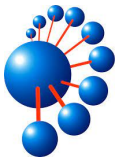
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA
Y CIENCIAS DE LA COMPUTACIÓN
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE CONCEPCIÓN



Soporte móvil de captura y transmisión de datos de tratamiento farmacológico

por

Vicente Schultz Solano

Memoria presentada para la obtención del título de

Ingeniero Civil Informático

Patrocinante: Gonzalo Eduardo Rojas Duran

Comisión Evaluadora: Lilian Salinas, Julio Godoy

Concepción, marzo de 2024

Quiero agradecer a mis compañeros universitarios Lucas, Darling, Franco, Joaquín, Eduardo, Giorgio, Diego, Hugo, Claudio y al resto de ellos por su ayuda y apoyo. A mi familia, mi hermano, mi hermana, mi papá, mi mamá, mis tíos, mis abuelos y mis primos por su apoyo y nunca perder la fe en mí. A mis amigos del colegio y a la Francisca, al Aníbal y al Felipe que siempre estuvieron ahí cuando los necesite y a Diego Muñoz, el ingeniero encargado del proyecto; y a Guillermo Vasquez, el diseñador gráfico que nos apoyó, por el trabajo que hicieron y finalmente al profesor Gonzalo Rojas por su invaluable asistencia.

Gracias a todos por su apoyo, compañía y fe.

Resumen

La adherencia a tratamientos farmacológicos es pieza fundamental en el tratamiento y mejoramiento de la calidad de vida de los pacientes. Y cuando hablamos de adultos mayores con enfermedades crónicas, la adherencia a dichos tratamientos se vuelve aún más relevante. La real adherencia a tratamientos farmacológicos es una información que normalmente no es compartida de forma fidedigna con los profesionales médicos en estos casos y puede hacer la diferencia entre una vida saludable y una vida llena de complicaciones médicas.

En la siguiente memoria de título se presenta el desarrollo de una aplicación móvil que apoya el proceso de recolección, análisis y visualización de datos que caracterizan la adherencia al tratamiento farmacológico de pacientes con enfermedades crónicas. Los datos en cuestión son precargados a la aplicación y posteriormente enviados a un servidor fundado siguiendo el standard FHIR creado por HL7 para el intercambio de datos electrónicos de atención médica.

Adicionalmente se hablará en menor medida sobre otras funcionalidades, como un reproductor de videos, la presentación de artículos de interés y un chat de texto que tienen como objetivo fomentar el uso de la aplicación para motivos lúdicos y de información.

Índice

Índice de figuras

Índice de cuadros

1. Introducción

La adherencia a tratamientos farmacológicos es un factor fundamental en el tratamiento y mejoramiento de la calidad de vida de los pacientes, por lo que tener datos objetivos de la adherencia constituye un aporte significativo a sus tratamientos. Sin embargo, en el caso de pacientes con patologías crónicas sometidos a tratamientos farmacológicos, la real adherencia a dichos tratamientos es una información que normalmente no es compartida de forma fidedigna con los profesionales médicos.

En el marco del proyecto FONDEF IT21I0097, "Seguimiento remoto e integración interoperable a ficha clínica de la adherencia a tratamiento antihipertensivo de personas mayores", se desarrolla una aplicación móvil con el objetivo de apoyar el proceso de recolección, análisis y visualización de datos que caracterizan la adherencia a los tratamientos farmacológicos de pacientes crónicos. Los datos en cuestión que serán analizados son recolectados haciendo uso de la aplicación móvil desarrollada en este trabajo.

El objetivo principal de la aplicación es entregarle a los pacientes con enfermedades crónicas un sistema intuitivo y de uso sencillo para recolectar los datos de las ingestas de sus medicamentos para su eventual análisis por sistemas externos, además de informarles de los momentos en los que deben ingerir los medicamentos. Entregarle a los pacientes un sistema intuitivo permitirá un aumento en la tasa de uso en cada uno de ellos, lo que entregará datos más precisos y actualizados que habilita la detección temprana de problemas en los pacientes. Los datos recolectados por la aplicación desarrollada son enviados a un sistema externo que los redistribuye a distintos paneles (o dashboards) donde serán analizados para ser presentados a los profesionales médicos para la toma de decisiones guiada por datos fidedignos.

Además de las características de la aplicación que asistirán a los pacientes en la ingesta de sus medicamentos, también se concibieron funcionalidades que tienen como objetivo fomentar el uso de la aplicación desde un ámbito lúdico como lo es un chat de texto, presentación de artículos informativos y un arreglo de videos educativos y de entretención.

1.1. Objetivos

1.1.1. Objetivo general

Apoyar la comunicación de datos objetivos de tratamiento farmacológico a personal sanitario, mediante el desarrollo de una aplicación móvil que capture, gestione y transmita información de datos de tratamiento de pacientes crónicos.

1.1.2. Objetivos específicos

1. Desarrollar una arquitectura de aplicación móvil para alerta, captura y transmisión de datos de tratamiento farmacológico a un servidor remoto, resguardando su privacidad.

2. Evaluar la factibilidad de la interoperabilidad de la aplicación móvil con sistemas de ficha clínica, mediante el análisis de las tecnologías existentes sobre estos sistemas y protocolos de comunicación específicos del dominio de salud.

3. Implementar y evaluar criterios de usabilidad móvil para personas mayores en el desarrollo de una interfaz móvil del ámbito de salud.

1.2. Metodología de trabajo

- Se adopta un método de desarrollo iterativo e incremental, con adaptaciones de propuestas ágiles a un entorno de desarrollo unipersonal.

- El desarrollo se basa en un análisis de tecnologías existentes en el área de sistemas de salud que sean pertinentes a los objetivos de la Memoria.

1.3. Estructura del informe

Posterior a la Introducción es presentado el marco teórico donde están listados conceptos previos claves dentro del informe y otras soluciones existentes de pastilleros digitales seguido de un análisis de los mismos.

Después, se listan los requerimientos funcionales y no funcionales definidos por el equipo de

investigadores del proyecto, además de aspectos que se consideraron necesarios después de evaluar las soluciones existentes de pastilleros electrónicos.

A continuación, se delinea la solución propuesta y la implementación de la solución. Donde se presenta la arquitectura que se siguió para levantar la aplicación, las consideraciones que se tuvieron a la hora de diseñar la interfaz gráfica y una descripción de las tecnologías usadas.

Se sigue con los resultados obtenidos, es decir, las funcionalidades que la aplicación posee, informando sus objetivos y ahondando en cada una de ellas informando las tecnologías destacables utilizadas.

Posteriormente, se presentan las condiciones de aceptación complidas, presentadas en los requerimientos, y los datos obtenidos con las pruebas de usuarios haciendo uso del cuestionario SUS.

Finalmente, es presentado el trabajo futuro que se debe llevar a cabo en la aplicación para un funcionamiento óptimo y las conclusiones obtenidas una vez finalizado el desarrollo.

2. Marco teórico

2.1. Conceptos previos

2.1.1. Términos médicos básicos

A continuación se listan algunos términos médicos clave extraídos del diccionario de la Real Academia Nacional de Medicina [16] que son referenciados numerosas veces en el informe.

1. **Medicamento:** Sustancia que, administrada interior o exteriormente a un organismo animal, sirve para prevenir, curar o aliviar la enfermedad y corregir o reparar las secuelas de ésta.

2. **Prescripción:** Disposición o conjunto de disposiciones que el médico da al enfermo, relativas al régimen de vida, la alimentación, el reposo, el ejercicio físico, los medicamentos,

etc.

3. **Posología:** Disciplina científica, rama de la terapéutica y de la farmacología, que se ocupa de determinar la dosis y el intervalo de administración correcto de los medicamentos.

4. **Tratamiento:** Conjunto de medidas médicas, farmacológicas, quirúrgicas, físicas o de otro tipo encaminadas a curar o a aliviar las enfermedades.

2.1.2. Ingesta prescrita y registro ingesta

A los pacientes se les permite ingerir medicamentos en momentos distintos para los cuales se les prescribió, e incluso pueden posponer u omitir por completo una ingesta. Esto puede generar complicaciones a la hora de llevar el registro de las ingestas.

Para llevar el registro de las ingestas particulares se han creado dos conceptos que sintetizan la información de la prescripción y su posología asociada. Con la única diferencia de que una de ellas no puede ser alterada por el paciente y la otra sí:

- **Ingesta prescrita:** Es la ingesta que el paciente debe realizar según la prescripción y su posología asociada sin cambios en la hora o la cantidad de medicamento a ingerir.

- **Registro ingesta:** El registro ingesta es la información de ingesta que el paciente realmente realizó y que la aplicación móvil recolecta, almacena localmente y posteriormente envía al backend. La ingesta que el paciente realmente realizó puede ser igual a la ingesta prescrita, puede tener una hora de ingesta distinta a ella o puede no haberse ingerido medicamento en absoluto.

2.1.3. FHIR

FHIR [5] (Fast Healthcare Interoperability Resources) es un estándar de datos para el intercambio de información de salud creado por HL7 (Health Level Seven International) con el objetivo de garantizar la segura y fácil interoperabilidad de los datos a través de los distintos sistemas donde sean usados.

2.2. Evaluación de soluciones existentes

A continuación se procede a evaluar las soluciones existentes en el mercado que buscan remediar el problema de la adherencia a tratamientos farmacológicos. Se evaluó la aplicación móvil *Medisafe* [15] creada por la empresa de Reino Unido con el mismo nombre y la aplicación *Salud* [18] exclusiva del sistema operativo IOS. Esta última fue la principal fuente de inspiración a la hora de crear la interfaz gráfica de esta memoria de título. Se presenta de Medisafe extensamente y de Salud en los ámbitos de interfaz los cuales inspiraron el desarrollo de la aplicación.

Los aspectos a evaluar de Medisafe y Salud fueron seleccionados en base a los requerimientos que se tenían para el proyecto y se corroboró su efectividad en base a la experiencia de uso de las mismas.

- **Adherencia a tratamientos farmacológicos:** Se estudian las funcionalidades que elevan la tasa de adherencia a los tratamientos farmacológicos.
- **Usabilidad e interfaz gráfica:** Revisar cómo se mantiene al usuario informado de los medicamentos que debe ingerir y cómo se le informa de los medicamentos que ya ha ingerido es clave para una interfaz gráfica efectiva. La capacidad de la aplicación para responder a los cambios externos a la aplicación también serán evaluados.
- **Hacer llegar datos de adherencia a profesionales médicos:** Se evalúan Los datos que recolecta y su capacidad para entregarlos a los profesionales médicos.

Medisafe busca solucionar el problema de la adherencia a tratamientos farmacológicos mediante múltiples funcionalidades en la aplicación:

- **Visualización de medicamentos:** La aplicación móvil muestra en todo momento los medicamentos que el usuario debe tomar, los que ya se ha tomado y los que se ha saltado.
- **Recordatorios de ingesta de medicamentos:** La aplicación móvil entrega recordatorios de ingesta de medicamentos por medio de notificaciones locales en el dispositivo móvil del paciente. Estas notificaciones son entregadas al momento que el paciente debe ingerir el medicamento y en momentos posteriores a la hora de ingesta si el paciente aún no ha informado de la toma del medicamento.

- **Entrega el control de las ingestas:** Se le entrega al paciente el control de la ingesta de medicamentos, dándole la capacidad de informar de la ingesta de medicamentos, posponer la ingesta de medicamentos a una hora que a él más le convenga o saltarse la ingesta del medicamento por una variedad de motivos que el paciente puede informar (se le acabó el medicamento, el medicamento le está causando efectos secundarios, entre otros).

- **Estado de ánimo general:** En menor medida se le pregunta al paciente cómo se siente en general. Esta pregunta puede entregar información adicional no relacionada a la adherencia a tratamientos farmacológicos, pero sí la respuesta del cuerpo del paciente a los medicamentos que está ingiriendo. Esta información podría habilitar a los profesionales médicos a cambiar el medicamento oportunamente para evitar el abandono del tratamiento por parte del paciente, por ejemplo.

La interfaz gráfica de Salud es intuitiva y fácil de navegar:

- **Iconografía y paleta de colores:** hace uso de iconos descriptivos que muestran el estado de la ingesta del medicamento del paciente, además de una paleta de colores que puede generar emociones de urgencia y calma según la situación de alguna ingesta particular lo amerite.

- **Facilidad de uso:** La aplicación tiene su contenido a pocos "clicks" de distancia y es clara a la hora de distinguir sus botones.

Medisafe tiene funcionalidades para entregar datos objetivos sobre la adherencia a tratamientos farmacológicos a los profesionales médicos:

- **Envío de archivo con registro de ingestas:** La aplicación es capaz de enviar un archivo .csv con los datos de ingesta. Los datos de ingesta enviados tienen datos como el nombre del medicamento, la fecha de registro, la fecha y hora de ingesta, entre otros.

Medisafe posee también características no funcionales que la hacen una aplicación móvil de calidad:

- **Compatibilidad con dispositivos móviles:** La aplicación es compatible con dispositivos móviles con sistema operativo Android.

- **Responsividad de la aplicación:** La pantalla de medicamentos se adapta a la situación en la que se encuentre la misma. Siempre mantiene actualizada la pantalla de medicamentos respondiendo a cambios de hora o a configuraciones que el usuario haya realizado fuera de la aplicación. La aplicación es capaz de responder a cambios de hora, e incluso de día, en tiempo real y sin necesidad de que el usuario interactúe con ella.

- **Localidad de los datos:** El sistema guarda los datos de ingesta de medicamentos en el dispositivo móvil del paciente, permitiendo al paciente usar la aplicación incluso si no tiene conexión a internet.

2.3. Adherencia a tratamientos farmacológicos

Existen diversas formas para medir la adherencia a tratamientos farmacológicos, la cual supone la consistencia de los pacientes en ingerir sus medicamentos prescritos a la hora indicada. La definición para referirse al término aún se debate, pero hay formas confiables para llegar a un estimado de su valor cuantitativo.

La empresa de análisis de datos del cuidado de la salud Komodo Health, llevó a cabo dos estudios utilizando datos de personas reales antes y después de utilizar la aplicación Medisafe [14]. La empresa encontró un aumento en la adherencia de los pacientes se espera aumentar de un 7 % además de otros datos estadísticos que comparten. Para fines de esta memoria, la adherencia a tratamientos farmacológicos en un porcentaje similar.

3. Requerimientos

La aplicación móvil tiene como objetivo apoyar a los pacientes con enfermedades crónicas con sus tratamientos farmacológicos. Se definió un único usuario que manejará la aplicación móvil:

- **Usuario/paciente** Ingiere los medicamentos que se le prescriben y es el responsable de informar de la ingesta de los mismos a la aplicación móvil, al igual que informar de su estado de ánimo general.

3.1. Historias de usuario

Ahora veremos las historias de usuario generadas a partir de los requerimientos que se tenían para el proyecto. Estas son:

1. Como **usuario/paciente** necesito que en todo momento se me muestre el estado de las ingestas que debo llevar a cabo a lo largo del día.

    - Criterios de aceptación:

        a) La aplicación debe mostrar las ingestas que se deben ingerir para el día actual.

        b) La aplicación debe mostrar las ingestas que se han ingerido en el día actual.

        c) La aplicación debe mostrar las ingestas que se han saltado y no fueron ingeridas en el día actual.

2. Como **usuario/paciente** requiero que la aplicación móvil me informe de los medicamentos que debo ingerir a la hora de ingesta estipulada.

    - Criterios de aceptación:

        a) La aplicación debe informar por medio de notificaciones locales los medicamentos que el paciente debe tomar.

3. Como **usuario/paciente** debo ser capaz de informar de las ingestas de medicamentos a la aplicación móvil.

    - Criterios de aceptación:

        a) La aplicación posee mecanismos que facilitan informar sobre la ingesta de medicamentos.

4. Como **usuario/paciente**, además de poder informar de la ingesta de mis medicamentos, quiero tener el control de los medicamentos que ingiero.

    - Criterios de aceptación:

        a) La aplicación debe permitir posponer la ingesta de medicamento a una hora posterior.

        b) La aplicación debe permitir saltarse la ingesta de medicamento.

5. Como **usuario/paciente** quiero ser capaz de informar sobre mi estado de ánimo general.

    - Criterios de aceptación:

        a) La aplicación le pregunta al paciente cuál es su estado de ánimo general una vez por día.

        b) Las opciones deben ir desde "muy mal" hasta "muy bien" o incluso "enfermo" o "enojado".

6. Como **usuario/paciente** quiero usar la aplicación para motivos lúdicos e informativos para tener otra razón para abrir la aplicación, además de para revisar los medicamentos que debo ingerir.

    - Criterios de aceptación:

        a) La aplicación debe mostrar una sección de artículos de interés.

        b) La aplicación debe ser capaz de reproducir videos desde una fuente de datos particular.

        c) La aplicación debe poseer un chat de texto.

7. Como **usuario/paciente** quiero usar la aplicación incluso si no tengo acceso a internet.

    - Criterios de aceptación:

        a) El sistema debe ser capaz de recordar los registros de las ingestas de medicamentos hechas en momentos donde no se tuvo internet para enviarlas al backend en una conexión posterior.

        b) El sistema debe ser capaz de recordar los registros de los estados de ánimo informados en momentos donde no se tuvo internet para enviarlos al backend en una conexión posterior.

        c) El sistema debe informar al usuario que sus datos serán registrados incluso si se encuentra sin conexión a internet.

3.2. Requerimientos no funcionales

1. La aplicación necesita ser compatible con dispositivos móviles que utilicen el sistema operativo Android.

2. La aplicación precisa ser compatible con el estándar FHIR para el intercambio de información de salud.

3. La aplicación debe ser capaz de precargar información del paciente, de sus tratamientos, medicamentos y posologías asociadas, desde un servidor que siga el estándar FHIR.

4. La aplicación requiere la capacidad para enviar información a un servidor que siga el standard FHIR.

5. La aplicación debe ser fácil de usar, comprensible y simple de navegar.

4. Solución propuesta

4.1. Arquitectura del sistema

El sistema completo que recupera, analiza y muestra los datos de ingesta de medicamentos está conformado por tres subsistemas que trabajan en conjunto.

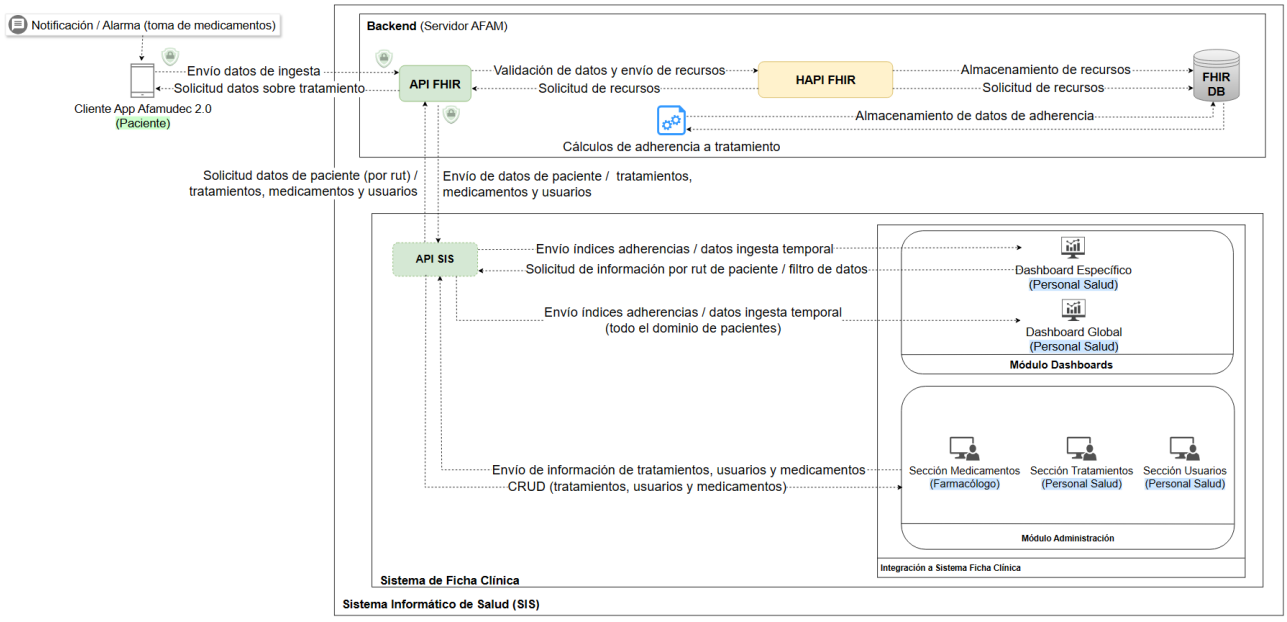


Figura 1: Arquitectura sistema AFAM 2.0

Los tres subsistemas son:

1. La **aplicación móvil** encargada de informar al paciente los medicamentos que debe ingerir y de recolectar sus datos de ingesta de medicamentos. Esta se comunica con la API FHIR encontrada dentro del backend.

2. El **servidor AFAM (el backend)** lleva a cabo cálculos y análisis de datos para determinar el nivel de adherencia del paciente. Los cálculos se llevan a cabo haciendo uso de los datos obtenidos desde la aplicación móvil los cuales llegan desde la **API FHIR**. Los datos serán posteriormente ordenados por el gestor **HAPI FHIR** para asegurar su integridad y, por acción del mismo, almacenados en la **FHIR DB**.

3. Finalmente, el **sistema de ficha clínica** funciona como la fuente de datos principal del sistema y se encarga de mostrar, analizar y ordenar los datos para los profesionales médicos y de administración. Estas acciones las lleva a cabo haciendo uso de:

    - La **API SIS** que posee los datos de adherencia y envía los que le solicitan, desde la API FHIR ubicada en el backend.
    - El **módulo administración** se encarga de almacenar y enviar los datos de medicamentos, tratamientos y pacientes.
    - Y el **módulo dashboards** se encarga de mostrar y llevar a cabo análisis de datos de adherencia para los profesionales médicos y de administración.

4.2. Arquitectura de la aplicación móvil

Se optó por usar una implementación de tres capas para la aplicación móvil. Se utilizó la arquitectura de tres capas llamada "Flutter Clean Architecture" que sigue estrictamente el principio de diseño de la separación de intereses. Ésta favorece el desarrollo de aplicaciones móviles y les entrega un alto grado de escalabilidad, legibilidad para el código fuente, testabilidad y se vuelve fácilmente mantenible en el tiempo. La arquitectura concibe una capa de características, una capa de dominio y una capa de datos las cuales funcionan de forma independiente entre sí.

La capa de características (o feature layer) se encarga de la presentación de la información

al usuario y es la capa más dependiente del framework que se use. En esta capa se encuentran las vistas (o screens) de la aplicación. Las vistas son las encargadas de mostrar la información que el usuario necesita para interactuar con la aplicación. También se encuentran en esta capa los controladores de las vistas, o state managements, que son los encargados de mostrar los datos en las vistas, además de manejar el estado de los datos que se encuentran actualmente en las vistas, como los datos ingresados por el mismo usuario. Además, se ocupa de llevar a cabo los cambios pertinentes a la hora de actualizar la interfaz gráfica cuando ocurra algún cambio que lo amerite.

La capa de dominio (o domain layer) es la encargada de la lógica de negocio de la aplicación móvil. Ésta no tiene dependencias con el resto de las capas y contiene las entidades (o entities), objetos de soporte que estructuran la información para su uso en los casos de uso (o use cases) los principales responsables de la lógica de negocio, la lógica particular para la aplicación que implementan las funcionalidades creadas para ella; y los repositorios (o repositories), que se ocupan de abstraer las funcionalidades de las capas externas, además de ser el puente entre los datos provenientes desde las fuentes de datos hasta las entidades, haciendo uso de objetos llamados "mappers", que mapean los datos en los modelos DTO (Data Transfer Object) a las entidades. Esta abstracción permite una migración de dependencias sencilla y elude la necesidad de llevar a cabo cambios mayores en el código si fuese necesaria tal migración.

La capa de datos (o data layer) es la capa más externa de la aplicación y se encarga de obtener la información de los sistemas distribuidores de estos. Los datos pueden ser solicitados a un servidor remoto usando una API y/o desde una base de datos local. Contiene además la implementación de los repositorios, los modelos DTO, que nos permiten manejar los datos que las fuentes de datos (o datasources) nos entregan; de las cuales existen dos tipos:

- Los remotos se encargan de obtener los datos de una API externa accediendo a ellas por medio de un URL y una API key, esta última solo si es necesario.
- Los locales obtienen los datos de una fuente de datos local, como una base de datos relacional programada dentro de la misma aplicación o desde sistemas de almacenamiento más simples que guardan datos de menor complejidad.

Existen dos capas más de soporte que son la de librerías compartidas y la de recursos (o assets).

La de librerías compartidas se encarga de reunir las librerías usadas por la aplicación para no tener que importarlas en cada uno de los archivos de la aplicación y la de recursos almacena archivos de imágenes, vídeos y audio, entre otros.

En la figura 1 podemos apreciar la arquitectura de tres capas que se utiliza para el desarrollo de la aplicación móvil. Podemos ver que las distintas capas van abstrayendo la implementación de las capas más externas. Mientras más avanzamos a las capas internas de la arquitectura, más independientes de la implementación de las capas externas se vuelven las internas.

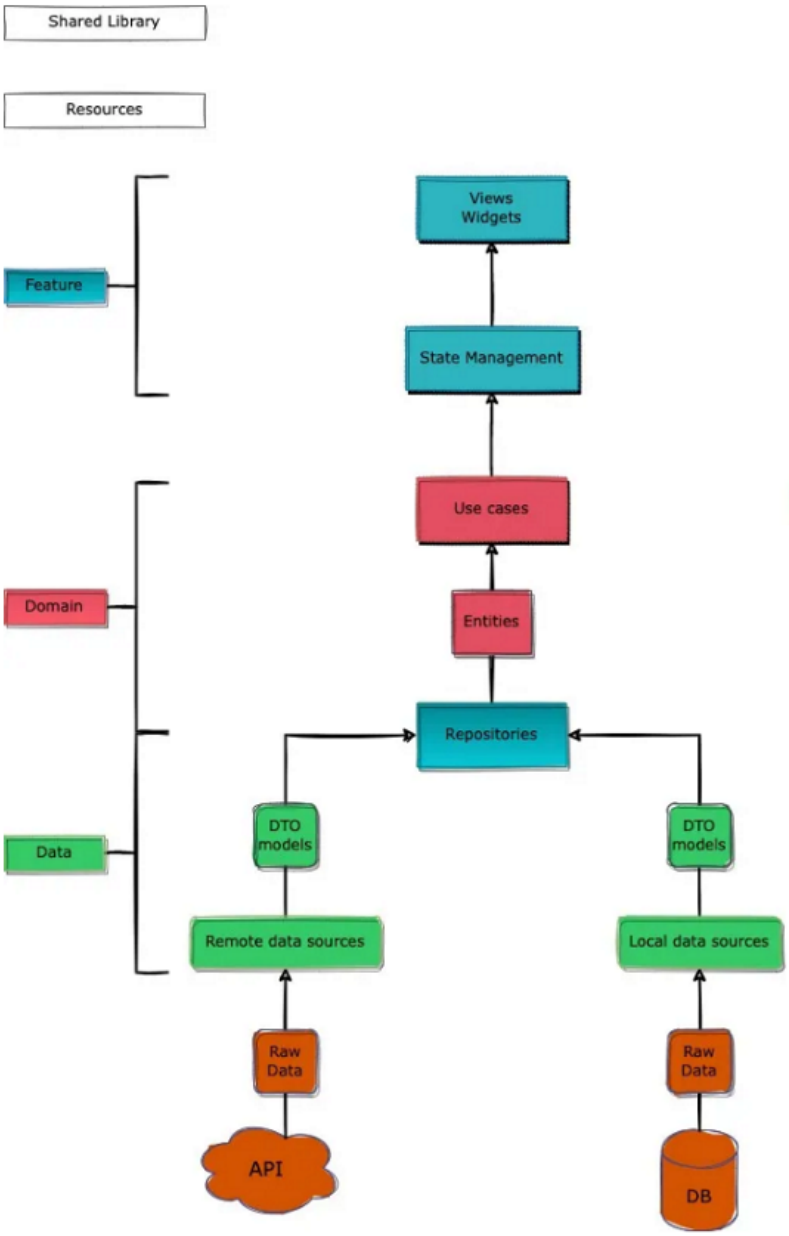


Figura 2: Diagrama de la arquitectura Flutter Clean Architecture [2]

Particularmente para la aplicación desarrollada, las APIs donde se obtienen los datos provienen del backend donde se obtienen los datos médicos del servidor FHIR y de la API de Youtube para recuperar datos de videos dentro de la plataforma. La primera fuente, el backend, es también la encargada de dar soporte a las características que distribuyen los artículos de información y entretención; y habilitar el uso del chat comunitario. Este nivel de abstracción también es aplicado para la base de datos local. Sin embargo, como ya veremos, muchos datos hacen su camino desde las distintas APIs hasta el repositorio y finalmente llegan hasta los sistemas de

almacenamiento local, una vez estructurados en entidades. De esta forma la aplicación hace uso únicamente de los datos que hayan llegado a éstas. Ésta fue una decisión de diseño elegida deliberadamente para cumplir con el requerimiento del funcionamiento offline de la aplicación.

En la capa de dominio los datos obtenidos desde la capa de datos es abstraída usando una clase abstracta (o abstract class) para cada una de las fuentes de datos utilizadas en la aplicación. Muchas de las entidades implementadas, que usamos para estructurar y manipular los datos dentro de la aplicación, siguen la misma estructura de datos que la base de datos local utiliza como filas (o rows) para almacenar la información. Es decir, las tablas de la base de datos local, la cual se presenta más adelante a fondo, contienen los mismos datos que sus entidades con el mismo nombre. Así, por dar un ejemplo, la tabla "Registro_Ingesta" tiene una entidad con el mismo nombre que nos auxilia a la hora de guardar datos en la base de datos local y a la hora de utilizar datos almacenados en la misma. Las entidades en la aplicación son utilizadas finalmente en los casos de uso, que se ocupan de guardar los datos localmente, enviar los datos que deban ser enviados o cualquier otra tarea que se le haya entregado y después hacer los cambios pertinentes para, posteriormente, informar de estos cambios, si es necesario, de vuelta a los state managers para que estos actualicen la UI.

Finalmente, en la capa de características se les entregan los datos a los state managers para que los muestren en las views. Además de ser los encargados de difundir los datos en las views, los state managers se encargan de "prestar atención" a las acciones que lleve a cabo el usuario en la UI. Por ejemplo, los botones que los usuarios presionen o los cambios que están fuera del control de la aplicación tales como los cambios de estado ligados a la hora. Una vez que los usuarios presionen ciertos botones, el o los state managers se ocupan de llevar a cabo los trabajos de solicitarle a los casos de uso los procesos mencionados en el párrafo anterior. La mayor parte del tiempo, los state managers van a comunicarse con los Widgets implementados. Estos son los responsables finales de mostrarle al usuario los datos y hacer aparecer las vistas, botones y demases con los cuales los usuarios utilizan la aplicación.

4.3. Fuentes de datos y repositorios

Los dos trabajos principales de la aplicación son informarle al paciente de los medicamentos que debe ingerir y entregarle la capacidad para informar cuando ha ingerido sus medicamentos. Por tanto, se requiere de una puerta que permita la entrada y salida de información, desde y hacia la aplicación; y que los entregue de tal forma que a la aplicación le sea fácil manejarlos.

Se optó por utilizar el patrón de diseño que implementa fuentes de datos y repositorios, mejor conocidos en inglés como datasources y repositories. Los datasources son una fuente de información particular que provienen de una localización en específico con las cuales hacemos las llamadas API. Por su parte, los repositorios son la abstracción de las fuentes de datos. Así, se pueden tener múltiples fuentes de datos que entreguen un mismo tipo de información, los cuales convergen a un repositorio en común, el que acepta un formato de datos genéricos para que al resto de la aplicación le facilite su uso.

4.4. Esquema de la base de datos local

Para soportar el requerimiento de la funcionalidad offline en la aplicación fue necesario implementar una base de datos local. Esta se encarga de almacenar los datos que son enviados desde el backend a la aplicación móvil y los datos que deben ser enviados al backend para su análisis. Se optó por implementar la base de datos más utilizada en el framework Flutter llamada SQFlite, una implementación de SQLite para el lenguaje de programación Dart usando Flutter. SQLite es una base de datos relacional muy utilizada en aplicaciones móviles por ser ligera y rápida.

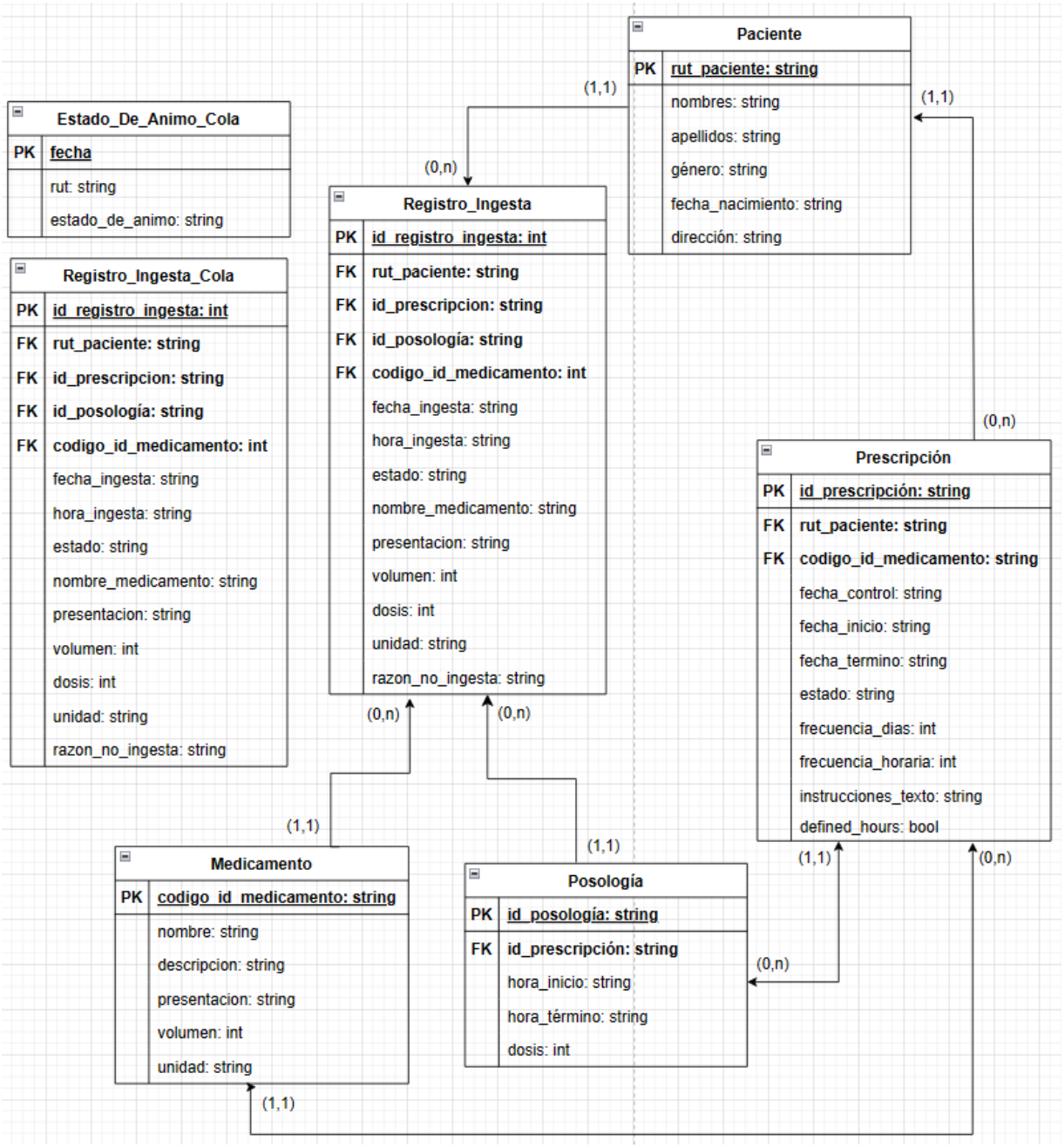


Figura 3: Diagrama UML de la base de datos local

Los pacientes tienen prescripciones asociadas a ellos. Y cada prescripción tendrá siempre un medicamento asociado y su posología. El medicamento asociado es el medicamento que el paciente debe ingerir y la posología indica la cantidad y la hora en la que el paciente debe ingerir el medicamento.

La tabla Registro_Ingesta guarda las ingestas que el paciente informa desde la aplicación móvil.

Esta tabla guarda los datos para un requerimiento futuro que permitirá analizar los datos de forma local para mostrar un resumen de la adherencia a los tratamientos farmacológicos del paciente. Esta funcionalidad no está contemplada en el alcance de la memoria, pero se aprovechó de implementar su requerimiento en la base de datos local para no tener que hacer cambios mayores en el futuro.

La tabla de Registro_Ingesta_Cola y Estado_De_Animo_Cola cumplen la función de una cola de envío de datos al backend. Los datos en ellas son enviados al backend al momento que éstas entran en ellas a menos que no haya conexión a internet. En ese caso los datos serán almacenados en las mismas tablas para ser enviados en una futura conexión a internet. La tabla de Registro_Ingesta_Cola no está relacionada con claves foráneas junto al resto de las tablas porque sus filas son una copia de las filas en Registro_Ingesta que no se lograron enviar por falta de conexión a internet. Esta copia de los datos no es necesaria para su correcto funcionamiento, pero su uso de esta forma permite ahorrar el tiempo de búsqueda de las filas que no pudieron ser enviadas por falta de conexión a internet. La tabla de Estado_De_Animo_Cola no tiene una tabla copia que esté relacionada con claves foráneas porque la aplicación no requiere una copia de los datos del estado de ánimo de los pacientes.

4.5. Diseño de la interfaz gráfica

La interfaz gráfica consideró las heurísticas de usabilidad que propuso Jakob Nielsen [13] las cuales guiaron el diseño de la interfaz de usuario. Estas heurísticas son:

1. Visibilidad del estado del sistema.
2. Coincidencia entre el sistema y el mundo real.
3. Control y libertad del usuario.
4. Consistencia y estándares.
5. Prevención de errores.
6. Reconocimiento en lugar de recuerdo.
7. Flexibilidad y eficiencia de uso.

8. Diseño estético y minimalista.

9. Reconocimiento, diagnóstico y recuperación de errores.

10. Ayuda y documentación.

El diseño de la interfaz gráfica se aferró fuertemente a las heurísticas para su desarrollo.

La interfaz gráfica pasó por múltiples cambios a lo largo de todo el desarrollo de la aplicación. Así, usando Flutter como el framework de trabajo se logró una interfaz gráfica que es intuitiva y fácil de usar.

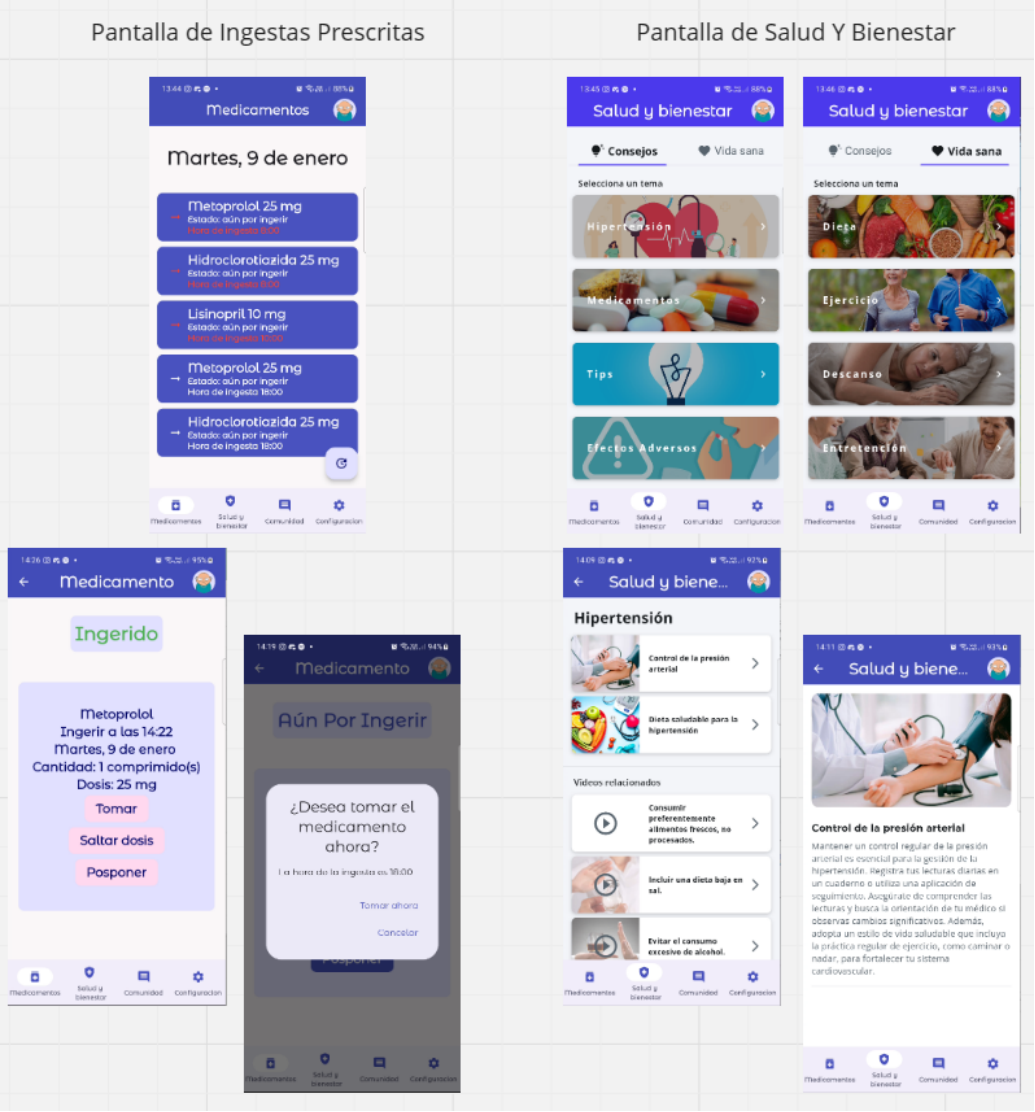


Figura 4: Versión prematura de la interfaz gráfica de la aplicación

La interfaz gráfica de la aplicación poseía originalmente un estilo simple, pero conveniente para el usuario y no se ahondó en el diseño de la misma para mantener el foco en el manejo de datos y la conexión con el backend externo que la aplicación debe llevar a cabo.

El diseño final de la aplicación móvil se decidió con la ayuda de un diseñador gráfico que se encargó de refinarla. Se llegó al resultado final después de múltiples iteraciones que utilizaron la interfaz original de la aplicación como punto de partida.

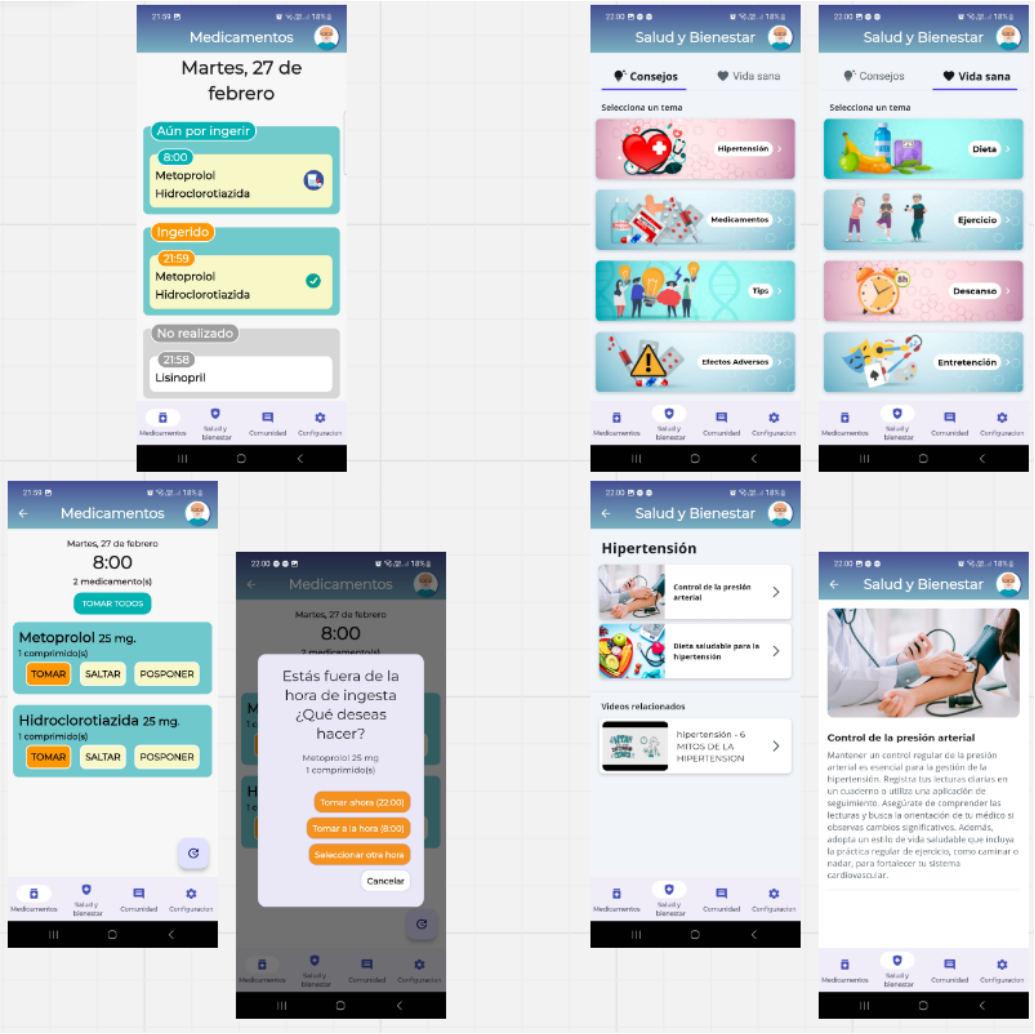


Figura 5: Versión final de la interfaz gráfica de la aplicación

El nuevo diseño utilizó tonos de color más claros, una iconografía más descriptiva y una paleta de colores más atractiva para el usuario. Se decidió también por una distribución de los elementos más clara y fácil de leer.

5. Implementación de la solución

Se decidió desarrollar la aplicación para dispositivos móviles Android por la demografía de los usuarios que van a hacer uso de la aplicación. No obstante, muchas personas hoy en día usan equipos de segunda mano con el sistema operativo iOS. Atendido lo anterior, se ha decidido desarrollar la aplicación en el framework Flutter el cual permite desarrollar una misma aplicación para dispositivos Android e iOS. Si bien este proyecto se enfoca en el desarrollo para su uso único en Android, si en el futuro se desea compilar la aplicación para dispositivos iOS se podrán hacer cambios menores en el código para permitir su uso en ambos sistemas operativos.

5.1. Flutter

Flutter [6] es un framework de código abierto creado por Google para desarrollar aplicaciones móviles, web y de escritorio. Fue lanzado en 2017 y es la opción favorita de múltiples profesionales y no profesionales para el desarrollo de aplicaciones móviles por su facilidad de uso y su capacidad para compilar su código para dispositivos Android e iOS, entre otros, con un solo código fuente. Flutter es escrito en el lenguaje de programación Dart y C/C++ siendo Dart el lenguaje que principalmente se usa para el desarrollo de aplicaciones en el framework. Su UI se muestra (se renderiza) en las pantallas como un árbol de Widgets. Todo dentro del framework gira en torno a los Widgets. Los Widgets son la pieza fundamental para crear la interfaz gráfica y se puede llegar a decir que todo dentro del framework es un Widget ya sea una imagen o una línea de texto simple, los cuales son posible de combinar entre sí para generar Widgets más complejos.

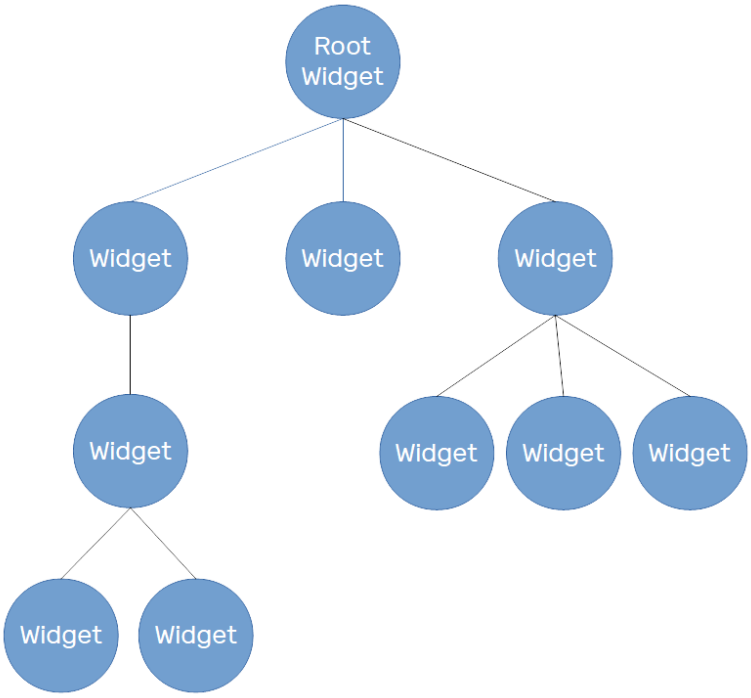


Figura 6: Cada Widget es un nodo en el árbol de Widgets

El framework permite el uso de librerías que pueden haber sido creadas por el mismo equipo responsable de Flutter o por terceros. Estas librerías facultan el uso de funcionalidades que el framework no posee por defecto. Para este proyecto se utilizaron las siguientes librerías:

- **Riverpod State Manager:** [17] Un state manager facilita la separación entre la lógica de negocio y la interfaz de usuario, lo cual mejora la mantenibilidad del código, facilita

la escritura de pruebas unitarias y de integración al proporcionar una forma sencilla de inyectar dependencias en las partes relevantes del código. La clave aquí es la clara definición de la fuente única de la verdad (el estado de la aplicación o State) y la manipulación de ese estado, o state, a través de un conjunto consistente de operaciones. Riverpod, al igual que otros State Managers, proporciona una estructura organizativa sólida para esta separación, permitiendo al desarrollador centrarse en la lógica subyacente sin preocuparse por los detalles de la interfaz de usuario.

- **SQFlite:**[21][22] SQFlite facilita la incorporación de la popular base de datos SQLite para su uso en Flutter. La librería entrega una API que permite la creación, lectura, actualización y eliminación de datos en la base de datos sin la necesidad de escribir sentencias SQL.

- **Dio:**[4] Paquete de red para Dart/Flutter utilizado para hacer solicitudes HTTP el cual puede implementar configuraciones globales, interceptores, Timeout, Request cancelation y más. Es el único paquete que la aplicación usa para sus solicitudes HTTP.

- **go_router:** [12] Un paquete de enrutamiento declarativo para Flutter que entrega una API basada en URLs conveniente para la navegación entre pantallas dentro de la aplicación. Permite definir patrones URL y navegar usando URLs. Otras funciones que este paquete entrega y que la aplicación usa son:

  - **Template syntax:** Permite expresar un URL por un valor entregado, como por ejemplo el id de una ingesta.
  - **Redirection:** Revisando el estado de la aplicación se puede redirigir al usuario a un URL diferente. Por ejemplo cuando el usuario se autentica exitosamente en la pantalla de login, la aplicación se percata de ello y lo redirige a otra ruta de la aplicación.
  - **ShellRoute:** Se usa para mostrar un navegador interno que permite acceder a vistas predefinidas. Es usado para implementar la barra de navegación que permite ir a las vistas principales de la aplicación en casi cualquier lugar que nos encontremos en ella.

Flutter posee una forma inherente para la navegación entre vistas y Widgets, llamada "Navigator", pero es limitada y puede llevar a errores en el futuro. Algunas de las limitaciones y potenciales errores de "Navigator" son la falta de funciones de enrutamiento tales como rutas con nombre y entregar parámetros de ruteo. Además, patrones de navegación inconsistentes y creciente complejización del código.

- **shared_preferences:** [**19**] Un paquete de persistencia de datos que permite guardar cualquier variable del tipo int, double, bool, String y listas de String. A las variables se les entrega un nombre y pueden ser llamadas o pueden ser eliminadas con los métodos que el paquete entrega.

- **socket_io_client:** [**20**] Es un port del paquete socket.io para Dart. Permite establecer comunicación bidireccional y de baja latencia entre un servidor y sus clientes. Es usado en la aplicación para implementar un chat de texto.

- **youtube_player_flutter:** [**24**] Paquete que implementa un reproductor de videos de Youtube usando la API oficial de Youtube iFrame Player API creada para introducir fácilmente videos que se albergan en la página.

- **connectivity_plus:** [**3**] Extensión que le permite a la aplicación detectar conexiones de red y de bluetooth. Es capaz de diferenciar entre conexiones de red WiFi y celular.

- **flutter_material_pickers:**[**8**] Paquete que implementa "pickers" los cuales son Widgets que nos dejan seleccionar una hora, una fecha o valores predeterminados como números o palabras permitiendo además la selección de una o múltiples opciones.

- **flutter_local_notifications:**[**7**] Extensión que implementa las notificaciones locales en la aplicación.

5.2. Github

Github [11] es una plataforma de manejo de código fuente basada en git, una herramienta creada por Linus Torvalds, que ayuda a desarrolladores a tener un espacio centralizado en el internet para colaborar y acceder al código de forma remota.

5.3. FlutterFlow

Flutterflow [10] es una herramienta low-code capaz de generar código frontend de una aplicación hecha con Flutter, además de permitir hacer consultas a APIs.

5.4. Pub.dev

Pub.dev [9] es el administrador de paquetes oficial de Flutter y Dart.

5.5. YoutubeDataApiV3

YoutubeDataApiV3 [25] permite recuperar datos de Youtube, como los videos de un canal particular o listas de reproducción particulares.

6. Resultados

A continuación se presenta el trabajo realizado, el cual está separado por las vistas principales que le fueron entregadas a la aplicación. Si bien esta memoria se concentra en la intercepción, emisión y actualización de datos desde un backend FHIR y posterior presentación al usuario de los mismos, las otras funcionalidades, al igual que la funcionalidad foco de esta memoria, utilizaron las mismas tecnologías y paquetes, por lo que se ve relevante su inclusión en el informe, incluso si solo es en menor medida.

6.1. Mapa de navegación

Se experimentó con múltiples mapas de navegación, los cuales tenían sus ventajas y desventajas. El mapa de navegación siguiente se considera como el más sencillo de navegar por su baja cantidad de "clicks" para llegar a los distintos contenidos que la aplicación ofrece.

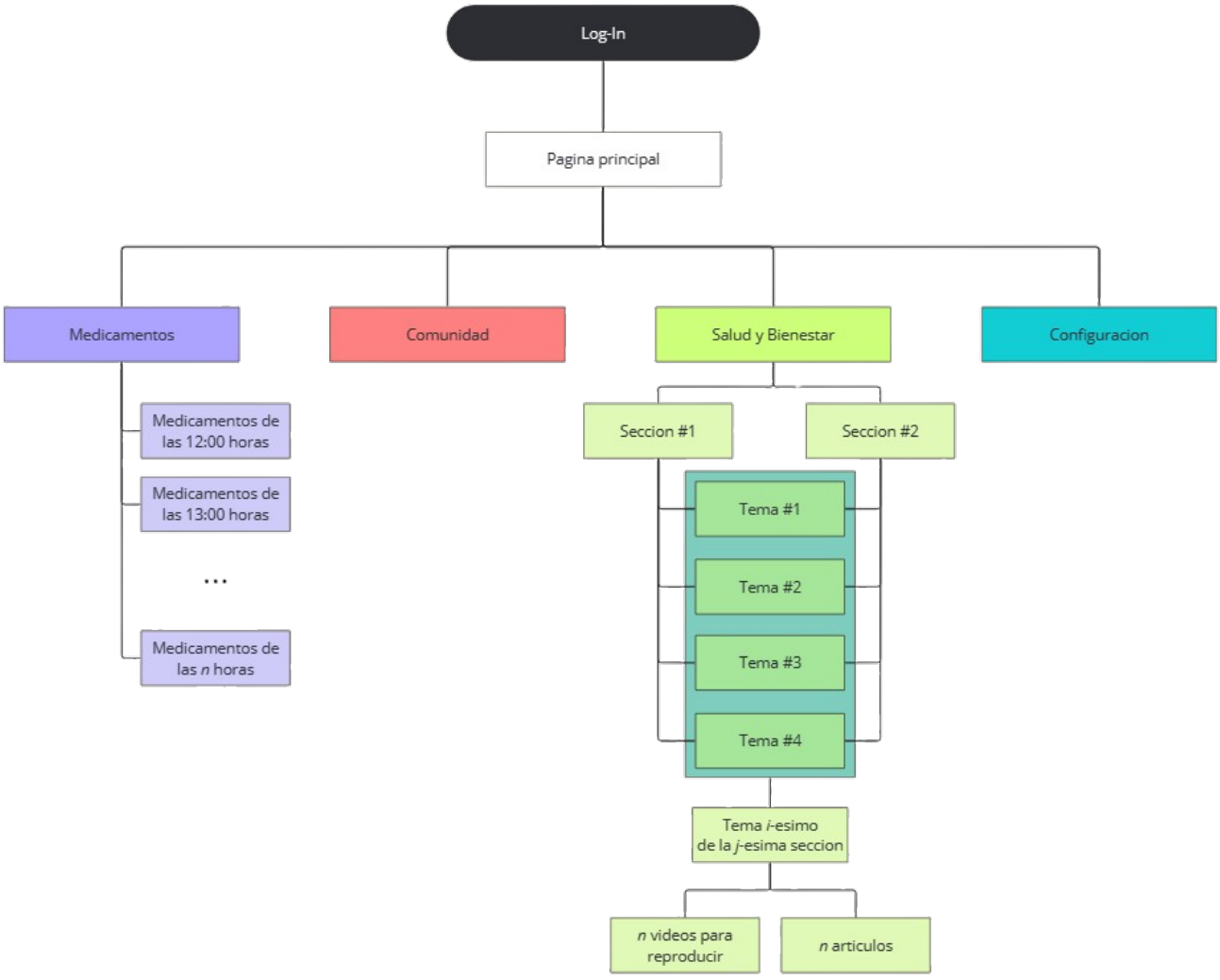


Figura 7: Mapa de navegación de la aplicación

La aplicación no posee una página principal como tal. La primera vista que se le presenta al usuario cuando abre la aplicación es siempre la de medicamentos (si éste ya ha iniciado sesión). Un Widget que se posiciona en la parte inferior de todas las ventanas, a excepción de la de inicio de sesión y en cualquiera de los videos para reproducir, nos permite navegar desde cualquier ventana a cualquiera de las tres ventanas principales: medicamento, comunidad y salud y bienestar.

Las funcionalidades y elecciones de diseño incorporadas en la aplicación pueden sobre complicar el diagrama y, por tanto, se obviaron las conexiones entre las páginas principales dentro del diagrama. En cambio, se solicita considerar el nodo “Página principal”, dentro del diagrama, como la ventana seleccionada actualmente, además de seguir siendo la misma dentro del diagrama. Esto significa que el usuario es capaz de navegar desde la ventana “Salud y Bienestar” a la misma ventana “Salud y Bienestar” y sus vistas de Selección #1 y Selección #2.

6.2. Medicamentos

Figura 8: Pantalla principal de “Medicamentos”

La pantalla de medicamentos le informa al usuario las ingestas que debe ingerir, las que ya ingirió y las ingestas que el mismo usuario informó que se saltó (no se tomó) en el día. Los medicamentos se agrupan por la hora en la que deben ser ingeridos, la hora a la que fueron ingeridos y la hora en la que fueron saltados dependiendo del estado de la ingesta.

En la parte inferior de la figura 8 podemos ver el Widget que nos permite navegar desde cualquier ventana, a excepción de la de inicio de sesión y del reproductor de videos, a cualquiera de las 4 ventanas principales de la aplicación. Para su implementación se usó la biblioteca GoRouter haciendo uso de ShellRoute, una de sus características de la que mostramos en la sección 6.

Las ingestas prescritas son obtenidas del backend y almacenadas en la base de datos local al momento que el usuario inicia sesión por primera vez y, posteriormente, son actualizadas de forma frecuente para mantenerse al día con nuevas prescripciones o cambios que los profesionales médicos puedan hacer en las prescripciones del paciente.

Figura 9: Vista presentada al apretar en alguno de los medicamentos agrupados por hora dentro del apartado "Aún por ingerir"de la Figura 7

En la Figura 8 se observa que hay cuatro botones que son únicos:

- **Tomar:** Presenta un pop-up preguntando al usuario si confirma la ingesta. Si el paciente está ingiriendo el medicamento a una hora distinta de la acordada, el pop-up cambia para

preguntarle si se lo tomó a la hora indicada, si se lo tomó en el momento en que lo está informando o si se lo tomó a una hora distinta. Si se lo tomó a una hora distinta, al paciente se le presenta un Widget que le permite seleccionar una hora específica.

- **Tomar todos:** Se presenta un pop-up preguntando al usuario si confirma la ingesta de todos los medicamentos de la hora seleccionada. También, tiene una variación por si ingiere los medicamentos a una hora distinta a la estipulada.

- **Saltar:** Aparece un pop-up de selección múltiple que le pregunta por la razón del salto de la ingesta. El usuario solo puede elegir una opción como respuesta.

- **Posponer:** Aparece en la pantalla un Widget que le permite al paciente seleccionar una hora. El medicamento será pospuesto para esa hora.

Los cuatro botones llevan a cabo un proceso de guardado y enviado de datos a la base de datos local y al backend respectivamente, a excepción del botón que pospone la ingesta, el cual sólo guarda información localmente. En el caso de que no haya internet el sistema guarda la información localmente para ser enviada en una conexión posterior.

Una vez que cualquiera de los botones es presionado, además de llevar a cabo el correspondiente manejo de los datos, el sistema actualiza las vistas para reflejar en tiempo real los cambios que se llevaron a cabo en la aplicación. Para esto, y para la actualización de los datos, el state manager "riverpod" fue clave para conservar las buenas prácticas de desarrollo sin comprometer el rendimiento de la aplicación.

El resto de las vistas de la funcionalidad de medicamentos y vistas alternativas de la pantalla principal de medicamentos son presentadas en el Anexo A. Un testimonio de la llegada de los datos enviados por la aplicación móvil al backend pueden ser vistos en el anexo B.

6.3. Estado de ánimo

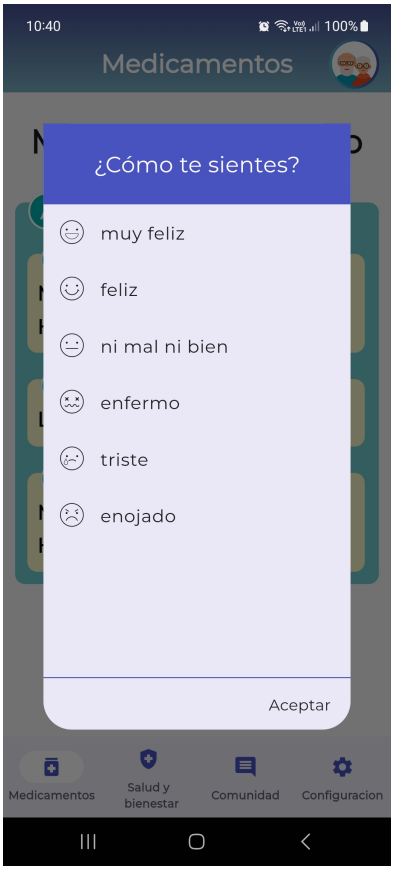


Figura 10: Widget "Estado de ánimo"

Una vez al día un Widget es presentado al usuario para preguntarle sobre su estado de ánimo general. El pop-up no desaparece hasta que el usuario seleccione una opción y presione aceptar.

Al igual que el registro ingesta enviado al informar de la toma, o no toma, de medicamentos, la funcionalidad del estado de ánimo también toma en consideración la posibilidad de que no haya internet. En ese caso la información también es guardada hasta que una nueva conexión se establezca.

6.4. Salud y bienestar

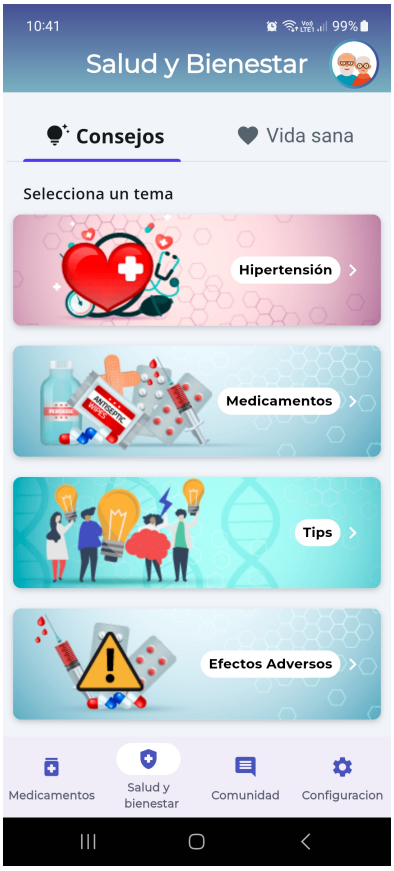


Figura 11: Pantalla principal de la vista “Salud y bienestar”

La pantalla de salud y bienestar tiene ocho temas. Cuatro son de la selección de consejos y el resto de la selección de vida sana. Cada tema posee artículos informativos y de entretención; y videos que están asociados con el tema tratado.

Los temas y artículos son obtenidos desde el backend y desde el mismo es posible integrar nuevos artículos. Por su parte, los videos son obtenidos desde la API “Youtube data api v3” que Youtube ofrece para obtener información de videos en la página. Los videos obtenidos provienen del canal “Afam Movil App” [1] y los títulos de cada video deben incluir el tema al cual pertenecen seguido del título deseado para el video o la aplicación no mostrará el video.

Para revisar las vistas que presentan los artículos y los videos vaya al anexo C de este informe.

6.5. Chat comunitario

Figura 12: Pantalla principal del chat comunitario

El chat comunitario es eso, un chat donde todos los pacientes pueden charlar en comunidad. Los mensajes son almacenados en el backend y redistribuidos a los usuarios de la aplicación usando socket_io en el backend y socket_io_client en Flutter para establecer conexión bidireccional y de comunicación continua para operar.

7. Evaluación de la propuesta

El testeo se llevó a cabo validando las condiciones de aceptación de las historias de usuario de la sección de requerimientos y las pruebas con usuarios se validaron utilizando tareas para los usuarios y un cuestionario SUS el cual respondieron una vez terminadas las tareas (ver anexo D).

7.1. Validación de las condiciones de aceptación

1. Como **usuario/paciente** necesito que en todo momento se me muestre el estado de las ingestas que debo llevar a cabo a lo largo del día.

    - La aplicación obtiene desde el backend las ingestas que el paciente debe llevar a cabo.
    - La aplicación muestra las ingestas que se han ingerido a lo largo del día gracias a la base de datos local que deja constancia de ellas.
    - La aplicación muestra las ingestas que no se han ingerido a lo largo del día gracias a la base de datos local que deja constancia de ellas y de su razón de no ingesta.

2. Como **usuario/paciente** requiero que la aplicación móvil me informe de los medicamentos que debo ingerir a la hora de ingesta estipulada.

    - La aplicación le informa, por medio de notificaciones locales, al usuario los medicamentos que debe tomar a lo largo del día.

3. Como **usuario/paciente** debo ser capaz de informar de las ingestas de medicamentos a la aplicación móvil.

    - Una vez que el usuario informa a la aplicación que se tomó (o no se tomó) un medicamento, esta se lo informa al backend por medio de su API de acceso (ver anexo B).

4. Como **usuario/paciente**, además de poder informar de la ingesta de mis medicamentos, quiero tener el control de los medicamentos que ingiero.

    - El usuario es capaz de posponer la ingesta del medicamento (anexo A figura 15). Se le es recordado si este entregó el permiso de acceso a las notificaciones.
    - El usuario puede decidir si tomar el medicamento o no (anexo A figura 14), Información que es enviada posteriormente al backend (ver anexo B).

5. Como **usuario/paciente** quiero ser capaz de informar sobre mi estado de ánimo general.

    - Una vez al día se le consulta al usuario por su estado de ánimo general.

- Las opciones son variadas y pueden ser vistas en la figura 9 de la subsección 6.3 “Estado de ánimo”.

6. Como **usuario/paciente** quiero usar la aplicación para motivos lúdicos e informativos para tener otra razón para abrir la aplicación, además de para revisar los medicamentos que debo ingerir.

    - En la vista principal “Salud y bienestar” se tiene acceso a artículos de interés para los usuarios, obtenidos desde el backend haciendo uso de la API de acceso a ella.
    - La aplicación es capaz de reproducir videos almacenados en el canal de Youtube “Afam móvil App” haciendo uso de la API de Youtube “Youtube data api v3”.
    - Haciendo uso del paquete socket_io_client de Flutter se pueden enviar y recibir los mensajes de texto desde el backend.

7. Como **usuario/paciente** quiero usar la aplicación incluso si no tengo acceso a internet.

    - Haciendo uso de la base de datos local, las ingestas realizadas en momentos donde no había conexión a internet son almacenadas para ser enviadas en una conexión posterior.
    - Haciendo uso de la base de datos local, los estados de ánimo informados en momentos donde no había conexión a internet son almacenados para ser enviados en una conexión posterior.
    - Si el usuario se encuentra sin internet recibe un mensaje en la parte superior de sus ingestas, informando sobre el estado del internet y que sus datos seguirán siendo registrados incluso sin internet.

7.2. Requerimientos no funcionales

1. La aplicación es totalmente compatible con el sistema Android incluyendo sus librerías y los sistemas implementados en ella.

2. El sistema es compatible con el estándar de reglas FHIR para el intercambio de información de salud.

3. La aplicación es capaz de precargar la información de ingesta del paciente desde el servidor FHIR.

4. La aplicación es capaz de enviar información a un servidor FHIR.

5. La aplicación cumple con los principios fundamentales que guían el diseño de interfaces de usuario creadas por Jakob Nielsen.

7.3. Pruebas de usuario

Para evaluar la usabilidad de la aplicación se llevó a cabo una encuesta utilizando el sistema de escalas de usabilidad SUS (*System Usability Scale* [23]).

Primero se les solicitó a los usuarios llevar a cabo unas tareas para que se familiarizaran con la aplicación y luego se les pidió completar una encuesta con afirmaciones sobre la aplicación. Cada afirmación podía ser punteada con un valor del 1 al 5 donde 1 es estar "totalmente en desacuerdo" con la afirmación y 5 es estar "totalmente de acuerdo" con la afirmación. Los encuestados fueron seleccionados del equipo de desarrollo del sistema creado además del equipo de investigación del proyecto.

	1	2	3	4	5
Creo que me gustaría utilizar esta aplicación con frecuencia					
Encontré la aplicación innecesariamente complejo					
Pensé que la aplicación era fácil de usar					
Creo que necesitaría el apoyo de un técnico para poder utilizar esta aplicación					
Encontré que las diversas funciones de esta aplicación estaban bien integradas					
Pensé que había demasiada inconsistencia en esta aplicación					
Me imagino que la mayoría de la gente aprendería a utilizar esta aplicación muy rápidamente					
Encontré la aplicación muy complicada de usar					
Me sentí muy seguro usando la aplicación					
Necesitaba aprender muchas cosas antes de empezar con esta aplicación					

Cuadro 1: Cuestionario SUS

Con los datos recolectados usando el formulario SUS y usando el método de puntuación que lo acompaña [23], se obtuvo la satisfacción de los usuarios que probaron la aplicación. La cantidad de encuestados, con un $n = 4$, entregó un puntaje de 84.5 sobre 100. Así, en base a la escala del método de puntuacion utilizado, esta cifra categoriza a la aplicación como "buena" en la escala.

Ir al Anexo D para ver las tareas creadas para los usuarios para familiarizarlos con la aplicación.

8. Trabajo futuro

Con respecto al desarrollo de la memoria, se encontraron múltiples complicaciones y desafíos a solucionar que requirieron agregar deuda técnica a la aplicación que deberá ser pagada. La recolocación y el orden general de los archivos que forman la aplicación movil deberán ser refactorizadas para encontrar más fácilmente los archivos deseados dentro de ella.

Queda pendiente la implementación de la funcionalidad que permita la visualización de ingestas histórica que se han llevado a cabo. Sin embargo, parte de este trabajo lo desarrolla este informe. En efecto, el informe introduce, explica e implementa los requerimientos de esta función, cuando hablamos sobre la base de datos en la subsección 4.4.

Ciertos problemas que dependen de errores o condiciones externas a la aplicación aún deben ser manejados apropiadamente. Ya hablamos de cómo la aplicación es capaz de manejar el hecho de que no haya una conexión a internet. Pero si la razón por la cual la aplicación no se está comunicando con el backend es porque el backend está caído, la aplicación no tiene un sistema para manejar este escenario. Este asunto es uno de los primeros que deben ser remediados.

Si bien se espera un aumento en el porcentaje de adherencia a los tratamientos en un 7 %, esta es solo una estimación obtenida de otras aplicaciones similares a la aplicación a que se refiere la presente memoria de título. El aumento de la adherencia por parte de los pacientes con sus tratamientos médicos aún debe ser calculada para esta aplicación en particular.

Cuando vimos la base de datos local implementada se habló de la tabla “Registro Ingesta Cola”. El objetivo final de ésta es optimizar la búsqueda de ingestas no enviadas llevadas a cabo en momentos donde no se tenía conexión a internet. Esta optimización puede ser reemplazada haciendo uso de una tabla auxiliar que identifique las ingestas que aún no han sido enviadas al servidor en la tabla “Registro Ingesta”.

9. Conclusiones

A lo largo de esta memoria de título, se ha logrado cumplir el objetivo inicial de entregar una nueva versión de la AFAM 1.0 para su uso en la captura y transmisión de datos de ingestas para contribuir con el aumento de la adherencia a tratamientos farmacológicos.

El problema que supone informar a pacientes sobre los medicamentos que deben ingerir aún no se ha solucionado. Si bien el sistema desarrollado apoyará en un avance del problema, muchos factores pueden llevar a su mal funcionamiento. Factores respecto de los cuales no tenemos control, como la falta de alfabetización digital de los usuarios de esta aplicación. Lo que puede mermar su capacidad para informar correctamente sobre los medicamentos que ingirió o que no ingirió.

La creación y evaluación de una interfaz gráfica agradable y fácil de usar para adultos mayores supuso uno de los mayores retos hasta los últimos días de desarrollo. Múltiples iteraciones de la interfaz fueron creados para terminar con un resultado que satisfaciera tanto las necesidades de lograr un aspecto agradable y acogedor como los de usabilidad que involucran el no caer dentro de las convenciones de interfaz, a los cuales los nativos digitales están acostumbrados y, en cambio gran parte de los futuros usuarios de esta aplicación no, sin exigirle a las vistas demasiado para no provocar un mal funcionamiento de ellas en los variados dispositivos donde será usada.

La retroalimentación entregada por parte de los integrantes del proyecto que probaron la aplicación ha dejado en claro que esta ha cumplico expectativas, validando así el trabajo y esfuerzo puestos en ella, entregando un futuro alentador para su desarrollo.

El desarrollo de la presente memoria de título dejó en claro los desafíos para comunicarse con un servidor estándar FHIR. La estructuración de los datos entregados por la API, necesitó manipulación exhaustiva previa para permitir su uso en la aplicación y las mismas complicaciones aparecieron a la hora de enviar los datos al servidor.

Mantener el orden y llevar a cabo los procesos de obtención y manipulación de los datos demostró ser pieza clave para asegurar un desarrollo que permitiera ciertas libertades y obtuviera resiliencia cuando cambios repentinos eran requeridos.

Referencias

[1] Afam Movil App Youtube Channel. https://www.youtube.com/@AfamMovilApp.

[2] Andreas Widijargo. An Introduction to Flutter Clean Architecture. https://medium.com/ruangguru/an-introduction-to-flutter-clean-architecture-ae00154001b0.

[3] Connectivity Plus. Flutter plugin for discovering the state of the network. https://pub.dev/packages/connectivity_plus.

[4] Dio. A powerful HTTP networking package for Dart and Flutter. https://pub.dev/packages/dio.

[5] FHIR. Fast Healthcare Interoperability Resources. https://fhir.org/.

[6] Flutter. Build for any screen. https://flutter.dev/.

[7] Flutter Local Notifications. A cross platform plugin for displaying and scheduling local notifications for Flutter. https://pub.dev/packages/flutter_local_notifications.

[8] Flutter Material Pickers. A flutter package for easily and consistently showing material themed picker dialogs. https://pub.dev/packages/flutter_material_pickers.

[9] flutter.dev. pub.dev. https://pub.dev/.

[10] FlutterFlow. Flutterflow. https://flutterflow.io/.

[11] Github. code hosting platform for collaboration and version control. https://github.com/.

[12] GoRouter. A declarative router for Flutter. https://pub.dev/packages/go_router.

[13] Komodo Health. 10 Usability Heuristics for User Interface Design. https://www.nngroup.com/articles/ten-usability-heuristics/.

[14] Komodo Health. Medisafe impact on medication persistence. https://shorturl.at/wBGP0.

[15] Medisafe. Medication management app. https://www.medisafe.com/.

[16] Real Academia Nacional de Medicina de España. Diccionario de términos médicos. https://dtme.ranm.es/index.aspx.

[17] Riverpod. A Reactive Caching and Data-binding Framework. https://riverpod.dev/.

[18] Salud, de IOS. Conócete a fondo con Salud. https://www.apple.com/es/ios/health/.

[19] Shared Preferences. Flutter plugin for reading and writing simple key-value pairs. https://pub.dev/packages/shared_preferences.

[20] Socket Io Cliente. Dartlang port of socket.io-client for flutter to use. https://pub.dev/packages/socket_io_client.

[21] SQFlite. Flutter plugin for SQLite. https://pub.dev/packages/sqflite.

[22] SQLite. C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. https://www.sqlite.org/index.html.

[23] SUS. La escala de usabilidad del sistema y cómo se usa en UX. https://ichi.pro/es/la-escala-de-usabilidad-del-sistema-y-como-se-usa-en-ux-202460535877815#:~:text=C%C3%B3mo%20calcular%20una%20puntuaci%C3%B3n%20SUS%20Al%20observar%20las,luego%20reste%205%20del%20total%20para%20obtener%20%28X%29.

[24] Youtube Player Flutter. Flutter plugin for playing or streaming inline YouTube videos using the official iFrame player API. https://pub.dev/packages/youtube_player_flutter.

[25] YoutubeDataApiV3. The YouTubeDataApiV3 is an API that provides access to YouTube data, such as videos, playlists, and channels. https://developers.google.com/youtube/v3?hl=es-419.

Anexo A: Vistas de “Medicamentos”

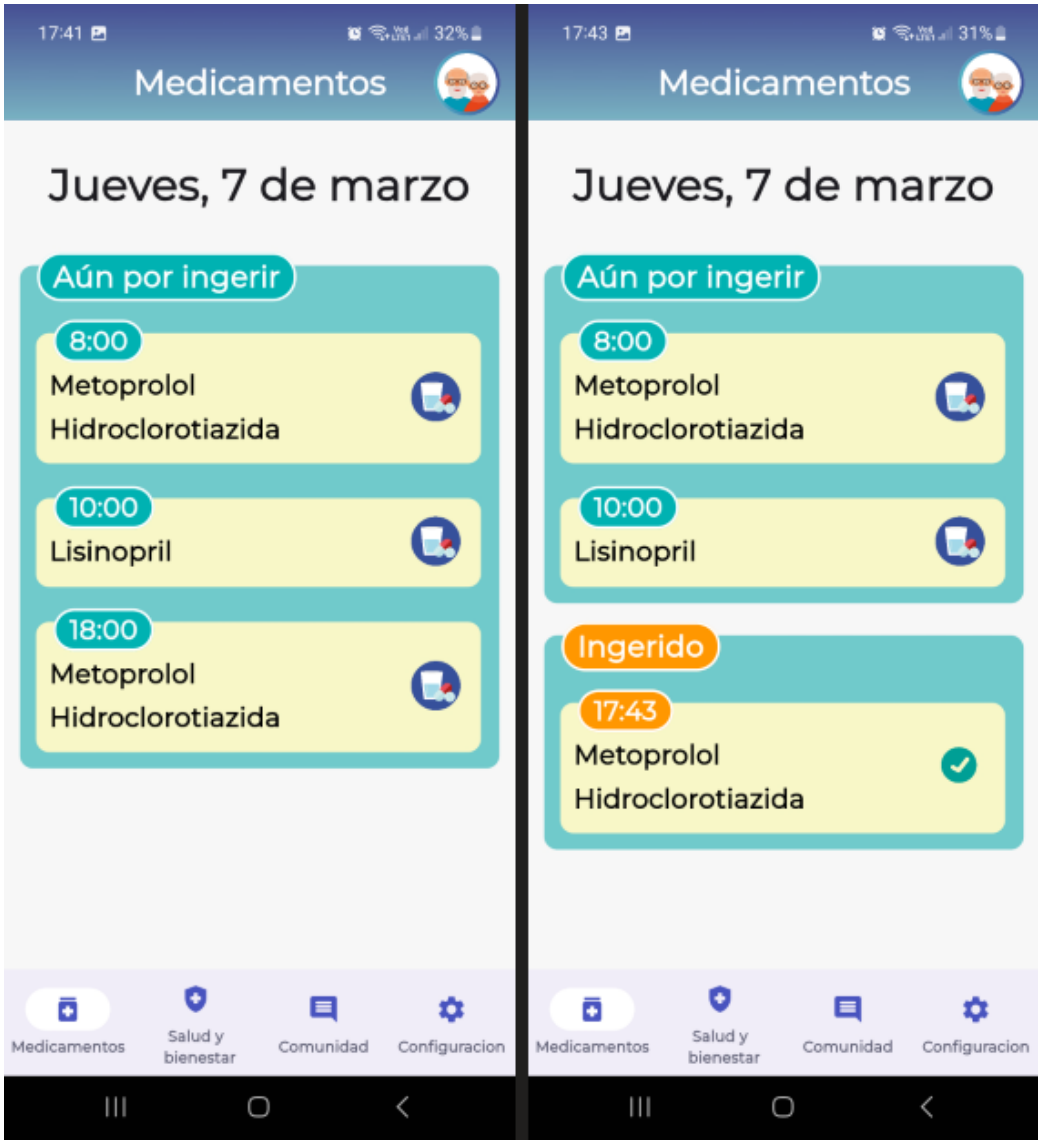


Figura 13: Vistas alternativas de la pantalla principal de “medicamentos”

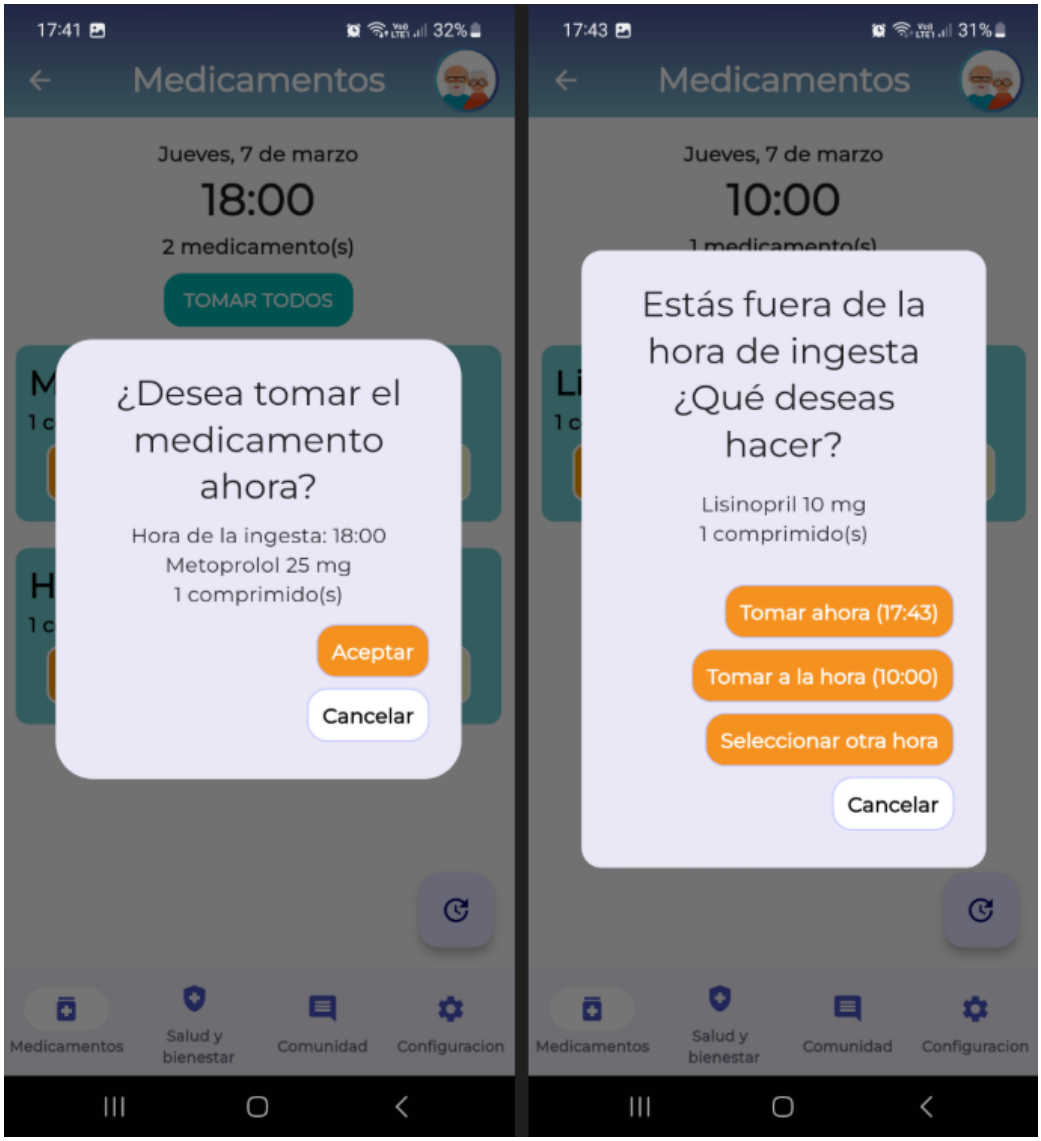


Figura 14: Vista de pop-ups para confirmar toma de medicamentos

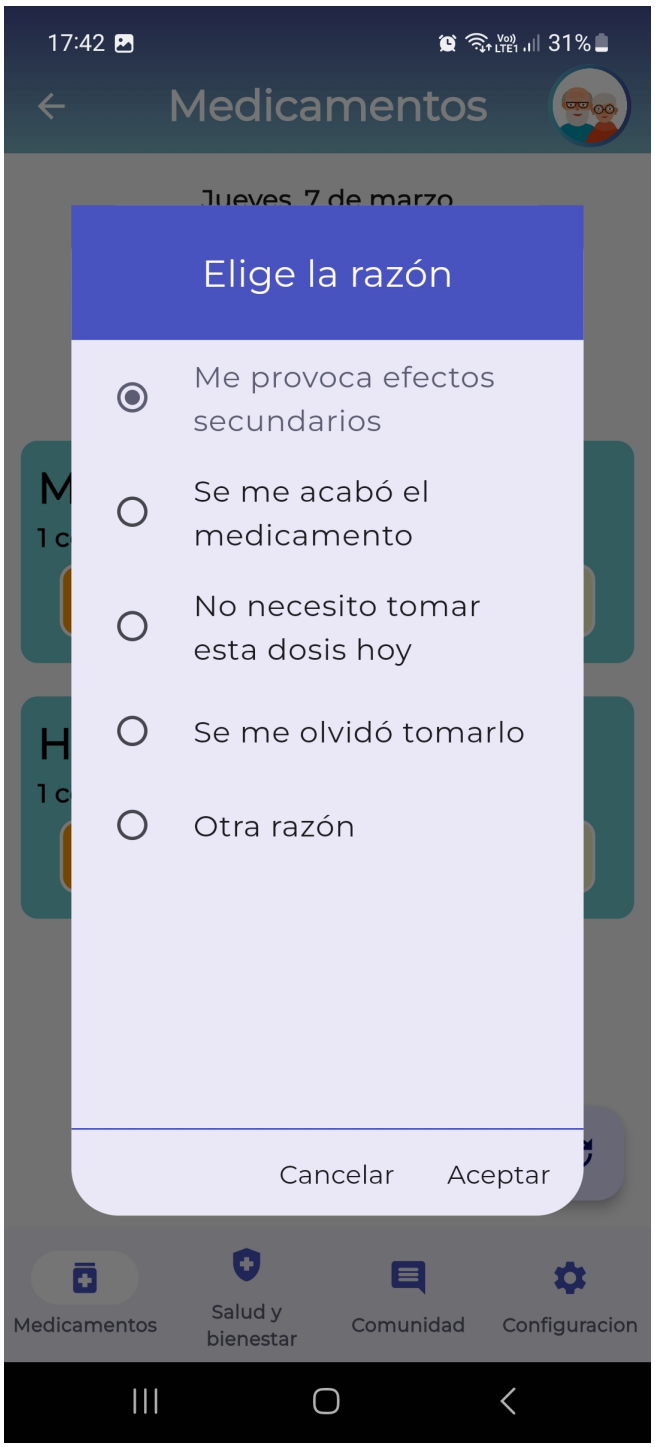


Figura 15: Vista de pop-up para seleccionar la razón de no ingesta

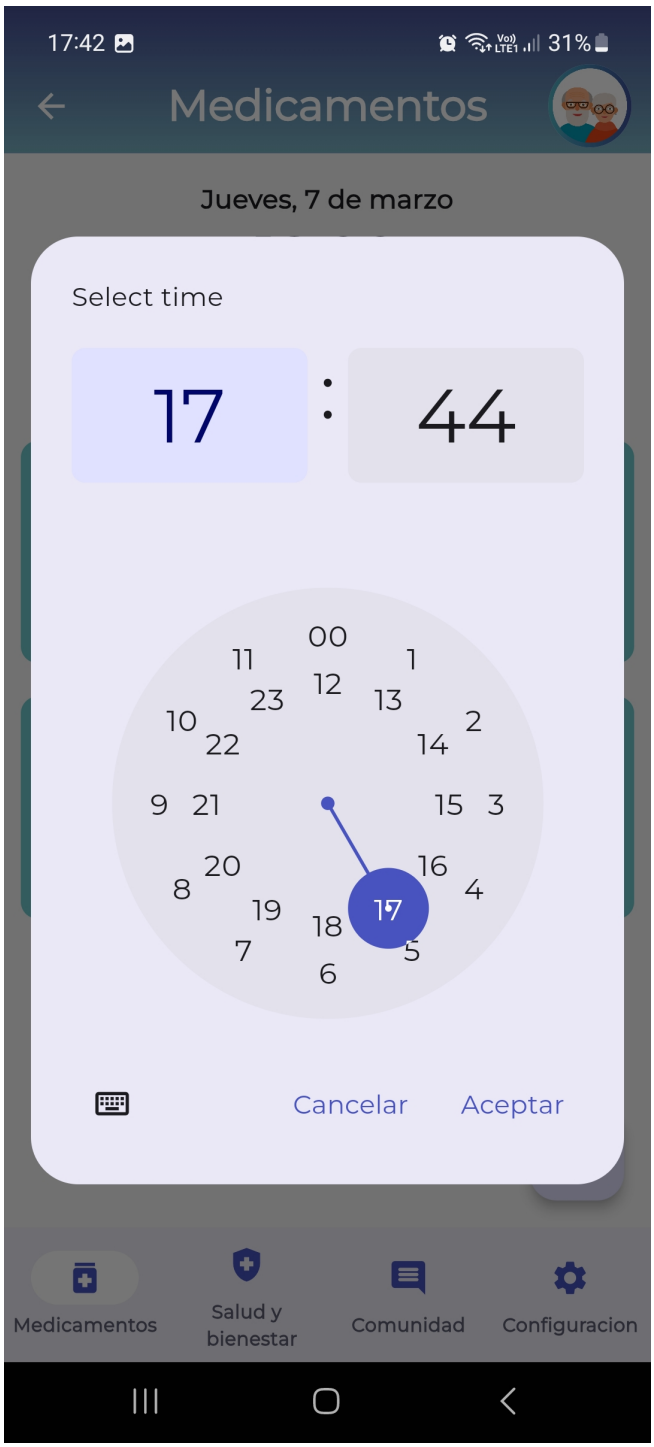


Figura 16: Vista de pop-up para posponer una ingesta indicando una hora

Anexo B: Testimonio de la llegada de datos desde la aplicación móvil al backend

Para comprobar que los medicamentos fueron informados correctamente desde la aplicación móvil al backend, se utilizan capturas de pantalla de la aplicación Postman la cual permite consultar y generar peticiones a APIs para revisar el estado de datos particulares que estas contengan.

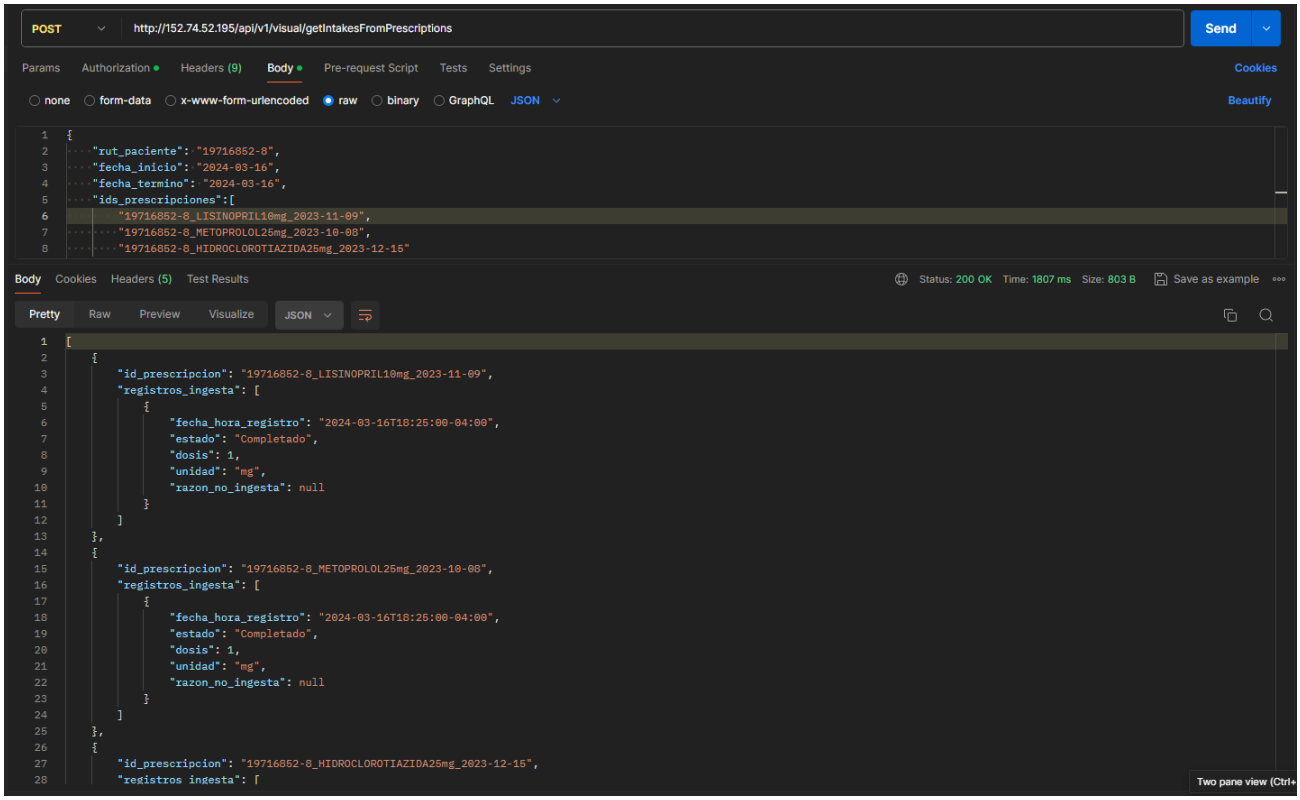


Figura 17: Ingestas realizadas del 16/03/2024 que fueron informadas al backend desde la aplicación móvil

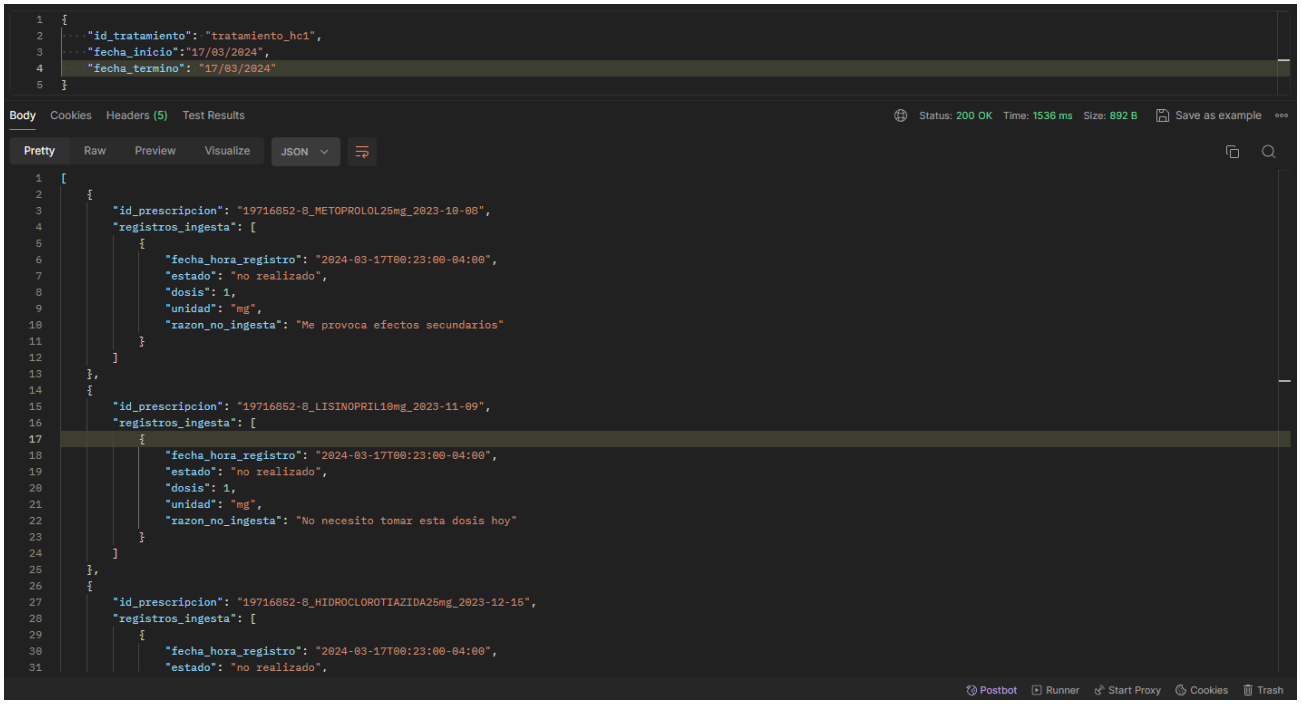


Figura 18: Ingestas no realizadas del 17/03/2024 que fueron informadas al backend desde la aplicación móvil

Anexo C: Vistas de “Salud y bienestar”

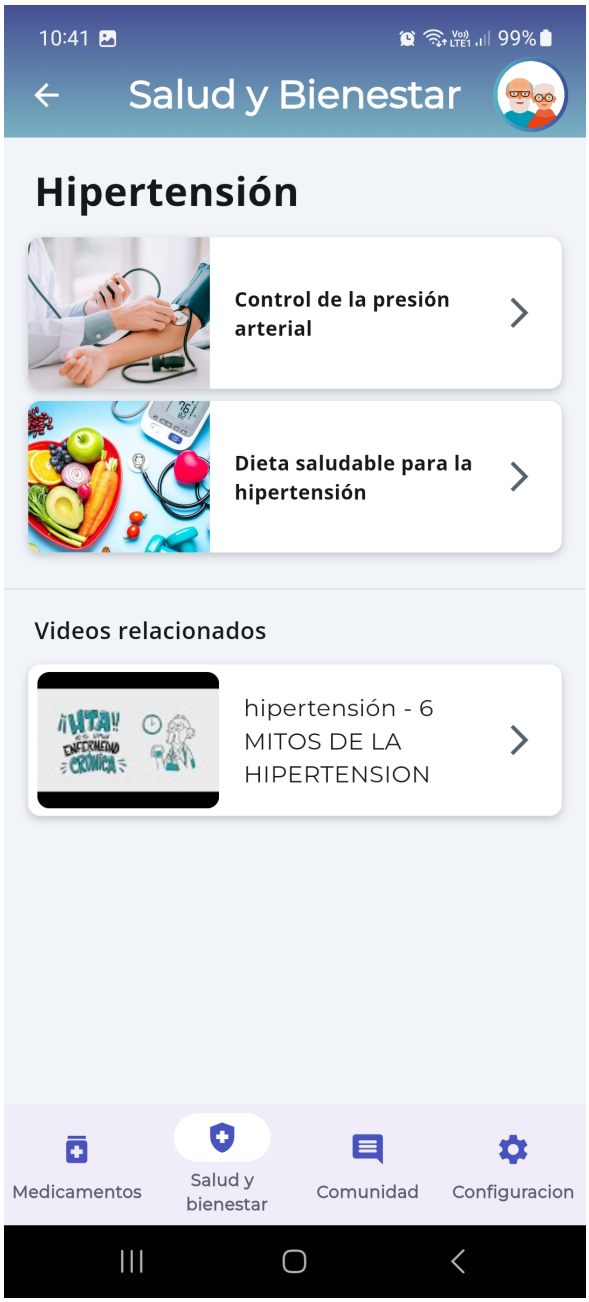


Figura 19: Vista de los elementos presentados en el tema “Salud y bienestar”

Figura 20: Vista del artículo “Control de la presión arterial”

Figura 21: Vista del video “6 Mitos de la hipertensión“

Anexo D: Diagrama de casos de uso

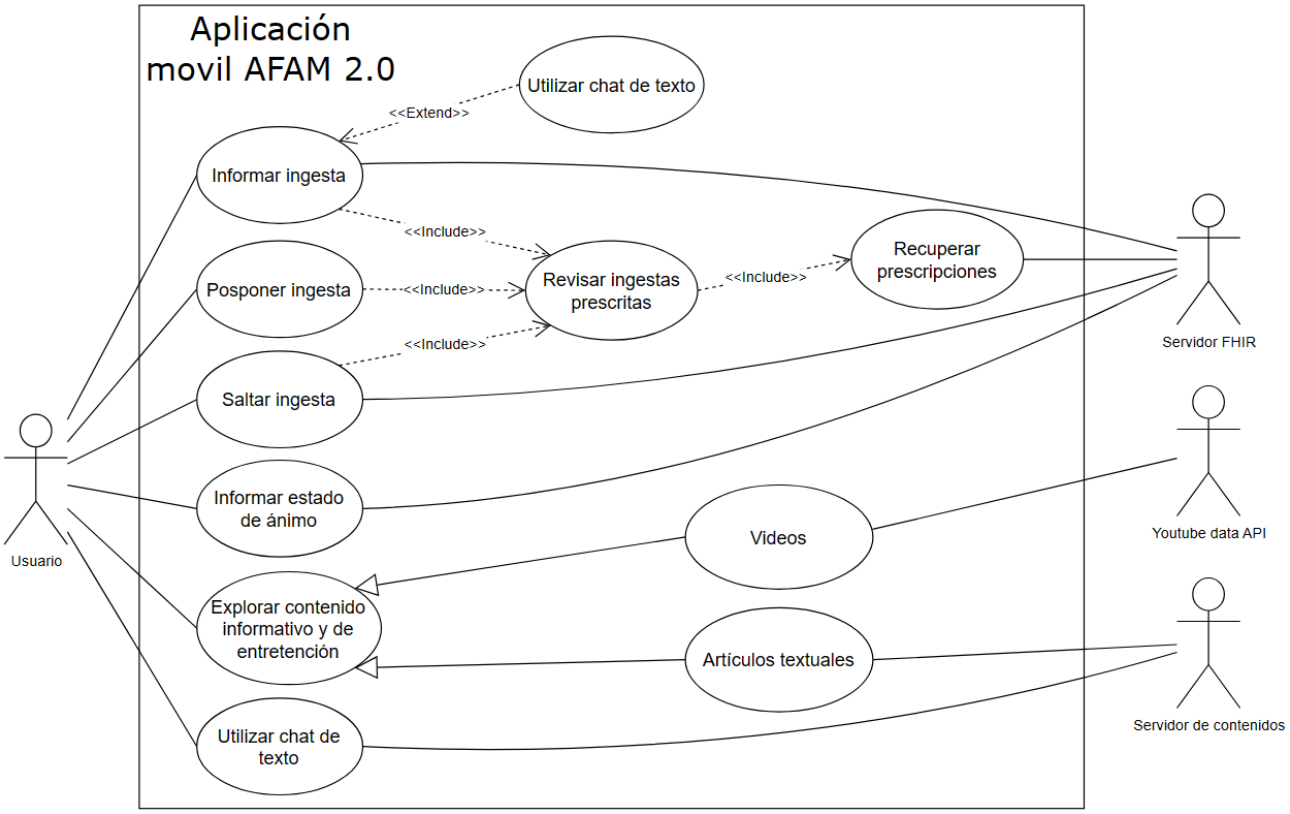


Figura 22: Diagrama de casos de uso de la aplicación móvil

Anexo E: Tareas cuestionario SUS

Tareas

Para instalar la aplicación en su dispositivo móvil primero debe descargar la APK desde este link.

Tarea 1: Login e identificación de los medicamentos a ingerir

1. Se solicita que le entregue permiso a la aplicación para enviarle notificaciones (leer la subsección "Sobre las notificaciones" de la sección "consideraciones previas").
2. Inicie sesión usando el usuario "10123456-7" y la clave "ingesta123".
3. Apenas inicie sesión se le preguntará sobre su estado de ánimo. Debe seleccionar un estado (puede elegir uno al azar).
4. Identifique los medicamentos que debe ingerir a lo largo del día, los cuales son todos los presentados en pantalla, y la hora a la cual estos deben ser ingeridos.
5. identifique el acceso a la vista de medicamentos de una hora particular.
6. En la vista de medicamentos de una hora particular, sean cuales sean los medicamentos agrupados que seleccionó, identifique los botones para "tomar todos" los medicamentos, "tomar" un medicamento particular, "saltar" un medicamento particular y "posponer" un medicamento particular.
7. Vuelva a la vista principal de los medicamentos a ingerir.

Tarea 2: Lleve a cabo las acciones de "tomar", "tomar todos", "posponer" y "saltar"

1. Informe a la aplicación que se *tomó todos* los medicamentos de las 23:59 horas.
2. Informe a la aplicación que se *tomó* la dosis de Lisinopril de las 23:00 horas.

3. Informe a la aplicación que decidió *saltar* la ingesta de Metoprolol de las 23:00 horas y seleccione una razón de porque lo hizo (puede elegir una al azar).

4. Informe a la aplicación que decidió *posponer* la ingesta de Hidroclorotiazida de las 23:00 horas y seleccione una hora (nota: la aplicación no permite elegir una hora anterior a la hora actual).

Tarea 3: Lleve a cabo las acciones de "tomar" y "tomar todos" para medicamentos los cuales su hora de ingesta pasó hace varias horas

1. Informe a la aplicación que se *tomó todos* los medicamentos de las 2:00 horas a la hora que el tratamiento le solicitó tomarlos (las 2:00 horas para este caso).

2. Informe a la aplicación que se *tomó* la dosis de Lisinopril de las 3:00 horas a la hora actual.

3. Informe a la aplicación que se *tomó* la dosis de Metoprolol de las 3:00 horas a una hora distinta a la hora actual y distinta a la hora que el tratamiento le solicitó tomarla.

(Opcional) Tarea 4: siga usando la aplicación

1. Presione "tomar", "tomar todos", "posponer" o "saltar" en cualquier vista de medicamentos de una hora particular que queden por ingerir.

2. Revise la vista de salud y bienestar y lea un artículo o vea un vídeo dentro de los temas a elegir de las selecciones "Consejos" o "Vida sana".

3. Revise la vista de chat comunitario y envíe un mensaje a la comunidad.

4. Revise la vista de ajustes y cambie el avatar que aparece arriba a la derecha.

Formulario

Una vez haya realizado las tareas, por favor conteste este corto formulario. Agradezco enormemente el tiempo que brindó para llevar a cabo estas tareas. Siéntase con libertad para seguir usando la aplicación y enviar cualquier sugerencia o comentario que tenga. Muchas gracias por su colaboración.