón BLoC y redux (Flutter, s.f.).

3.1.16.10 Streams. Un stream es una secuencia de eventos asíncronos. Es como un Iterable asíncrono donde, en lugar de obtener el siguiente evento cuando se pide, el stream indica que hay un evento cuando está listo, en pocas palabras manejar un flujo de información como un caudal (Dart, s.f.).

3.1.16.11 Widget. Elemento fundamental de Flutter inspirado en React que consiste en cualquier elemento que compone la interfaz, es decir una caja de texto, un botón o la pantalla completa son widgets, los widgets se componen a su vez de widgets, así que estos pueden ser muy complejos o abstractos y personalizables.

3.1.17 Material Design

Estándar de diseño establecido por Google para tener unas buenas prácticas de diseño en las aplicaciones modernas (Google Inc., s.f.).

3.1.18 Adobe XD

Programa de prototipado de interfaces para sitios web y aplicaciones móviles proporcionadas por la suite de adobe (Adobe, s.f.).

3.1.19 Google Fit

Google Fit es un proveedor de Google que permite realizar conexiones y consultar información por medio de una API REST, proporcionando datos de los usuarios de su salud, composición corporal, frecuencia cardiaca (Google, s.f.).

3.1.20 Node.js

Ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables. (Node.js, s.f.).

Permite gestionar múltiples conexiones al mismo tiempo, evitando el bloqueo de procesos. Node.js utiliza un modelo de entrada y salida sin bloqueo controlado por eventos que lo hace ligero y eficiente. Esto permite manejar una gran cantidad de conexiones simultáneas con un alto nivel de rendimiento (Jesuites educación , s.f.).

3.1.20.1 Express. Express es una infraestructura de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles. Con miles de métodos de programa de utilidad HTTP y middleware a su disposición, la creación de una API sólida es rápida y sencilla (Node.js, s.f.).

3.2 Estado del arte

3.2.1 Introducción y problemática

Las TIC (Tecnologías de información y comunicaciones) han incidido de manera contundente en el diario vivir, influyendo en las actividades del ser humano y facilitando la atención a ciertas necesidades, unido a esto, la democratización de smartphones y tablets, junto con la mejora de recursos de hardware ha conllevado a la proliferación de aplicaciones de distinta índole. Según (AppBrain, 2022) actualmente Google Play cuenta con un poco más de 2.5 millones de aplicaciones y sólo el 3.90% son de la categoría HealthCare lo que es desafortunado teniendo en cuenta la cantidad de problemas en el ámbito de salud que existen hoy en día como, por ejemplo, la diabetes y la hipertensión.

La organización panamericana de la salud informó que entre el 20 y el 40% de la población adulta en América latina sufre de hipertensión, principal causa de muerte por enfermedades cardiovasculares, además, un gran porcentaje de esta población no lleva un control adecuado (OPS/OMS, s.f.). “Aproximadamente 62 millones de personas en América sufren de diabetes, la diabetes sin un control apropiado puede aumentar las posibilidades de daño ocular, enfermedad renal, enfermedades cardiovasculares, neuropatías, trombosis cerebrales, que pueden desencadenar en mortalidad prematura” (OPS/OMS, s.f.).

En este proyecto se van a indagar diferentes maneras tecnológicas enfocadas en telemedicina para facilitar el monitoreo y atención de pacientes. Es importante tener una noción de algunas plataformas existentes hoy en día para monitorear a pacientes hipertensos y/o diabéticos, por ejemplo, aplicaciones como SocialDiabetes, GluQUO, Bant, son especiales para pacientes diabéticos con funcionalidades como conexión a Google Fit, ver el historial de los datos propiciados por los dispositivos wearables en gráficas, etc. (GaeaPeople, 2020). Por otro lado,

Instant Heart Rate es una aplicación que sirve como aliado para las personas hipertensas, ya que les facilita llevar un registro de sus datos, por ejemplo, el monitoreo de la frecuencia cardiaca (Apps para controlar la hipertensión arterial, 2019).

3.2.2 SocialDiabetes

SocialDiabetes es una plataforma médica que consiste en una aplicación que realiza un control a pacientes diabéticos con distintas funcionalidades como un seguimiento a la dieta, acceso a Google Fit para tener control de la actividad física y acceso a otros dispositivos como baumanómetro o glucómetros para control de los síntomas. También posee una plataforma web con el fin de tener acceso a los datos y visualización de estos para que el profesional médico se pueda apoyar mejor (SocialDiabetes, s.f.).

SocialDiabetes se centra de una manera muy eficiente en pacientes diabéticos, pero no toma en cuenta pacientes hipertensos, tampoco cuenta con una funcionalidad de geolocalización por medio de mapas para un seguimiento continuo de la ubicación del paciente en caso de emergencia.

Es una aplicación enfocada en pacientes hipertensos, permite ver la frecuencia cardiaca en tiempo real, lo hace al detectar el pulso del usuario cuando cubre por completo la lente de la cámara del smartphone con su dedo índice, como si se tratara de un pulsioxímetro (Garcia, 2016).

La aplicación es una buena aliada para un paciente hipertenso, pero la metodología que implementa al medir el ritmo cardiaco no es directamente de un dispositivo wearable que, si puede llegar a ser más preciso, además no toma en cuenta a pacientes diabéticos y no cuenta con una plataforma web la cual es esencial para llevar un control por parte del profesional de la salud.

Tampoco cuenta con una conexión a un tercero que proporcione modelos predictivos con el fin de generar alertas para los pacientes si sobrepasan sus respectivos umbrales

3.2.3 Instant Heart Rate

Es una aplicación enfocada en pacientes hipertensos, permite ver la frecuencia cardiaca en tiempo real, lo hace al detectar el pulso del usuario cuando cubre por completo la lente de la cámara del smartphone con su dedo índice, como si se tratara de un pulsioxímetro (Apps para controlar la hipertensión arterial, 2019).

La aplicación es una buena aliada para un paciente hipertenso, pero la metodología que implementa al medir el ritmo cardiaco no es directamente de un dispositivo wearable que, si puede llegar a ser más preciso, además no toma en cuenta a pacientes diabéticos y no cuenta con una plataforma web la cual es esencial para llevar un control por parte del profesional de la salud.

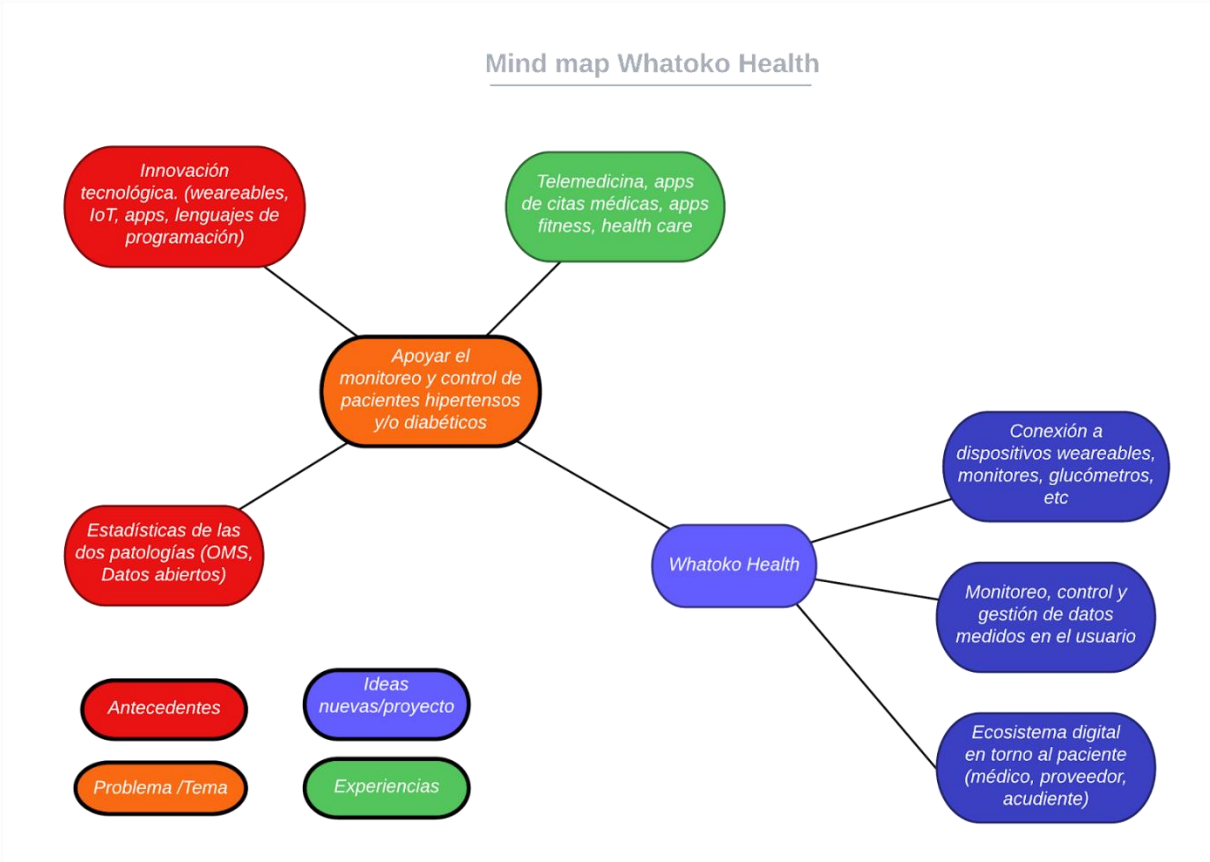
Tampoco cuenta con una conexión a un tercero que proporcione modelos predictivos con el fin de generar alertas para los pacientes si sobrepasan sus respectivos umbrales.

3.2.4 Mapa mental Whatoko Health

A continuación, se muestra un mapa mental de cómo apoyar el autocuidado, el monitoreo y la gestión de pacientes de hipertensión y/o diabetes a través de una aplicación móvil y una plataforma web.

Figura 5

Mapa mental Whatoko Health



4. Desarrollo del proyecto

4.1 Requerimientos

Tabla 1

Alistamiento del proyecto

Prioridad	Alta
Identificación del Requerimiento	RF00
Nombre	Alistamiento del proyecto.
Descripción	Se requiere inicializar el proyecto en Flutter, Node y Angular, crear los pipelines para la documentación automatizada e implementar por medio de integración continua con Firebase App Distribution un release de la aplicación para que los miembros del equipo puedan realizar las respectivas pruebas y retroalimentación.

Tabla 2

Autenticación de usuarios en plataforma web

Prioridad	Alta
Identificación del Requerimiento	RF01
Nombre	Autenticación de usuarios en plataforma web.
Descripción	La plataforma web permitirá a usuarios de rol administrador autenticarse y acceder a todos los servicios que esta ofrece**.**

Tabla 3

Visualización de dashboard

Prioridad	Alta
Identificación del Requerimiento	RF02
Nombre	Visualización de dashboard
Descripción	Los usuarios administradores podrán visualizar el dashboard que contiene información relacionada a los usuarios pacientes de la aplicación.

Tabla 4

Agregar usuarios en la plataforma web

Prioridad	Alta
Identificación del Requerimiento	RF03
Nombre	Agregar usuarios en la plataforma web
Descripción	Los usuarios administradores con acceso a la plataforma podrán agregar usuarios nuevos.

Tabla 5

Inactivar usuarios en la plataforma web

Prioridad	Alta
Identificación del Requerimiento	RF04
Nombre	Inactivar usuarios en la plataforma web
Descripción	Los usuarios administradores con acceso a la plataforma podrán inactivar a otros usuarios.

Tabla 6

Editar usuarios en la plataforma web

Prioridad	Alta
Identificación del Requerimiento	RF05
Nombre	Editar usuarios en la plataforma web
Descripción	Los usuarios administradores con acceso a la plataforma podrán editar información básica de otros usuarios.

Tabla 7

Conectar con Firebase

Prioridad	Alta
Identificación del Requerimiento	RF06
Nombre	Conectar con Firebase
Descripción	El backend de la aplicación se va a implementar en Node.js, además el uso de Firebase permite para autenticar a los usuarios por terceros con OAuth para verificar su identidad y además usar App Distribution para las pruebas de integración continua con el equipo.

Tabla 8

Inicio de la aplicación

Prioridad	Alta
Identificación del Requerimiento	RF07
Nombre	Inicio de la aplicación
Descripción	Dado un archivo con los wireframes en el programa Adobe XD, diseñar las respectivas pantallas responsivas sin funcionalidad, donde se incluyen las pantallas de splash, inicio y login.

Tabla 9

Registro de la aplicación

Prioridad	Alta
Identificación del Requerimiento	RF08
Nombre	Registro de la aplicación
Descripción	Se van a maquetar las pantallas de registro dadas por el archivo XD, donde se incluyen los campos de nombre, apellido, fecha de nacimiento, correo electrónico o teléfono y contraseña.

Tabla 10

Inicio de sesión con credenciales

Prioridad	Alta
Identificación del Requerimiento	RF09
Nombre	Inicio de sesión con credenciales
Descripción	Además del inicio de sesión por terceros, se podrá iniciar sesión por credenciales previamente registradas en el formulario de la aplicación y podrá iniciar por correo electrónico o número celular.

Tabla 11

Integración con huella

Prioridad	Alta
Identificación del Requerimiento	RF10
Nombre	Integración con huella
Descripción	Se dará la opción al usuario de poder ingresar a la aplicación con la huella.

Tabla 12

Vinculación y desvinculación de cuentas

Prioridad	Media
Identificación del Requerimiento	RF11
Nombre	Vinculación y desvinculación de cuentas
Descripción	El usuario podrá añadir múltiples terceros (Google, Facebook, Twitter o Microsoft) y huella para iniciar sesión con cualquiera de las cuentas vinculadas y podrá desvincularlas igualmente.

Tabla 13

Módulo recuperar contraseña

Prioridad	Alta
Identificación del Requerimiento	RF12
Nombre	Módulo recuperar contraseña
Descripción	El usuario podrá recuperar la contraseña (si está registrado) y la ha olvidado recibiendo un correo electrónico y reiniciándola tanto en la aplicación móvil como en la plataforma web.

Tabla 14

Módulo de conexiones

Prioridad	Alta
Identificación del Requerimiento	RF13
Nombre	Módulo de conexión
Descripción	Se requiere conectar por API con Google Fit para obtener mediciones del paciente y comunicarse con el backend en ambas vías con los respectivos datos e información.

4.2 Análisis y diseño

4.2.1 Modelado de base de datos de usuarios

En la plataforma web existen diferentes tipos de usuarios, por ejemplo, profesionales de la salud, desarrolladores, administradores, cada uno tiene un flujo de información que incluye el manejo de diferentes perfiles, roles y permisos.

Usuario, Son las personas que cuentan con acceso a la plataforma web.

Rol, Profesión o papel que tiene el usuario en el sistema. Por ejemplo, un rol podría ser médico o desarrollador.

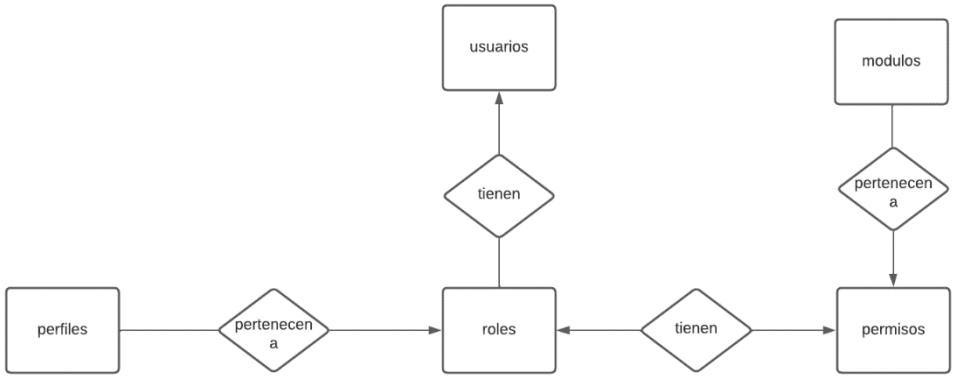
Perfil, Agrupa a varios roles, por ejemplo, los roles de enfermero y médico general pertenecen al perfil de profesionales de la salud.

Módulo, Las funciones que existen en el sistema. No todas las funcionalidades están disponibles para todos los usuarios, así que por ende es necesario controlar por medio de permisos y roles.

Permiso, Los permisos se otorgan a los usuarios en base al rol y al módulo en el que se encuentren, para realizar ciertas acciones.

Figura 6

Diagrama Modelo Entidad Relación de base de datos de usuarios



El principal reto en el modelado de la base de datos fue la falta de conocimiento del problema, se tuvo que investigar arduamente antes de poder iniciar. Luego, debido a la falta de compresión de cada uno de los términos mencionados anteriormente (roles, permisos, módulos) y su función en el flujo de la base de datos surgieron errores y modelos sin mucho sentido, los cuales se fueron iterando y atendiendo las correcciones de los tutores en la empresa se llegó al resultado final.

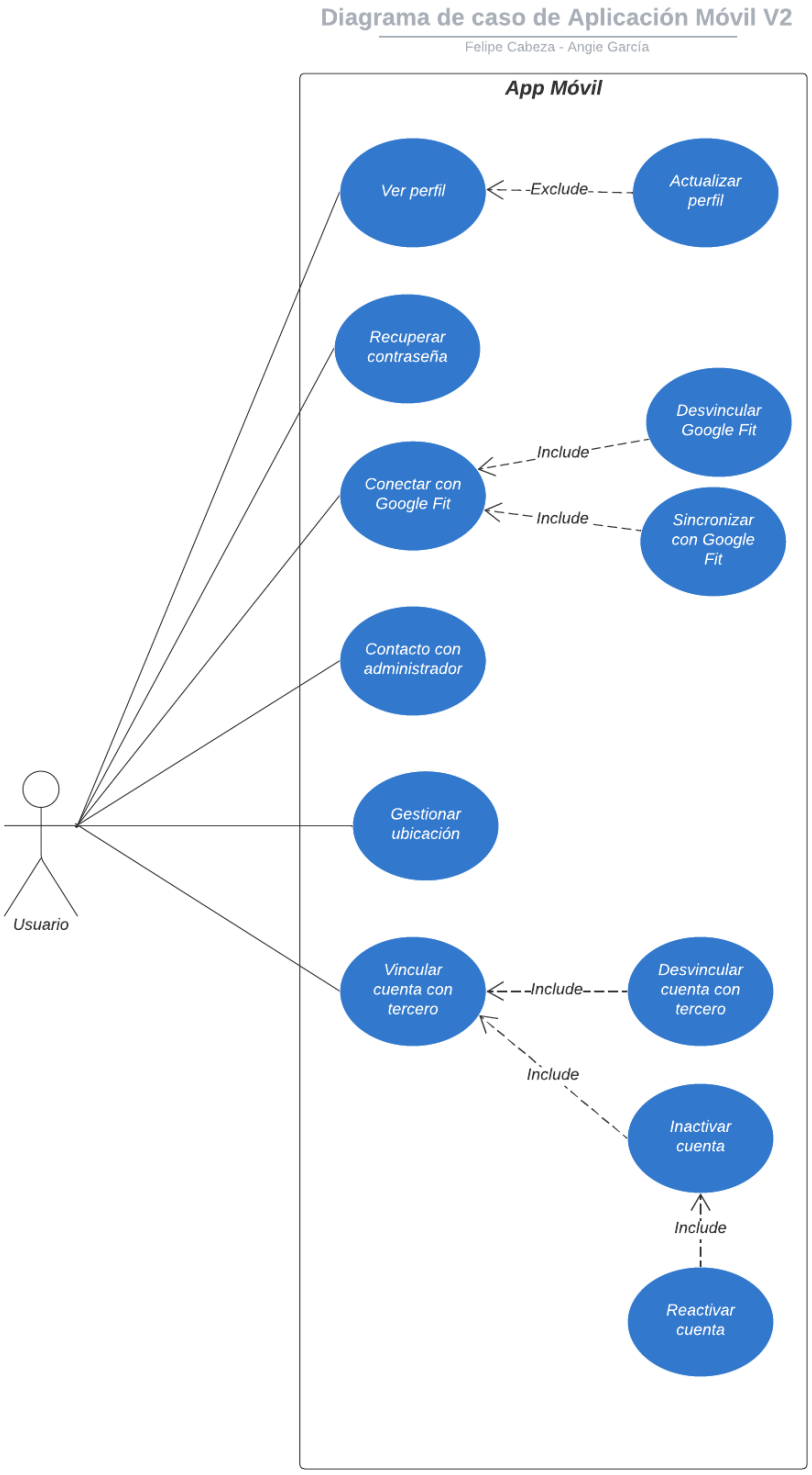
Al comienzo se realizaron bosquejos en herramientas de diagramación sencillas. Luego, debido a las sugerencias de los tutores se identificó que existían herramientas más robustas como lo eran MySQL Workbench y LucidChart, ya que como se iba a trabajar con una base de datos relacional en MySQL, LucidChart permitía la opción de exportar el código en SQL, luego de haber implementado la relación entre tablas con los modelos e incluso importar ese mismo código en MySQL Workbench para poder observar también el diagrama relacional.

4.2.2 Casos de uso

4.2.2.1 Casos de uso aplicación móvil. A continuación, se muestran los casos de uso de la aplicación móvil, se omite el tema del inicio de sesión y registro.

Figura 7

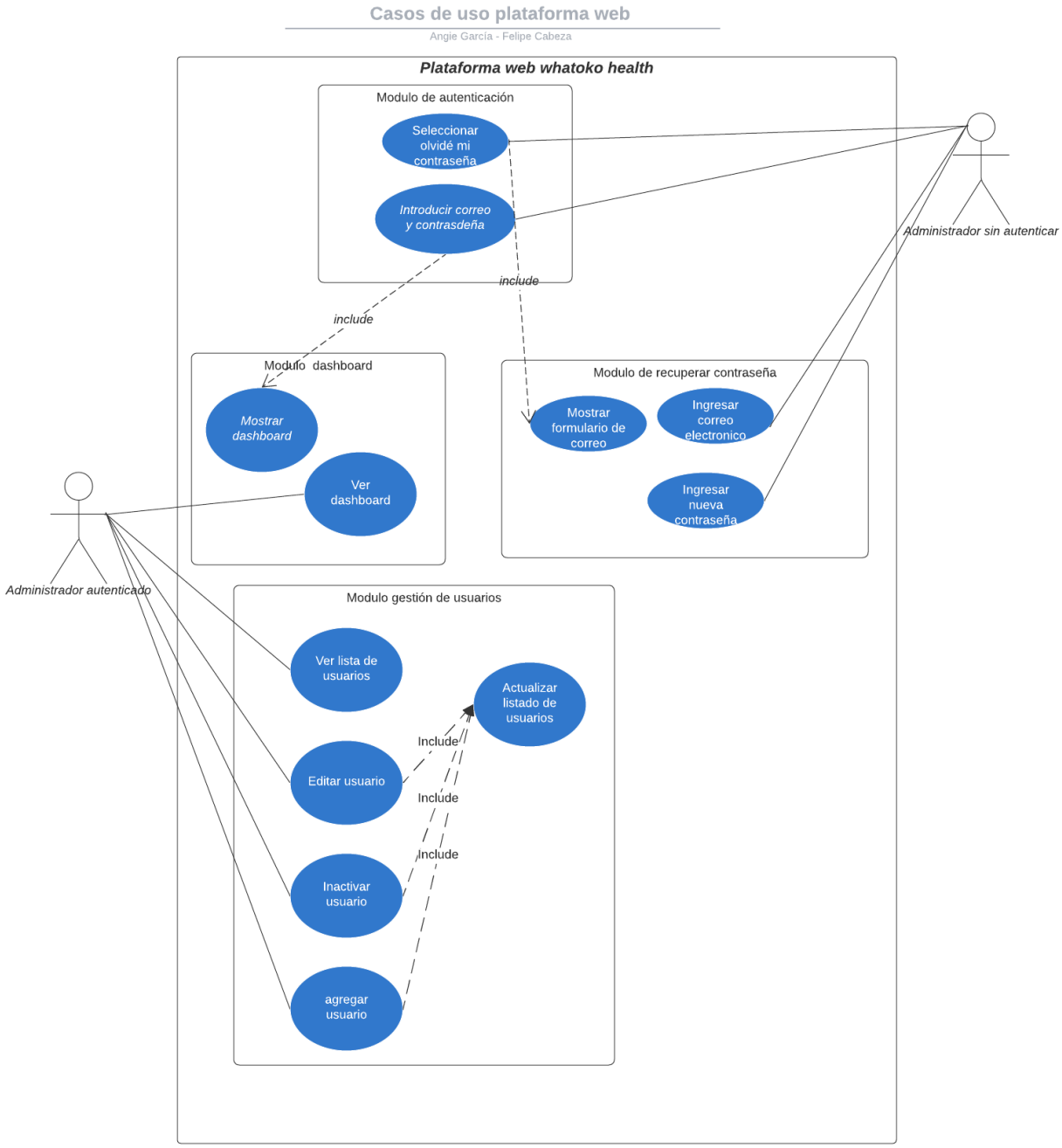
Diagrama casos de uso de la aplicación móvil



4.2.2.2 Casos de uso plataforma web. El siguiente diagrama representa los casos de uso de la plataforma web de Whatoko Health, la cual está enfocada en usuarios administradores.

Figura 8

Diagrama casos de uso de la plataforma web.

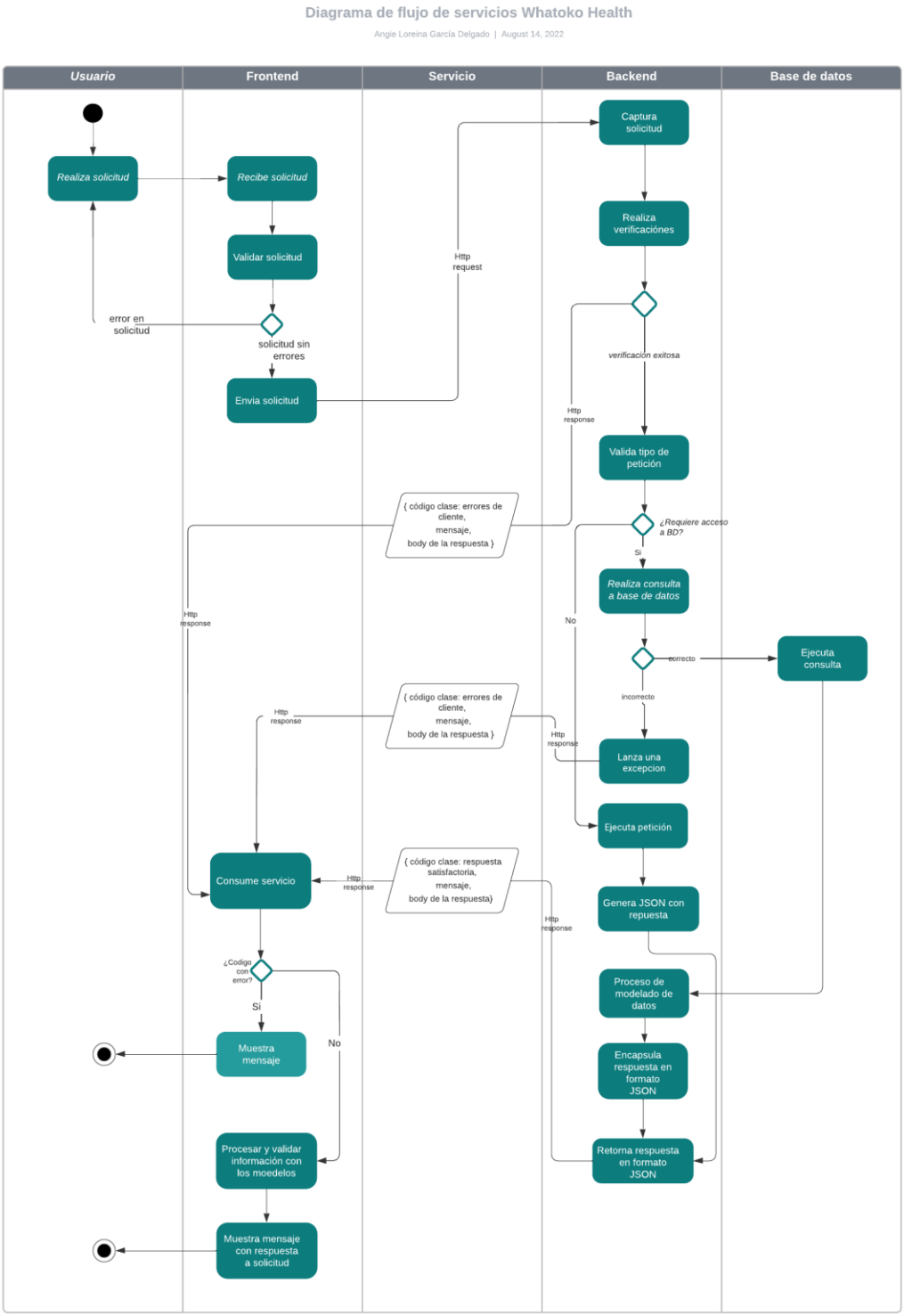


4.2.3 Diagrama de flujo de servicios

El siguiente diagrama representa el flujo de los servicios de Whatoko Health entre los diferentes componentes del proyecto, es una generalización del comportamiento de los servicios que existen.

Figura 9

Diagrama de flujo servicios Whatoko Health



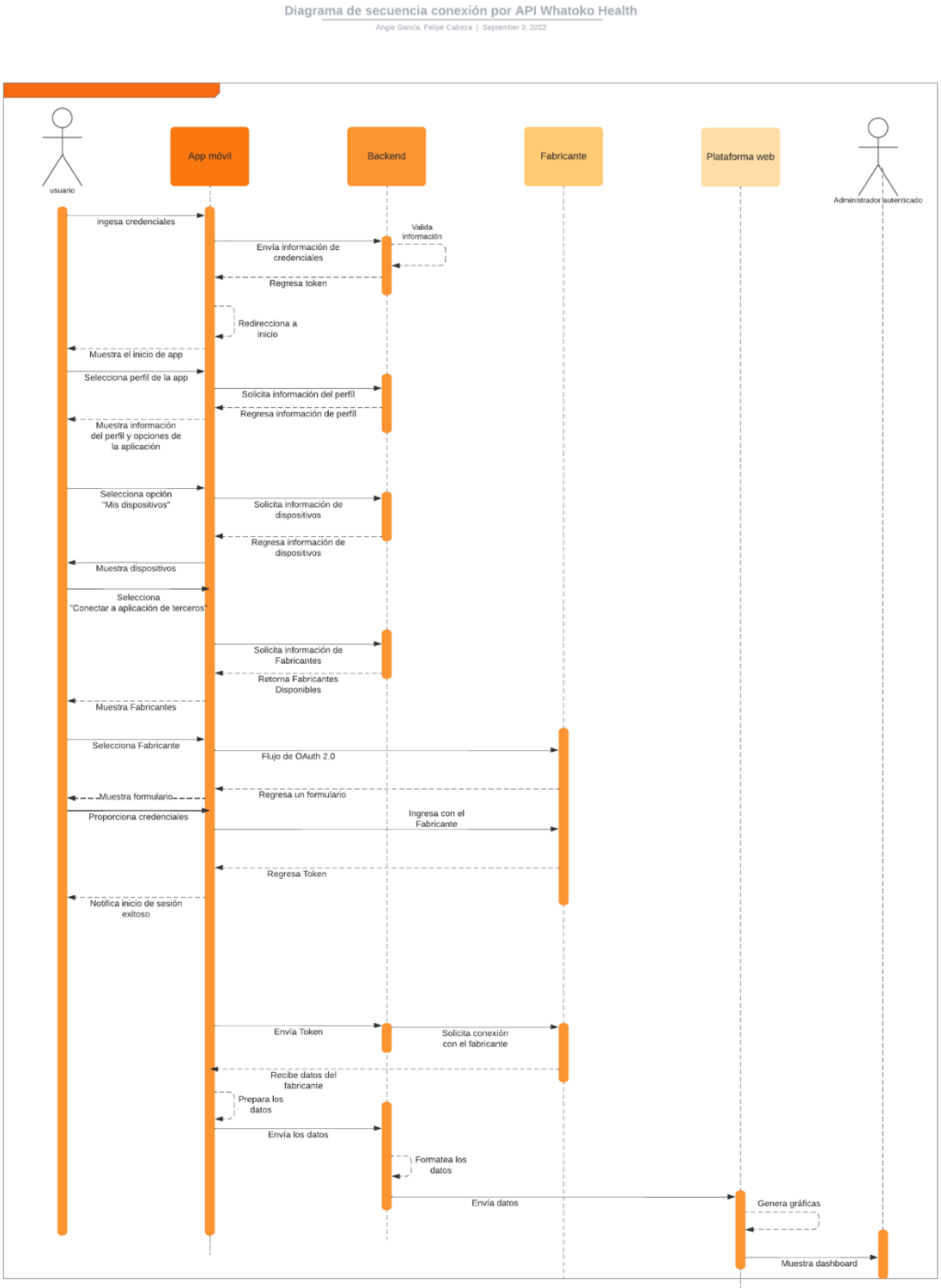
4.2.4 Diagrama de actividades

Existen dos maneras de acceder a los datos que están proporcionando los dispositivos wearables, y el usuario podrá escoger entre una de estas dos opciones que se explicaran a continuación.

4.2.4.1 Diagrama de conexión de dispositivos por API. Con esta opción se va a realizar la conexión de manera transitiva, es decir, se va a conectar a la API del fabricante y realizar un proceso de autenticación por medio del protocolo OAuth, en donde la aplicación móvil recibe los datos que luego se enviarán al backend, en donde se realizará su respectivo procesamiento mostrado con más detalle en la figura 10.

Figura 10

Diagrama de secuencia, conexión de dispositivos por API

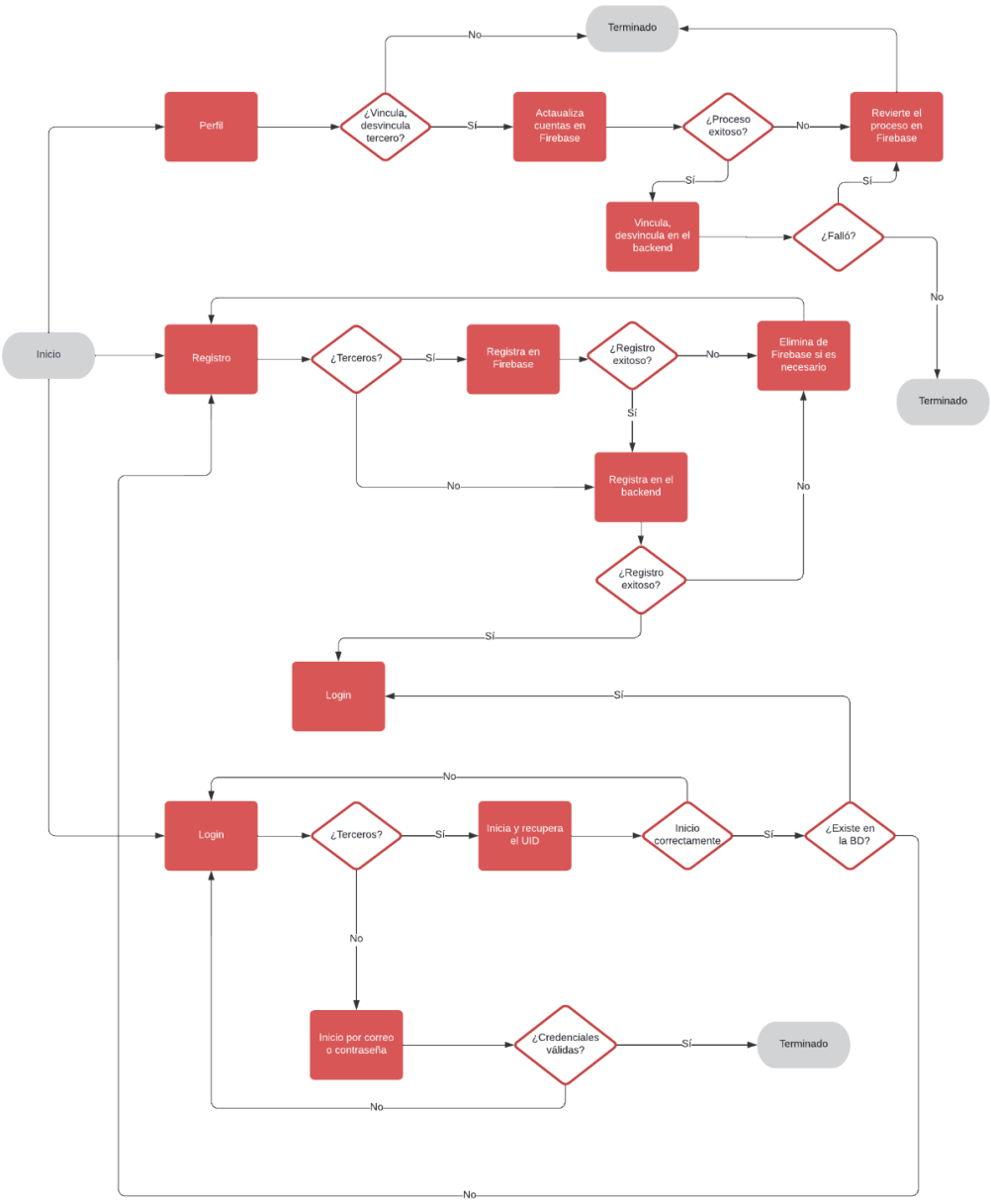


4.2.5 Diagramas de autenticación

4.2.5.1 Diagrama de autenticación y vinculación de terceros. En la sección de aplicaciones móviles, 4.3.3, se explicará con mayor profundidad el proceso de autenticación con credenciales y la vinculación con terceros. La abstracción del flujo se implementó de la siguiente manera.

Figura 11

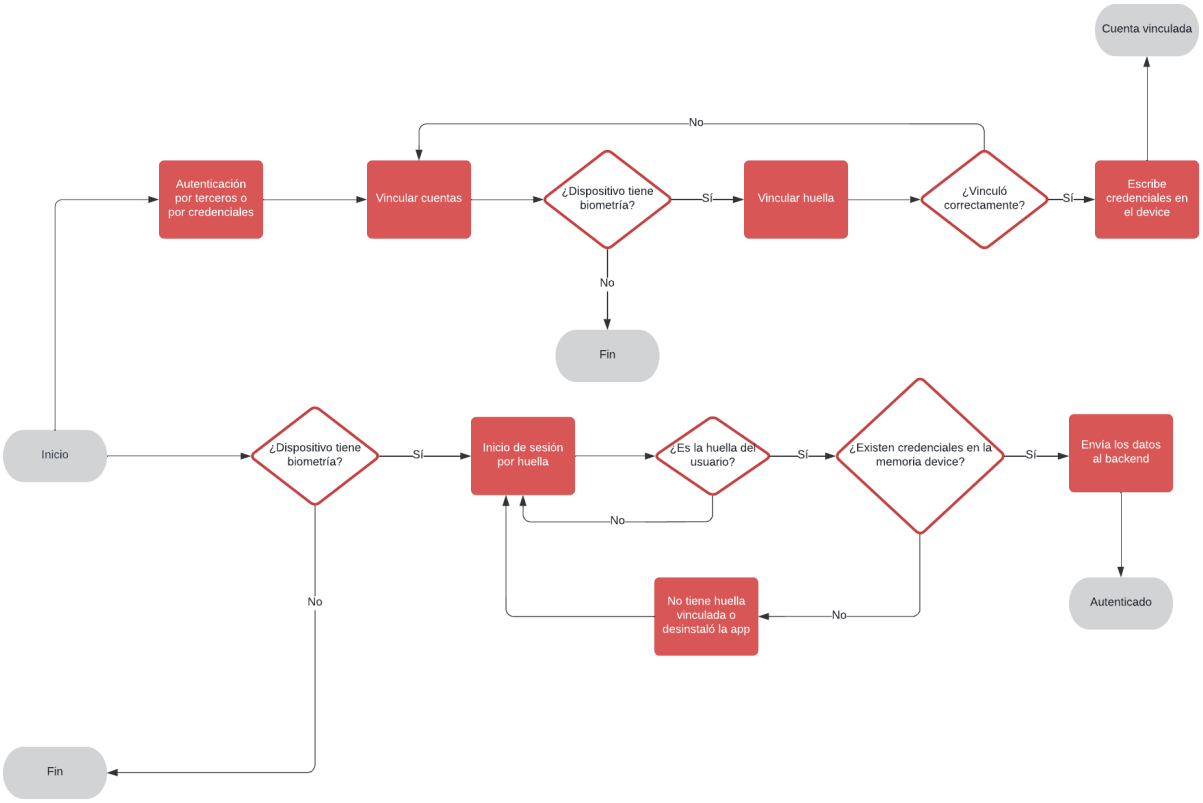
Flujo de autenticación



4.2.5.2 Diagrama de autenticación por huella. La aplicación tendrá la opción para que el usuario pueda vincular su huella digital (si así lo desea), y si el dispositivo lo soporta. El flujo se explica a continuación.

Figura 12

Flujo de autenticación por huella



4.3 Product Backlog

El proyecto se desarrolló con base al marco de trabajo Scrum, por ello se expondrá a continuación el Product Backlog de este, conformado por Epics y el desarrollo de sus incidencias.

4.3.1 Alistamiento del proyecto

En este Epic se realizó el despliegue inicial de los ambientes, y estructura de los proyectos.

4.3.1.1 Arquitectura TI. En esta incidencia se realizaron una serie de pruebas de concepto para la confirmación de las tecnologías que se utilizaron.

4.3.1.2 Estructura del proyecto. Basados en los estándares de calidad definidos por la empresa, se creó la respectiva estructura de proyecto tanto para la plataforma web como para la aplicación móvil.

4.3.1.2.1 Proyecto Node.js. Una parte importante del proyecto fue el desarrollo del backend, debido a que este se encarga de todos los procesos necesarios para que la plataforma web y la aplicación móvil funcionen de forma correcta. Algunos de estos procesos son el control de sesiones, la validación de información que los clientes envían y solicitan a la base de datos.

El backend se desarrolló en el entorno de ejecución Node.js por todas las características que ofrece, como se mencionó en el apartado 3.1.22. Además, se complementó con TypeScript que es un super conjunto de JavaScript.

La estructura del proyecto quedó definida como se muestra en la figura 13, los directorios que se encuentran dentro services corresponden a los microservicios del proyecto. El presente trabajo de grado desarrolló el microservicio de autenticación, usuarios y seguimiento de Google Analytics.

Cada microservicio tiene una estructura similar a la que se expone en la figura 14, esta es una de las más usadas en desarrollo, MVC (Modelo vista controlador), esta arquitectura busca separar el código según la responsabilidad otorgada en cada uno de los archivos.

Figura 13

Estructura de proyecto Node.js

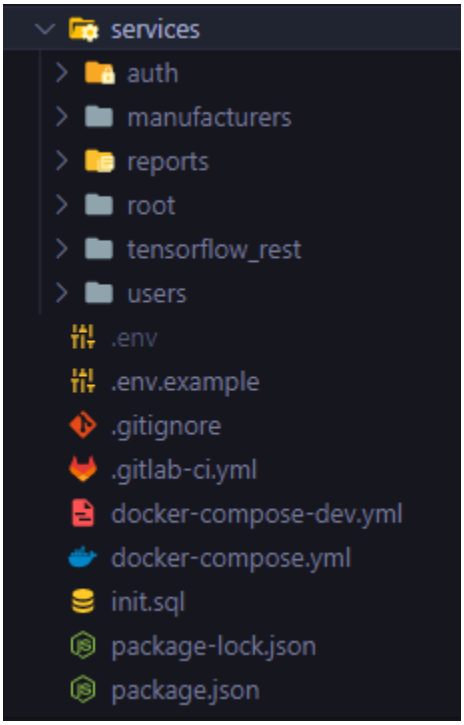
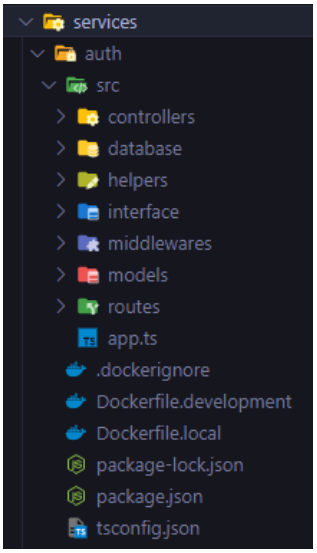


Figura 14

Estructura de microservicios



A continuación, se dará una breve explicación de los directorios definidos en la estructura del microservicio:

Controllers, contienen los controladores que manejan toda la lógica detrás de los parámetros de solicitud de validación, consulta y envió de respuestas con los estados indicados.

Helpers, contiene los bloques de código que resuelven problemas repetitivos. Al definirlos de esta forma se puede acceder a estos desde cualquier parte del proyecto, evitando repetición de código.

Middlewares, contiene los middlewares que son un bloque de código encargado de proteger las peticiones que realiza un cliente al servidor.

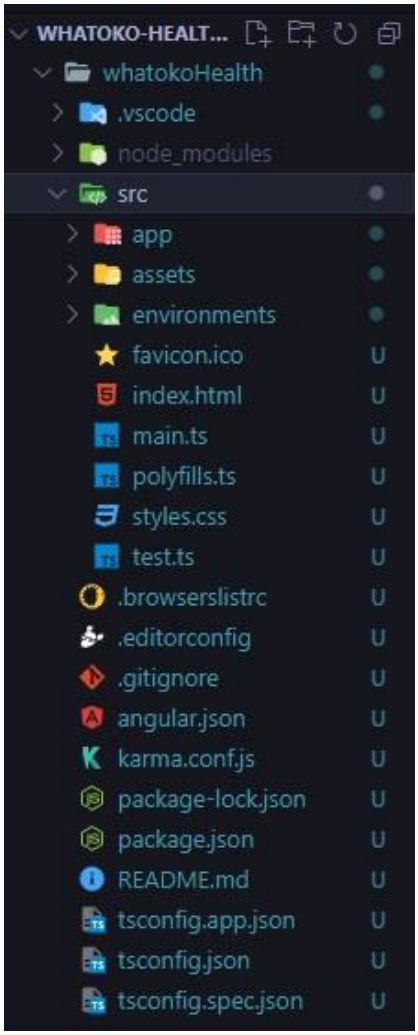
Routes, contiene los todos los endpoints establecidos, creados y funcionales.

Database, contiene el archivo de configuración de la base de datos. Para realizar esta configuración se utilizó el ORM sequelize, ya que este permite agilizar el desarrollo en bases de datos relacionales como la que se usa en el presente proyecto.

4.3.1.2.2 Proyecto Angular. La plataforma web fue implementada en Angular, el framework de JavaScript. Inicialmente se realizó una instalación básica del proyecto y en la sección 4.3.5.1 se muestra el proceso de integración con el template NobleUI - Angular Admin

Figura 15

Estructura de proyecto Angular

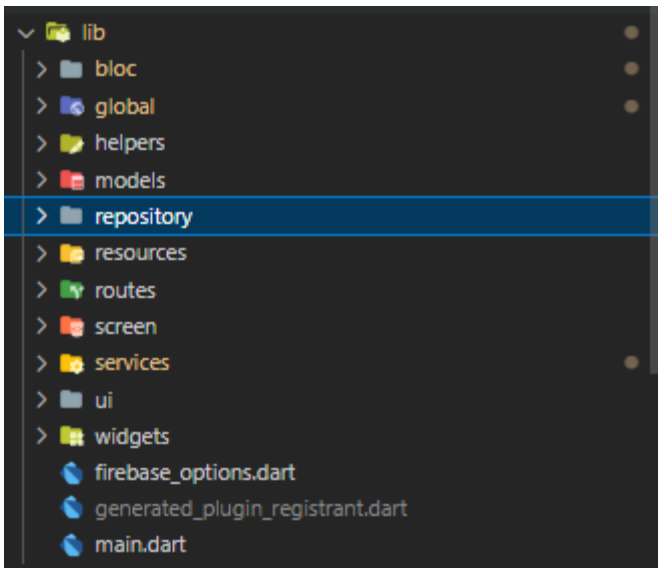


4.3.1.2.3 Proyecto Flutter. Para este proyecto se realizaron las respectivas pruebas en Android y se trabajó con el catálogo de widgets de Material Design, debido a que por cuestión de recursos no se contaba con un iPhone o un dispositivo móvil para implementar el desarrollo iOS, ni tampoco una Mac con Xcode. Sin embargo, con el código implementado la empresa podrá realizar la migración a dispositivos iOS en un tiempo considerablemente menor para que la aplicación móvil sea completamente multiplataforma.

A partir de lo mencionado anteriormente se creó el proyecto de manera local, la estructura de directorios y el nombre de la aplicación, también es importante aclarar que para estándares de la empresa se usó el editor de código Visual Studio Code, y para el manejo de estados de la aplicación el patrón BLoC.

Figura 16

Estructura de directorios móvil



A continuación, se dará una explicación de la funcionalidad general de los directorios raíz de la carpeta lib definidos en la aplicación móvil:

Bloc, Como se mencionó en la sección 3.1.18.1 en este directorio se va a manejar la lógica de negocio para el manejo de estado de la aplicación.

Global, Se encuentran las variables globales.

Helpers, En otros proyectos sería análoga a la carpeta utils, que son ficheros que facilitan funcionalidades requeridas en la aplicación.

Models, Los modelos que convierten y serializan los datos e información al backend se gestionan en este directorio.

Resources, En este directorio se encuentran configuración de los temas y recursos adicionales.

Routes, Las rutas de navegación de la aplicación.

Screen, Las diferentes pantallas de la aplicación.

Services, Repositorios usados en los BLoC's para su funcionamiento.

Ui, Componentes de las interfaces gráficas.

Widgets, Los widgets abstraídos reutilizables.

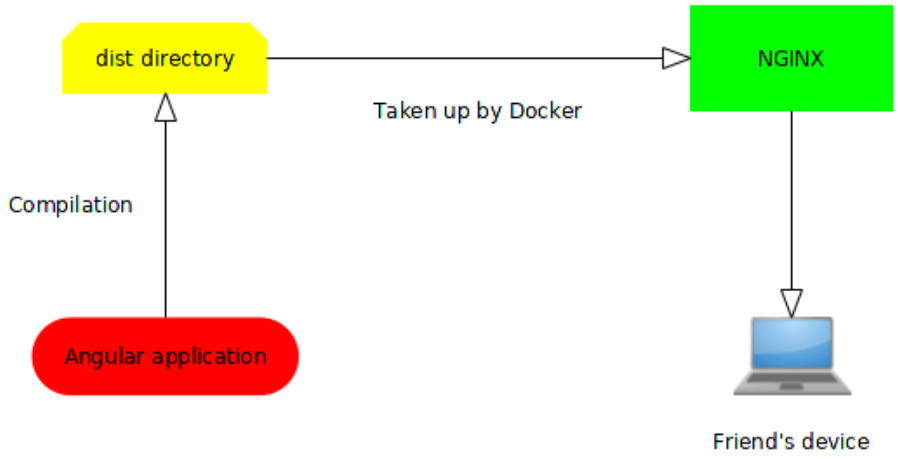
4.3.1.2.4 Docker. El backend del proyecto está orientado a microservicios, por ende, es importante que se realice un adecuado despliegue de este, para lograrlo se debe contenerizar el proyecto, se creó un archivo llamado Dockerfile como el de la figura 18 en la raíz de cada microservicio, este archivo contiene las instrucciones que crean la imagen del microservicio y a partir de esta imagen se construye un contenedor.

Debido a que son varios contenedores se usó la herramienta dedicada a la orquestación de contenedores de cada microservicio, para ello se añadieron un conjunto de reglas en un archivo .yaml, estas reglas corresponden a las configuraciones que cada microservicio requiere para su correcto despliegue.

En el frontend del proyecto también se implementó Docker para realizar el despliegue de angular en el servidor web ngnix, el dockerfile contiene las instrucciones encargadas de compilar la aplicación de angular y generar un directorio de salida que contendrá todos los archivos y carpetas que se alojaran en el servidor de ngnix. El proceso se pude evidenciar mejor en la figura 17.

Figura 17

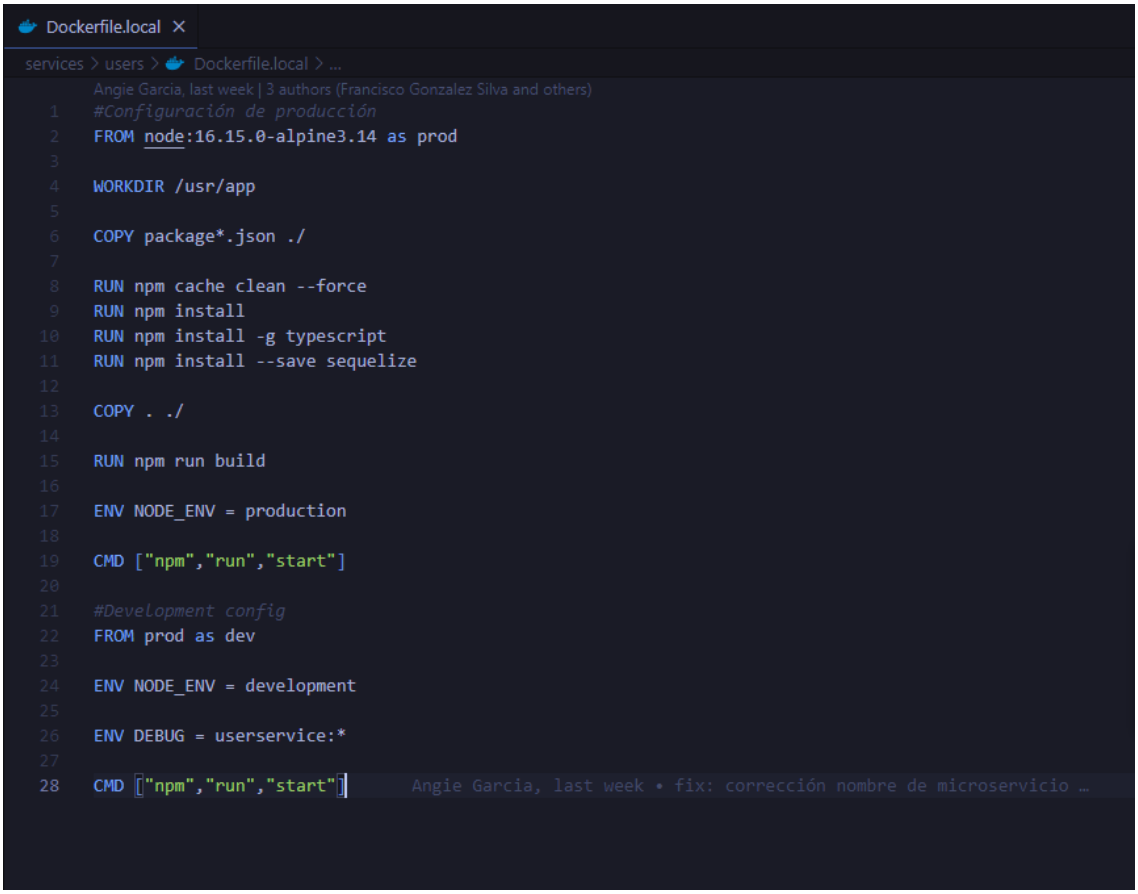
Construyendo y sirviendo una aplicación Angular desde NGINX



Nota. Patrut, C. (2020, febrero 24). CI & CD With Docker and NGINX for an Angular Application. The Startup. https://medium.com/swlh/ci-cd-with-docker-and-nginx-for-an-angular-application-a74ea1027a85

Figura 18

Dockerfile con instrucciones de la imagen del microservicio.



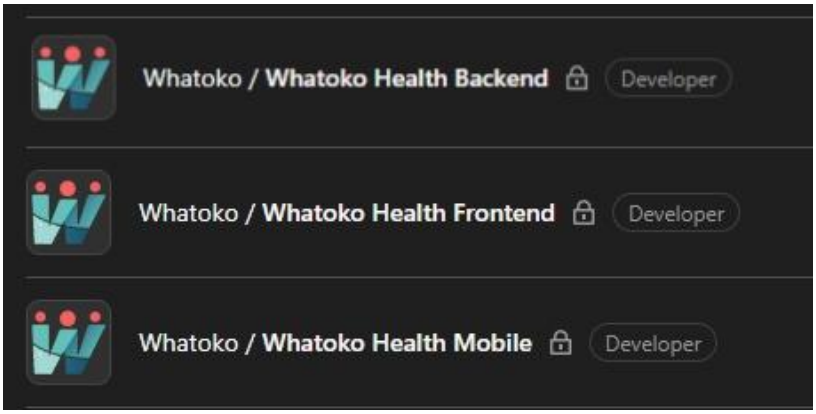
4.3.1.3 Setup CI/CD. En esta incidencia se realizó la configuración de la integración y entrega continuas del proyecto en todos sus ambientes de desarrollo.

4.3.1.3.1 Creación de repositorios. El Product Owner del equipo definió los repositorios en la plataforma GitLab, se trabajó con tres repositorios: Whatoko Health Backend, Whatoko Health Frontend y Whatoko Health Mobile. Se establecieron los accesos y permisos al resto del equipo de desarrollo.

Los proyectos definidos en la sección anterior fueron clonados en sus respectivos repositorios mediante la llave SSH.

Figura 19

Repositorios del proyecto



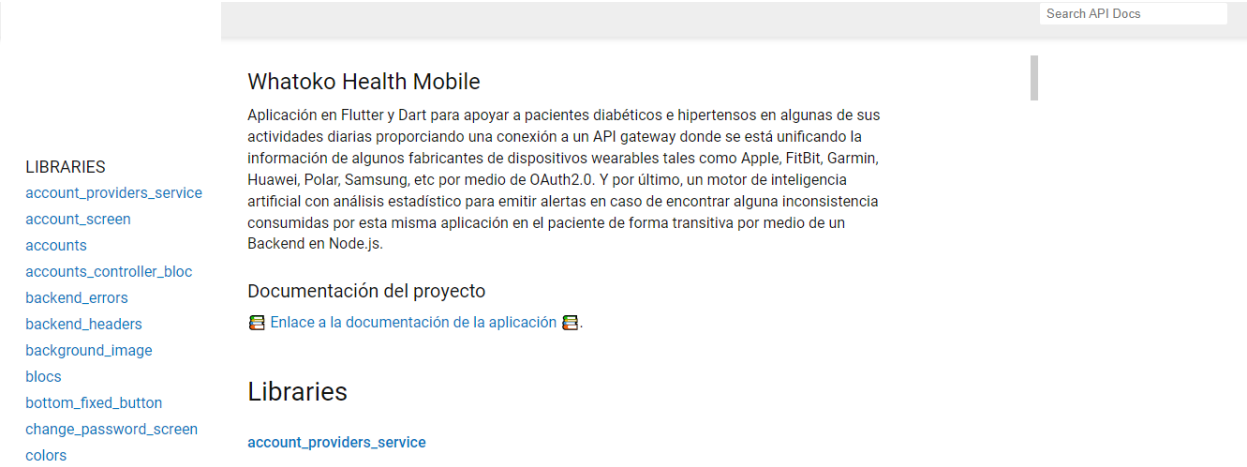
4.3.1.3.2 Creación de GitLab Page para repositorio mobile. Hoy en día el éxito de los proyectos de software está muy ligado a la comunicación y el trabajo colaborativo, por tanto, es fundamental tanto para el mismo desarrollador, como para los colaboradores del proyecto la documentación para ahorrar tiempo en la compresión del código y que así los proyectos sean más eficientes.

Gitlab brinda la opción de ejecutar pipelines con ciertas configuraciones y la opción de almacenar la documentación generada del proyecto como un artefacto por medio de integración continua, donde los miembros del equipo puedan acceder de manera rápida y ubicua.

Por los estándares de calidad de la empresa, para el repositorio de la aplicación móvil se ejecutó un pipeline para generar la documentación automatizada y según el manual de documentación efectiva de Dart (Dart, s.f.), se siguieron una serie de principios, así cada vez que se ejecutó el pipeline, las funciones, atributos o clases de los ficheros en Dart generaban un artefacto con la documentación en el repositorio, donde se puede consultar el catálogo de widgets personalizables usados en la aplicación y las respectivas funciones implementadas, así fueron fácilmente comprensibles a los colaboradores del proyecto para una mayor eficiencia.

Figura 20

Inicio a la documentación automatizada generada con el pipeline



4.3.1.3.3 Creación de GitLab Page para repositorio backend y frontend. Para el caso de la plataforma web y el backend se integró una herramienta de código abierto llamada Compodoc para la generación de documentación automatizada.

Su instalación se hizo agregando un paquete llamado compodoc usando el administrador de paquetes npm y realizando cambios en los archivos de configuración de cada proyecto indicando el nivel de cobertura que tendrá Compodoc.

Como resultado se obtuvo un sitio web con la documentación que incluye funcionalidades de búsqueda y temas agradables. Dicho sitio web se agregó a la configuración y entrega continuas usando la herramienta GitLab pages, permitiendo tener un sitio web estático conectado directamente al código del repositorio.

Figura 21

Documentación de Compodoc en Gitlab Page

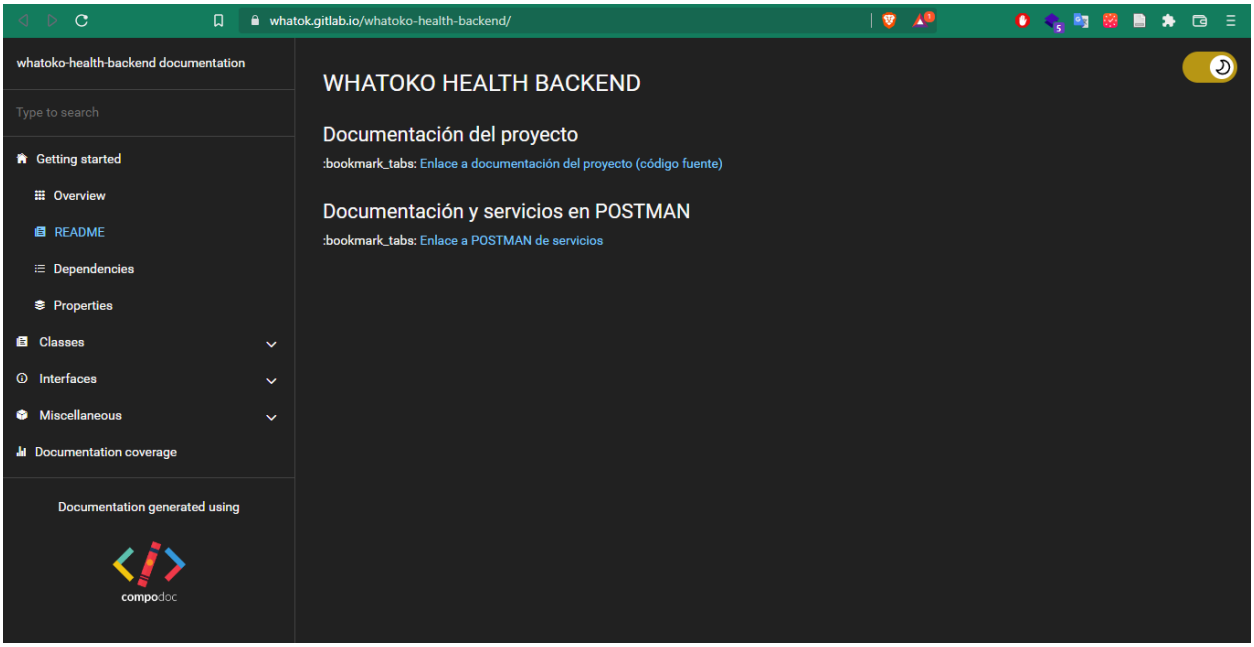
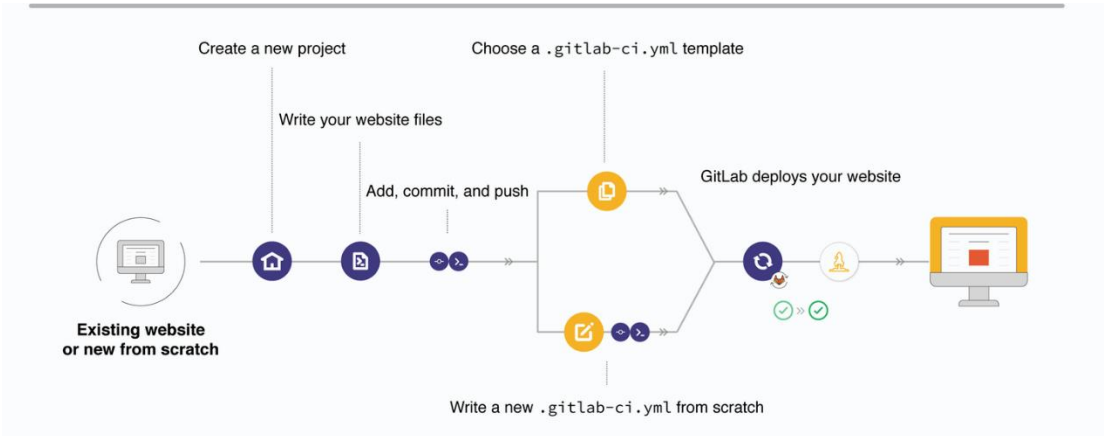


Figura 22

Diagrama de flujo del trabajo con GitLab Pages

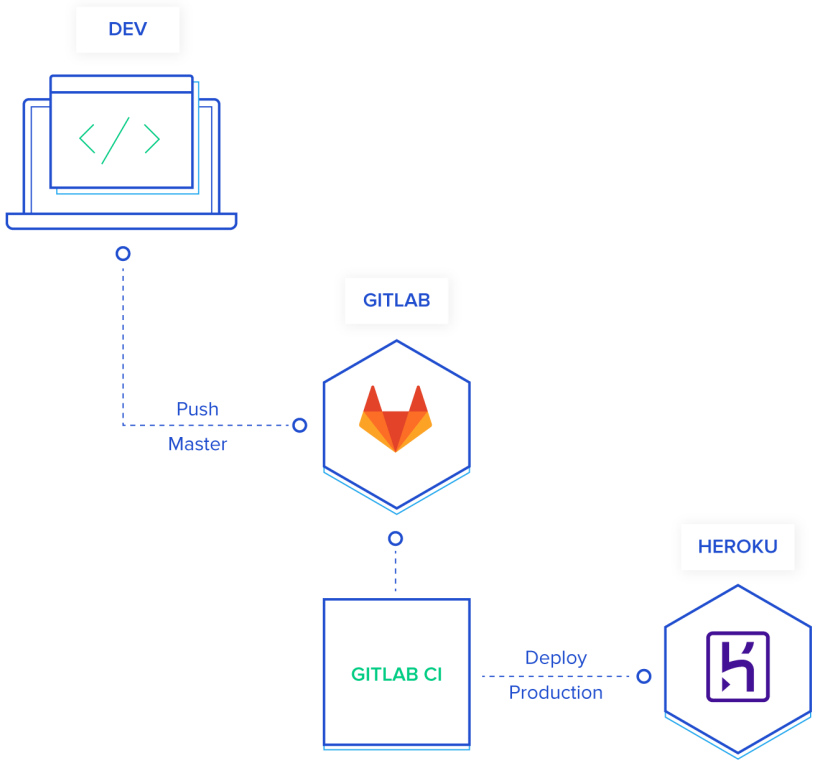


Nota. CI/CD entre GitLab y un sitio web. Tomado de *GitLab Pages | GitLab. (s. f.)*. Recuperado 3 de septiembre de 2022, de https://docs.gitlab.com/ee/user/project/pages/

4.3.1.3.4 Despliegue Heroku. Aprovechando el código fuente que se encuentra en el repositorio de GitLab se hicieron configuraciones para realizar el despliegue de la plataforma web en Heroku, el proceso se resume en la figura 23.

Figura 23

Estructura del despliegue de Heroku con Gitlab CI/CD



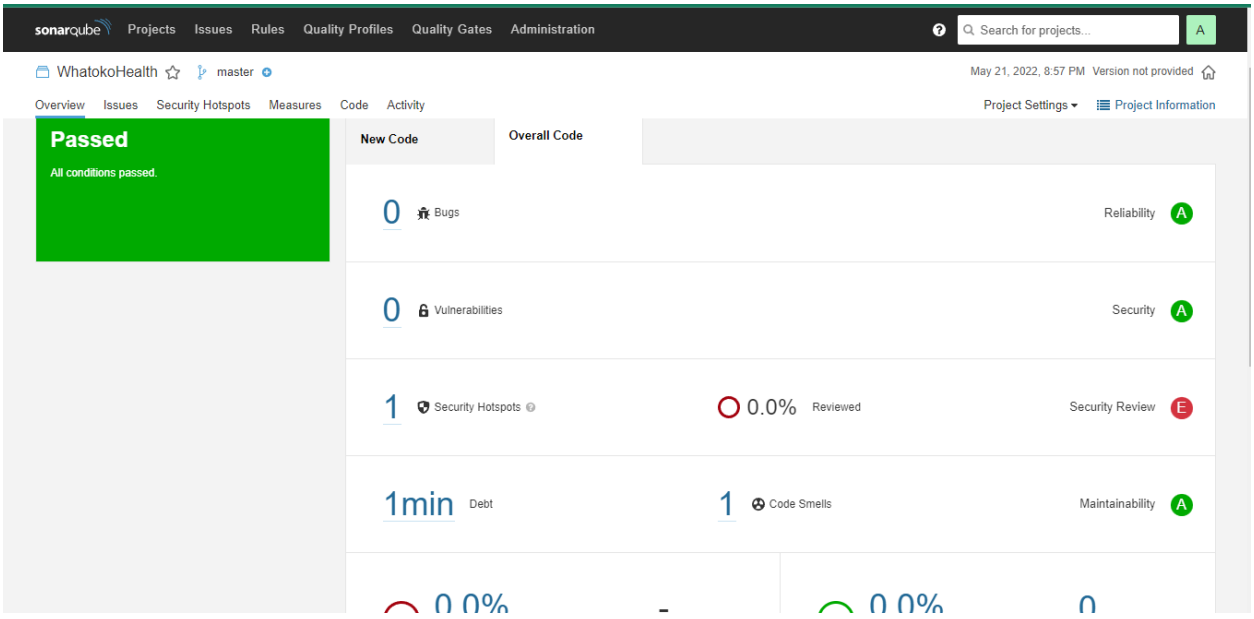
Nota. Estructura del despliegue de Heroku con Gitlab CI/CD. Tomado de *How to Build an Effective Initial Deployment Pipeline. (s. f.). Toptal Engineering Blog. Recuperado 18 de septiembre de 2022, de https://www.toptal.com/devops/effective-ci-cd-deployment-pipeline*

4.3.1.4 Testing. En el proceso de desarrollo de un proyecto software es normal encontrar errores, algunos pueden ser más complejos que otros, para evitar este tipo de inconvenientes se establecieron herramientas que ayuden a identificar estos errores de forma rápida y que midan el nivel de vulnerabilidad que causan. Para el desarrollo del presente proyecto, se desplegaron y documentaron las siguientes herramientas para pruebas en etapas futuras.

4.3.1.4.1 Sonarqube. Es una herramienta de revisión automática de código para detectar errores, vulnerabilidades y malas prácticas en el código. Se integró principalmente con el backend del proyecto permitiendo inspeccionar continuamente cambios, para lograr esto se agregó a la configuración realizada anteriormente en Docker.

Figura 24

Ejemplo de revisión realizada por Sonarqube al proyecto.



4.3.1.4.2 Appium. Es un entorno de pruebas disponible para iOS, Android y Windows para realizar pruebas en aplicaciones nativas, híbridas y web. Su filosofía sigue los siguientes principios:

- No se requiere recompilar la aplicación para realizar las pruebas, sino simplemente seleccionar la ruta del apk.

- El lenguaje de programación es indiferente, es decir, no es necesario escoger un lenguaje de programación en específico y por tanto, ninguna configuración para cada tecnología que se esté utilizando.

- No hay que reinventar la rueda para las pruebas móviles, es decir, ya están las API's y documentación para realizar las pruebas.

- Las pruebas móviles deben de ser a código abierto para facilitar a la comunidad un desarrollo más eficiente (appium, s.f.).

Appium posee una aplicación llamada Appium Desktop para la realización de pruebas con una interfaz y también permite realizar las pruebas por terminal, como se muestra a continuación.

Figura 25

Iniciar Appium por terminal

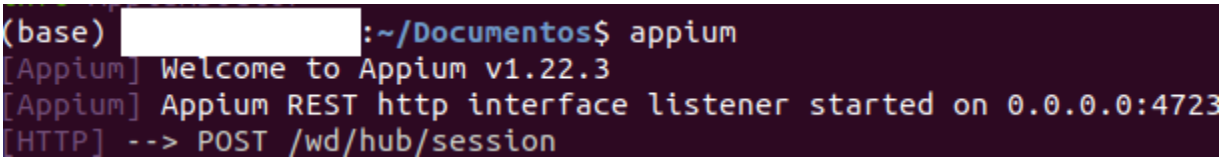
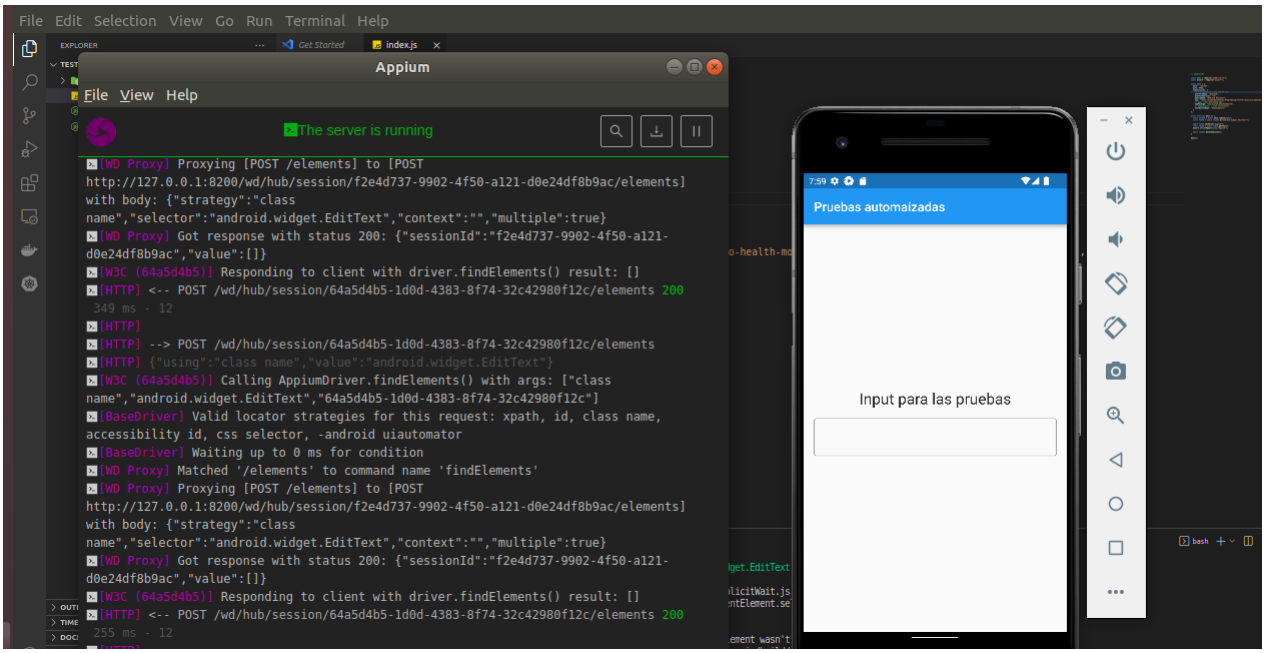


Figura 26

Appium Desktop

Figura 27

Ejemplo de prueba satisfactoria con Appium Desktop



Así se realizó la respectiva configuración en entorno local y su respectiva documentación

para el fácil acceso a los miembros del equipo.

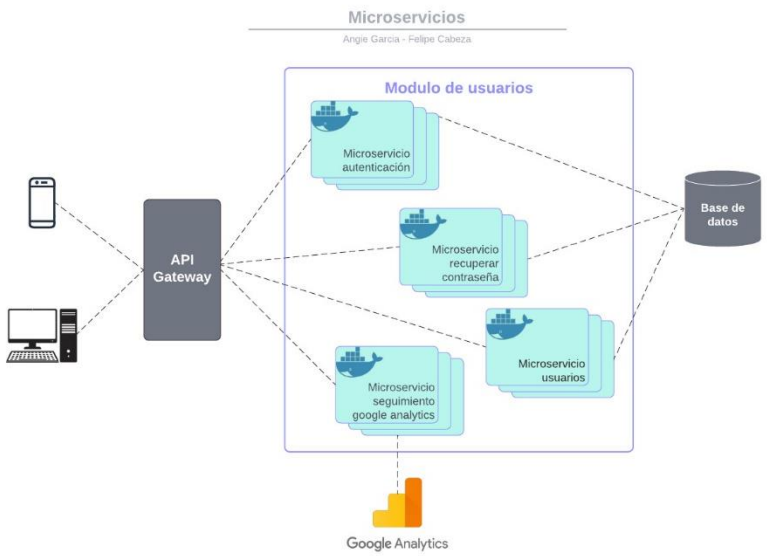
4.3.2 Módulo usuarios

Este módulo agrupa la construcción de microservicios que en conjunción reúnen las acciones necesarias para que se pueda crear, editar, desactivar un usuario y que tenga la opción de autenticar, registrar y recuperar su contraseña. También cuenta con un microservicio que hace seguimiento a las acciones de los usuarios en la aplicación móvil consumiendo los datos desde Google Analytics.

En la figura 28 se observan todos los microservicios que conforman el módulo de usuarios y los otros componentes del proyecto con los que se relacionan. La comunicación entre estos se realiza gracias a la construcción de web services, a través de ellos viaja la información hasta API Gateway y esta a su vez la transfiere a la aplicación móvil y la plataforma web. Esta información se mueve en ambos sentidos.

Figura 28

Estructura microservicios del módulo usuarios.



4.3.2.1 Microservicio de autenticación. Este microservicio contiene servicios que permiten autenticar y registrar un usuario validando diferentes escenarios, además para mejorar la seguridad tanto de la aplicación móvil, como de la página web se integró el uso de JWT (JSON Web Token). Con el manejo de estos se permite validar la identidad de quien ingresa al sistema y controlar el tiempo que va a durar la sesión antes de pedirle que se autentique de nuevo.

4.3.2.1.1 Servicio de registro. Al momento de registrar un usuario su nombre es obligatorio, su correo, número de documento, número de celular, nombre de cuenta de Twitter e identificador de terceros (Google, Twitter, Facebook y Microsoft) son únicos en caso de existir.

Como se mencionó anteriormente existen varios escenarios para tener en cuenta en un registro:

- El usuario se registra con correo y contraseña: En este caso se valida que la contraseña sea de mínimo 8 caracteres y que el correo no exista.

- Registro con celular y contraseña: Nuevamente se debe validar el mínimo de caracteres de la contraseña y verificar que ese número de celular no exista.

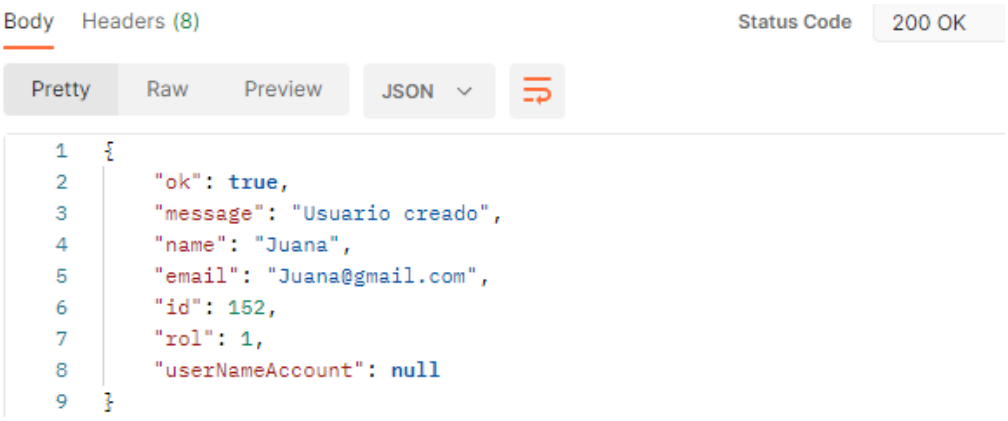
- Registro por terceros: Es el caso del registro por medio de proveedores como Google, Facebook, Twitter y Microsoft, para este tipo de registro se cuenta con un identificador numérico único para cada cuenta y su usuario, permitiendo así que no se puedan registrar varios usuarios con la misma cuenta y que identifique si un usuario ya se encuentra registrado.

Un error frecuente que captura este servicio es el intento de registro de un usuario que se encuentra desactivado, cuando se encuentra con este problema el servicio le sugiere al usuario que siga otro proceso para su activación.

Al finalizar este servicio proporciona una respuesta que indica si fue correcta o no la petición.

Figura 29

Ejemplo de respuesta exitosa de registro en Postman.



4.3.2.1.2 Servicio de login. Cuando un usuario intenta autenticarse puede hacerlo de varias formas:

- Ingreso con correo y contraseña: en este caso se verifica que el correo exista y que la contraseña corresponda.

- Ingreso con número de celular y contraseña: nuevamente se valida que correspondan los datos a un usuario.

- Ingreso por medio de terceros: en este caso lo que debe coincidir es el identificador numérico asociado al usuario que intenta ingresar.

En la figura 30 se observa un ejemplo de los campos que contine el cuerpo del mensaje de un usuario que intenta autenticarse por medio de terceros y en la figura 31 un ejemplo de respuesta incorrecta en caso de un usuario inactivo.

Figura 30

Ejemplo del cuerpo de mensaje en autenticación por terceros en Postman.

Figura 31

Ejemplo de respuesta incorrecta por usuario inactivo en Postman.

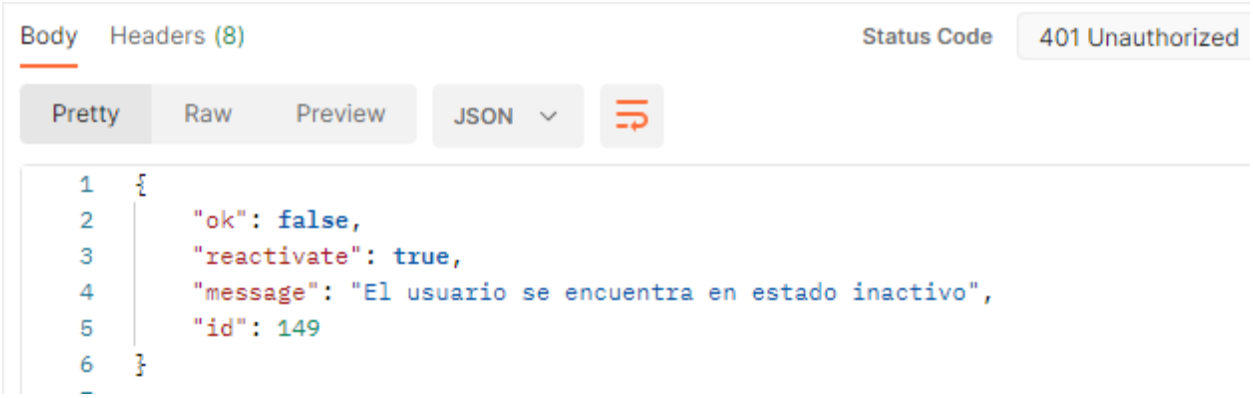


Figura 32

Ejemplo de respuesta correcta en Postman.

4.3.2.1.3 Servicio de validación de token. Este servicio es el encargado de tomar el token existente y verificar que no esté corrupto y que aún siga vigente.

También es frecuente que se presente un error indicando que el usuario que intenta ingresar esta inactivo y nuevamente se le indica que debe seguir el proceso de activación. Cuando la autenticación es exitosa esta retorna un token de JWT para agregar seguridad en el proceso.

Figura 33

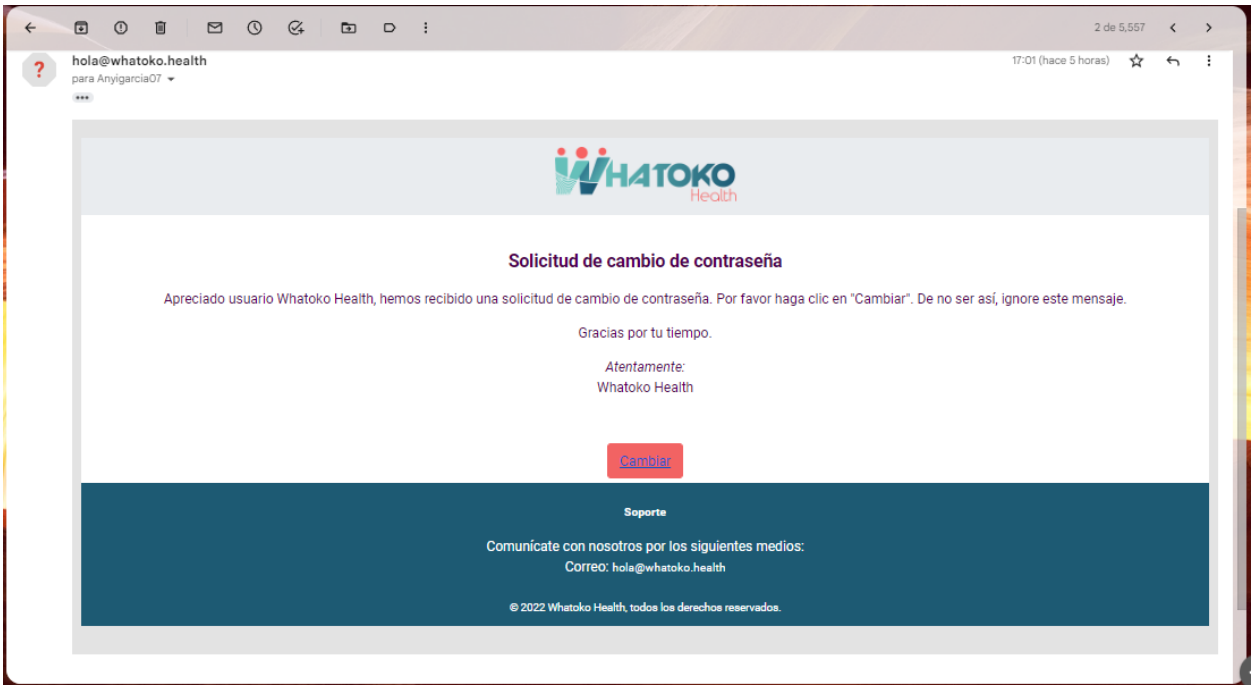
Ejemplo de respuesta incorrecta en validación del token.

4.3.2.2 Microservicio de recuperar contraseña. Está encargado de permitirle a un usuario recuperar su contraseña.

4.3.2.2.1 Servicio encargado de enviar el correo electrónico. Se realiza una validación por medio del correo electrónico, el usuario debe ingresar el correo que tiene registrado en el sistema, el servicio validará que el correo exista y que también sea un usuario activo. Luego, mediante el uso de un módulo de node.js llamado Nodemailer se enviará un correo como el de la figura 34.

Figura 34

Ejemplo de correo de recuperación de contraseña



4.3.2.2.2 Servicio encargado de cambiar contraseña. El correo que envía el servicio del paso anterior contiene un link que proporciona acceso al servicio encargado de cambiar contraseña, al acceder direcciona a una nueva página que permite el ingreso de la nueva contraseña y su confirmación, el backend valida el link y luego asigna al usuario la nueva contraseña.

4.3.2.3 Microservicio gestión de usuarios. Este microservicio gestiona la creación, obtención, edición e inactivación de usuarios, es decir el CRUD del proyecto. Inicialmente se mapearon los datos suministrados por la base de datos del proyecto, para lograrlo se crearon modelos correspondientes a las tablas de esta con el uso del ORM sequelize, los modelos implementados se pueden ver en la figura 35. En la figura 36 se evidencia un ejemplo de la construcción del modelo de usuarios.

Figura 35

Modelos del microservicio usuario.

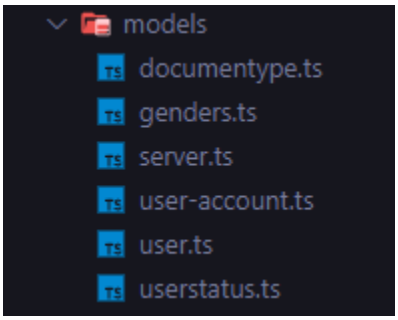


Figura 36

Ejemplo de la construcción del modelo usuario

Adicionalmente se encuentran otros servicios encargados de tareas más específicas relacionadas con el usuario, los cuales se expondrán a continuación.

4.3.2.3.1 Servicio encargado de vincular y desvincular cuentas de terceros. Este servicio permite a usuarios que estén autenticados asociar sus cuentas de terceros, cuenta con validaciones como el no permitir relacionar las cuentas de otros usuarios como en la figura 37 y pasar el estado del usuario a inactivo cuando desvincula su última cuenta y su registro fue por terceros.

Figura 37

Ejemplo de respuesta incorrecta del servicio en Postman.

4.3.2.3.2 Servicio encargado de consultar las cuentas de terceros de un usuario. Este servicio permite saber las cuentas de terceros asociadas a un usuario, nuevamente para acceder a este servicio el usuario debe estar autenticado.

4.3.2.3.3 Servicio encargado de activar un usuario. En el caso de que un usuario que haya desactivado su cuenta quiera volver, este servicio le permitirá volver a quedar activo y podrá tener acceso de nuevo a los demás servicios.

4.3.2.4 Microservicio de seguimiento Google Analytics. Se implementó con la finalidad de realizar seguimiento a las acciones del usuario en la aplicación móvil, estas acciones son inicialmente almacenadas por Google Analytics. El microservicio realiza una conexión con su API e inicia con el consumo de información para luego mapearla y entregarla de forma estándar a los demás componentes del proyecto.

4.3.3 Aplicación móvil

En este Epic se implementaron principalmente las pantallas de la aplicación, su conexión con Firebase y su proceso de autenticación.

4.3.3.1 Conexión con Firebase. A pesar de que el manejo de las sesiones, el envío y consumo de datos e información de los usuarios y dispositivos wearables se realizó por medio del backend, para este proyecto se requirió el uso de Firebase para un par de funcionalidades, las cuales incluyeron lo siguiente:

La autenticación con terceros (Facebook, Google, Twitter y Microsoft) que se muestra en la sección 4.3.3.3, pero con el fin de verificar que el usuario que está ingresando a la aplicación sea real y facilitando el proceso de registro e inicio de sesión en la aplicación. Si el usuario no desea registrarse con el formulario nativo de la aplicación, tiene la opción de iniciar con terceros y poder acceder a las funcionalidades de la aplicación de una manera más rápida y sencilla.

La integración continua con App Distribution que es una herramienta de Firebase para enviar al correo electrónico de ciertos colaboradores del equipo, una versión de la aplicación para realizar las pruebas de manera continua. Por estándares de calidad de la empresa y por medio de integración continua, se configuró un pipeline para que cada vez que se añadiera una funcionalidad y se fusionara con la rama establecida por la empresa en el repositorio se creara un .apk con una nueva versión de la aplicación, con el fin de que el equipo realizara pruebas y reportara fallos. Para este proyecto, la empresa cuenta con un equipo de QA los cuales se encargaban de examinar minuciosamente los detalles de la aplicación, reportar fallos y realizar retroalimentación para una mejora continua y mejor control de calidad en el producto.

Además, Firebase también posee una herramienta llamada Firestore Collection y Real Time Database, las cuales son unas bases de datos no relacionales que permiten el almacenamiento de información, pero para este proyecto la información de los usuarios y los dispositivos wearables no se decidió almacenar allí, sino en una base de datos propia gestionada por el backend.

Por último Google requiere el uso de una firma digital con una llave SHA-1 y/o SHA-256 para conceder a la aplicación ciertos permisos y que pueda gozar de todas las funcionalidades, pero por defecto la ruta de esta llave es la ruta de la máquina local y así el despliegue solo va a servir localmente, es decir, que para que los otros colaboradores pudieran utilizar de manera eficiente la aplicación sin ninguna inhibición, fue necesario realizar la configuración del keystore para que la huella digital no fuera configurada con la ruta de la máquina local y las pruebas de integración continua fueran fácilmente desplegables a todo el equipo.

4.3.3.2 Inicio de la aplicación.

En el diseño de las pantallas para Flutter se realiza por medio de widgets mediante código, es decir, no se usan herramientas para el diseño de las pantallas.

Para el desarrollo profesional de aplicaciones móviles es necesario un proceso de maquetación de wireframes y/o mockups antes de proceder al desarrollo de las pantallas.

Hoy en día una de las herramientas más populares para el proceso de prototipado es Adobe XD, herramienta proporcionada por la suite de Adobe que es bastante robusta y fácil de usar.

Para este caso el equipo de diseño del proyecto proporcionó los wireframes con los temas, los colores y las pantallas incluyendo como requerimiento el diseño responsivo de las pantallas en cada dispositivo.

A continuación, se muestra el archivo proporcionado por el equipo de diseño para el sprint y cómo fue su resultado final en Flutter.

Figura 38

Archivo de Adobe XD con primeros wireframes de la aplicación

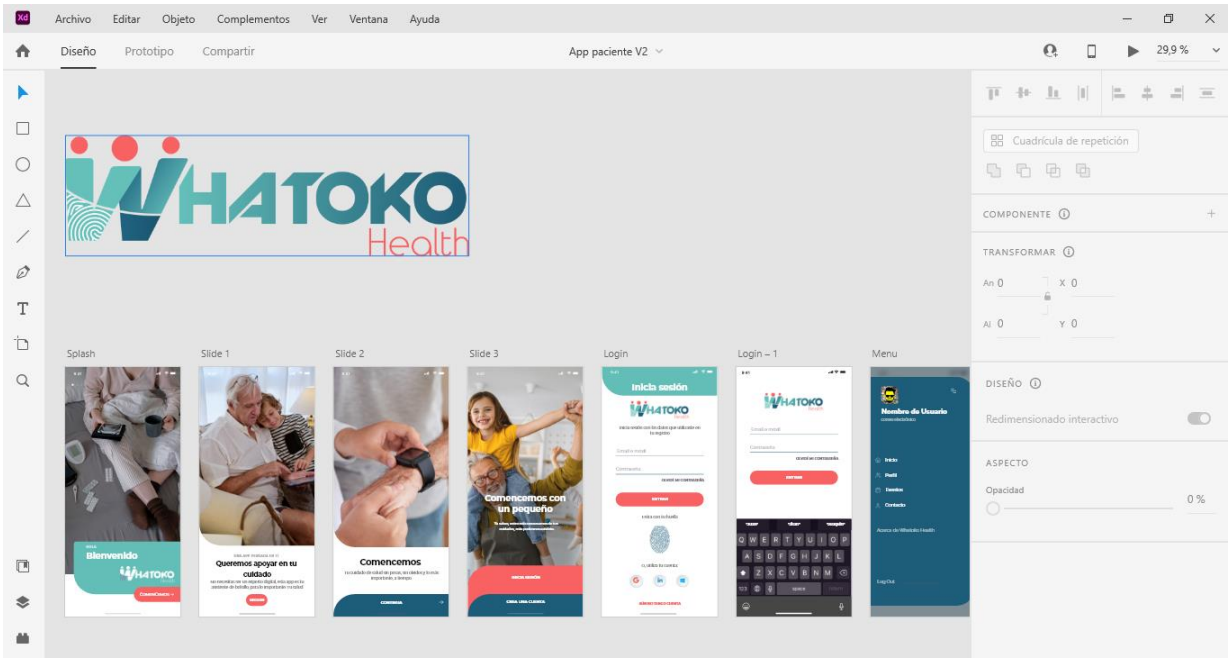


Figura 39

Pantalla de splash en Flutter en el emulador de Android Virtual Device (AVD)

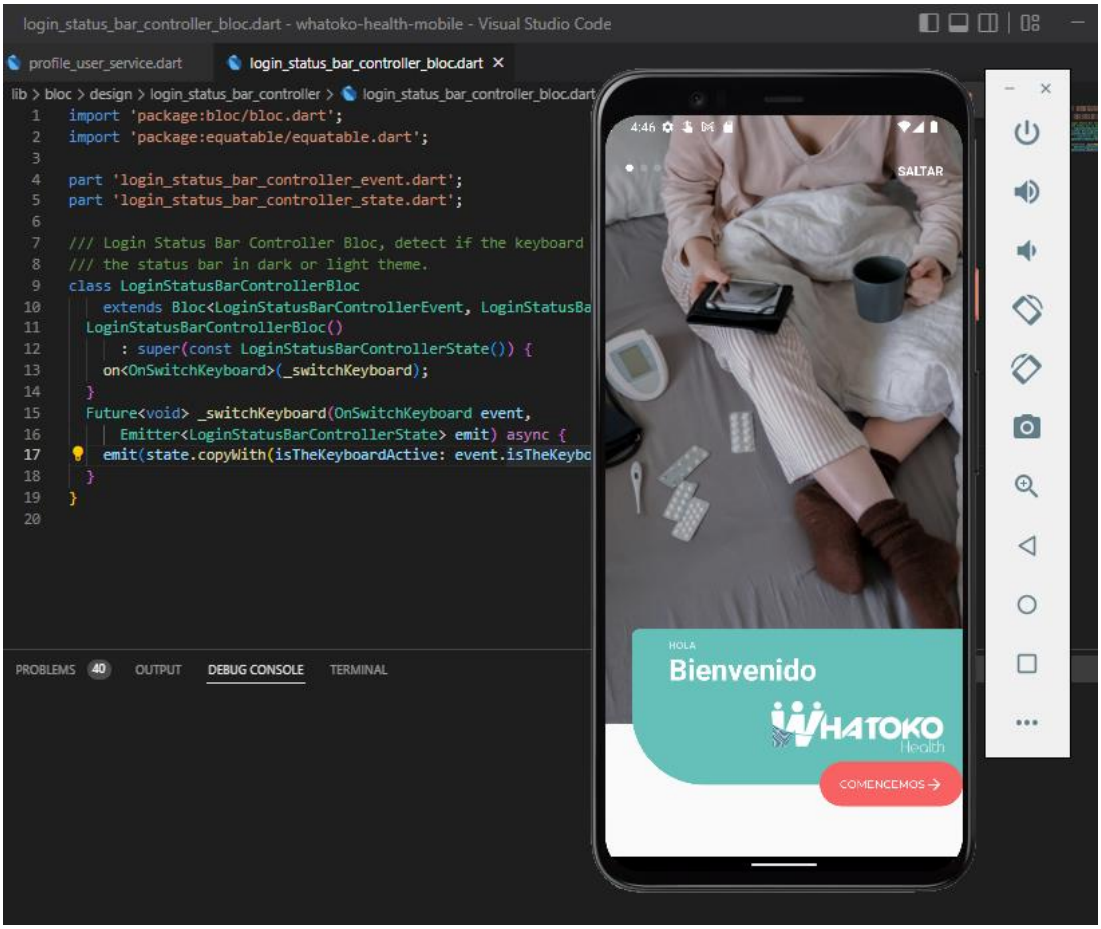


Figura 40

Pantalla de Slide 3 en Flutter en el AVD

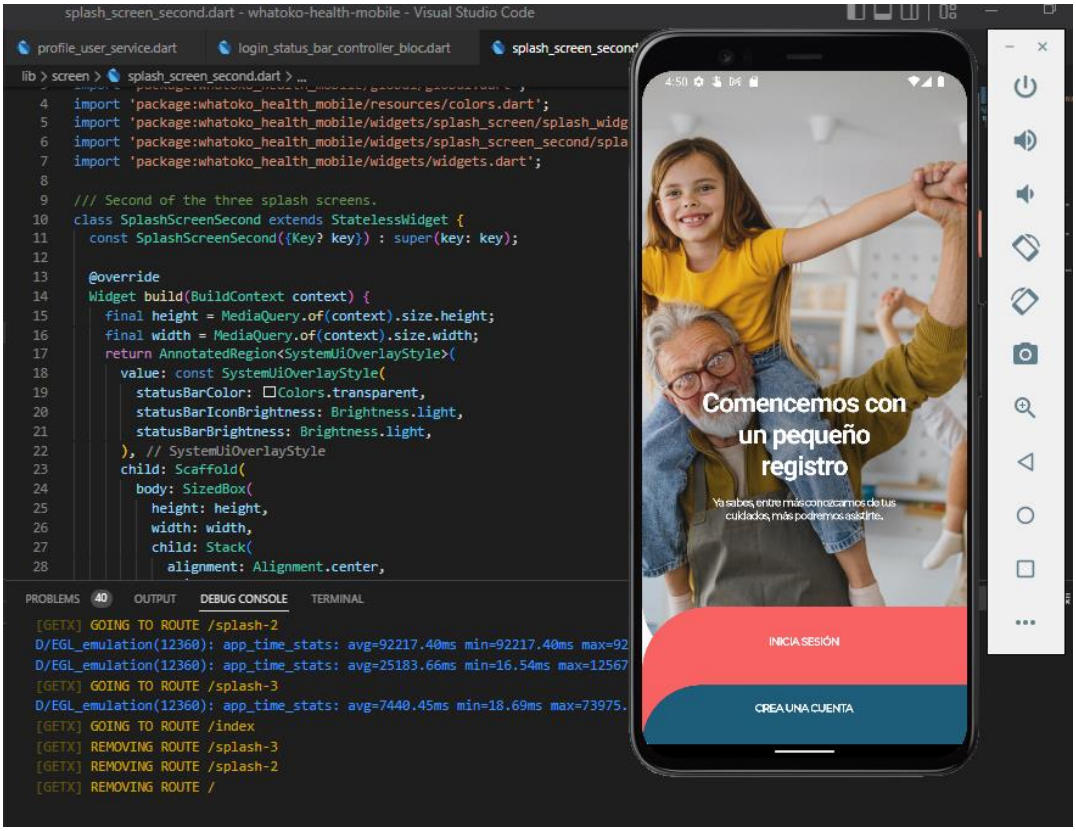


Figura 41

Pantalla de Login en Flutter

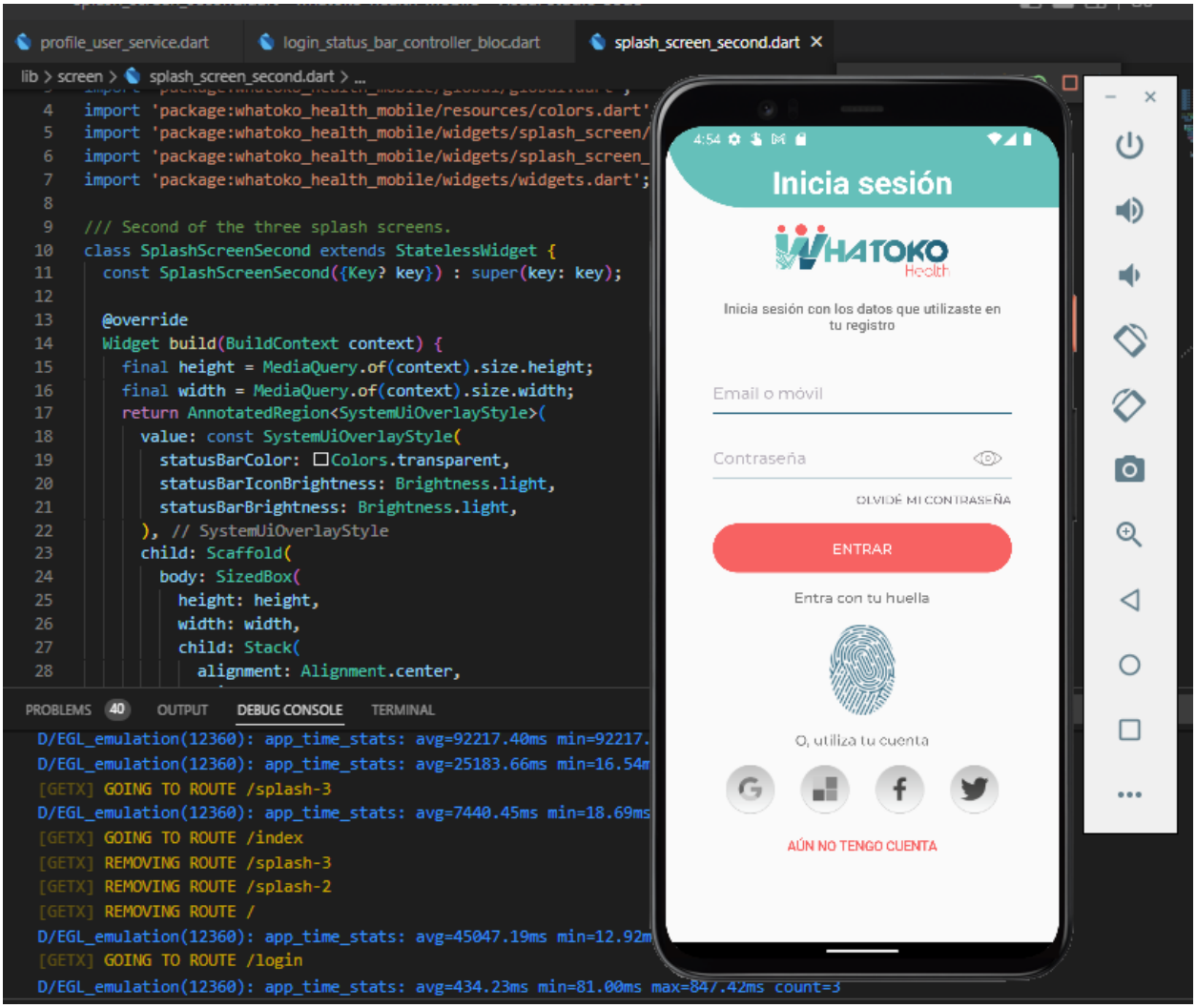


Figura 42

Pantalla de Login – 1 en AVD

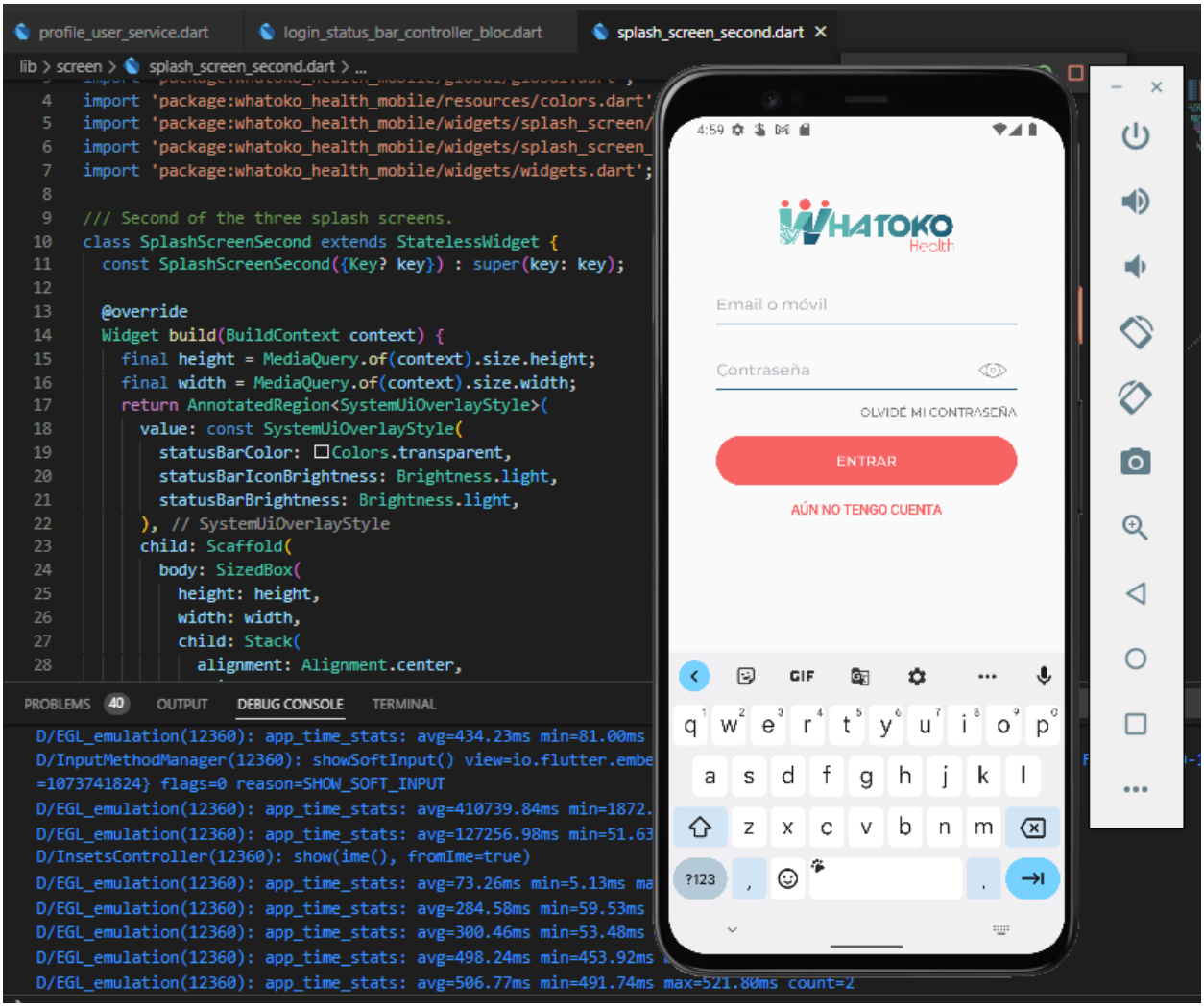
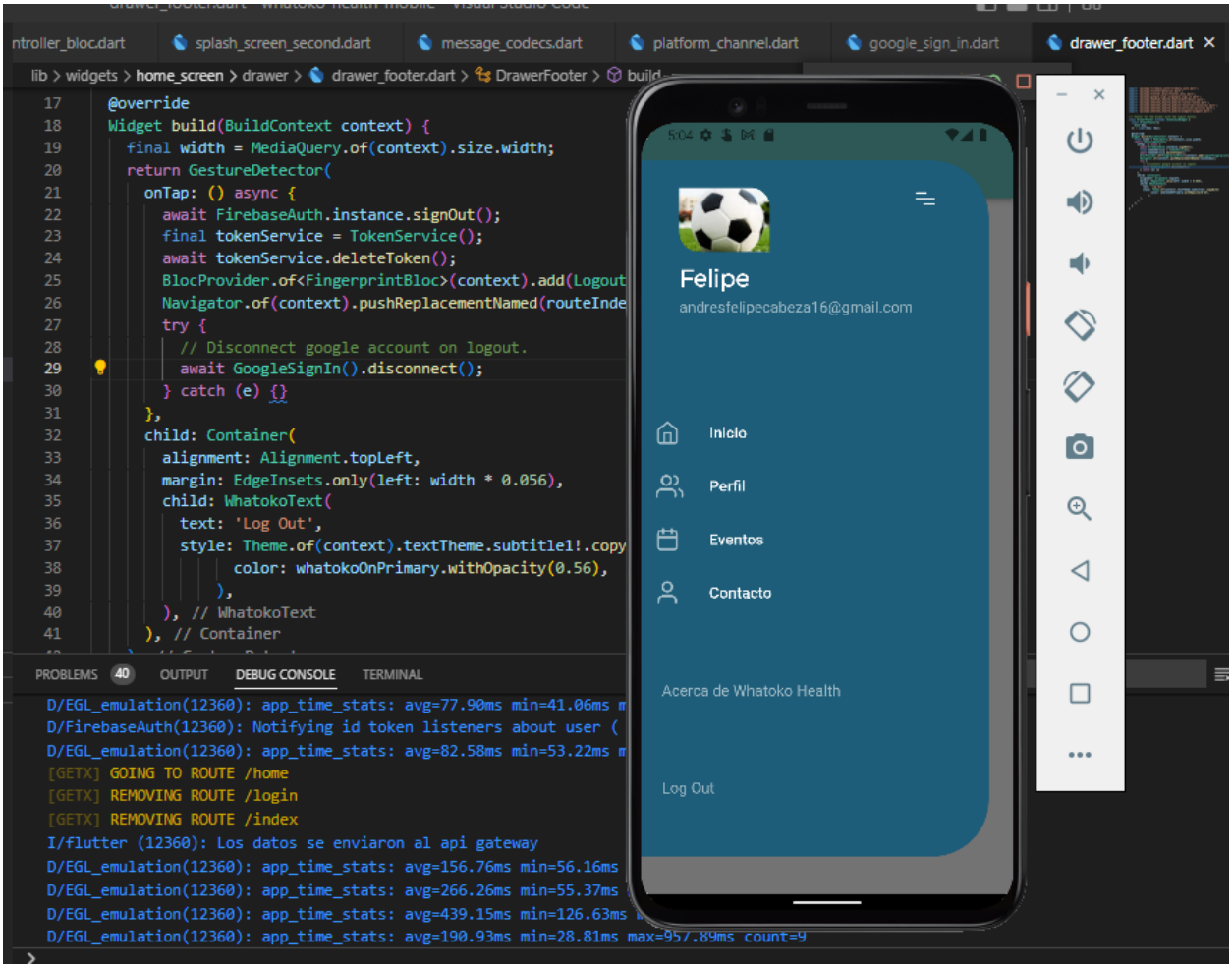


Figura 43

Pantalla de Menú de inicio en Flutter con un usuario con foto de perfil



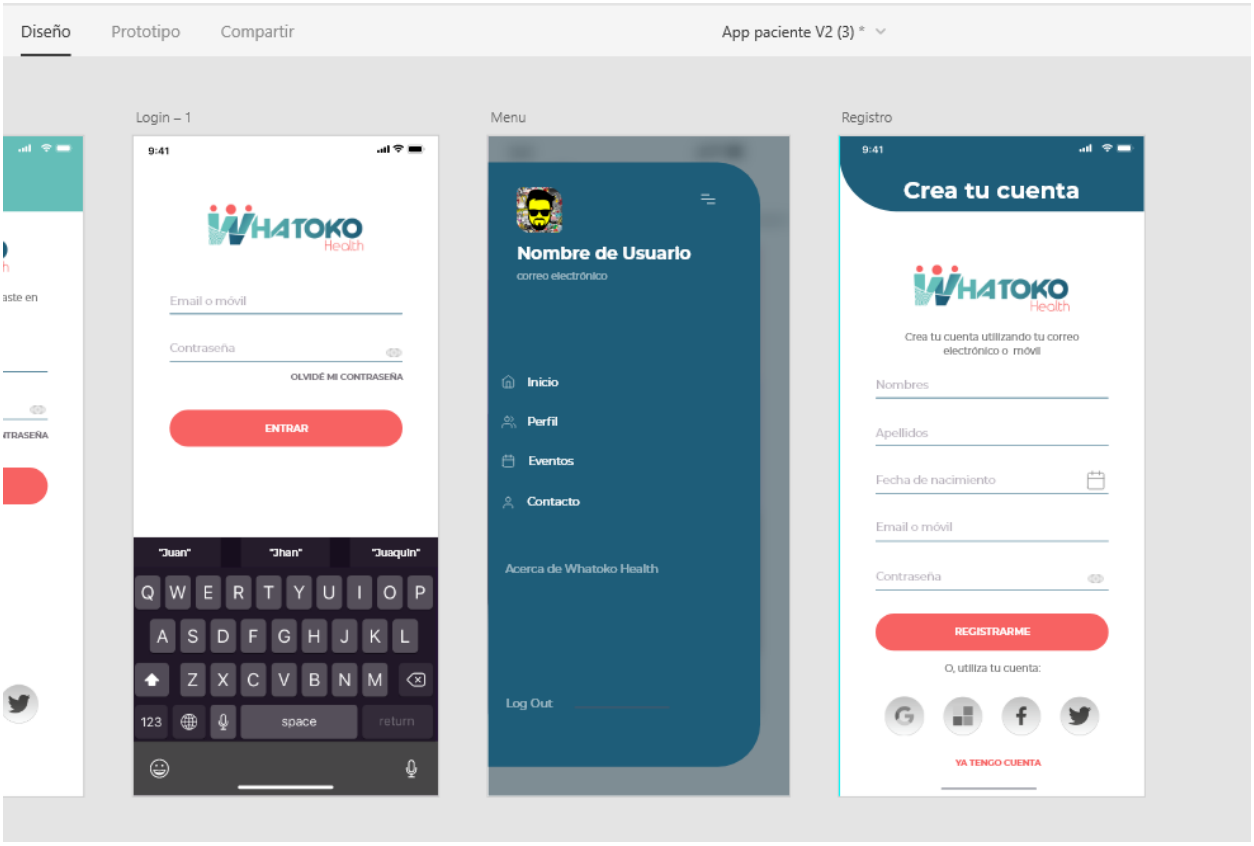
Algo quizás no evidente es que en la pantalla de inicio de sesión, se están realizando una serie de acciones cuando el usuario despliega u oculta el teclado, por ejemplo, en la parte superior donde se muestran las notificaciones y la hora del dispositivo, se evidencia que el tema es claro, pero en la pantalla de inicio de sesión al desplegar el teclado, si continuara con el tema claro, no se vería ya que se solaparía con el color de fondo, así que se muestra con el tema oscuro, mismo que se muestra en el archivo XD, y al ocultar el teclado debe volver al tema claro, por eso fue necesario implementar un controlador para escuchar cuando el teclado estaba desplegado o cuando

no estaba visible para ir cambiando el Status bar (Material Design, s.f.) en donde no se solape con el color de fondo.

4.3.3.3 Registro de la aplicación. A partir de otra pantalla en el archivo XD, similar a la del inicio de sesión, pero ahora para registrar el usuario se solicitó un formulario con los siguientes datos: correo electrónico o teléfono, nombres, apellidos, contraseña y fecha de nacimiento. Para realizar el proceso de registro, el usuario puede hacerlo llenando los campos del formulario, o por medio de terceros, los cuales incluyen a Facebook, Twitter, Microsoft o Google.

Figura 44

Formulario de registro en Adobe XD

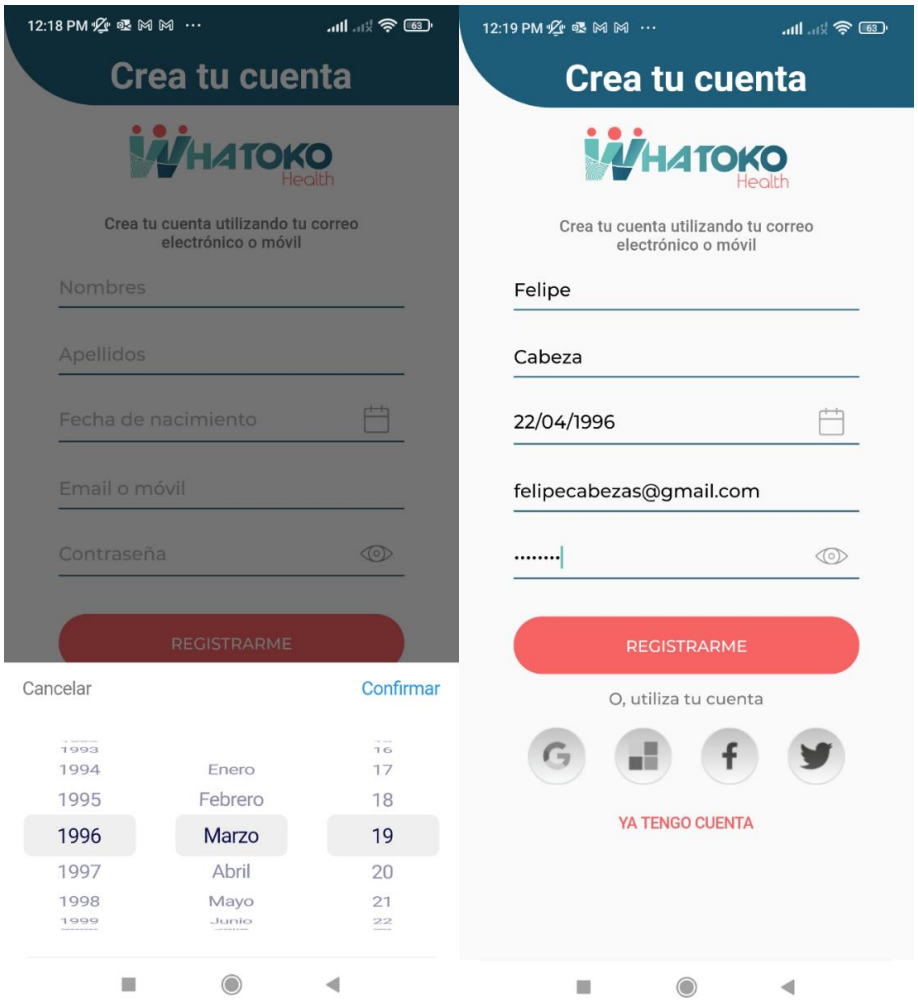


En el caso que el usuario quiera registrarse por medio del formulario, se hará la validación de los campos, por ejemplo, en los nombres y apellidos solo se pueden ingresar caracteres del

abecedario, es decir, no se aceptan dígitos ni símbolos, para la fecha de nacimiento se escogió un plugin, el cual permite hacer scroll para seleccionar el día, mes y año de nacimiento, y validaciones como que el usuario tenga una edad mínima, además el plugin inicialmente estaba configurado en inglés, así que se tuvo que hacer la respectiva configuración para que el plugin funcionara en español y por último la contraseña debe tener al menos 8 dígitos.

Figura 45

Fecha de nacimiento en pantalla de registro y pantalla de registro



Nota. Con el ícono de mostrar la contraseña se puede establecer si quiere que sea visible o no.

Otro punto importante, para mejorar la experiencia de usuario fue una serie de mensajes para notificar el estado de la aplicación, e indicadores en caso de que la aplicación estuviera cargando, cuando se ejecutara alguna operación que requiriera algún tiempo considerable.

Figura 46

Indicador de cargando y envío de formulario con datos erróneos

Nota. Las validaciones se realizan en el campo que esté el usuario actualmente y en caso de que alguna de estas falle, no lo dejará avanzar.

Para el registro por medio de terceros se realizó un proceso con el siguiente flujo: Se selecciona el icono del tercero a registrar, luego se muestra un webview de la aplicación por medio de un plugin, el cual redirige al usuario a la pantalla de autenticación del tercero donde el usuario diligencia las credenciales de acceso, si son correctas, posteriormente se mostrará un mensaje al usuario si desea aceptar el uso de la aplicación, el cual contiene los términos de uso y la política de privacidad, si el usuario acepta se regresa un token a Firebase y este entrega unos datos básicos del usuario a la aplicación. Este proceso de autenticación se implementó por medio de OAuth (OAuth, s.f.), el cual se explica en la sección 4.3.4 de conexiones más a detalle para cada tercero, sin embargo, en esta sección es importante aclarar que el usuario se autenticó y es real, por lo tanto, se entregan unos datos a Firebase que provee el tercero, y estos datos son enviados al backend, posteriormente el usuario se registra exitosamente si sigue este proceso. Si el usuario diligencia mal las credenciales, no acepta el uso de la aplicación, cancela el proceso o ya se encuentra registrado, el proceso de registro no va a ser exitoso.

Por último, la aplicación controla otros mensajes de error, por ejemplo, cuando el usuario se intenta registrar y no tiene conexión a internet, muestra un mensaje advirtiendo que no tiene conexión, atrapando la excepción y evitando que la aplicación falle.

Figura 47

Inicio de sesión o registro por Microsoft y Facebook

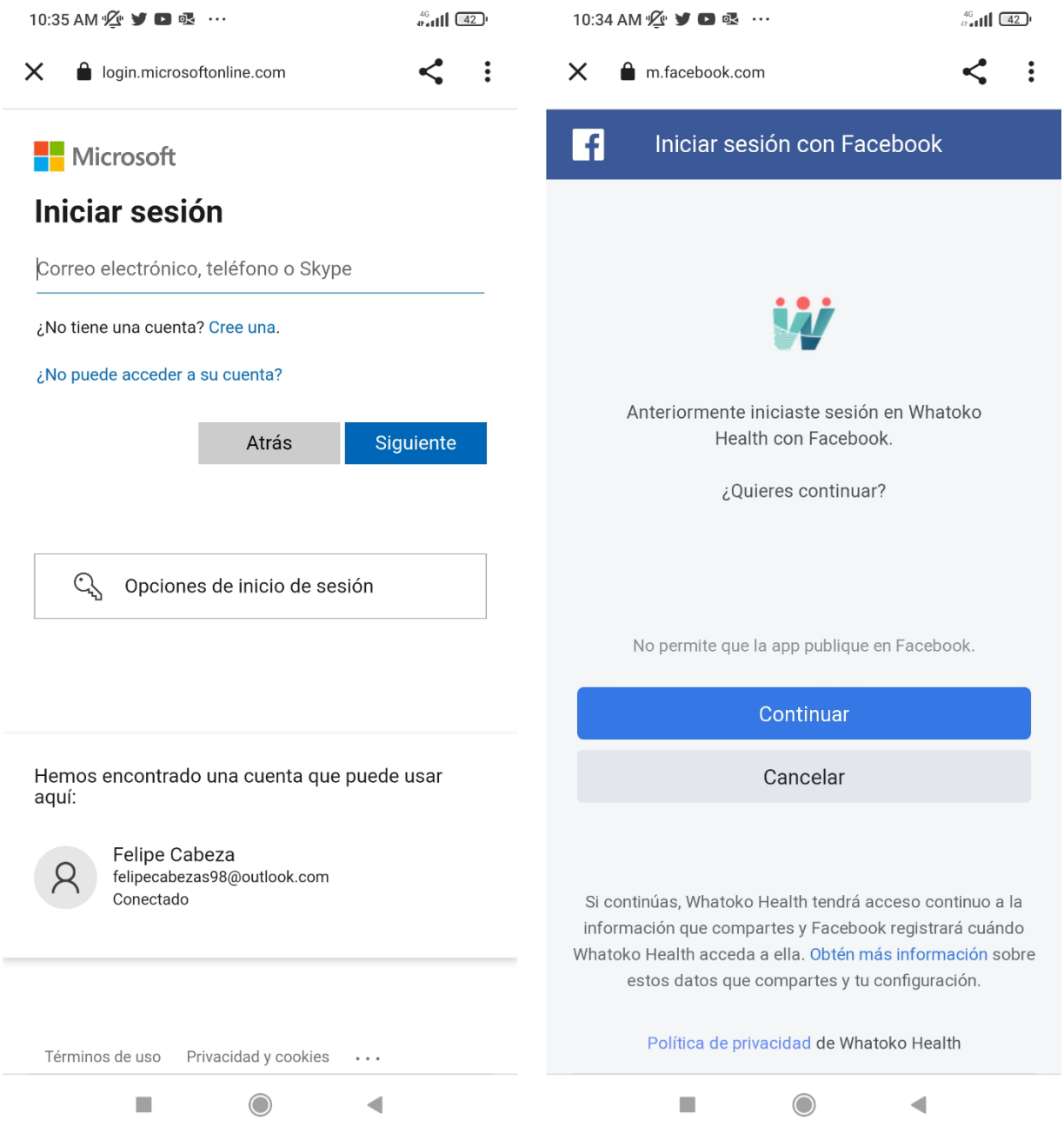
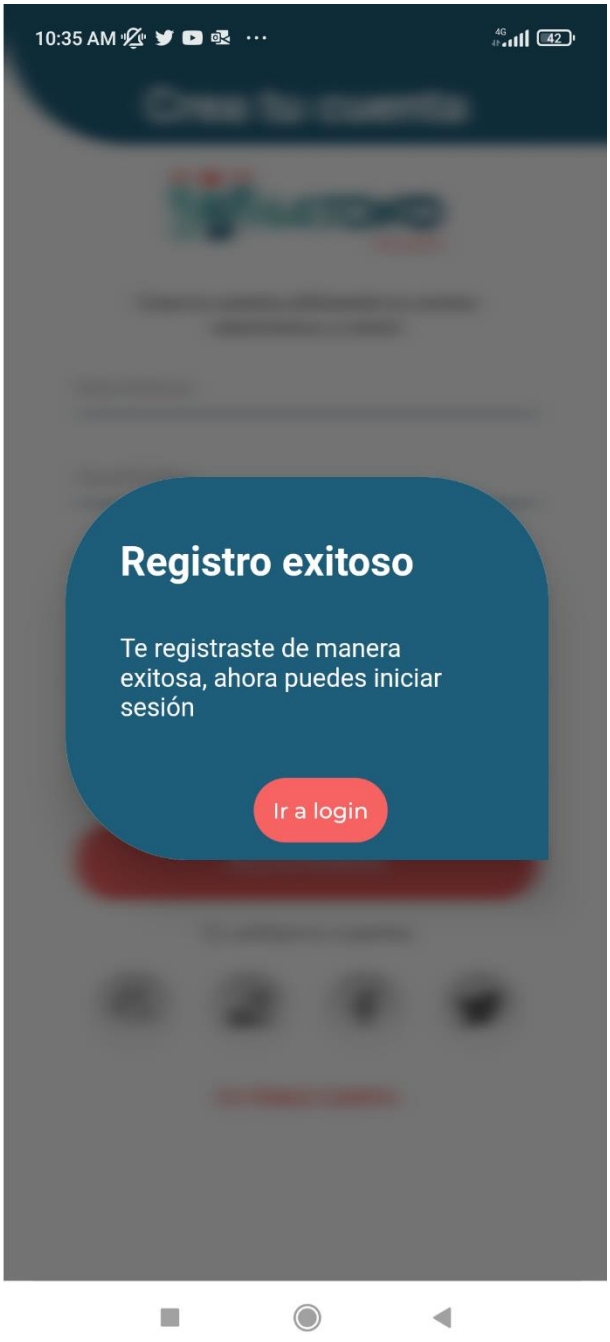


Figura 48

Inicio de sesión o registro con Twitter y modal de registro exitoso



4.3.3.4 Mi perfil de la aplicación. En esta sección de perfil primero se hizo la maquetación con datos del usuario ficticios inspirados también del Adobe XD, esta pantalla contiene en la parte superior los datos que suministra el usuario, ya sean suministrados por medio del tercero que se registró o por el formulario nativo de la aplicación, cabe aclarar que no todos los terceros proveen los mismos datos, ya que por ejemplo en Twitter está la opción de registrarse sin correo electrónico, o entrega el nombre de usuario que posee en Twitter.

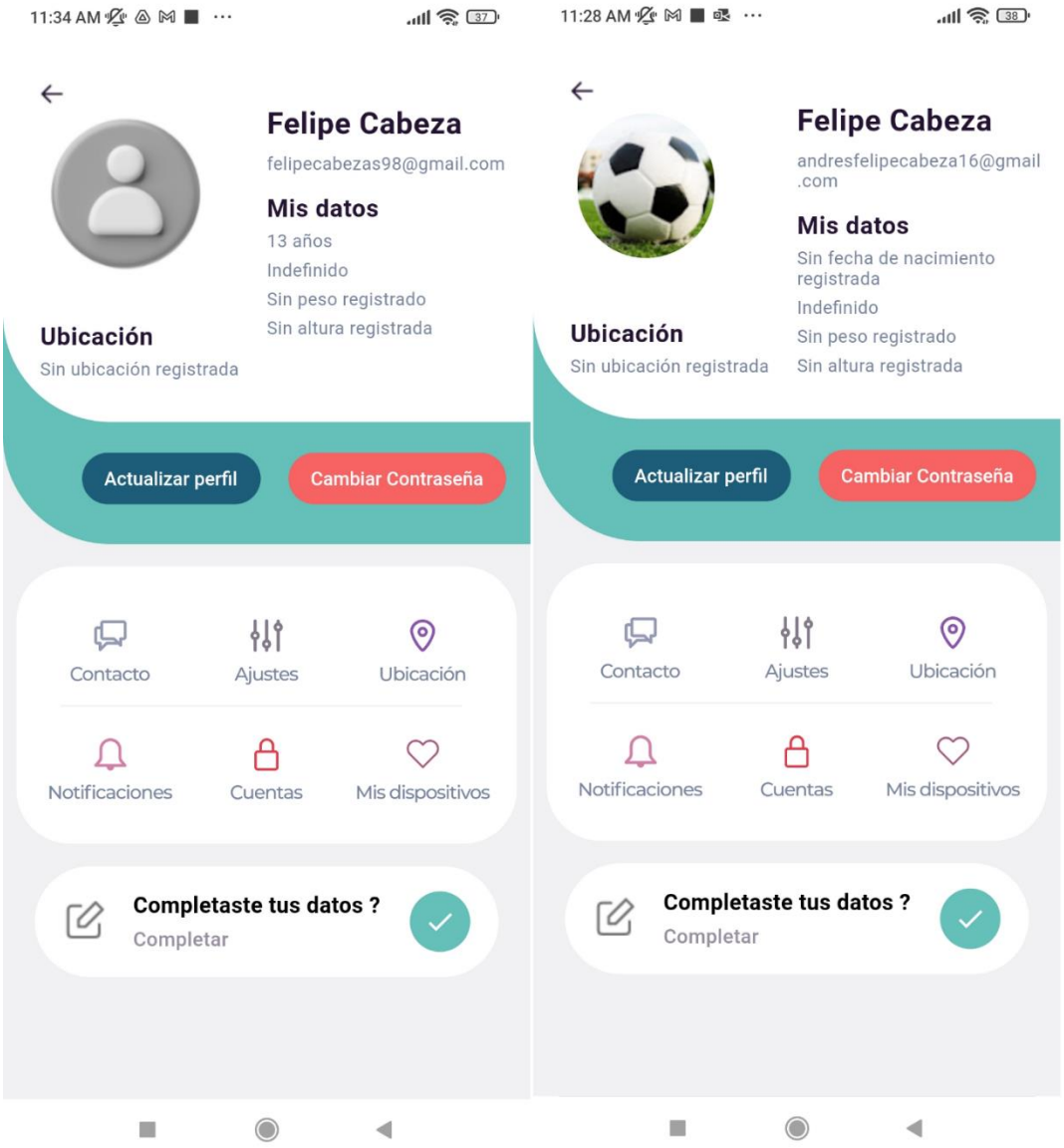
Esta pantalla contiene los datos iniciales del usuario después del proceso de registro, ya que más adelante se van a realizar conexiones con otros fabricantes que pueden proveer datos adicionales del usuario y completar estos datos que hacen falta.

Así, se muestra el correo o el teléfono suministrado, nombres, apellidos, y la edad del usuario calculada a partir de la fecha de nacimiento, hay algunos terceros como Facebook, Twitter y Google que suministran una foto de perfil si el usuario tiene. A la foto de perfil se le realizó un procesamiento para cambiar las dimensiones ya que era entregada en un formato de 48x48 en casi todos los terceros y lo cual hacía que se mostrase de manera pixelada.

Debajo las opciones de actualizar perfil y cambiar contraseña, es importante aclarar que es diferente la recuperación de contraseña a la funcionalidad de actualizar contraseña, donde para este caso se hace hincapié a actualizar contraseña, y por otro lado en actualizar perfil se pueden actualizar algunos datos, pero no se podrá cambiar el correo electrónico ni el teléfono con el que se registró, luego, debajo están las opciones que tiene la aplicación, las cuales incluyen la conexión a dispositivos, la vinculación con terceros, las opciones de notificaciones, ajustes, ubicación y contacto. Y, por último, la opción de completar los datos, pero en esta sección solo se va a realizar el tratamiento con los datos suministrados por el formulario o por los terceros y el diseño de la pantalla.

Figura 49

Perfil a un usuario registrado y usuario con foto de perfil



4.3.3.5 Vincular cuentas. Luego de realizar el proceso de registro, ya sea por credenciales o por medio de un tercero, existe la opción de añadir otros métodos adicionales de autenticación, por ejemplo, si se inicia sesión por credenciales, existe la opción de vincular cada uno de los terceros que se mencionó en la sección 4.3.3.3, o vincular la huella, que se explicará en la sección 4.3.3.7, también existe la opción de desvincularlos.

Esta opción está presente luego de haber iniciado sesión, desplegando el menú, la opción de perfil y por último seleccionando la opción de cuentas. Si el usuario hizo el proceso de registro por medio de Google, es decir, por un correo de Gmail, la pantalla va a mostrar que la cuenta de Google ya se encuentra vinculada, y dará la opción únicamente desvincular. Pero mostrará las otras opciones disponibles que son: la huella, Facebook, Twitter y Microsoft, en las cuales el proceso de vinculación es igual que como se explicó en el registro. Luego de vincular la nueva cuenta podrá iniciar sesión por cualquiera de las cuentas vinculadas.

Figura 50

Cuenta de Google vinculada por defecto y vinculación de Facebook

Nota. Después de vincular podrá iniciar sesión con cualquiera de las cuentas que tiene vinculadas.

Existen un par de casos específicos que se tuvieron en cuenta en el proceso de vincular y desvincular cuenta: el primer caso tiene que ver con usuarios que intentan vincular una cuenta de terceros diferente con el mismo correo asociado, para este caso Firebase arroja un error, por ejemplo, si se intenta vincular un correo sandra@gmail.com por Google y la cuenta de Twitter tiene este mismo correo, arrojará un error. El segundo caso, el cual consiste en un usuario que haya ingresado por un tercero, si ingresó por ejemplo por Twitter e intenta desvincular esta cuenta, que es la única cuenta que el usuario tiene en el momento para iniciar sesión, se mostrará un modal advirtiendo al usuario si está seguro de querer desvincular esta cuenta ya que al hacerlo su cuenta quedará inactiva, debido a que está desvinculando la única cuenta que tiene disponible. Así que si el usuario la desvincula, esta quedará inactiva, y la aplicación le mostrará un modal al usuario indicando que su cuenta se inactivó, luego lo redireccionará a la pantalla inicial de la aplicación como si hubiera cerrado la sesión.

Ahora, si el usuario desea reactivar tiene dos posibilidades, ya sea por inicio de sesión o por registro, si lo hace por registro ingresando por el mismo tercero el cual inactivó la cuenta se mostrará un modal advirtiendo al usuario que la cuenta está inactiva y si desea activarla, el usuario al aceptar reactivará la cuenta y podrá iniciar sesión por medio de este tercero. La otra opción es que lo haga por medio de la pantalla de iniciar sesión, pero con la única diferencia de que el momento de hacerlo y se pregunte en el modal si desea reactivar la cuenta, esta se reactiva y lo redirecciona a inicio, es decir, como si hubiera iniciado sesión normalmente.

4.3.3.6 Flujo de recuperar contraseña en la aplicación. En esta sección se creó una funcionalidad en la pantalla de inicio de sesión, precisamente con el botón de “olvidé mi contraseña”, el cual permite a los usuarios que previamente se registraron por correo electrónico, por medio del formulario de la aplicación, es decir, que el usuario no se haya registrado por medio de un tercero, la opción de que pueda recuperar su contraseña en caso de que la haya olvidado. En donde el flujo consiste en que en el formulario de inicio de sesión requiere que el usuario ingrese el campo de correo electrónico y se hace la respectiva validación que se creó por medio de una expresión regular, para que se verifique que sea un correo electrónico válido, posteriormente se enviará la consulta al servicio del backend, en el cual se envía al correo electrónico un link para la respectiva recuperación de la contraseña que llevará al usuario a un servidor web donde ingresará la nueva contraseña para que posteriormente pueda iniciar sesión nuevamente. En la sección 4.3.5.3.2 se muestra como se ve el formulario de recuperación de contraseña del servidor web.

4.3.3.7 Integración con huella. La aplicación también permite la autenticación por huella digital, en donde los dispositivos que posean el hardware de biometría requerido, específicamente por huella digital, ya que existen otras opciones para el inicio de sesión por biometría, como por ejemplo el reconocimiento facial. La opción de realizar dicha integración al dispositivo permite la opción de vincular la huella luego de se haya realizado previamente el respectivo flujo de registro e inicio de sesión, ya sea por medio de credenciales o por medio de terceros. Dicha funcionalidad se encuentra en la opción de perfil y luego cuentas. El flujo consiste en disparar primero un modal preguntado al usuario si desea vincular la huella digital, en caso de ser afirmativo se muestra un widget notificando al usuario que debe acercar la huella y en caso de que la huella coincida con la que está configurada en el dispositivo actual en la configuración de Android se vinculará, y la próxima vez que el usuario inicie sesión podrá iniciar por medio de su huella digital, además

permite la opción de desvincularla en cualquier momento y pudiendo vincular en cualquier otro momento que se requiera. Es importante aclarar que la integración por huella solo se asocia a un usuario, es decir, si cinco usuarios usan la aplicación, sólo uno puede gozar de esta funcionalidad en el dispositivo actual, mientras que los otros usuarios sólo verán una advertencia que el modal está asociado a otro usuario al intentar realizar un intento de integrar la huella, es decir, no podrán iniciar sesión en este dispositivo por huella a menos de que el usuario la desvincule. Además, cabe resaltar que dicha autenticación claramente es funcional para el dispositivo en el que se realizó la respectiva configuración, en otras palabras, no se va a poder iniciar con la misma huella en otro dispositivo, a menos de que la misma huella esté configurada en el otro dispositivo Android para desbloquearlo, y no esté asociada a otro usuario en el otro dispositivo específicamente en la aplicación móvil de Whatoko Health. La integración con huella permite una manera rápida de autenticación como lo realizan los dispositivos modernos.

Figura 51

Vinculación de huella digital

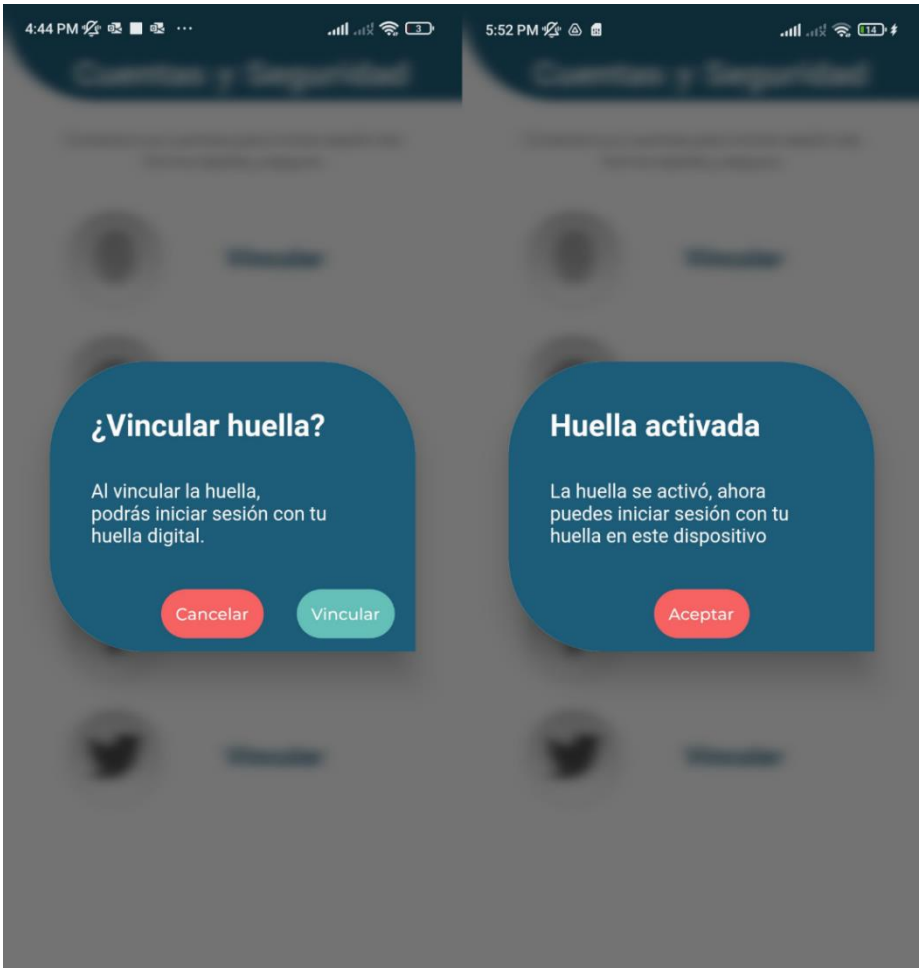
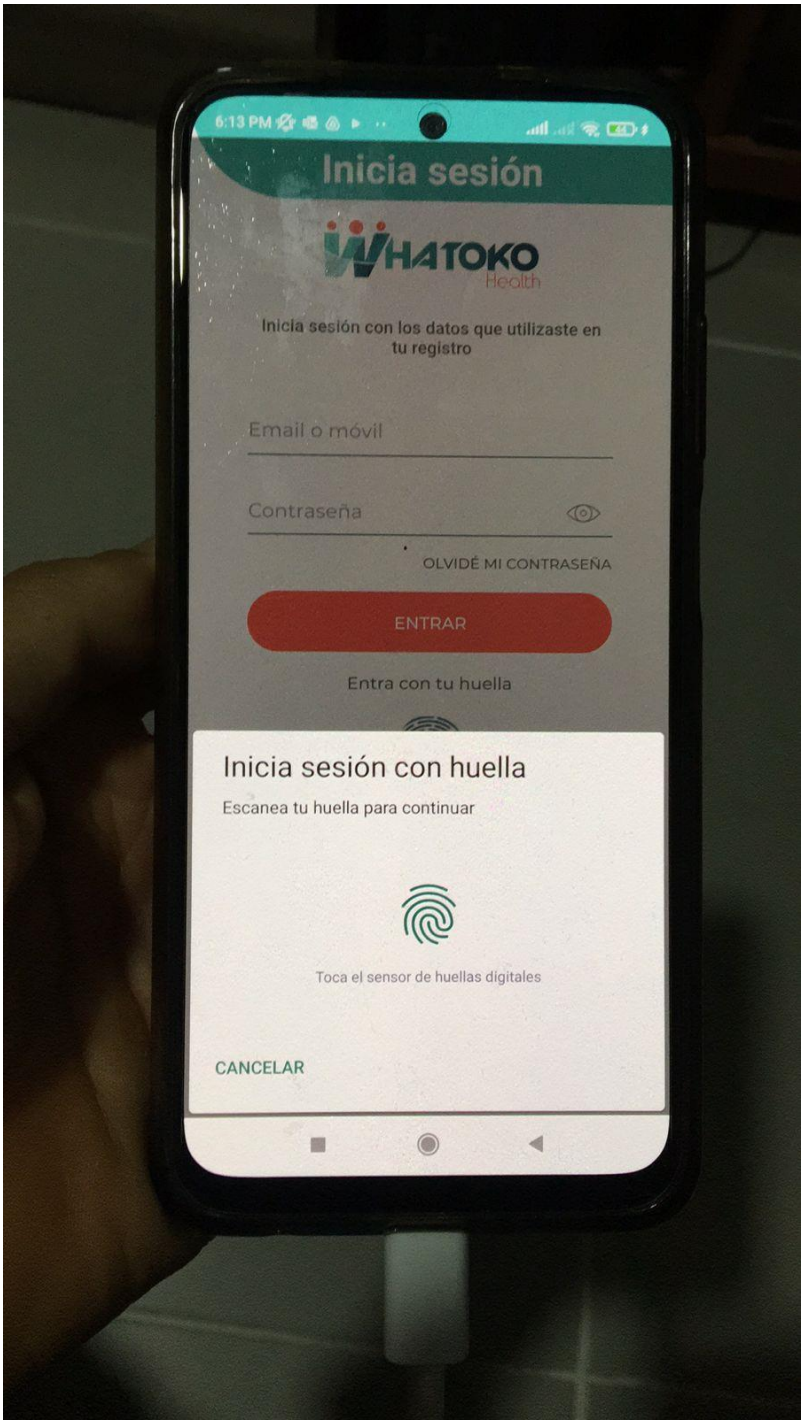


Figura 52

Autenticación por huella

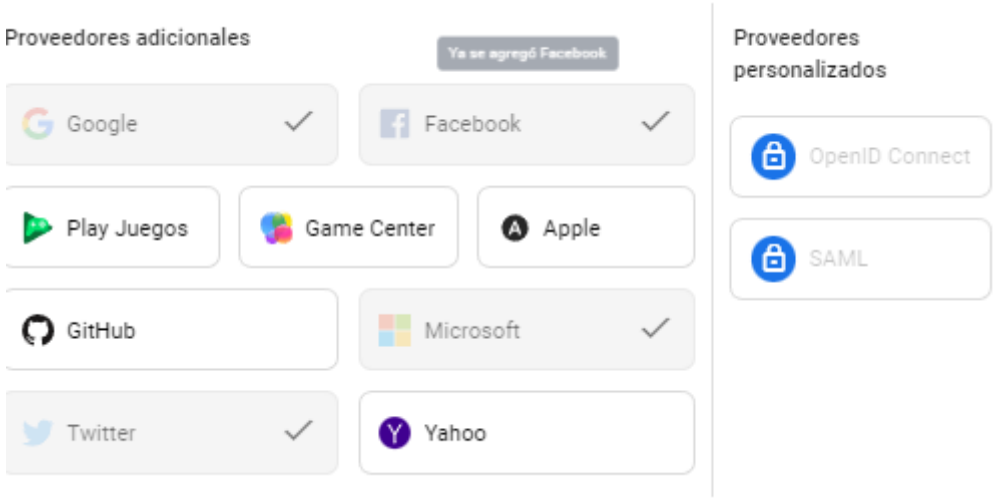


4.3.4 Módulo de conexiones

4.3.4.1 Conectar con librería de autenticación con terceros. Después de realizar la respectiva configuración con Firebase, se proporciona un módulo de autenticación en la consola de Firebase, donde existen una serie de proveedores para el inicio de sesión por medio de OAuth, en los cuales se incluyen los siguientes

Figura 53

Proveedores disponibles para autenticación por OAuth en Firebase



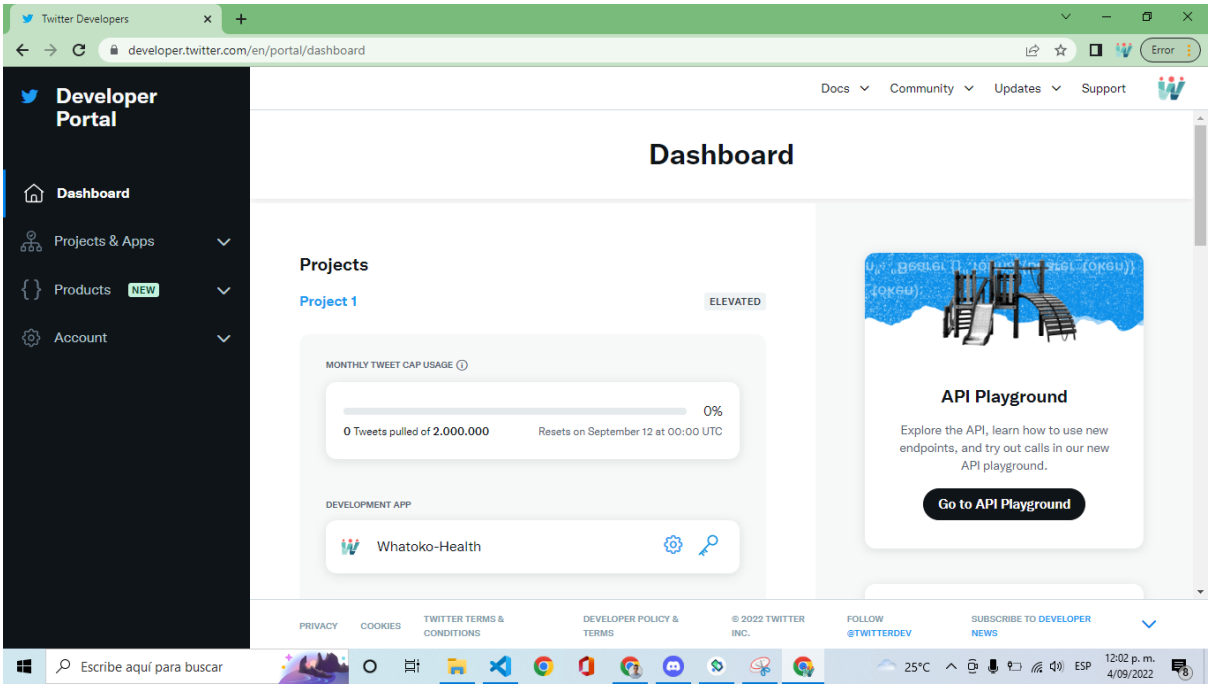
Luego se procede a realizar la respectiva configuración de cada tercero.

Para cada uno de estos proveedores a excepción de Google, fue necesario ir al sitio de desarrollo, crear una cuenta, configurar las políticas de privacidad y crear una aplicación con la configuración de OAuth a la cual Firebase se pudiera conectar.

Después de realizado este proceso, los usuarios que se autentiquen por el tercero, se mostrarán en el módulo de autenticación de Firebase con el correo, teléfono o nombre de usuario entregado y su respectivo identificador.

Figura 54

Portal de Developers Twitter con la aplicación de Whatoko Health



4.3.4.2 Captura de datos. Como bien se mencionó en la sección 4.2.4 existen dos maneras de capturar los datos del usuario, por medio de API, para la captura de datos por API, es decir, sin conectar un dispositivo wearable directamente, se implementó Google Fit.

4.3.4.2.1 Conexión por API. La conexión por API involucra solicitar los datos del usuario a un proveedor que los facilite siempre y cuando haya consentimiento del usuario. Para este caso se realizó la conexión con Google Fit.

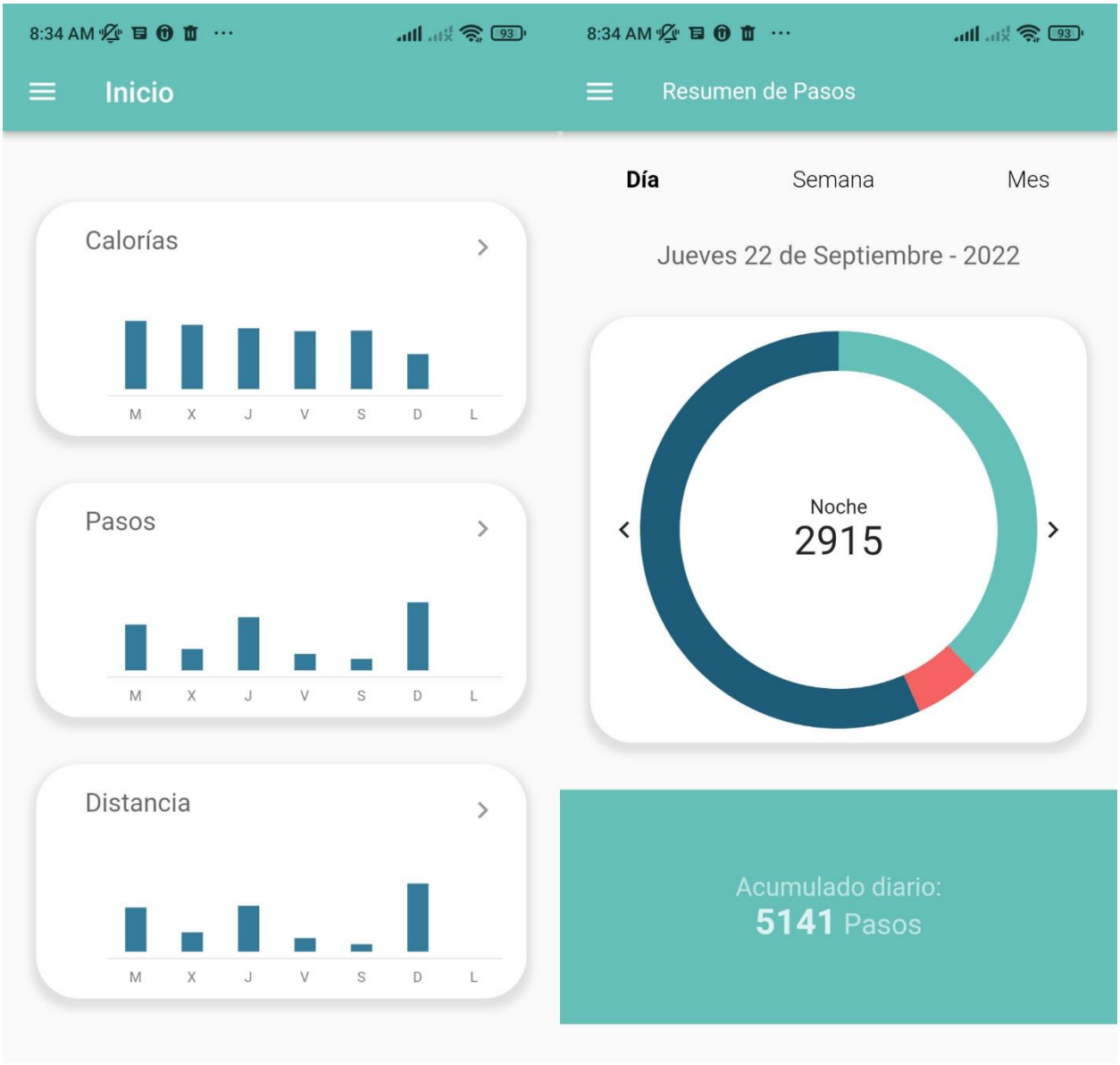
Google Fit es un servicio de Google que generalmente recopila datos de otras aplicaciones, como, por ejemplo, toma datos de actividad física con aplicaciones como polar (Polar, 2022), y datos nutricionales como con MyFitnessPal (myfitnesspal, 2022), los datos recolectados dependen de las aplicaciones que el usuario haya conectado previamente con su cuenta de Google.

Al establecer ciertos permisos de la aplicación, se pueden recopilar los datos que se requiere capturar (Google, 2021). Si el usuario no tiene datos registrados en la API de Google de glucosa y se hace una petición solicitando estos datos, lógicamente la API no retorna ningún dato, así que la API retorna los datos disponibles del usuario que esté solicitando, en un determinado intervalo de tiempo. Luego de capturar esos datos y mapearlos como lo requiere el módulo del API Gateway, estos datos se envían a los diferentes endpoints del API Gateway que este proporciona con las fechas de las mediciones con el formato requerido por el módulo. Después de que el API Gateway hace su respectivo formateo y envío de datos al módulo de reportería que también es otro módulo al cual se conecta la aplicación para el envío de datos, posteriormente estos datos son consumidos por la aplicación y mostrados a través de gráficas, en la pantalla de inicio.

De esta manera se ilustra al usuario de una manera sencilla y fácilmente comprensible el historial de sus mediciones de actividad física como el contador de pasos, calorías, distancia recorrida, etc, en varios intervalos de tiempo a conveniencia del usuario.

Figura 55

Gráficas resumen de actividad y resumen diario Whatoko Health



Para realizar el respectivo proceso de sincronización es necesario iniciar sesión, ir al perfil, ajustes, Apps y servicios vinculados y añadir la aplicación de Google Fit.

Figura 56

Flujo para vincular cuenta de Google Fit

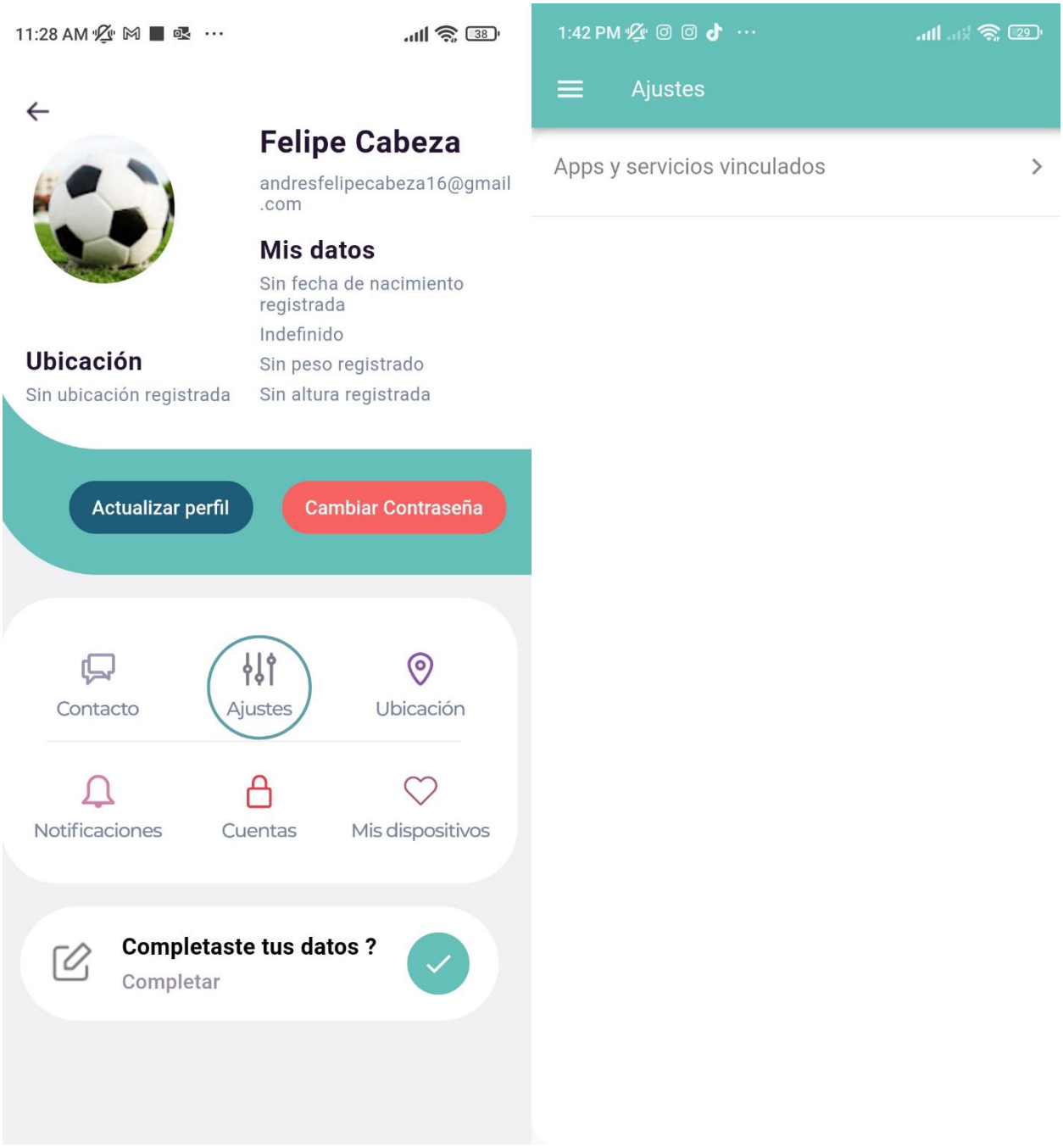
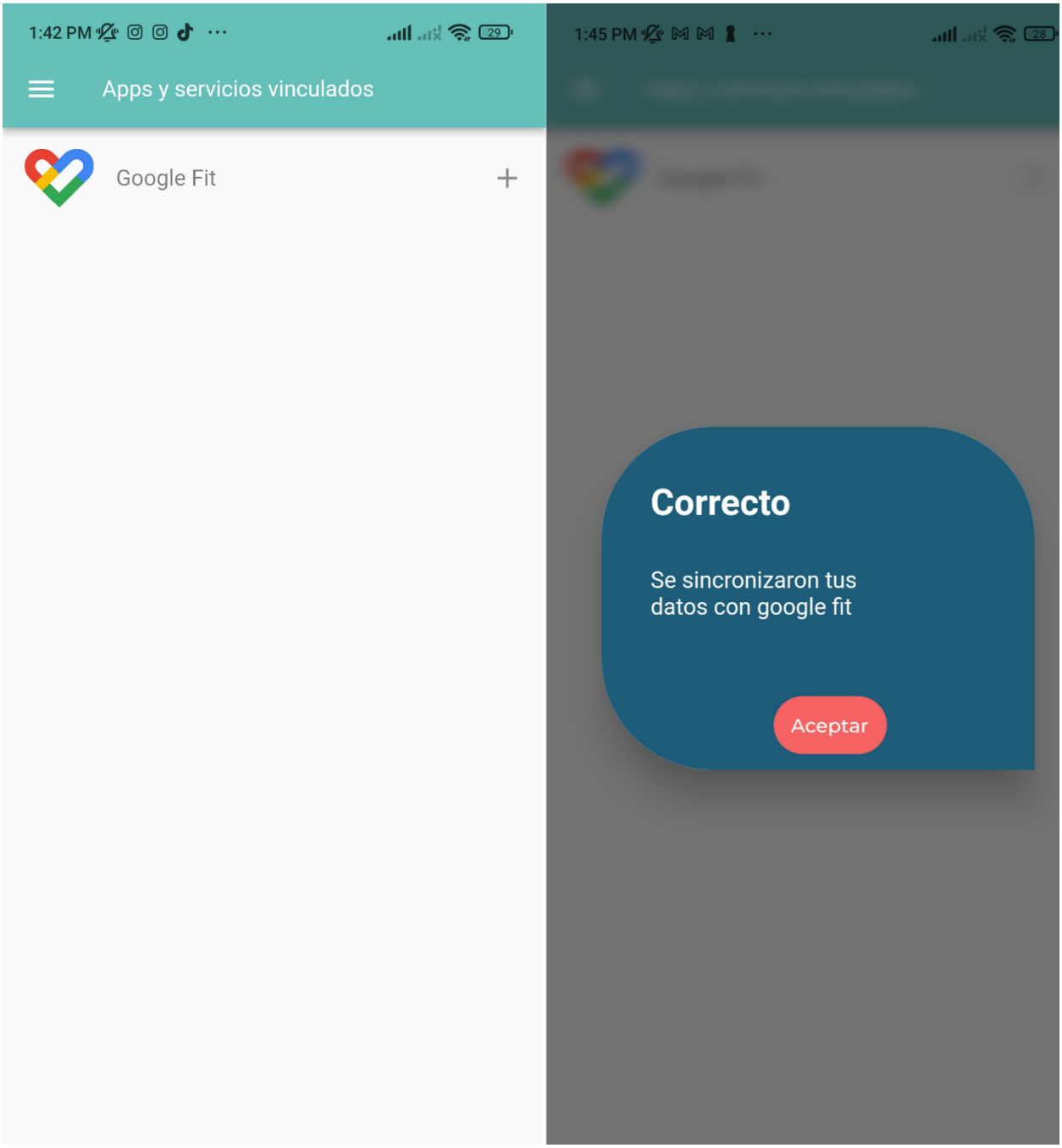


Figura 57

Vinculación de cuenta de Google Fit

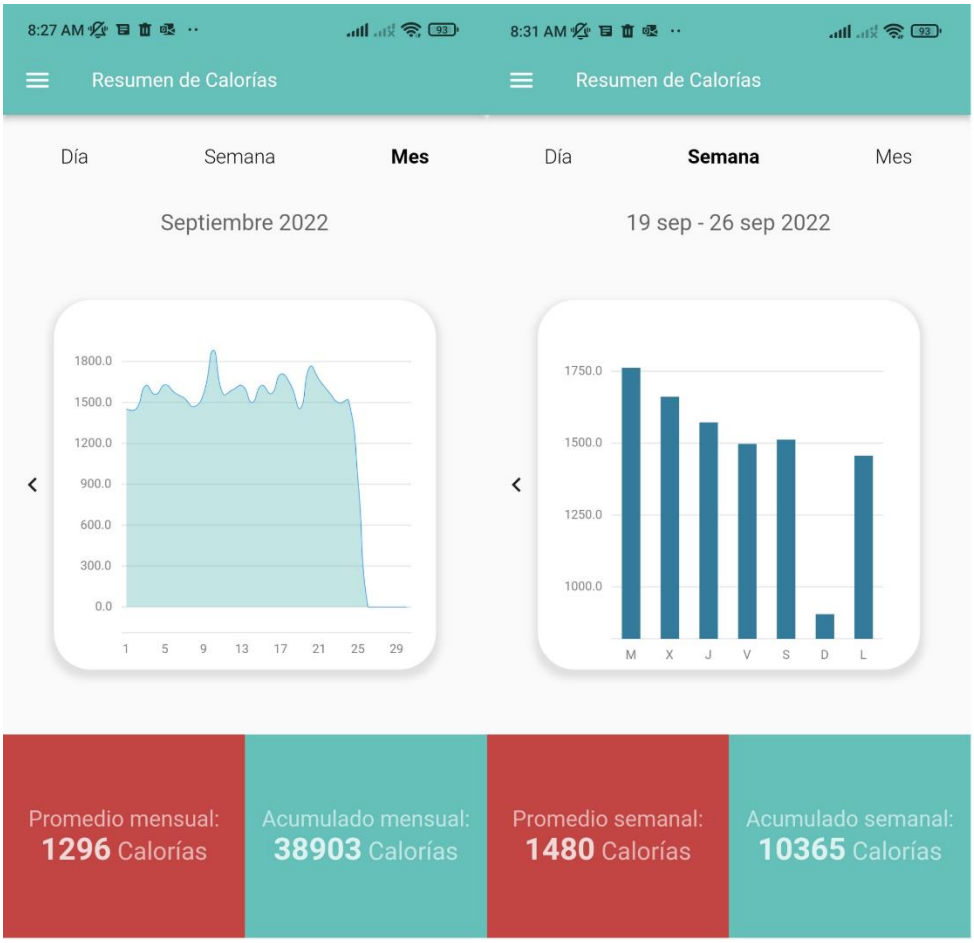


En caso de no tener mediciones no se van a mostrar las gráficas respectivas a la medición.

La aplicación por defecto muestra gráficas de resumen semanal en la pantalla de inicio y gráficas con más detalle al querer acceder a la pantalla con más información, en donde se muestra el resumen con datos diarios, mensuales y semanales en diferentes periodos de tiempo que el usuario puede seleccionar a conveniencia.

Figura 58

Gráficas de actividad física por mes y semana (calorías)



Otra característica importante es que al recibir los datos de Google Fit, estos vienen crudos, en fechas de nanosegundos, así que fue importante realizar un procesamiento a estos datos y además garantizar que no se interrumpiera el flujo en caso de que la captura de un dato falle o

estuviera corrupto, es decir, si de mil datos recibidos, falla la conversión de un dato, no se va a detener el flujo, sino se va a ignorar esa medición errónea o ese dato corrupto y se va a realizar el respectivo procesamiento de los otros datos.

Para finalizar, la aplicación permite luego de que ya se haya conectado con Google Fit, la opción de volver a sincronizar los datos desde la última fecha de sincronización hasta la fecha actual, esto para que no exista redundancia en los datos, y la opción de desvincular la cuenta de Google Fit.

4.3.5 Plataforma web

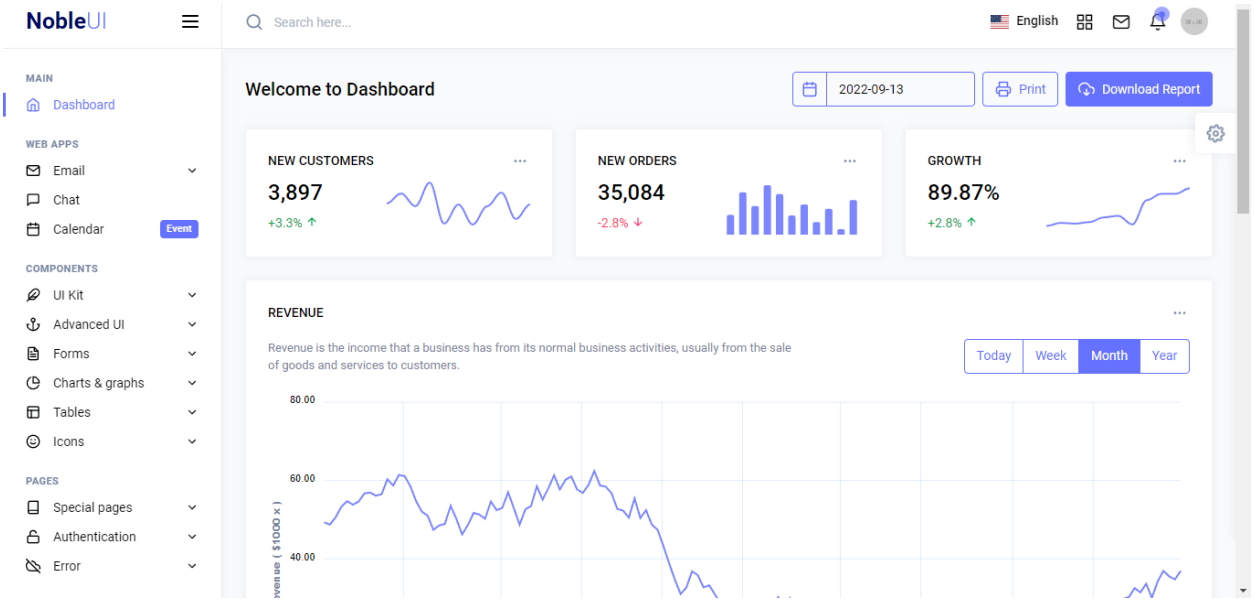
En este Epic se abordará el proceso de construcción de la plataforma web de Whatoko Health, la cual tiene como fin ser visualizada por usuarios administradores y profesionales de la salud, cuenta con una sección que permite la gestión usuarios, otra que contiene un dashboard con la información recolectada y gestionada por el módulo de reportería, también incluye un tablero con gráficos de Google analytics y otro tablero que permite visualizar las predicciones que realiza el módulo de inteligencia artificial de Whatoko health.

4.3.5.1 Integración con el template NobleUI - Angular Admin. Para lograr un mejor acabado en el diseño de la plataforma, se usó un témplate que contenía algunos componentes gráficos que se pudieron reutilizar, como por ejemplo formularios, botones y el diseño de la estructura base.

Para integrar el template con el proyecto existente se añadieron a la estructura de directorios los nuevos componentes y luego se subieron los cambios al repositorio. Gracias a la integración continúa implementada en el item 4.3.1.3.4 los cambios se ven reflejados en desarrollo y el resultado es el de la figura 59.

Figura 59

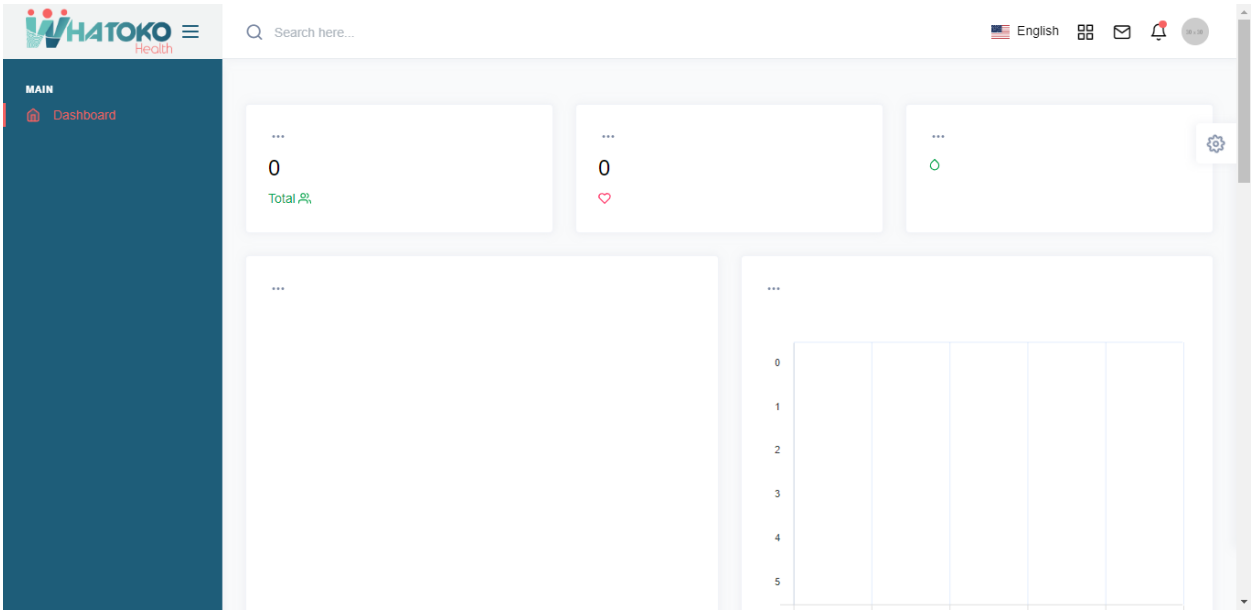
Template NobleUI - Angular Admin integrado



4.3.5.2 Personalización del template. Con base en las sugerencias del equipo de diseño se realizaron cambios al template establecido, los cambios involucran color, logos y la eliminación de los elementos que no se estén usando. Con esto se consiguió la primera versión de la plataforma, se puede apreciar en la figura 60.

Figura 60

Primera versión de la Plataforma web Whatoko Health



4.3.5.3 Servicios del módulo usuario. Para su funcionamiento la plataforma web estableció conexión con los servicios ofrecidos por el backend del proyecto descritos en el inciso 4.3.2. En la figura 61 se encuentran los servicios que fueron implementados y a continuación se resaltaran algunos de ellos.

Figura 61

Servicios de la plataforma Whatoko Health

4.3.5.3.1 Servicio de autenticación. La plataforma web solo permite el ingreso de usuarios con el rol de administrador y que están previamente establecidos, es decir que no se permite el registro. Al consumir el servicio del inciso 4.3.2.1.2 se envía un correo y una contraseña como se muestra en la figura 62 y con base en la respuesta del backend permite el ingreso al usuario o se le indica el error. Una vez el usuario logra ingresar con credenciales validas se guarda en el LocalStorage el token que el servicio suministra (figura 32)

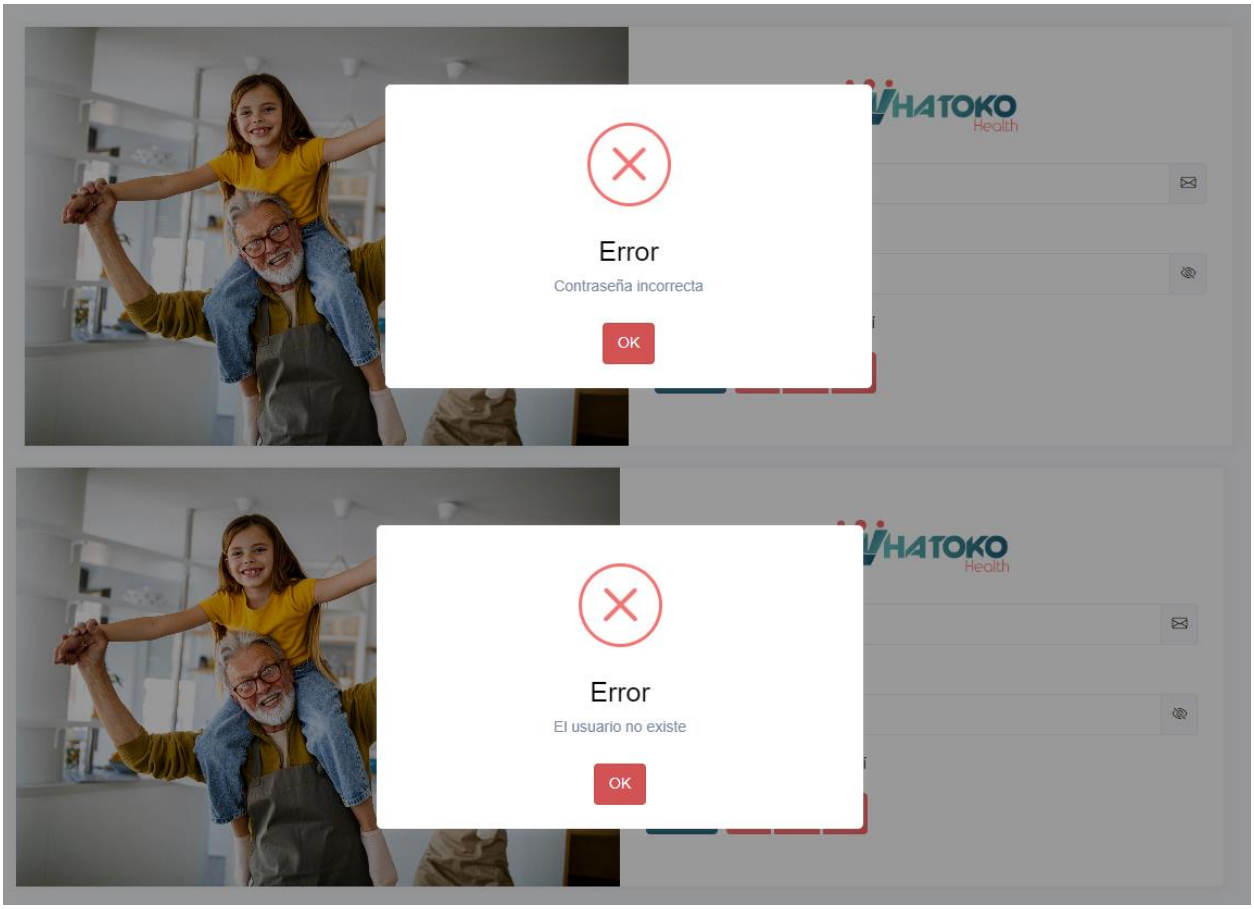
Para tener un control de la sesión del usuario también se hizo uso del servicio del inciso 4.3.2.1.3, el cual valida el token existente en el LocalStorage. Este servicio se implementó en un guard de angular, el cual protege el acceso a las otras rutas de la plataforma si el token no es válido o esta vencido.

Figura 62

Sección de autenticación de la plataforma Whatoko Health

Figura 63

Ejemplos de respuesta en autenticación de plataforma web



4.3.5.3.2 Servicios de recuperar y cambiar contraseña. En caso de que un usuario haya olvidado su contraseña, la plataforma ofrece la posibilidad de recuperarla ingresando el correo electrónico. Si el correo es válido el usuario recibirá un mensaje como el de la figura 34, con un enlace que lo direccionará a una pantalla donde podrá ingresar su nueva contraseña.

Figura 64

Sección de la plataforma para ingresar la nueva contraseña.

4.3.5.3.3 Servicios CRUD. En esta sección se abordan el conjunto de servicios que permiten las acciones básicas del manejo de usuarios, como lo son el crear, editar, mostrar e inactivar. Estas acciones se pueden realizar desde el apartado usuarios de la plataforma web como se muestra en la figura 65 y sus diferentes funcionalidades están indicadas por los botones, un ejemplo es el de la figura 66 que muestra el servicio que permite agregar un nuevo usuario.

Figura 65

Apartado de usuarios en la plataforma web

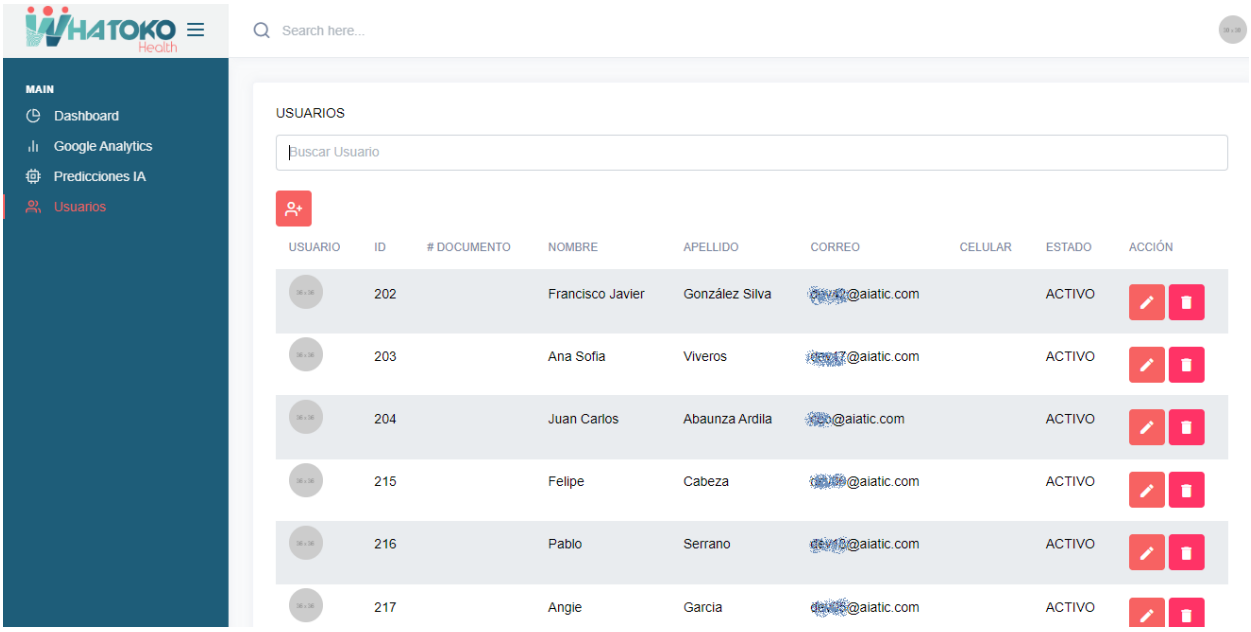
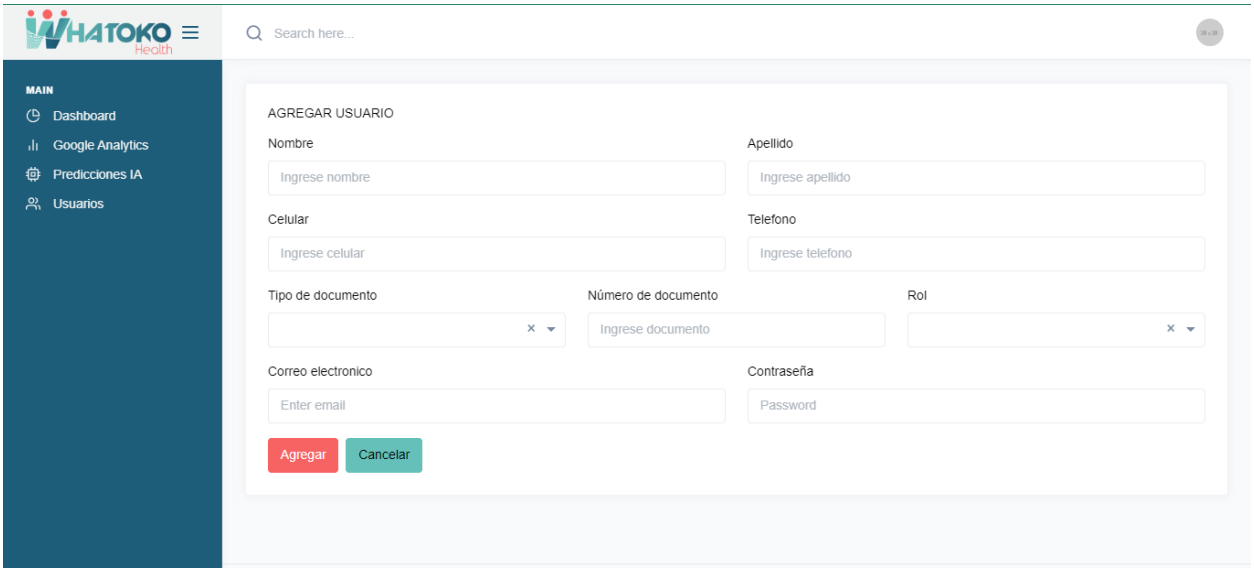


Figura 66

Pantalla para crear un nuevo usuario



4.3.5.4 Consumo del módulo reportes de Whatoko Health. El módulo reportería retorna informes de los usuarios por medio de una API, la plataforma web consume estos servicios y le realiza un tratamiento a la información que llega con el fin de construir graficas.

Para la construcción de las gráficas se integró al proyecto la librería ApexCharts, se escogió esta librería porque provee graficas en formato SVG, es moderna, gratuita y de código abierto. Las gráficas que permite realizar son agradables e interactivas, además la librería contiene en su documentación ejemplos que pueden ser usados.

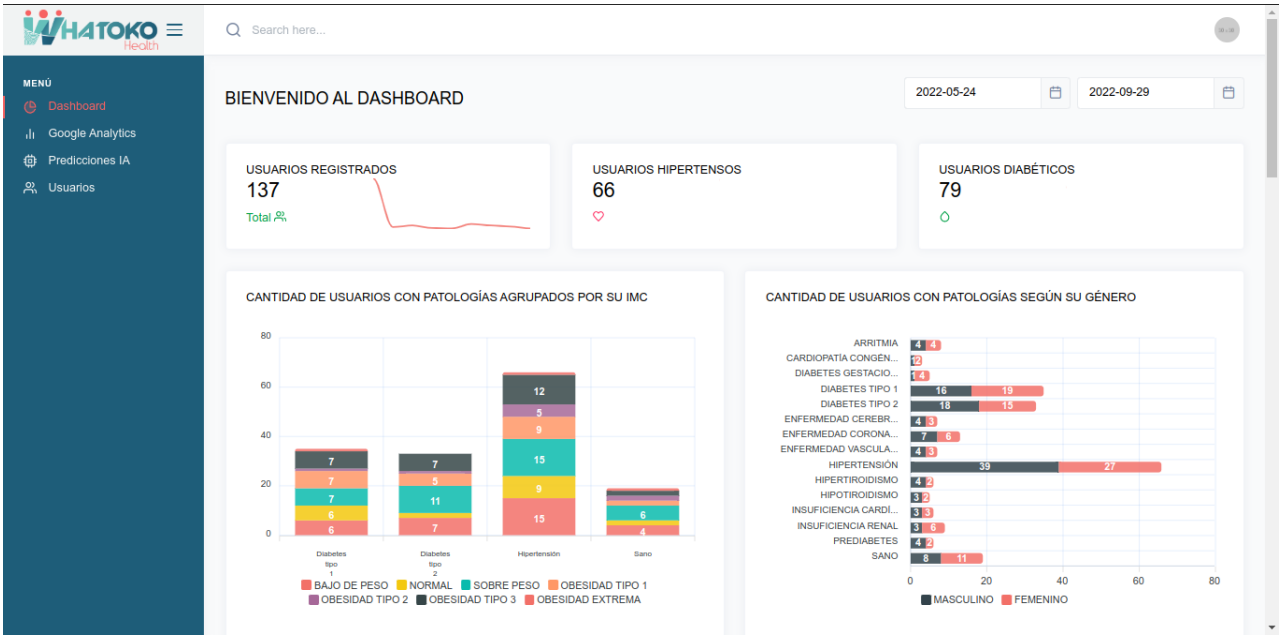
Figura 67

Ejemplos de graficas realizadas con la librería ApexCharts

Las gráficas que se construyeron en el dashboard informan sobre la división de usuarios en patologías por su grupo sanguíneo, la cantidad de usuarios con patologías según su género, la cantidad de usuarios con patologías agrupados por su IMC, el porcentaje de usuarios por patología, deportes practicados por los usuarios y número de usuarios por departamento. Toda esta información puede ayudar a un profesional de la salud que ingrese a la plataforma a saber el estado general de los pacientes que están usando la aplicación de Whatoko Health.

Figura 68

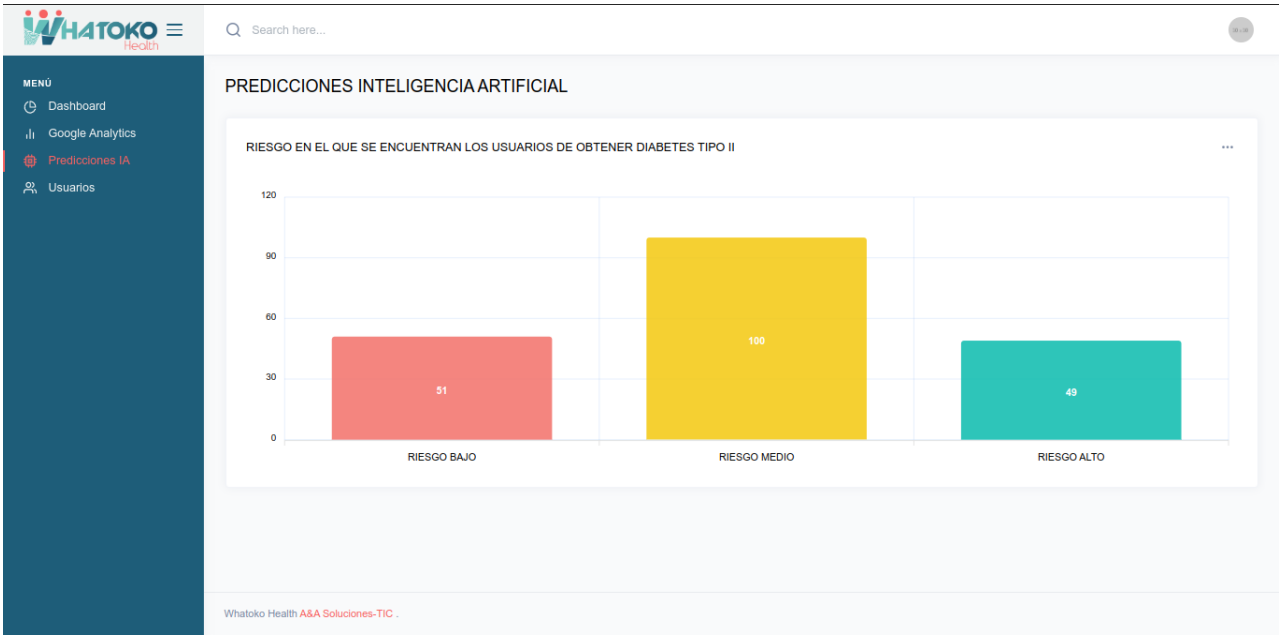
Dashboard de plataforma Whatoko Health



4.3.5.5 Consumo del módulo predicciones de Whatoko Health. Este módulo es el encargado de entregar predicciones, la plataforma web realiza el consumo de los servicios que entrega y grafica los resultados arrojados para una mayor comprensión de los mismos. Se puede ver un ejemplo en la figura 69.

Figura 69

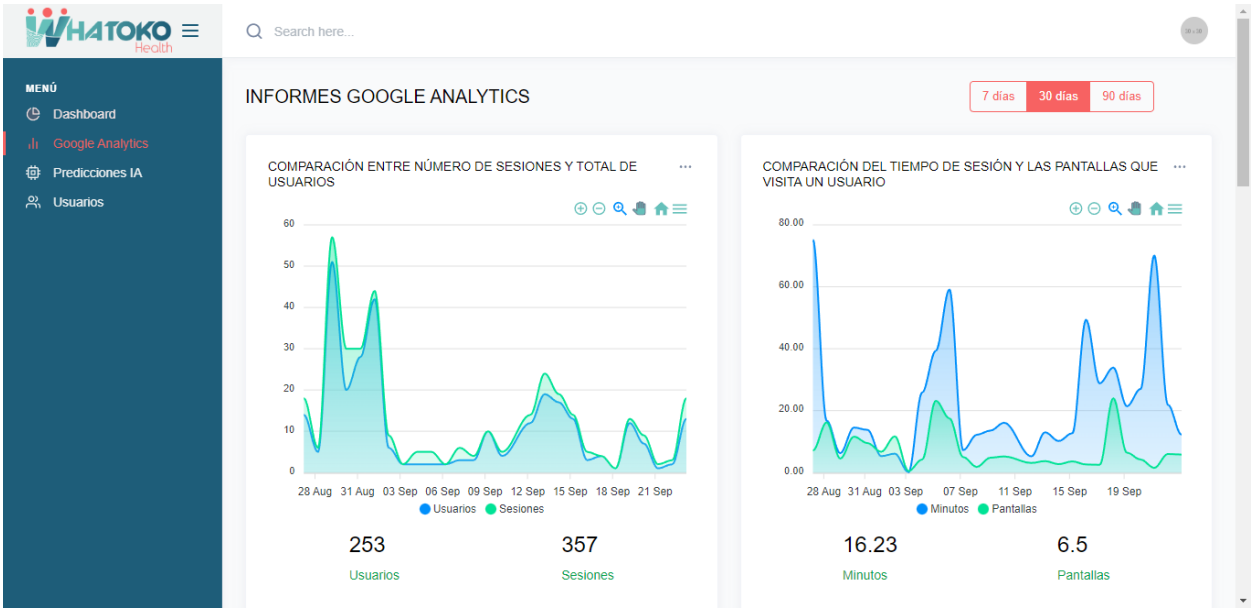
Plataforma Whatoko Health en la sección de predicciones.



4.3.5.6 Consumo de servicio Google Analytics. Este servicio es fundamental para realizar un seguimiento a la interacción de los usuarios con la aplicación móvil, consume toda la información que entrega el microservicio de la sección 4.3.2.4 y representa de forma gráfica los resultados obtenidos. La plataforma permite aplicar filtros de fecha a todas las gráficas.

Figura 70

Plataforma Whatoko Health, sección de informes Google Analytics



5. Conclusiones

Con el avance de la tecnología han surgido herramientas que permiten capturar datos de nuestra salud como lo son los signos vitales y los datos de composición corporal, lo que antes solo se podía realizar en un consultorio médico ahora está al alcance de nuestras manos. Un claro ejemplo son ciertos dispositivos wearables que además de que capturan los datos, permiten crear persistencia a través de los mismos. Con dicho volumen de datos se quiso fomentar el apoyo en el cuidado de la salud del usuario y su estilo de vida en el presente proyecto.

Con la construcción de la aplicación móvil y la plataforma web se logró un traspaso de datos e información entre todos los módulos del proyecto, los datos capturados en la aplicación móvil se transmiten al módulo del API Gateway, el cual hace un formateo, y posteriormente los envía al servicio de reportería, donde luego la plataforma web realiza un consumo de esta

información para mostrar una visualización a través de un dashboard incluyendo un resumen de los reportes y una generación de predicciones, donde los profesionales de la salud pueden apoyarse generando valor a los datos que se están entregando.

Tal y como lo hemos podido comprobar la principal razón de Whatoko Health es apoyar la salud de las personas por medio de las herramientas tecnológicas proporcionadas y orientando a la prevención, también apoyando a los profesionales de la salud añadiéndole un valor agregado a los datos que se capturan.

En cuanto a la implementación técnica del proyecto contribuyó a consolidar e integrar los conocimientos que se adquirieron durante el desarrollo profesional de los estudiantes dentro de la institución, fue muy importante el trabajo en equipo, la inclusión de metodologías agiles, la aplicación de un control de versiones y otras buenas prácticas.

En última instancia la integración de varias tecnologías modernas y robustas como Flutter, Docker, Angular, Node.js, etc, APIs en el desarrollo de proyecto dejó como resultado un producto mínimo viable escalable con un sinfín de aplicabilidad para trabajo futuro.

Referencias Bibliográficas

ADEN. (21 de Septiembre de 2021). *Metodologías ágiles: ¿Qué son y cuáles son las más utilizadas?* Aden Business Magazine: https://www.aden.org/business-magazine/metodologias-agiles/

Adobe. (s.f.). *Adobe XD*. Herramienta rápida y poderosa de diseño y colaboración de experiencias e interfaces de usuario: https://www.adobe.com/co/products/xd.html

Amazon. (s.f.). *¿Qué es DevOps?* Amazon Web Services (AWS): https://aws.amazon.com/es/devops/what-is-devops/

Angelov, F. (s.f.). *Bloc State Management*. https://bloclibrary.dev/#/

Angular. (28 de Febrero de 2022). *Angular*. What is Angular?: https://angular.io/guide/what-is-angular

Apache NetBeans. (2022). *Apache NetBeans Releases*. https://netbeans.apache.org/download/index.html

AppBrain. (28 de 04 de 2022). *Number of Android apps on Google Play*. AppBrain: https://www.appbrain.com/stats/number-of-android-apps

appium. (s.f.). *Introduction* . Appium: https://appium.io/docs/en/about-appium/intro/?lang=es

Apps para controlar la hipertensión arterial. (29 de Julio de 2014). https://tevafarmacia.es/herramientas/apps-de-salud/apps-para-controlar-la-hipertension-arterial

Apps para controlar la hipertensión arterial. (29 de Julio de 2019). TEVAFarmacia: https://tevafarmacia.es/herramientas/apps-de-salud/apps-para-controlar-la-hipertension-arterial

Apps para controlar la hipertensión arterial. (29 de Julio de 2019). TEVAFarmacia: https://tevafarmacia.es/herramientas/apps-de-salud/apps-para-controlar-la-hipertension-arterial

Araya-Orozco, M. (2004). Hipertensión arterial y diabetes mellitus. *Revista Costarricense de Ciencias Médicas*, *25*(3-4), 65-71. https://doi.org/https://www.scielo.sa.cr/scielo.php?script=sci_arttext&pid=S0253-29482004000200007&lng=en&tlng=es

Atlassian. (s.f.). *Flujo de trabajo de Gitflow*. Atlassian Git Tutorial: https://www.atlassian.com/es/git/tutorials/comparing-workflows/gitflow-workflow#:~:text=%C2%BFQu%C3%A9%20es%20Gitflow%3F,vez%20y%20quien%20lo%20populariz%C3%B3.

Bastidas, W. (03 de 11 de 2020). *medium.com*. Eventos del ciclo de vida en Angular: https://medium.com/williambastidasblog/eventos-del-ciclo-de-vida-en-angular-41ddf766273b#:~:text=Un%20componente%20Angular%20entra%20en,espec%C3%ADficos%20del%20ciclo%20de%20vida

Carlos, A. (2019). Guías ALAD sobre el Diagnóstico, Control y Tratamiento de la Diabetes Mellitus Tipo 2 con Medicina Basada en Evidencia Edición 2019. *Revista de la ALAD Asociación Latinoamericana de Diabetes*, 1-2. https://doi.org/https://www.revistaalad.com/guias/5600AX191_guias_alad_2019.pdf

CloudFare. (s.f.). *What is multitenancy? | Multitenant architecture*. CloudFare: https://www.cloudflare.com/learning/cloud/what-is-multitenancy/

Dart. (s.f.). *Asynchronous programming: futures, async, await*. Dart: https://dart.dev/codelabs/async-await

Dart. (s.f.). *Asynchronous programming: Streams*. Dart: https://dart.dev/tutorials/language/streams

Dart. (s.f.). *Dart programming language*. Dart: https://dart.dev/

Dart. (s.f.). *Effective Dart: Documentation*. Dart: https://dart.dev/guides/language/effective-dart/documentation

Escoffier, C. (30 de Junio de 2017). *https://developers.redhat.com/blog/2017/06/30/5-things-to-know-about-reactive-programming*. https://developers.redhat.com/blog/2017/06/30/5-things-to-know-about-reactive-programming

Eusko Jaurlaritza. (21 de Diciembre de 2021). *Hipertensión arterial en la diabetes.* Eusko Jaurlaritza - Gobierno Vasco: https://www.osakidetza.euskadi.eus/enfermedad-hta/-/hipertension-arterial-en-la-diabetes/

Firebase. (s.f.). *Firebase*. https://firebase.google.com/

Flutter. (2022). *Flutter*. Build apps for any screen: https://flutter.dev/

Flutter. (13 de Julio de 2022). *StatefulWidget class*. widgets library - Dart API: https://api.flutter.dev/flutter/widgets/StatefulWidget-class.html

Flutter. (13 de Julio de 2022). *StatelessWidget class*. widgets library - Dart API: https://api.flutter.dev/flutter/widgets/StatelessWidget-class.html

Flutter. (s.f.). *List of state management approaches*. Flutter: https://docs.flutter.dev/development/data-and-backend/state-mgmt/options

GaeaPeople. (21 de Enero de 2020). *Las Apps que harán la vida más fácil a las personas con diabetes*. Magazine. Soluciones para la Diabetes: https://www.solucionesparaladiabetes.com/magazine-diabetes/apps-personas-diabetes/

Git SCM. (s.f.). *Git—Acerca del Control de Versiones*. git-scm: https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Acerca-del-Control-de-Versiones

Gonçalves, M. J. (13 de 10 de 2021). *¿Qué es Angular y para qué sirve?* Hiberus: https://www.hiberus.com/crecemos-contigo/que-es-angular-y-para-que-sirve/

Google. (s.f.). *Google Fit*. Google Developers: https://developers.google.com/fit

Google Inc. (s.f.). *Firebase*. https://firebase.google.com/

Google Inc. (s.f.). *Design*. Material Design: https://material.io/design

Gutierrez, V. (2014). ANALISIS Y SELECCIÓN DE MODELOS DE CERTIFICACION PARA UNA EMPRESA DESARROLLADORA DE SOFTWARE [Tesis para obtener el título de maestro en ciencias en ingeniería de sistemas, Instituto politécnico nacional]. Repositorio Digital IPN - Instituto Politécnico Nacional https://tesis.ipn.mx/bitstream/handle/123456789/18034/Analisis%20y%20seleccion%20de%20modelos%20de%20certificacion%20para%20una%20empresa%20desarrolladora%20de%20software.pdf?isAllowed=y&sequence=1

Hipertensión y diabetes: Descuido fatal. (9 de Abril de 2018). Vive con Diabetes: https://www.vivecondiabetes.com/viviendo-con-diabetes/complicaciones/8207-hipertensi%C3%B3n-y-diabetes-descuido-fatal.html

IBM. (s.f.). *¿Qué es una API REST? - Colombia*. IBM: https://www.ibm.com/co-es/cloud/learn/rest-apis

IBM. (5 de Marzo de 2021). *What is a web service?* IBM documentation: https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-

south.containers.appdomain.cloud/docs/en/cics-ts/5.1?topic=services-what-is-web-service

Jesuites educación . (s.f.). *fp.uoc.fje.edu*. Qué es Node.js y cuáles son las ventajas de usar esta tecnología: https://fp.uoc.fje.edu/blog/que-es-node-js-y-cuales-son-las-ventajas-de-usar-esta-tecnologia/#:~:text=utilizar%20esta%20plataforma.-,Node.,evitando%20el%20bloqueo%20de%20procesos

Las Apps que harán la vida más fácil a las personas con diabetes. (21 de Enero de 2020). https://www.solucionesparaladiabetes.com/magazine-diabetes/apps-personas-diabetes/

Maida, E. & Pacienzia, J (2015). Metodologías de desarrollo de software [Tesis de Licenciatura en Sistemas y Computación, Universidad Católica Argentina]. El Repositorio Institucional UCA. https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf

Maria, D., & Mikhail, O. (5 de MAyo de 2022). *Why to Use Flutter for Building Cross-Platform Apps?* https://www.cleveroad.com/blog/why-use-flutter/

Material Design. (s.f.). *Android bars*. Material Design: https://material.io/design/platform-guidance/android-bars.html

Microsoft Azure. (s.f.). *Comparación entre Kubernetes y Docker*. Microsoft Azure: https://azure.microsoft.com/es-es/topic/kubernetes-vs-docker/

Microsoft Azure. (s.f.). *Comparación entre Kubernetes y Docker*. Microsoft Azure: https://azure.microsoft.com/es-es/topic/kubernetes-vs-docker/

Microsoft. (s.f.). *Estilo de arquitectura de microservicios*. Microsoft: https://docs.microsoft.com/es-es/azure/architecture/guide/architecture-styles/microservices

Mozilla. (3 de Marzo de 2022). *WebSockets*. MDN web docs: https://developer.mozilla.org/es/docs/Web/API/WebSockets_API

NGINX. (s.f.). *What Is Load Balancing? How Load Balancers Work*. NGINX: https://www.nginx.com/resources/glossary/load-balancing/

Node.js. (s.f.). *expressjs*. Infraestructura web rápida, minimalista y flexible para Node.js: https://expressjs.com/es/

Node.js. (s.f.). *nodejs.org*. Acerca de Node.js: https://nodejs.org/es/about/

Novidá. (s.f.). *BLE (Bluetooth Low Energy): ¿qué es y cómo usarlo en IoT?* Novidá: https://www.novida.com/es/blog/ble/

OAuth. (s.f.). *OAuth 2.0*. OAuth: https://oauth.net/2/

OPS/OMS. (s.f.). *Diabetes—OPS/OMS* . Organización Panamericana de la Salud: https://www.paho.org/es/temas/diabetes

OPS/OMS. (s.f.). *Hipertensión—OPS/OMS*. Organización Panamericana de la Salud: https://www.paho.org/es/temas/hipertension

Organización Mundial de la Salud. (10 de Noviembre de 2021). *Diabetes*. Organización Mundial de la Salud: https://www.who.int/es/news-room/fact-sheets/detail/diabetes

Organización Mundial de la Salud. (25 de Agosto de 2021). *Hipertensión*. Organización Mundial de la Salud: https://www.who.int/es/news-room/fact-sheets/detail/hypertension

Organización Mundial de la Salud. (s.f.). *Hipertensión*. Organización Mundial de la Salud: https://www.who.int/es/news-room/fact-sheets/detail/hypertension

Pérez, M., Jorge, L., & Fernández, M. (2011). El control de la hipertensión arterial: un problema no resuelto. *Revista Cubana de Medicina, 50*(3), 311-323.

https://doi.org/http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0034-75232011000300009

Red Hat. (31 de Enero de 2018). *¿Qué son la integración/distribución continuas (CI/CD)?* Red Hat: https://www.redhat.com/es/topics/devops/what-is-ci-cd

Sáenz, C. (Noviembre de 2017). *La diabetes sería la séptima causa de mortalidad para el año 2030*. el Hospital: https://www.elhospital.com/temas/La-diabetes-seria-la-septima-causa-de-mortalidad-para-el-ano-2030,-segun-proyecciones-de-la-OMS+122682

Salesforce. (s.f.). *¿Qué es Cloud Computing?* Salesforce: https://www.salesforce.com/mx/cloud-computing/

SCRUM GUIDES. (Noviembre de 2020). *The 2020 Scrum GuideTM*. SCRUM GUIDES: https://scrumguides.org/scrum-guide.html

SocialDiabetes. (s.f.). *Gestión integral para el control de la diabetes*. SocialDiabetes: https://www.socialdiabetes.com/

StackOverflow. (5 de Enero de 2017). *Life cycle in flutter*. StackOverflow: https://stackoverflow.com/questions/41479255/life-cycle-in-flutter

Universidad Internacional de Valencia. (30 de Abril de 2019). *Qué es wearable y qué tipos de dispositivos existen*. VIU: https://www.universidadviu.com/es/actualidad/nuestros-expertos/que-es-wearable-y-que-tipos-de-dispositivos-existen