

Memoria Final



FISIOMAP IA ASSISTANT



Alexia Ainhoa Fernández Losper

Alexiafeernandez1234@gmail.com

https://github.com/a-fernandez21/Proyecto_Tfg.git

https://github.com/bdominguez6/fisiomap_pro.git

Colegio Vivas

ÍNDICE

Plataforma de destino	3
Storyboard	3
Elementos utilizados	14
Problemas encontrados	14
Tutorial de instalación	15

Plataforma de destino

Fisiomap IA Assistant concluye siguiendo la línea establecida desde su concepción inicial: una ampliación funcional de la aplicación web Fisiomap, adaptada al entorno móvil para ofrecer una experiencia más accesible y eficiente. No obstante, el proyecto sigue en evolución y ha logrado una buena aceptación, consolidándose como una extensión sólida y coherente con la versión web original.

Descripción funcional

Desde su planteamiento inicial, Fisiomap IA Assistant se desarrolló con el objetivo de mejorar la gestión clínica y facilitar la comunicación entre profesionales sanitarios y pacientes mediante el uso de inteligencia artificial. La aplicación se compone principalmente de dos partes: por un lado, el personal sanitario, que puede grabar, transcribir y generar informes automáticos de las sesiones; y por otro, los pacientes, que reciben una atención más rápida y personalizada. Además, el sistema incorpora una parte administrativa fundamental que se encarga de gestionar y proteger los datos clínicos, garantizando su seguridad, trazabilidad y correcto almacenamiento.

Storyboard

Al abrir la aplicación, el usuario se encuentra con un formulario de inicio de sesión donde debe ingresar el correo electrónico y la contraseña asociados a su cuenta. Tras completar los campos y pulsar el botón "Iniciar sesión", el sistema valida las credenciales y redirige al usuario a la interfaz correspondiente según su perfil, en este caso, el del profesional sanitario.

En caso de que el usuario ingrese un correo electrónico no válido o deje el campo vacío, se muestra una **SnackBar** indicando que el correo electrónico no puede estar vacío. Lo mismo ocurre con la contraseña: si está vacía o no cumple los requisitos, se muestra un mensaje de error similar.

En la misma pantalla también se encuentra el botón "¿Has olvidado la contraseña?", que permite recuperar el acceso en caso de no recordar la contraseña. Al seleccionarlo, el usuario es dirigido a un nuevo formulario en el que debe introducir su correo electrónico, para recibir

un enlace que le permitirá restablecer su contraseña.




Correo electrónico

Contraseña

[¿Has olvidado la contraseña?](#)

Continuar





¿Has olvidado la contraseña?

Introduce la dirección de correo electrónico asociada a tu cuenta y te enviaremos un enlace para restablecer tu contraseña.

Correo electrónico

Send email to reset password



Error de identificación 

No hay ningún usuario asociado a esta dirección de correo

Continuar

Continuar

El correo electrónico no puede estar vacío

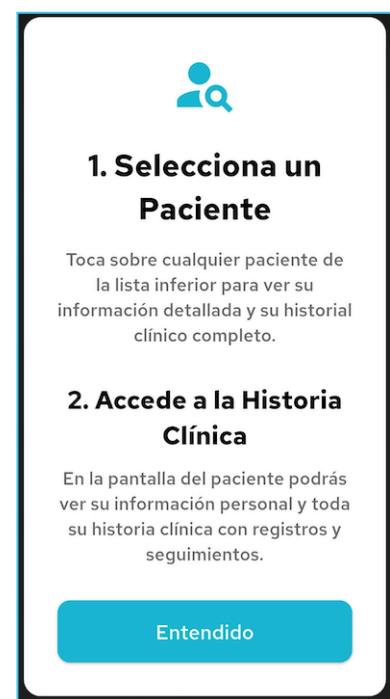
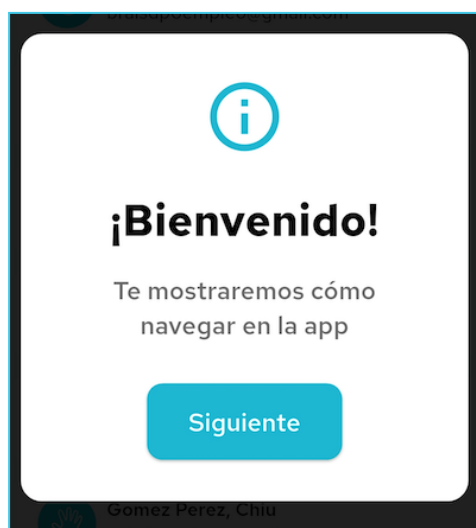
La contraseña no puede estar vacía

Tras darle a Continuar, al usuario se le muestra un modal para activar el login biométrico. Esta funcionalidad aún no está habilitada para el usuario, ya que su implementación y mantenimiento están a cargo de Brais.



A continuación, se presentan dos wizards consecutivos, que solo aparecen la primera vez que se abre la aplicación tras su instalación:

1. El primer wizard da la bienvenida al usuario y nos muestra un mensaje de cómo navegar por la aplicación.
2. El segundo wizard indica que el usuario puede seleccionar un paciente de la lista y acceder a su historia clínica.

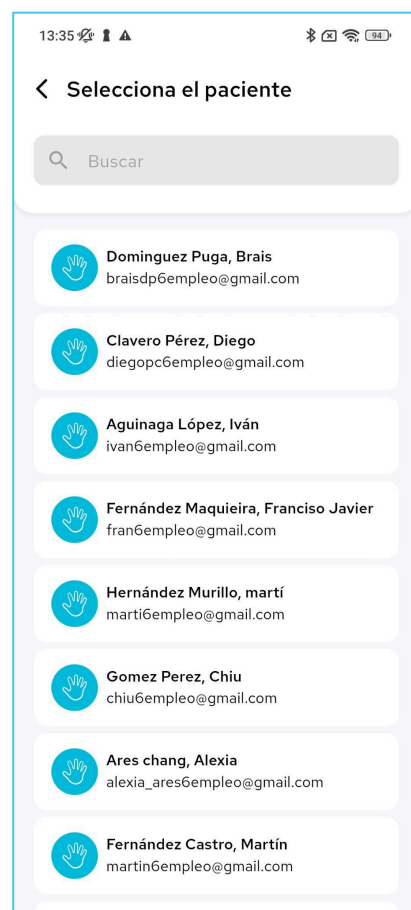


Una vez finalizados los wizards, el usuario accede a la pantalla que muestra la lista de pacientes asociados a su cuenta, presentada de forma paginada con un máximo de 15 pacientes por página para facilitar la navegación.

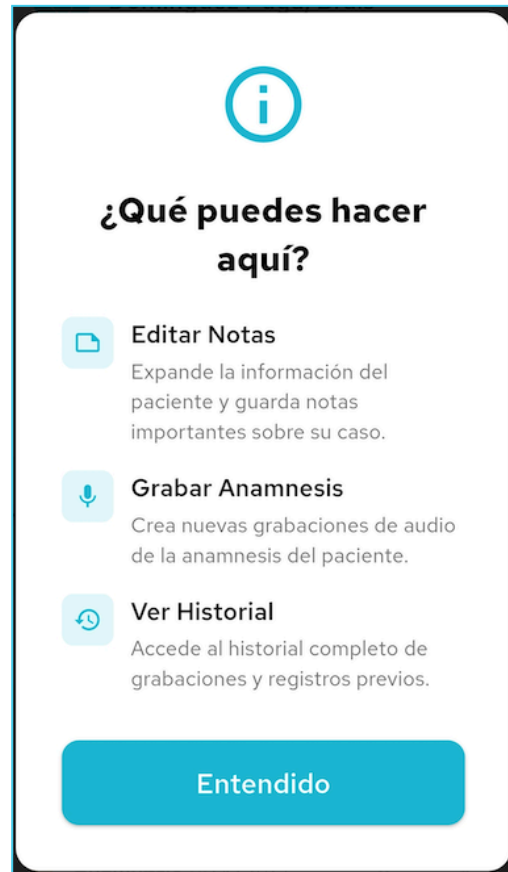
En la parte superior de la lista se encuentra una barra de búsqueda, que funciona como un sistema de filtrado, permitiendo localizar rápidamente un paciente concreto introduciendo su nombre o cualquier otro criterio disponible.

En caso de no encontrar resultados que coincidan con los criterios de búsqueda, el sistema muestra un mensaje indicativo informando que no se han encontrado pacientes que cumplan con los filtros aplicados.

Actualmente, tanto el login biométrico como algunas funcionalidades de la lista de pacientes no están habilitadas, ya que su desarrollo está a cargo de Brais.



Una vez seleccionado un paciente, antes de acceder a la información detallada, se muestra un wizard introductorio que explica al usuario las funcionalidades disponibles en esta pantalla.



Al finalizar este wizard, el usuario accede a la pantalla completa, que incluye:

- Un Card con la información del paciente, mostrando todos los datos registrados en la base de datos. Esta Card se despliega al hacer scroll hacia abajo, permitiendo visualizar más detalles según se navega por la pantalla.

13:59

< Dominguez Puga, Brais

Dominguez Puga, Brais

Última visita: Sin información

Fecha de nacimiento: 2000-05-24 (25 años)

Sexo: Masculino

Teléfono: +34624809256

Email: braisd6empleo@gmail.com

Dirección: Calle Doctor Corbal, 129,, Vigo, 36207

Anotaciones importantes

Un crack

Historia Clínica 5 registros

Anamnesis 15 Oct 2025

Anamnesis 08 Oct 2025

Anamnesis 01 Oct 2025

Anamnesis 22 Sep 2025 Nueva anamnesis

- Una sección para anotar observaciones importantes sobre el paciente, que se realiza a través de un modal para añadir la información de manera rápida y organizada.

Anamnesis 01 Oct 2025

Nueva anamnesis

Notas guardadas correctamente

Anotaciones

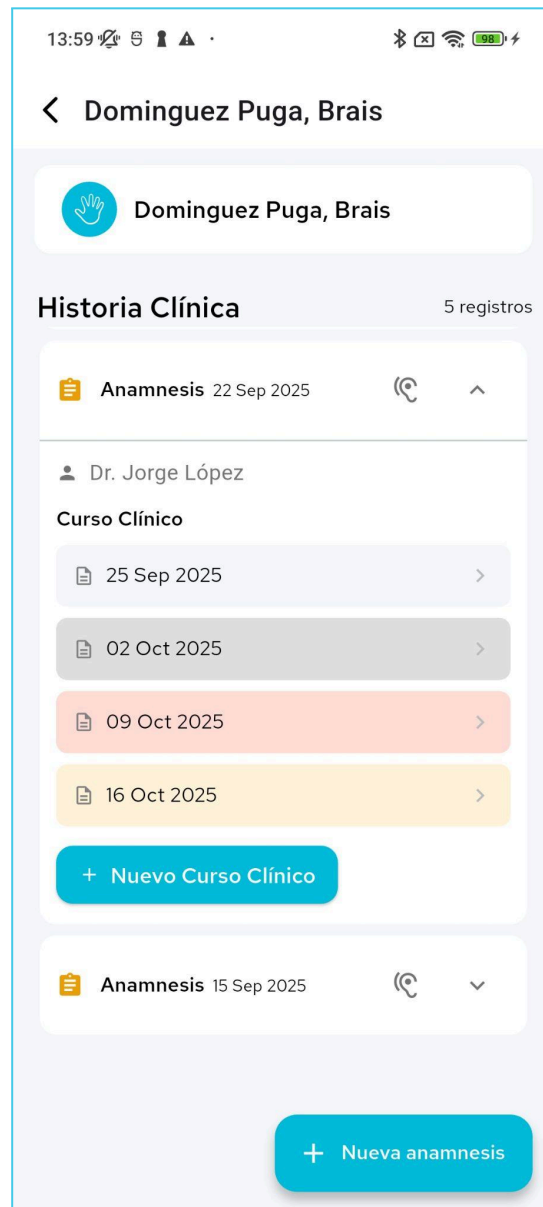
Apunta información importante sobre el paciente

Un crack

8/500

Cancelar

- El historial clínico completo, presentado como una lista scrollable de todas las anamnesis registradas. Cada anamnesis se puede desplegar para consultar su contenido, incluyendo los cursos clínicos asociados a esa anamnesis, facilitando un seguimiento detallado de la evolución del paciente.

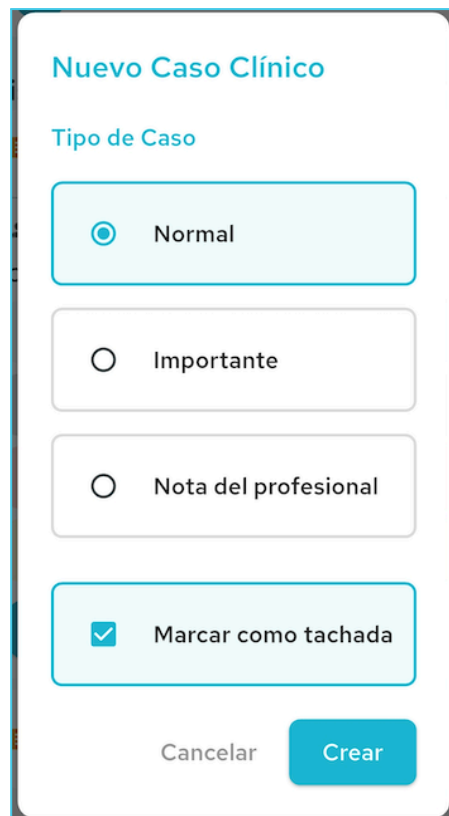


En esta sección también se dispone de un botón para crear un nuevo curso clínico. Al seleccionarlo, se abre un modal donde el usuario puede elegir entre tres tipos mediante radiobuttons:

Normal, Importante o Nota del profesional, además de un checkbox para indicar si el curso clínico debe marcarse como tachado.

La visualización del curso clínico sigue estas reglas:

Si se marca la opción Tachada, el curso clínico siempre se mostrará en gris medio oscuro, independientemente del tipo seleccionado.



El formulario 'Nuevo Caso Clínico' presenta la siguiente estructura:

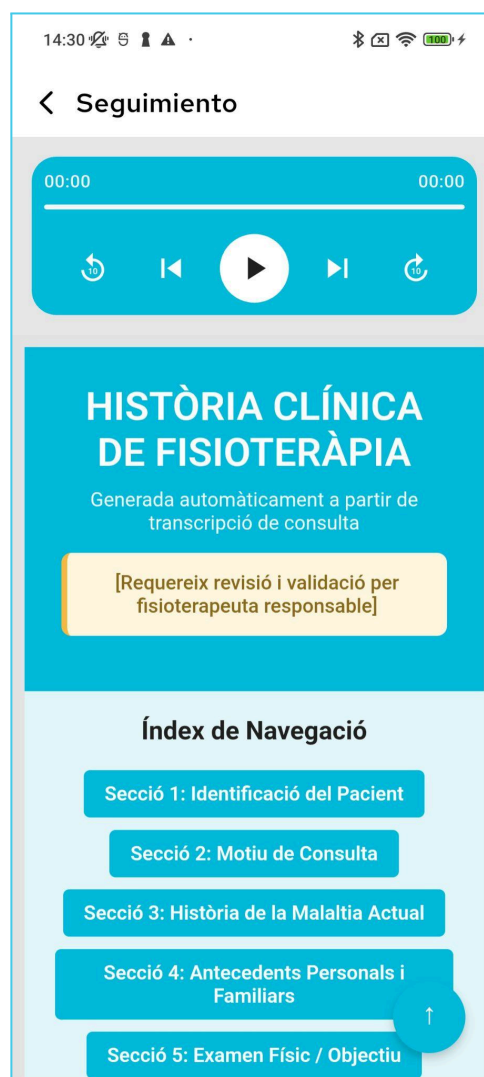
- Título:** Nuevo Caso Clínico
- Encabezado de sección:** Tipo de Caso
- Opciones de Tipo de Caso:**
 - ☒ Normal
 - ☐ Importante
 - ☐ Nota del profesional
- Opción de Estado:**
 - ☒ Marcar como tachada
- Botones de Acción:**
 - Cancelar
 - Crear

- Si no está tachado:
 - Normal: se crea con color blanco.
 - **Importante:** se crea con color rojo coral.
 - **Nota del profesional:** se crea con color amarillo.

Estas reglas permiten identificar visualmente la importancia y el estado de cada curso clínico, garantizando que la información más relevante o ya revisada se destaque de manera clara y coherente.

Al seleccionar el icono de la oreja, el usuario accede a una pantalla que incluye un reproductor de audio, donde puede escuchar la grabación correspondiente a la anamnesis o al curso clínico seleccionado. Sin embargo, actualmente el backend aún no permite recibir ni enviar el audio, por lo que el reproductor solo puede funcionar con grabaciones creadas recientemente dentro de la propia aplicación, en pruebas locales. Esto se debe a que el equipo de backend —responsable de los endpoints y de la gestión de los datos— todavía no ha integrado completamente esta funcionalidad.

Junto al reproductor, se encuentra un WebView que muestra el informe en formato HTML, generado automáticamente a partir de un flujo de n8n desarrollado por el equipo de backend. Esta disposición permite al usuario revisar tanto el audio como el informe correspondiente a la misma anamnesis o curso clínico en una sola pantalla, facilitando la interpretación de la información clínica.



Al crear un nuevo curso clínico o anamnesis, se muestra un wizard introductorio que explica al usuario cómo realizar la grabación. Una vez que se pulsa "Entendido", se abre la pantalla de la grabadora, donde se solicitará el permiso de acceso al micrófono para poder llevar a cabo la grabación.

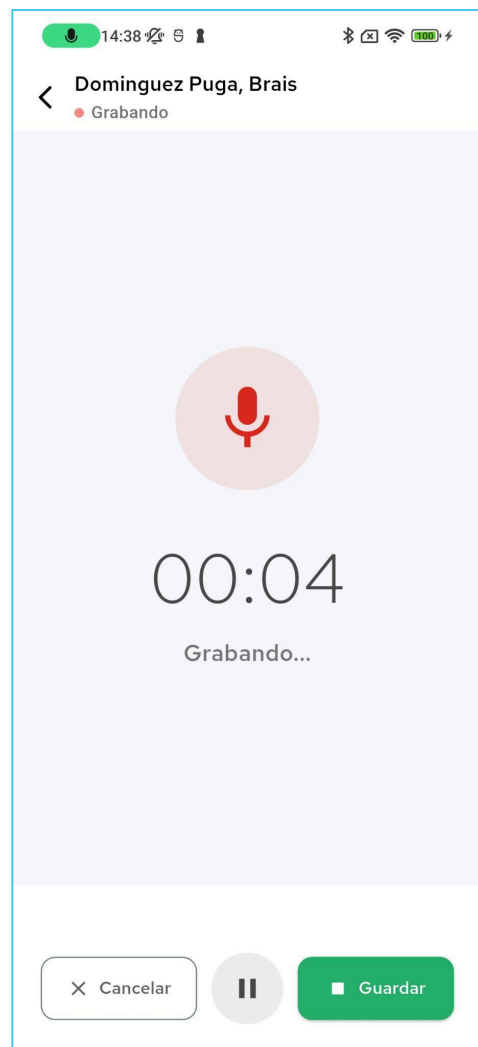
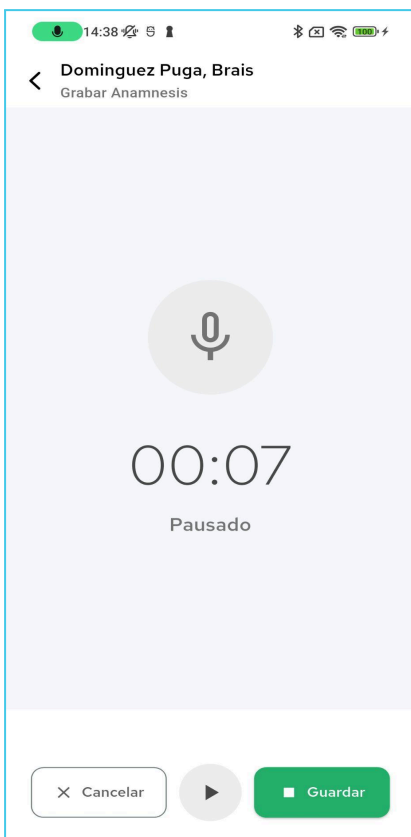
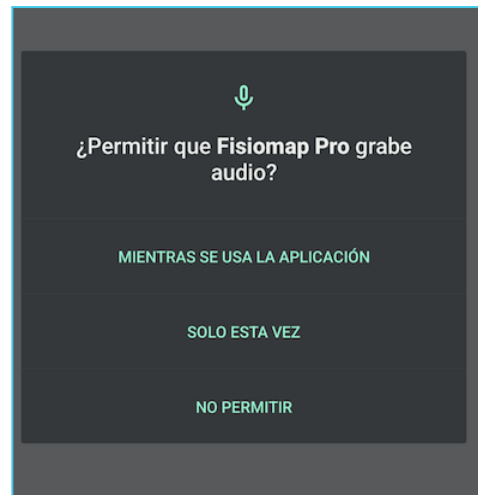
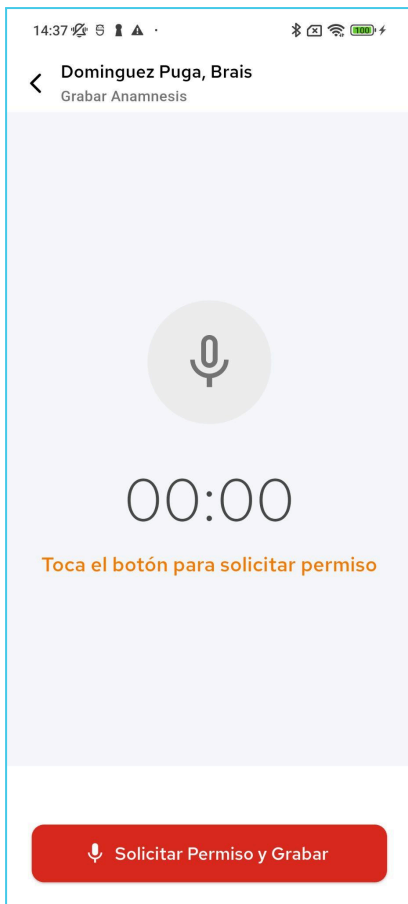
En el primer uso, la grabación comienza automáticamente tras aceptar el permiso; en usos posteriores, el usuario debe pulsar el botón de iniciar grabación manualmente. La interfaz incluye una AppBar que muestra el nombre del paciente seleccionado y el tipo de contenido que se está creando (anamnesis o curso clínico). Al pulsar el botón de volver, se regresa a la pantalla del historial clínico y la información del paciente.

Dentro de la interfaz de la grabadora se encuentran tres botones principales:

- **Cancelar:** detiene la grabación y vuelve automáticamente a la pantalla anterior.
- **Play/Pausa:** permite reproducir o pausar la grabación. El icono cambia según el estado actual.
- **Guardar:** al pulsarlo, la grabación se guarda y se regresa a la pantalla anterior, generándose automáticamente una nueva Card para la anamnesis o el curso clínico.

Originalmente, la intención era que al guardar la grabación se mostrase un estado de subida mientras se almacenaba en la base de datos, para que luego pudiera visualizarse en la pantalla del reproductor de audio. Sin embargo, debido a limitaciones actuales del backend, esta funcionalidad aún no está activa.





Elementos utilizados

En el desarrollo de Fisiomap IA Assistant he trabajado con Dart y el framework Flutter, que son las tecnologías que utiliza la empresa para sus aplicaciones. Nada más llegar, estuve aproximadamente tres semanas realizando un curso de Flutter de Udemy dentro de la propia empresa, lo que me permitió aprender la base necesaria antes de empezar con el proyecto.

Flutter me ha ayudado a hacer la creación de las distintas pantallas de la aplicación, utilizando sus widgets para construir una interfaz clara y funcional. Toda la lógica y el comportamiento del asistente se implementaron en Dart, organizando el código por módulos y archivos para mantenerlo ordenado y fácil de mantener. En especial, la estructura se ha desarrollado siguiendo el patrón MVVM, que es el que se utiliza internamente en la empresa.

Además, utilicé varios paquetes de Flutter para gestionar aspectos como la navegación entre vistas, el estado de la aplicación y la conexión con servicios externos. Esto permitió añadir funcionalidades más completas sin complicar en exceso la estructura del proyecto.

Finalmente, configuré el entorno de compilación para generar los ejecutables y poder probar la aplicación tanto en emuladores como en dispositivos reales, dejando preparada la base para futuras mejoras o nuevas versiones.

Librerías usadas:

Categoría	Librerías clave	Función / Uso
Core & Estado	flutter, provider, flutter_bloc, pumpun_core	Framework principal y gestión de estado.
Autenticación & Seguridad	firebase_core, firebase_messaging, flutter_secure_storage, local_auth	Firebase, notificaciones, almacenamiento seguro y autenticación biométrica.
Navegación & Web	go_router, webview_flutter, url_launcher, web_browser_detect	Navegación entre pantallas y manejo de contenido web.
Multimedia	image_picker, camera, google_mlkit_face_detection, google_mlkit_text_recognition, flutter_exif_rotation	Captura de imágenes, detección facial y OCR.
Video	video_player, chewie, media_kit, vimeo_player_flutter	Reproducción de videos, con controles y soporte Vimeo.
Audio	record, audioplayers	Grabación y reproducción de

		audio
Documentos & PDF	syncfusion_flutter_pdfviewer, flutter_pdfview, pdfx, file_picker, open_file, path_provider	Visualización y manejo de PDFs y archivos.
Calendario & Fecha	syncfusion_flutter_calendar, syncfusion_flutter_datepicker, scroll_date_picker	Calendario y selección de fechas.
UI & Diseño	cupertino_icons, font_awesome_flutter, google_fonts, lottie, delayed_display, loading_indicator, carousel_slider, photo_view, showcaseview	Iconos, animaciones, carruseles, tutoriales y zoom en imágenes.
Chat & Mensajería	chat_bubbles	Interfaz de chat con burbujas.
Formularios & Input	pin_code_fields, html_editor_enhanced	Campos PIN y editor HTML.
Internacionalización	flutter_localizations, intl, intl_utils	Localización y soporte multilenguaje.
Utilidades	dio, event_bus, synchronized, path, package_info_plus, device_info_plus, android_id, permission_handler, flutter_dotenv, idle_detector_wrapper	HTTP, eventos, sincronización, info de dispositivo, permisos y variables de entorno.
Otros Widgets	expandable, responsive_grid, fluttertoast, smooth_star_rating_null_safety, flutter_widget_from_html	Widgets expandibles, grids, toasts, ratings y renderizado HTML.
Configuración	flutter_launcher_icons, flutter_native_splash, accessibility_tools	Iconos, splash screen y accesibilidad.
Testing	flutter_test, flutter_lints	Testing y linter de código.

Problemas encontrados

Una vez finalizadas todas las pantallas y la parte visual de Fisiomap IA Assistant, surgieron varios problemas relacionados con el backend, tanto por parte de mis compañeros de prácticas como por parte del backend de la empresa. Estos retrasos hicieron que no me diera tiempo a integrar los datos reales en la aplicación.

A día de hoy, lo único que he podido recoger correctamente de la base de datos son los campos básicos de información del paciente. Sin embargo, mi intención era implementar muchas más funcionalidades conectadas a datos reales: reproducir los audios almacenados, gestionar el sistema de marcas dentro del audio, y mostrar la información de las anamnesis junto con sus correspondientes cursos clínicos.

De hecho, en una de las primeras versiones que presenté, tenía las anamnesis y los cursos clínicos separados. Chiu me comentó que cada anamnesis debía estar relacionada con su curso clínico, así que realicé los cambios necesarios y ahora esa parte está estructurada correctamente a nivel de interfaz y lógica.

El problema principal es que en la última semana antes de entregar el proyecto todavía no disponíamos de todo lo necesario desde el backend: no habían terminado los endpoints que debíamos consumir y, además, el acceso a la API nos lo proporcionaron muy tarde, prácticamente en la misma semana de la entrega.

En conclusión, considero que hubiera tenido tiempo de sobra para dejar la aplicación completamente funcional y trabajando con datos reales si el backend hubiese estado listo a tiempo.

Tutorial de instalación

El proyecto se encuentra dentro de un repositorio de GitHub, y la rama donde he trabajado es `evo-anamnesis-module`. Para poder ejecutar la aplicación, es necesario clonar el repositorio, acceder a la rama correspondiente y abrir el proyecto en un entorno de desarrollo compatible como puede ser Visual Studio Code.

1.- Clonar el repositorio

- Clonar el repositorio de GitHub usando Git o descargarlo como archivo ZIP. Una vez descargado, situarse en la rama `evo-anamnesis-module`, que contiene la versión desarrollada del proyecto.

2.- Abrir el proyecto en un IDE

- El proyecto está desarrollado en Flutter, por lo que debe abrirse en un IDE compatible, como Android Studio, Visual Studio Code o IntelliJ IDEA, con los plugins de Flutter y Dart instalados.

3.- Instalar dependencias

- Al abrir el proyecto, ejecutar el comando:
 - `flutter pub get`
- Esto descargará todas las dependencias necesarias para que la aplicación funcione correctamente.

4.- Ejecutar la aplicación

- La aplicación puede ejecutarse en un dispositivo Android real conectado por USB (con la depuración USB activada).
- Una cosa a tener en cuenta es que, por el momento, no está habilitada para ejecutarse en iOS, ya que falta que la empresa configure correctamente el archivo `GoogleServices-Info.plist` necesario para su funcionamiento en dispositivos Apple.