



Medical Reminder

Andrés Signes Grimalt

Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles

Eduard Martin Lineros

05/2022



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada a [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	Medical Reminder: Tu asistente medico
Nombre del autor:	Andrés Signes Grimalt
Nombre del consultor:	Eduard Martin Lineros
Fecha de entrega (mm/aaaa):	05/2022
Titulación:	<i>Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles</i>
Resumen del Trabajo (máximo 250 palabras):	
<p>Las pautas a la hora de tomar medicamentos siempre es difícil de recordar. ¿A qué hora he tomado la última dosis? ¿Me he tomado la pastilla esta mañana?</p> <p>Son preguntas que me he hecho cuando estaba a mitad de algún tratamiento, Medical Reminder viene a solucionar este problema.</p> <p>Con esta app podremos añadir nuestra medicación y programar los periodos de ingesta de nuestro medicamento, además podremos consultar toda la información sobre nuestra medicación (prospecto, fotos, dosis ...).</p> <p>La app constará de un sistema de autenticación, de esta forma podremos guardar toda la app del usuario en la nube para ser accesible desde cualquier dispositivo sin tener que guardar la información en el propio dispositivo.</p> <p>Una vez creado el perfil podremos añadir un tratamiento, indicando inicio y fin de la receta médica y definir a que horas tomaremos la medicación a lo largo del día.</p> <p>Este proyecto se desarrollará con el framework Flutter utilizando el lenguaje Dart que nos permitirá obtener una app para las plataformas dominantes en el ámbito móvil (Android e iOS), el framework también permite generar una web, aunque en principio esta no será una prioridad en el desarrollo.</p>	
Abstract (in English, 250 words or less):	
<p>The guidelines when taking medication are always difficult to remember. What time did I take the last dose? Did I take the pill this morning?</p> <p>These are questions that I have asked myself when I was in the middle of a treatment, Medical Reminder comes to solve this problem.</p> <p>With this app we can add our medication and schedule the periods of intake of our medication, we can also consult all the information about our medication (leaflet, photos, doses ...).</p> <p>The app will consist of an authentication system, so we can save all the user's data in the cloud to be accessible from any device without having to save the information on the device.</p>	

Once the profile is created, we will be able to add a treatment, indicating the start and end of the medical prescription and define at what time we will take the medication throughout the day.

This project will be developed with the Flutter framework using the Dart language that will allow us to obtain an app for the dominant platforms in the mobile field (Android and iOS), the framework also allows us to generate a web, although in principle this will not be a priority in the development.

Palabras clave (entre 4 y 8):

flutter, android, ios, medicacion, recordatorio, salud

Índice

1. Introducción	8
1.1 Contexto y justificación del Trabajo	8
1.1.1 Contexto	8
1.1.2 Justificación del Trabajo	8
1.2 Objetivos del Trabajo	9
1.2.1 Objetivos Principales	9
1.2.2 Objetivos Funcionales	9
1.2.3 Objetivos No Funcionales	9
1.3 Enfoque y método seguido	10
1.4 Planificación del Trabajo	10
1.4.1 Recursos Disponibles	10
1.4.2 Planificación	11
Plan de trabajo	12
Diseño	12
Implementación	12
Entrega final	13
1.5 Breve resumen de productos obtenidos	13
1.6 Breve descripción de los otros capítulos de la memoria	13
2. Diseño	14
2.1 Diseño centrado en el usuario	14
2.1.1 Usuarios y contexto	14
2.1.1.1 Análisis Competitivo	14
2.1.1.2 Entrevistas en Profundidad	15
2.1.1.3 Perfiles Usuarios	15
2.1.2 Diseño conceptual	16
2.1.2.1. Usuarios	16
2.1.2.2 Flujos de interacción	17
2.1.3 Prototipo	18
Inicio de sesión	18
Crear cuenta	18
Recuperar contraseña	19
Perfil de usuario	19
Agenda	20
Buscar medicamento	20
Detalle tratamiento	21
Detalle medicamento	21
2.1.4 Evaluación	22
2.2.1 Definición de casos de uso	22
Crear cuenta	22
Recuperar cuenta	22

Inicio de sesión	23
Cerrar sesión	23
Crear tratamiento	24
Marcar toma del tratamiento como completada	24
2.2.2 Diseño de arquitectura de la aplicación	25
2.2.2.1 Arquitectura	25
2.2.2.2 Base de datos	26

Lista de figuras

Metodología en Cascada	9
Diagrama de Gantt, plan de trabajo	11
Diagrama de Gantt, Diseño	11
Diagrama de Gantt, Implementación	11
Diagrama de Gantt, Entrega Final	12
Flujo de interacción	17
Prototipos	18-22
Casos de uso	23
Diseño de Arquitectura	27

1. Introducción

1.1 Contexto y justificación del Trabajo

1.1.1 Contexto

Desafortunadamente cada vez somos una sociedad más medicada y más ajetreada. No obstante, todos llevamos dispositivos conectados que nos proporcionan información de nuestro entorno y nos ayudan en nuestro día a día.

Actualmente, para recordar la toma de un medicamento se puede solucionar de los siguientes modos:

- creando recordatorios o alarmas a mano en el teléfono
- Añadiendo eventos a nuestro calendario
- Existen muchas apps en el mercado que nos permiten programar la toma de medicamentos, no obstante no he visto que ninguna se integre con una Base de datos de medicamentos aportando más información.
 - Medisafe Pill & Med Reminder
 - MyTherapy
 - Medicine time!
- También existen pastilleros donde podemos colocar las pastillas para saber que debemos tomar en cada momento



1.1.2 Justificación del Trabajo

La idea de esta app viene a solucionar este problema.

Añadir un tratamiento a nuestra app, con cada medicamento que debemos tomar para crear recordatorios de toma de nuestra medicación y anotando cada toma para completar la pauta todos los días. Además, también añadimos el prospecto del medicamento en la app y más información útil para el usuario. El objetivo es que la aplicación informe mediante una notificación a la hora de tomar nuestra medicación.

Permitirnos buscar un medicamento y programar las tomas de estas, pudiendo añadir un evento al calendario o programar una notificación al móvil.

1.2 Objetivos del Trabajo

El objetivo de este trabajo es desarrollar una app para las plataformas Android e iOS que nos permita buscar un medicamento y definir un periodo y unos horarios de toma de este medicamento.

1.2.1 Objetivos Principales

- Aplicar los conocimientos obtenidos durante el máster.
- Producir una aplicación completa desde su definición y diseño hasta su implementación.
- Aplicar patrones de desarrollo para que la aplicación pueda ser escalable y entendible por otros desarrolladores que se puedan unir al proyecto.
- Permitir a los usuarios almacenar sus tratamientos y notificarles correctamente el momento en el que deben tomarse su dosis.
- Dar toda la información posible a los usuarios sobre un medicamento.

1.2.2 Objetivos Funcionales

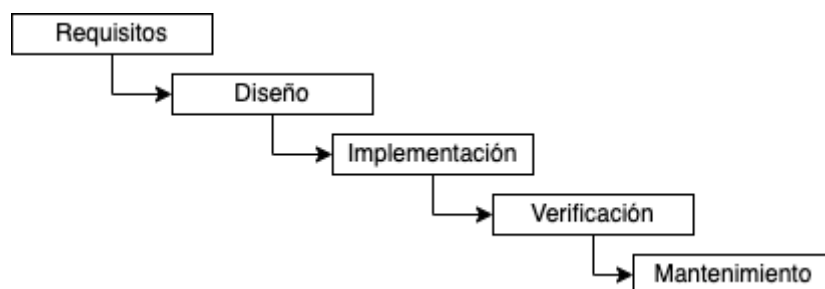
- Implementar un sistema de autenticación para el usuario utilizando Firebase.
- Integrar la API de AEMPS para poder obtener los medicamentos y su información adicional.
- Integrarse con la BD de Firebase para almacenar los tratamientos de los usuarios.
- Permitir buscar medicamentos mediante el nombre o escaneando el código de barras de la caja.
- Añadir un tratamiento indicando un inicio y un fin (opcional) y hora y minuto para cada toma durante el día.
- Marcar medicamentos como favoritos y guardarlos en una lista específica.
- Enviar notificaciones push al usuario para recordar la toma del medicamento.
- Desarrollar la aplicación para que pueda ser multiidioma (aunque inicialmente solo este disponible en español)

1.2.3 Objetivos No Funcionales

- Crear una interfaz fácil de usar y fácilmente navegable
- Utilizar un estilo de colores y fuentes coherente por toda la aplicación
- Hacer un uso responsable de la conexión a internet del dispositivo sin abusar de la conexión.
- Hacer la aplicación responsiva para que pueda ser exportada para web o para que se vea de forma correcta en tablets en modo horizontal.
- Informar al usuario de posibles errores ajenos a la aplicación.
- Realizar pruebas unitarias para verificar el correcto funcionamiento.
- Garantizar la seguridad de los datos del usuario y su privacidad.

1.3 Enfoque y método seguido

Normalmente hubiera elegido la metodología ágil para hacer el proyecto, ya que nos permite iterar sobre el mismo y ver como evoluciona teniendo siempre un producto funcional. No obstante, debido a la naturaleza del mismo y los plazos de inicio y fin del proyecto creo que es mejor seguir la metodología en cascada. Cada etapa corresponde a una PEC (a excepción del mantenimiento) y podemos iterar sobre estas PEC, sabiendo en cada momento en que fase del proyecto estamos.



El framework a utilizar será Flutter, ya que nos permite obtener dos compilaciones con solo un código base.

Nuestro backend estará alojado en Firebase, por el hecho de que es gratuito para nuestro objetivo y además tiene una muy buena integración con Flutter.

Todo el código estará alojado en GitHub de manera pública, allí también gestionaremos las posibles incidencias que detectemos y el seguimiento de las mismas.

1.4 Planificación del Trabajo

1.4.1 Recursos Disponibles

- Recursos Humanos
 - ◆ Andrés Signes Grimalt: Project manager, diseñador, desarrollador, tester, etc.
 - ◆ Eduard Martin Lineros: Revisor Proyecto

- Recursos Físicos:
 - ◆ MacBook pro como herramienta de trabajo diaria
 - ◆ Teléfono Nexus 5x para realizar pruebas
- Recursos de Software
 - ◆ El IDE utilizado para el desarrollo será IntelliJ, por mi experiencia utilizándolo durante años.
 - ◆ Draw.io para el diseño de diagramas
 - ◆ Figma para el prototipado de interfaces
 - ◆ Google Docs para la elaboración de la memoria y gestor documental.
 - ◆ GitHub como repositorio de código
 - ◆ Gantt Project como editor de diagrama de Gantt
 - ◆ Firebase como gestor de usuarios y base de datos online
 - ◆ [CIMA](#) como base de conocimiento para obtener información de medicamentos.

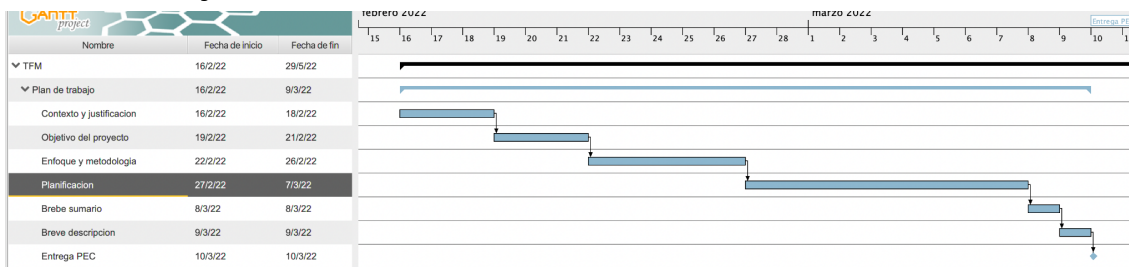
1.4.2 Planificación

Contando con que ya trabajo una jornada completa y mis obligaciones familiares, la dedicación diaria aproximada es de 2 horas diarias en días laborables y 4 horas los días no laborables.

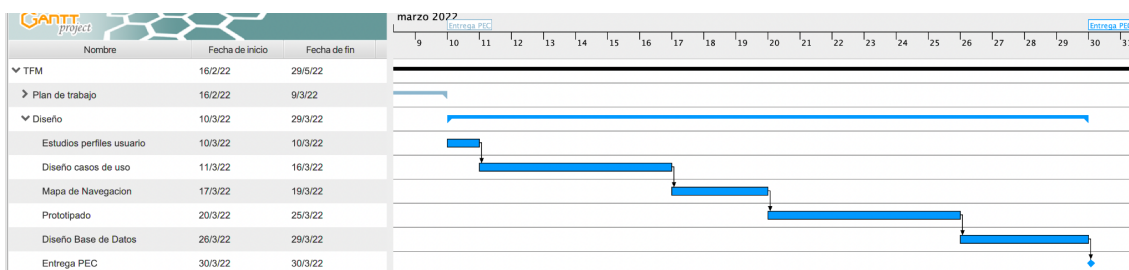
PEC	INICIO	FIN	Horas
PAC 1 (PLAN DE TRABAJO)	16/02/2022	09/03/2022	54 Horas
Contexto y justificación	16/02/2022	18/02/2022	6
Objetivo del proyecto	19/02/2022	21/02/2022	10
Enfoque y método seguido	22/02/2022	26/02/2022	12
Planificación	27/02/2022	06/03/2022	22
Breve sumario	07/03/2022	07/03/2022	2
Breve descripción	08/03/2022	08/03/2022	2
PAC 2 (DISEÑO)	10/03/2022	30/03/2022	54 horas
Estudio perfiles de usuario	10/03/2022	10/03/2022	2
Diseño casos de uso	11/03/2022	16/03/2022	16
Mapa navegación	17/03/2022	19/03/2022	10
Prototipado	20/03/2022	25/03/2022	14

Diseño base de datos	26/03/2022	29/03/2022	12
PAC 3 (IMPLEMENTACIÓN)	31/03/2022	11/05/2022	100 Horas
Autenticación	31/03/2022	08/04/2022	22
API Externa	09/04/2022	16/04/2022	22
Acceso a Base de Datos	17/04/2022	21/04/2022	12
Interfaces	22/04/2022	03/05/2022	26
Pruebas	4/05/2022	10/05/2022	18
ENTREGA FINAL	12/05/2022	30/05/2022	50 Horas
Corrección de Errores	12/05/2022	15/05/2022	14
Documentación	16/05/2022	20/05/2022	10
Video Presentación	21/05/2022	25/05/2022	14
Memoria PEC	26/05/2022	29/05/2022	14

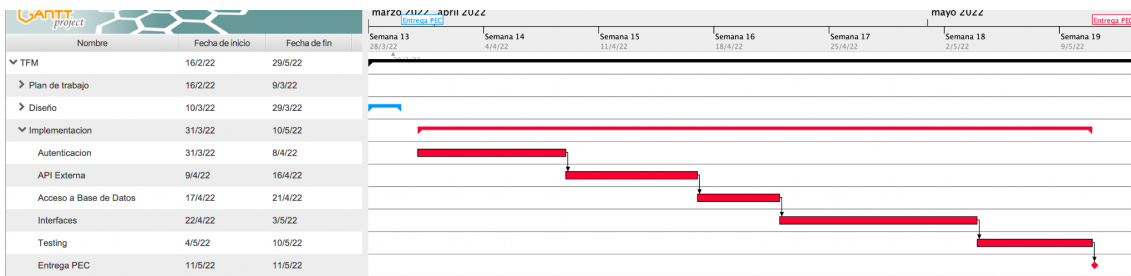
Plan de trabajo



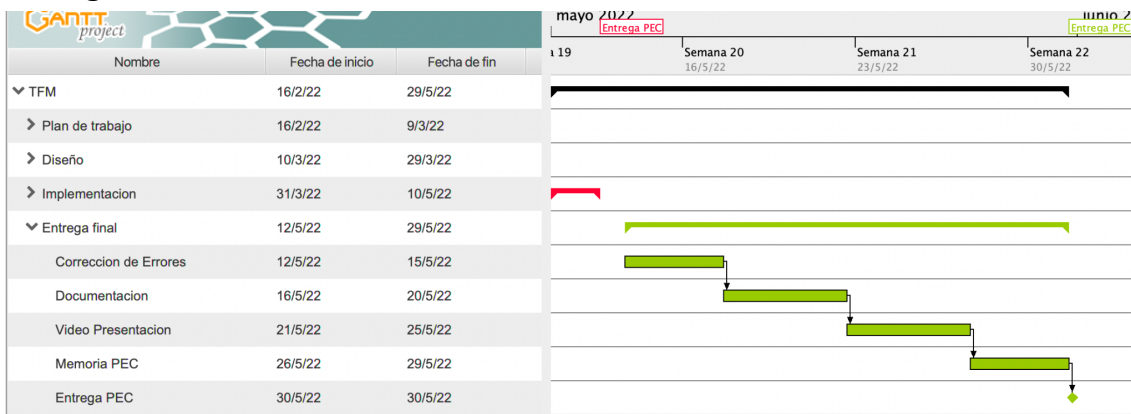
Diseño



Implementación



Entrega final



1.5 Breve resumen de productos obtenidos

- El Objetivo del proyecto es obtener los siguientes productos:
- Compilaciones de la aplicación para la plataforma iOS
 - Compilación de la aplicación para la plataforma Android
 - Memoria final de todo el proyecto
 - Vídeo descriptivo del funcionamiento de la app
 - Código fuente en un repositorio público alojado en GitHub

1.6 Breve descripción de los otros capítulos de la memoria

En el siguiente capítulo se tratará el diseño conceptual de la aplicación, definiremos los casos de uso específicos, la navegación para llegar a esos casos de uso y como ubicarlos dentro de la app, también diseñaremos las interfaces mediante prototipos y también diseñaremos nuestro modelo de datos para almacenarlo en una base de datos.

En el capítulo posterior realizaremos el desarrollo de la app, centrándonos el código y aplicando lo establecido en el capítulo anterior.

Empezaremos por la autenticación de usuario y la gestión del perfil de usuario, seguiremos construyendo el acceso a una API externa, luego el acceso a nuestra API para almacenar datos del usuario, más tarde crearemos las interfaces que integren todas estas conexiones y efectuaremos pruebas integrales de la app para verificar su estabilidad.

En el último capítulo, empezaremos corrigiendo errores que podamos haber descubierto en la última fase del capítulo anterior, documentaremos todo nuestro código y elaboraremos un video descriptivo de la app. Finalmente, terminaremos de elaborar la Memoria final del TFM.

2. Diseño

2.1 Diseño centrado en el usuario

El objetivo en esta fase es conocer las características de los usuarios, sus necesidades y objetivos, así como el contexto de uso. En esta fase detectaremos las funcionalidades que deberemos aplicar para satisfacer las necesidades de los usuarios.

2.1.1 Usuarios y contexto

Para detectar las necesidades de los usuarios se utilizan varias técnicas de indagación, en nuestro caso no utilizaremos todas las técnicas debido al tiempo y recursos limitados de los que disponemos.

Nos centraremos en el análisis competitivo y entrevistas en profundidad.

2.1.1.1 Análisis Competitivo

En la primera fase ya analizamos el contexto en el que se encontraba el proyecto y que opciones similares había en el mercado. Si nos centramos en el entorno móvil, para el cual está pensada esta aplicación, encontramos gran variedad de aplicaciones que ofrecen lo mismo o similar que queremos ofrecer a nuestros usuarios.

MyTherapy Pill Reminder y *Medisafe Pill Reminder* son las principales aplicaciones que encontramos para las dos plataformas dominantes (Android e iOS). Existen muchas otras, pero en mi opinión estas son las dos más avanzadas y dominantes en el mercado.

Analizándolas detenidamente podemos extraer las siguientes conclusiones y los puntos en que debemos centrarnos para intentar mejorar tanto su interfaz como funcionalidad.

- Demasiadas funcionalidades. Hay veces que menos es más, si nuestra intención es hacer una aplicación para recordarnos la toma de un

medicamento, creo que debemos centrarnos en esta funcionalidad y no sobrecargar los menús e interfaces con más opciones.

- Localización de nuestra aplicación. Nuestra aplicación está concebida para el mercado español y proporcionará información específica de la AEMPS, con sus prospectos en español.
- Ofrecer una aplicación totalmente gratuita, sin suscripciones, ni secciones de pago.

2.1.1.2 Entrevistas en Profundidad

Preguntas:

- ¿Tomas algún medicamento o suplemento de forma habitual?
- ¿Has olvidado alguna vez tomar tu medicamento?
- ¿Alguna vez has dudado si habías tomado el medicamento en alguna de las dosis previas?
- ¿Sueles revisar el prospecto de los medicamentos antes de tomarlos por primera vez?
- ¿Te preocupa la forma farmacéutica del medicamento que vas a tomar? (pastilla efervescente, jarabe, cápsula ...)
- ¿Has utilizado alguna vez una aplicación o método alternativo para recordar la toma de un medicamento? ¿Cuál?

Resumen respuestas:

- La mayoría de personas entrevistadas han olvidado tomar alguna dosis de su tratamiento en alguna ocasión, aunque habitualmente lo suelen recordar.
- Las personas con una medicación habitual ya tienen interiorizado la toma del medicamento y asumido en su rutina diaria, no obstante, creen que la aplicación hubiese sido muy útil al inicio de su tratamiento para adoptar esa rutina.
- También han visto muy útil consultar el prospecto del medicamento directamente en el teléfono para poder realizar consultas rápidas sobre posibles efectos secundarios o incompatibilidades con otros medicamentos.
- Sobre la forma farmacéutica, casi ninguna persona ha creído útil esta información, solo algunos usuarios para medicamentos infantiles y las complicaciones de administrarlo a los niños.
- Las personas más jóvenes han utilizado alguna vez aplicaciones similares o métodos digitales para recordar las tomas (alarmas en el móvil o eventos en el calendario), las personas de edad más avanzada optan por anotarlo en la nevera o utilizar pastilleros físicos.

2.1.1.3 Perfiles Usuarios

- Usuario esporádico:
 - ◆ Usuario que añade un medicamento para una enfermedad en concreto, con un inicio y fin determinado
- Usuario diario:
 - ◆ Persona que tiene una pauta sin un fin específico y que toma un medicamento todos los días.


→ Usuario intenso:


- ◆ Persona con una o varias enfermedades que le obligan a tomar varios medicamentos durante el día y que además no tiene un final de tratamiento, además tiene periodos en que se le puede modificar las pautas o dosis del tratamiento.


2.1.2 Diseño conceptual

A partir de toda la información recopilada en la primera fase, pasamos a definir las necesidades de los usuarios tipo y sus objetivos a cumplir. Aquí definiremos unos usuarios tipo para ejemplarizar de manera gráfica a quien va dirigida nuestra aplicación.

2.1.2.1. Usuarios

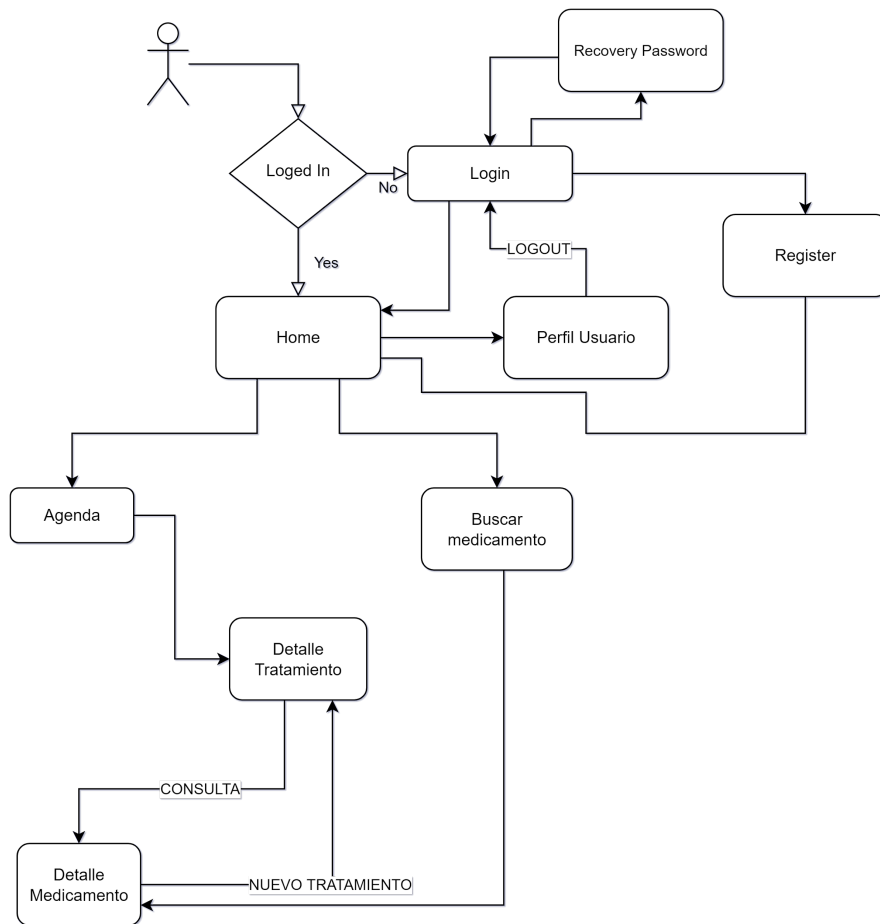
Águeda Rosselló	
	Edad: 33 años
	Situación laboral: Responsable de RRHH
	Situación familiar: pareja estable, embarazada
Debido a su recién embarazo tiene que tomar una medicación recetada por su doctor, es una medicación que tiene que tomar hasta que de a luz, todos los días por la mañana.	

Lorena Pérez	
	Edad: 24 años
	Situación laboral: Estudiante
	Situación Familiar: sin pareja estable
Lorena es diabética y tiene que tomar varios medicamentos a diario, dependiendo de su situación hay periodos en los que tiene que modificar sus dosis o cambiar algún medicamento.	

Vicente Ginestar	
	Edad: 61 años
	Situación laboral: Conductor de autobuses
	Situación Familiar: Separado, 3 hijos
Vicente es un hombre con colesterol que tiene que tomar su medicación todos los días antes de cada comida.	

2.1.2.2 Flujos de interacción

Mediante la aplicación web [Draw.io](https://draw.io) hemos realizado un mapa que define las posibles interacciones y caminos que puede tomar un usuario por las diferentes pantallas de la aplicación. Este apartado nos ha servido para revisar todas las decisiones tomadas anteriormente. De este modo quedan estructuradas y organizadas los posibles flujos de entrada y salida de una pantalla.



2.1.3 Prototipo

En esta fase crearemos unos diseños conceptuales de las pantallas que contendrá nuestra aplicación para después tomarlas de referencia a la hora de implementar la versión final y para también situar toda la lógica de navegación en pantalla. Este prototipado también ha sido realizado mediante la aplicación web Draw.io

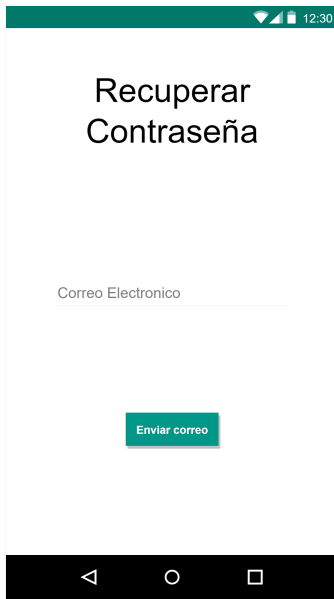
Inicio de sesión

The screenshot shows the login interface for the 'Medical Reminder' app. At the top, there is a green status bar with the time '12:30' and icons for signal strength, Wi-Fi, and battery. The main title 'Medical Reminder' is centered in a large, black font. Below the title are two input fields: 'Correo Electronico' and 'Contraseña'. A prominent green 'Login' button is centered below the fields. Underneath the button are two links: 'Crear Cuenta' and 'Recordar Contraseña', both in a smaller green font. At the bottom of the screen is a black navigation bar with three white icons: a back arrow, a home circle, and a recent apps square.

Crear cuenta

The screenshot shows the account creation interface for the 'Medical Reminder' app. At the top, there is a green status bar with the time '12:30' and icons for signal strength, Wi-Fi, and battery. The main title 'CREAR CUENTA' is centered in a large, black font. Below the title are three input fields: 'Correo Electronico', 'Contraseña', and 'Repetir Contraseña'. A green 'Crear Cuenta' button is centered below the fields. At the bottom of the screen is a black navigation bar with three white icons: a back arrow, a home circle, and a recent apps square.

Recuperar contraseña

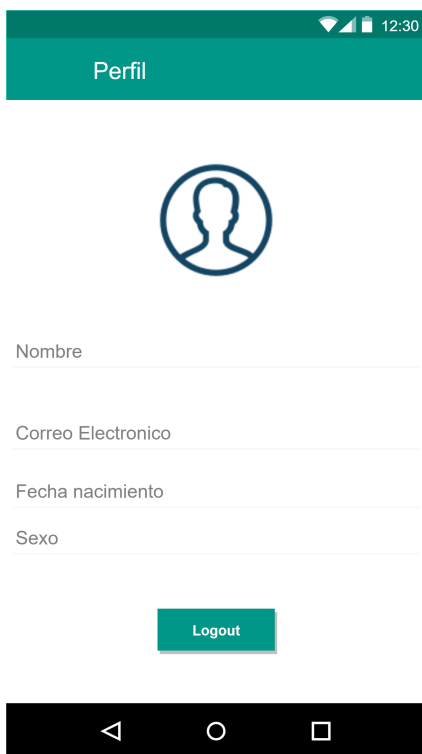


Recuperar
Contraseña


Correo Electronico

Enviar correo

Perfil de usuario



Perfil



Nombre

Correo Electronico

Fecha nacimiento

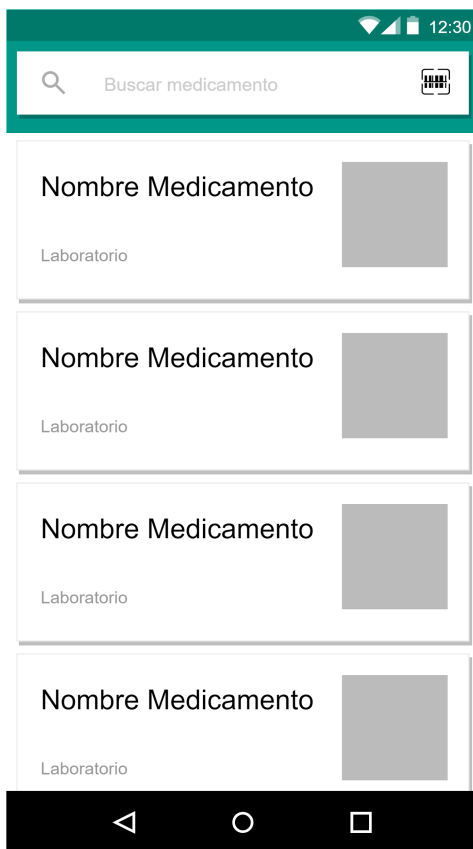
Sexo

Logout

Agenda



Buscar medicamento



Detalle tratamiento

Detalle tratamiento

Medicamento
Laboratorio

Inicio

Fin

Frecuencia

- 4 horas
- 6 horas
- 8 horas

Detalle medicamento

22/04/2022 08:00	22/04/2022 14:00	23/04/2022 08:00

Detalle medicamento

Detalle medicamento



Nombre Medicamento

Laboratorio

Ficha Tecnica

Prospecto



Forma farmaceutica
comprimido

Dosis

100 mg

Medicament sujeto a prescripcion medica

Crear tratamiento

2.1.4 Evaluación

Tras exponer todo lo definido en esta fase, se ha presentado a posibles usuarios de la aplicación y han valorado satisfactoriamente el funcionamiento. Han destacado las siguientes funcionalidades:

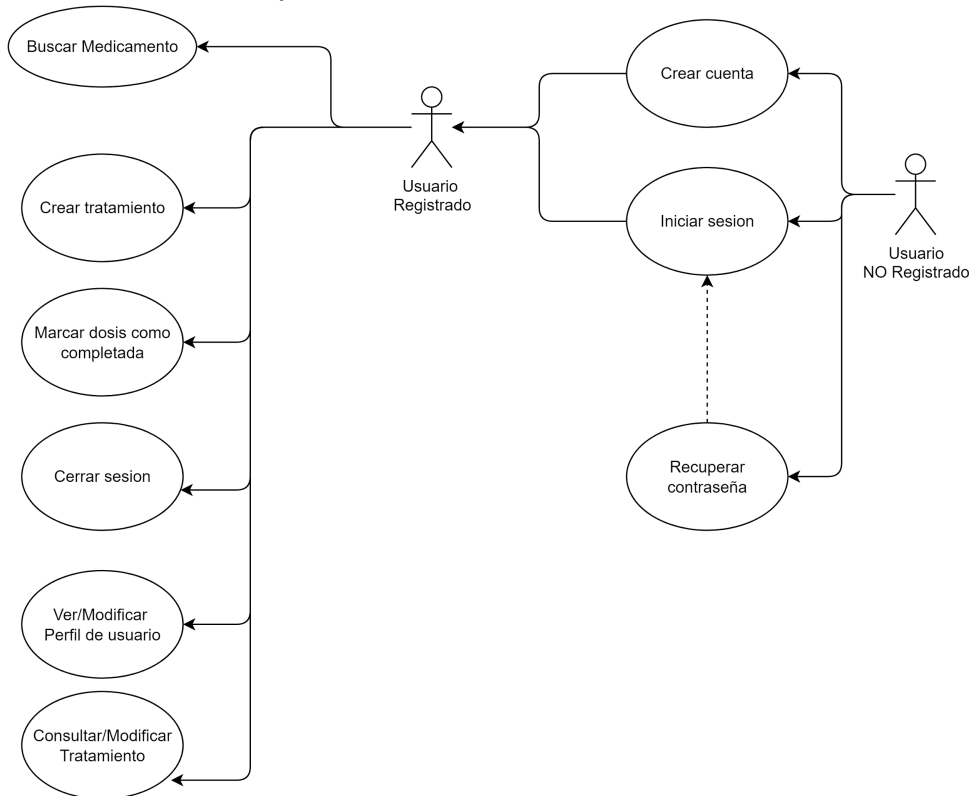
- Recibir notificaciones a la hora de tomar el medicamento
- Poder consultar el prospecto del medicamento directamente en la aplicación.

Como posibles mejoras han indicado lo siguiente:

- Poder marcar ciertos medicamentos como favoritos para utilizarlos de forma recurrente para consultarlos, aunque no los tengamos añadidos a ningún tratamiento.

2.2.1 Definición de casos de uso

Los casos de uso nos sirven para documentar, que acciones están disponibles para el usuario y opciones tiene el usuario para interactuar dependiendo de su perfil o del estado de la aplicación.



Crear cuenta	
Actores	Usuario no registrado
Precondiciones	El usuario llega a esta pantalla desde la pantalla de iniciar sesión.

Flujo	<ol style="list-style-type: none"> 1. Desde la pantalla de crear cuenta, pulsa sobre el botón que indica crear cuenta. 2. En la pantalla de registro deberá indicar el correo electrónico y la contraseña que desea utilizar dos veces para confirmar que introduce la misma contraseña. 3. La aplicación comprueba que la contraseña tiene una seguridad mínima y que el email tiene un formato válido. 4. La aplicación envía los datos al servidor, si el usuario ya existe emite un error. 5. Si el registro es correcto, se abre la pantalla principal de la aplicación. 6. Si hay algún error en el proceso de registro, se muestra mediante un snackbar en la parte inferior de la pantalla.
Postcondiciones	El usuario queda registrado en la aplicación y con una sesión iniciada para futuros accesos a la aplicación

Recuperar cuenta	
Actores	Usuario ya registrado
Precondiciones	El usuario está registrado en la aplicación, pero no recuerda la contraseña de inicio de sesión
flujo	<ol style="list-style-type: none"> 1. Desde la pantalla de inicio de sesión el usuario pulsa en recuperar contraseña. 2. Introduce el correo electrónico con el que se registró. 3. El usuario recibe en su correo electrónico su nueva contraseña. 4. Vuelve a la pantalla de iniciar sesión e introduce la contraseña que ha recibido en su correo electrónico. 5. Accede a la pantalla principal y almacena la sesión en el dispositivo para futuros accesos.
Postcondiciones	El usuario ha recuperado la contraseña y ha accedido de nuevo a la aplicación.

Inicio de sesión	
Actores	Usuario ya registrado
Precondiciones	El usuario está registrado en la aplicación, pero no tiene una sesión arrancada en el dispositivo.
Flujo	<ol style="list-style-type: none"> 1. Desde la pantalla de inicio de sesión el usuario introduce su correo electrónico y la contraseña que definió en el

	<p>registro.</p> <p>2. Accede a la pantalla principal de la aplicación y almacena la sesión en el dispositivo para futuros accesos.</p>
Postcondiciones	El usuario ha accedido a la aplicación y queda conectado en la aplicación.

Cerrar sesión	
Actores	Usuario ya registrado
Precondiciones	El usuario está registrado en la aplicación y con la sesión iniciada.
Flujo	<ol style="list-style-type: none"> 1. Desde la pantalla de principal de la aplicación, pulsamos sobre el botón de perfil de usuario. 2. Pulsamos sobre el botón de cerrar sesión. 3. Se cierra la sesión y nos redirige a la pantalla de inicio de sesión.
Postcondiciones	El usuario ha cerrado la sesión y borrado los datos de usuario del dispositivo.

Crear tratamiento	
Actores	Usuario ya registrado
Precondiciones	El usuario está registrado en la aplicación y con la sesión iniciada.
Flujo	<ol style="list-style-type: none"> 1. Desde la pantalla de principal de la aplicación, pulsamos sobre el botón crear tratamiento. 2. Buscamos el medicamento 3. Revisamos el detalle del medicamento y pulsamos en crear tratamiento 4. Indicamos el inicio y fin (si existe) del tratamiento 5. Indicamos la rutina de dosis que tomaremos durante el día. 6. Guardamos el tratamiento.
Postcondiciones	El tratamiento queda almacenado y se añade a nuestra agenda para que lo revisemos durante el día.

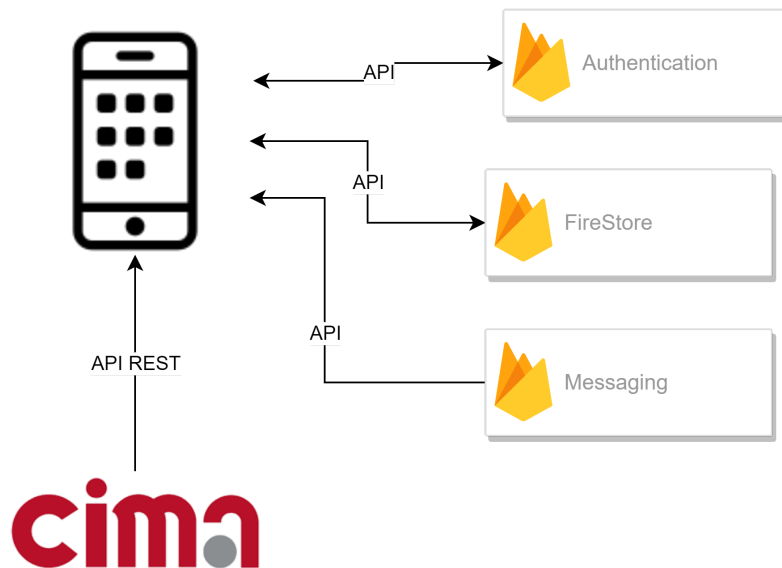
Marcar toma del tratamiento como completada	
Actores	Usuario ya registrado
Precondiciones	El usuario está registrado en la aplicación y tiene un tratamiento en curso
Flujo	<ol style="list-style-type: none"> 1. Desde la agenda situada en la pantalla principal 2. Disponemos de dos opciones <ol style="list-style-type: none"> a. Desde la fila de la dosis que hemos tomado, podemos marcar directamente la toma como completada con un botón específico. b. También podemos pulsar sobre la fila de la toma y nos mostrará todo el detalle del tratamiento. Pulsamos sobre la dosis que hemos tomado y se marca como completada.
Postcondiciones	La dosis queda completada y almacenada.

2.2.2 Diseño de arquitectura de la aplicación

En el proyecto hemos utilizado la arquitectura cliente-servidor, donde el cliente es la aplicación móvil desarrollada en Flutter y utilizaremos dos servidores, uno para almacenar todos los datos del usuario y que gestionaremos nosotros y otra la API de CIMA que nos proveerá de toda la información de los medicamentos.

2.2.2.1 Arquitectura

Para el sistema de autenticación usaremos Firebase Autenticathion. Nos facilitará toda la infraestructura para almacenar los usuarios, recuperar cuenta e incluso en un futuro añadir métodos alternativos de autenticación (Google, Apple...).



Para el almacenamiento de la información utilizaremos FireStore como base de datos NO-SQL. Guardaremos todo lo referente a tratamientos y dosis de los usuarios.

Firestore Cloud messaging nos servirá para el envío de mensajería push a los usuarios. De esta forma podremos avisar a los usuarios a la hora de tomar la medicación.

El patrón de diseño que emplearemos es el patrón BLoC, presentada por el equipo de Google en 2018.

Nuestra aplicación seguirá la siguiente estructura

- **Acceso a Datos:** Capa de acceso a datos a través de la API.
- **Repositorio:** Capa de transformación de los datos en crudo a nuestro modelo de datos.
- **Lógica de negocio:** Capa de negocio e interpretación de los datos obtenidos
- **Vista:** Capa visible donde se generaran las vista con los datos que mostraremos al usuario

2.2.2.2 Base de datos

La base de datos será No-Sql, almacenará los datos en formato JSON en Firebase Firestore.

Tendremos varias colecciones.

Usuarios

users
+ id: uuid
+ email: String
+ Name: String
+ birth_date: DateTime
+ sex: String

Tratamientos

treatments
+ id: uuid
+ id_medication: int
+ id_user: uuid
+ start: DateTime
+ end: DateTime
+ frequency_hours: int

Dosis

doses
+ id: uuid
+ treatment_id: uuid
+ taking_time: datetime
+ state: String

3. Implementación

3.1 Creación del proyecto

Para la creación del proyecto he utilizado el paquete [very good cli](#) creada por la empresa [Very Good Ventures](#).

Esta es una empresa puntera en el desarrollo de software utilizando el framework Flutter y ha colaborado con Google en muchos proyectos, promueve el open source y cuenta con muchas librerías muy recomendables que también he utilizado para crear este proyecto.

Primero debemos instalar el paquete en el sistema

```
dart pub global activate very_good_cli
```

El comando para la creación del proyecto es el siguiente

```
very_good create . --org-name com.asignes --project-name  
medical_reminder --desc "medical reminder app"
```

He dividido el proyecto en diferentes funcionalidades, he intentado encapsular esas funcionalidades en paquetes para gestionar la lógica.

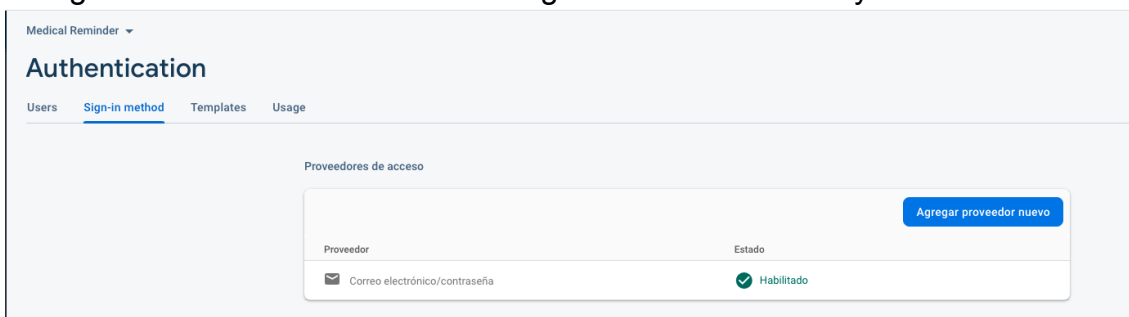
3.2 Modelos de datos

Todos los modelos de datos se obtienen en formato JSON. Para la generación de estos modelos hemos utilizado la web quicktype.io que facilita mucho la creación del modelo a partir de un objeto de ejemplo.

También utilizamos los paquetes [json_serializable](#) y [json_annotation](#) que facilitan mucho el trabajo con modelos, ya que permiten crear los objetos fácilmente y las conversiones en ambos sentidos.

3.3 Autenticación

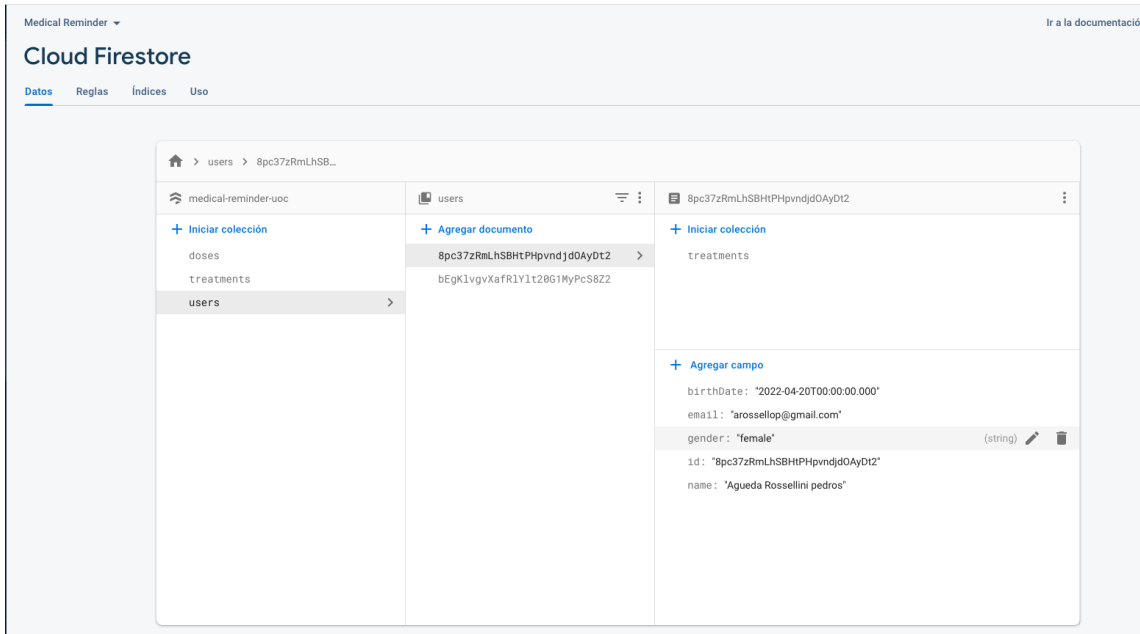
La autenticación de la aplicación se realiza mediante Firebase, en la configuración inicial solo admitimos registro mediante email y contraseña.



Toda la lógica se encapsula dentro del paquete `authentication_repository`.

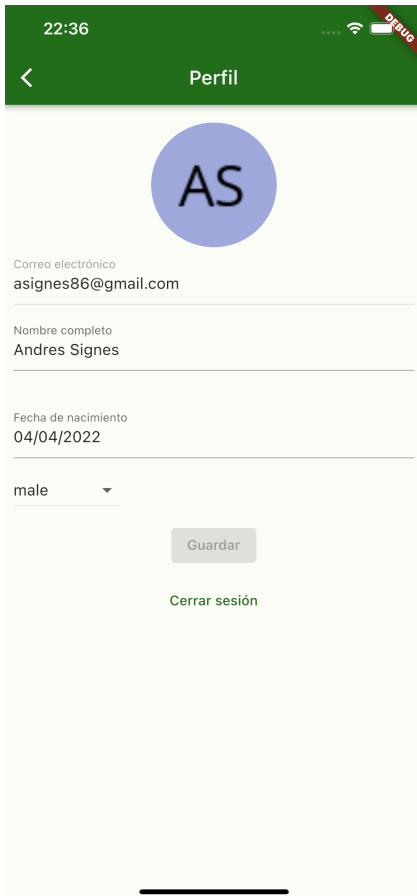
Este paquete contiene las dependencias a los paquetes de Firebase que gestionan el core del plugin, la autenticación y el acceso a firestore.

Al registrarse el usuario, se genera un registro en la colección `users` en la que almacenamos otras propiedades del usuario. El identificador del registro es el mismo en Authentication y en la colección.



La aplicación permite el registro de usuario, recuperar contraseña, abrir sesión y cerrar sesión. Queda pendiente realizar la modificación de la contraseña una vez el usuario está registrado.

Desde el perfil de usuario se pueden modificar los datos del usuario y cerrar sesión.



3.4 API Externa (CIMA)

Para la obtención de información de medicamentos he utilizado la API pública de CIMA.

Para ello he creado varios paquetes para gestionar el modelo de datos, la llamada a la API a través de HTTP y el repositorio.

Para realizar las búsquedas se utiliza el endpoint

```
https://cima.aemps.es/cima/rest/medicamentos?nombre=<TEXTO_BUSQUEDA>
```

y para consultar un medicamento en concreto lo haremos mediante su número de registro

```
https://cima.aemps.es/cima/rest/medicamento?nregistro=<NUM_REGISTRO>
```

Estas llamadas se realizan mediante el paquete [http](#), al recuperar los datos convertimos la respuesta en JSON al modelo de datos que hemos creado y podemos mostrar un listado de medicamentos o el detalle de uno en específico.



Una idea que ha quedado pendiente es la búsqueda de medicamentos mediante el código de barras que aparece en la caja del mismo.

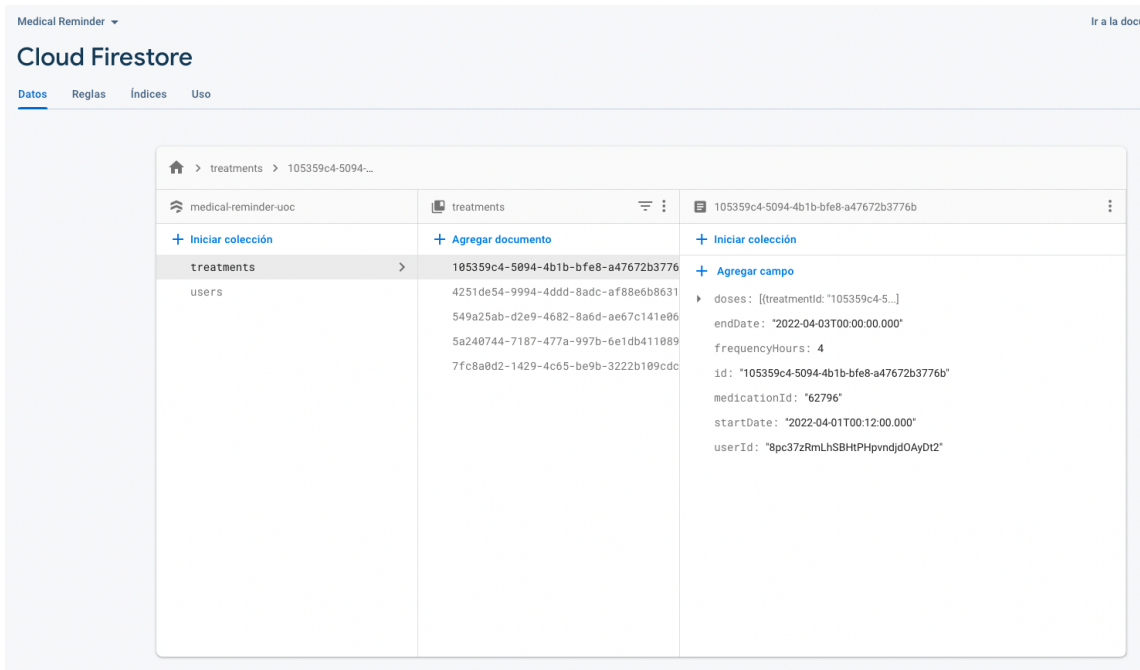
3.5 Tratamientos y Dosis

Los tratamientos y las dosis del mismo se almacenan en Firestore de Firebase.

Para ello he creado un modelo que representa la estructura que ya definimos en su momento.

He almacenado todo dentro de una misma colección, ya que a la hora de realizar las operaciones de creación y consulta resulta más útil tener todo dentro de un mismo objeto para poder operar con él.

También he añadido un objeto Medicamento que se serializa al obtener los datos a través de la API de CIMA, en la base de datos de CIMA solo almacenamos el ID.



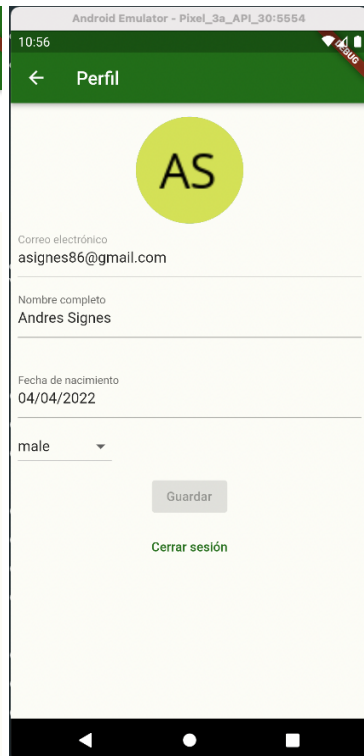
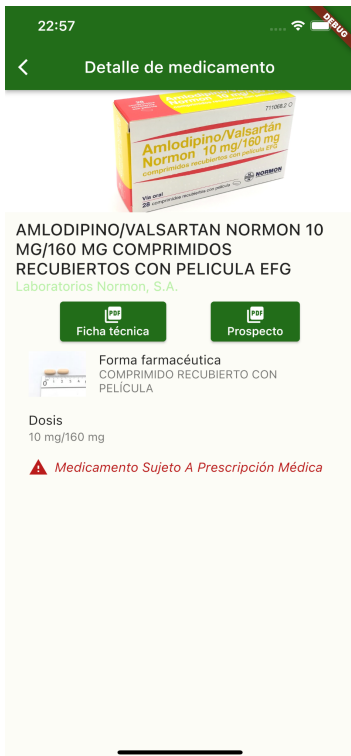
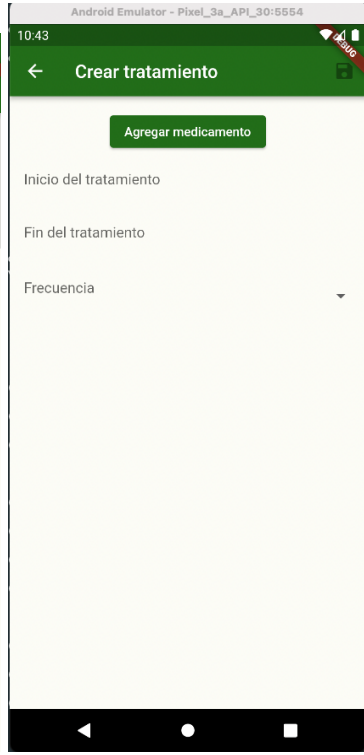
3.6 Interfaces

La aplicación utiliza el estilo [Material Design](#), he intentado mantener unas interfaces limpias y fáciles de utilizar y que siguiesen con lo definido en los mockups iniciales.

En los diferentes formularios que hay en la app he implementado validaciones para que se completen todos los campos y notificar al usuario si hay algún error y en que campo está el error. Las notificaciones de estos errores se realiza mediante SnackBars que aparecen de forma temporal en la parte inferior de la aplicación indicando el error.

Para el estilo he definido un color base y a partir de ese el propio sistema genera una paleta de colores.

Algunas de las capturas de las interfaces en ambas plataformas.

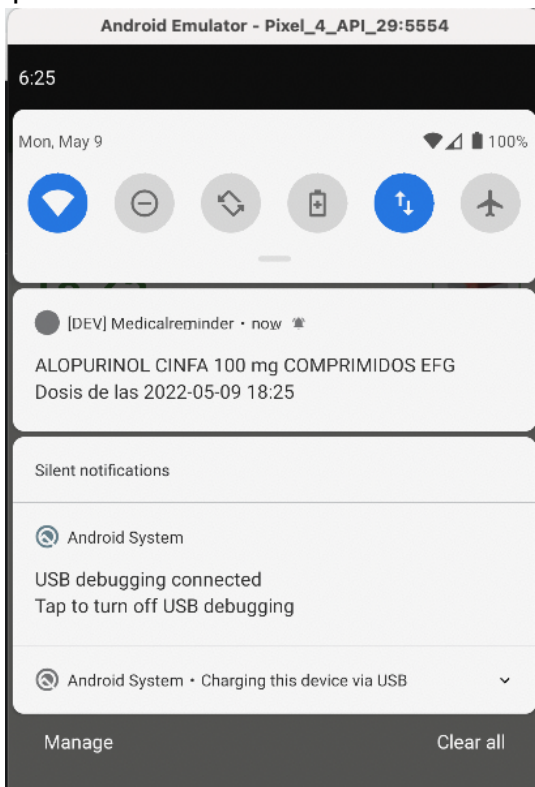


La aplicación está pensada para ser utilizada en modo vertical, no obstante, he intentado que todas las interfaces sean usables en formato horizontal.

3.7 Notificaciones

Las notificaciones se realizan de forma local. Al obtener todos los tratamientos, el sistema revisa todas las dosis y programa una notificación para la hora de la toma.

Lo ideal sería realizar el envío de las notificaciones a través del servidor, pero no he encontrado una forma gratuita que programe una notificación al añadir las dosis a la Base de datos. Esta hubiese sido mi opción preferida y creo que más óptima, ya que si el usuario tiene la aplicación en varios dispositivos puede que no reciba todas las notificaciones.



3.8 Multiidioma

Todos los textos de la aplicación se almacenan en un fichero ARB. En cada fichero se almacenan las localizaciones por cada idioma. Así podremos añadir nuevos idiomas a la aplicación de manera sencilla y rápida.

4 Pruebas

4.1 Pruebas automatizadas

4.1.1 Github Actions

El paquete que he utilizado para la creación del proyecto, `very_good_cli` trae ya una GitHub Action que analiza nuestro código, y ejecuta los test que tengamos implementados, [very_good_workflows](#).

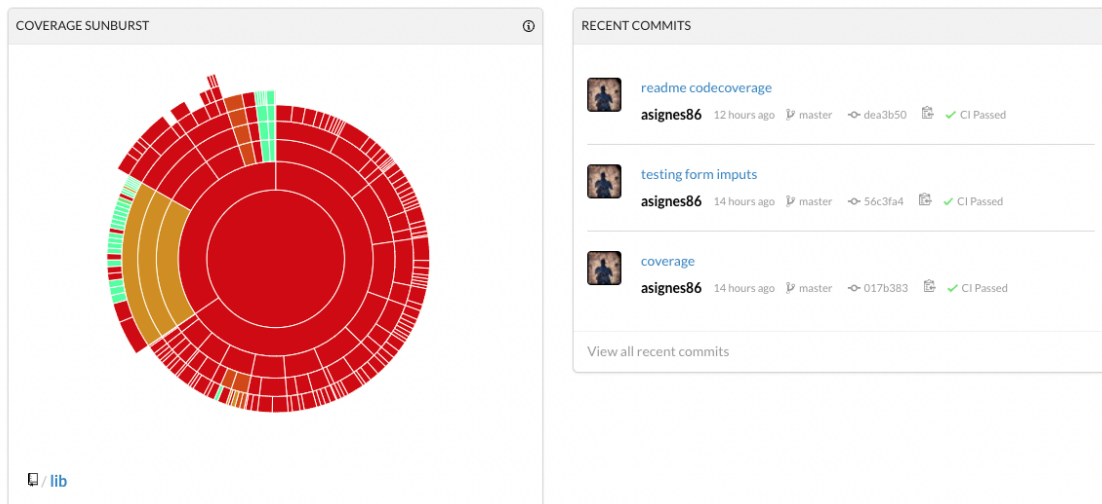
Los pasos que ejecuta son los siguientes:

- Install Dependencies
- Format
- Analyze
- Run tests
- Check Code Coverage

4.1.2 Pruebas Unitarias

Debido a la falta de tiempo, solo he podido realizar test unitarios de algunos de los paquetes de la aplicación.

Este es el [coverage](#) actual de la app.



Files	Files	Green	Yellow	Red	Coverage
lib	1625	173	0	1452	10.65%
packages	857	443	0	414	51.69%
Project totals (150 files)	2482	616	0	1866	24.82%

Sobre todo me he centrado en ir de dentro hacia fuera, empecé por la API externa que consulta a CIMA y los campos de formulario que se utilizan en la creación de la cuenta y el perfil de usuario. Según el coverage hay aproximadamente un 25% de la aplicación testeada.

4.2 Pruebas manuales

El resto de validaciones las he ejecutado con los dispositivos que tenía a mi disposición y los emuladores y simuladores de mi PC.

Físicamente, dispongo de los siguientes dispositivos:

- LG Nexus 5x con Android API 27
- iPhone 12 Pro

Al disponer de un PC con macOS he podido hacer pruebas en dispositivos iOS.

En los simuladores/emuladores he utilizado los siguientes dispositivos:

- Pixel 3a con Android API 30
- Pixel 4 con Android API 29
- iPhone 13 Pro Max

Las pruebas manuales que he realizado incluyen:

Sobre la parte de autenticación inicial del usuario

- Creación de cuenta
- Recuperación de contraseña
- Inicio de sesión.

Sobre la gestión de tratamientos, He revisado los siguientes flujos

- Creación de un tratamiento y posibles errores y avisos que se pueden producir durante la creación.
- Revisión de las diferentes pantallas de consulta
 - Consulta de tratamiento
 - Consulta de medicamento del tratamiento.
 - Abrir enlaces del medicamento (ficha técnica y prospecto)
 - Marcar una toma como completada
 - Eliminación del tratamiento.
- Perfil de usuario
 - Consulta de datos
 - Actualización de datos
 - Cierre de sesión

También he validado que las notificaciones se envían correctamente tanto con la aplicación abierta, en segundo plano o cerrada totalmente.

5. Conclusión

Una vez finalizado el trabajo, analizaré el trabajo realizado y los resultados que he obtenido, además de una visión a futuro sobre el proyecto.

Empezando por el principio, la elección de la temática sanitaria creo que ha sido un acierto, es el ámbito en el que me muevo laboralmente y creo que esto me ha ayudado a entender mejor las necesidades de los usuarios. Sobre la metodología utilizada considero que me hubiera sentido más cómodo siguiendo la metodología ágil para poder iterar sobre lo ya realizado e ir mejorando todo el conjunto.

La planificación que planteé en un inicio se ha cumplido en su mayoría, no obstante ha habido algunas semanas en las que no se ha podido trabajar todas las horas planificadas y se ha tenido que recuperar en otra jornada. También ha habido funcionalidades e ideas que por falta de tiempo han quedado relegadas, que espero completar en un futuro no muy lejano.

Los diseños y prototipados que hice en su momento me han servido de mucha ayuda en la hora del desarrollo. Es una parte que no me entusiasmaba, pero después agradecí. Al realizar todo el prototipado y diseño tuve una visión más global y estructurada de toda la aplicación antes de empezar con el desarrollo.

Pasando a la fase de implantación, tengo que agradecer mucho la elección de Flutter como framework, su comunidad es muy buena y me ha ayudado mucho. He aprendido mucho sobre gestores de estado, estructuración de proyectos y Widgets que no conocía.

La elección de Firebase como backend me pareció lo más lógico en su momento, y creo que acerté. La inclusión de authenticator para registrar al usuario fue bastante sencilla. La gestión de Firestore no lo fue tanto, ha sido mi primer proyecto utilizando una base de datos NO SQL y al tener tan interiorizado las base de datos relacionales fue complicado al principio. No obstante, después de algunos cambios en el diseño, pienso que el resultado no ha sido malo y ha servido de aprendizaje para este tipo de base de datos que era nuevo para mí.

Respecto a los objetivos que me planteé al inicio del proyecto, en mi opinión considero que el resultado es una aplicación funcional, que cumple con las funcionalidades que definí al inicio y que además es visualmente aceptable. Hay aspectos mejorables y funcionalidades que se han quedado por el camino, pero el objetivo principal era aprender de todo este proceso.

Siguientes pasos

Una vez finalizada la entrega del proyecto, toca pensar en cuáles son los siguientes pasos. Revisar las funcionalidades que creo que son mejorables y otras que se han quedado por el camino.

Mejoras

- Notificaciones desde la nube a todos los dispositivos del usuario
- Edición de un tratamiento ya creado.
- Multiidioma (ya está preparada, pero solo están los textos en español)
- Mejor gestión de los tratamientos finalizados/completados.
- Mejoras de diseño en formato horizontal
- Permitir crear tratamientos sin una fecha de fin.

Nuevas funcionalidades

- Integrar los recordatorios de dosis con el calendario del dispositivo.
- Buscar un medicamento a través del código de barras.
- Añadir sección de medicamentos favoritos.
- Poder gestionar tratamientos de familiares.
- Versión Web
- Almacenamiento offline de los tratamientos actuales.

6. Glosario

- **AEMPS:** Agencia española de medicamentos y productos sanitarios.
- **CIMA:** Centro de información online de medicamentos de la AEMPS.
- **GitHub:** Web de alojamiento de repositorio de código.
- **Flutter:** Framework de desarrollo de aplicaciones multiplataforma creado por Google.
- **Dart:** Lenguaje de programación de código abierto creado por Google.
- **Firebase:** Es una plataforma para el desarrollo de aplicaciones web
- **BLoC:** Busines Logic Component
- **ARB:** Application Resource Bundle. se trata de un formato de fichero creado por google para la gestión de traducciones. Sigue el estándar de JSON
- **Coverage:** Se trata del porcentaje de cobertura del código que está siendo cubierto por las pruebas. Es decir, la parte automatizada de testeo de la aplicación, si existen líneas del código de la aplicación que no se han evaluado automáticamente, indicara que esa línea no está cubierta.

5. Bibliografía

- <https://es.wikipedia.org/>
- <https://flutter.dev/>
- <https://firebase.google.com/>
- <https://cima.aemps.es/>
- <https://draw.io/>
- <https://es.wikipedia.org>
- <https://bloclibrary.dev/>
- <https://verygood.ventures/>
- <https://icon.kitchen/>
- <https://www.deepl.com/>

6. Anexos

- [Manual API CIMA](#)