

# **BlutTruck, PÁGINA WEB PARA MONITORIZACIÓN**

## **BlutTruck, WEBSITE FOR CARDIAC MONITORING**



**VNiVERSiDAD  
D SALAMANCA**

**Autor:** Carlos León Arjona

**Curso académico:** 2024-2025

**Tutor:** Iván Álvarez Navia



## Certificado del/los tutor/es TFG

D./Dña. Iván Álvarez Navia, profesor/a del Departamento de Ingeniería Informática de la Universidad de Salamanca,

HACE/N CONSTAR:

Que el trabajo titulado “**BlutTruck: PÁGINA WEB PARA MONITORIZACIÓN**”, que se presenta, ha sido realizado por Carlos León Arjona, con DNI 45135849M y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado en Ingeniería Informática en esta Universidad.

Salamanca, 3 de Julio de 2025

Fdo.:

## Resumen

El desarrollo de la herramienta propuesta consistirá en la creación de una aplicación web que permitirá a los usuarios monitorizar sus propios datos de salud y generar gráficos de progreso. Además, ofrecerá la posibilidad de que un usuario supervise la salud de otras personas, como familiares o amigos, lo que puede ser especialmente útil para adultos mayores o personas con poca familiaridad con la tecnología.

El objetivo de esta herramienta es mejorar el seguimiento de métricas de salud mediante el uso de dispositivos inteligentes y algoritmos de inteligencia artificial, permitiendo detectar patrones o posibles riesgos. Los datos de salud serán recogidos a través de *smartwatches* y sincronizados mediante Health Connect, almacenándose en Firebase para su posterior análisis en la plataforma web.

La aplicación web permitirá a los usuarios registrarse de manera autónoma, contando con un área personal donde podrán visualizar su progreso a lo largo del tiempo mediante gráficos interactivos. También tendrán la opción de monitorear a terceros, con diferentes niveles de acceso: observador, que solo podrá visualizar datos y recibir alertas, y administrador, que podrá configurar parámetros y gestionar la información.

El sistema de inteligencia artificial analizará los datos recopilados y proporcionará un porcentaje de riesgo o alertas en caso de detectar anomalías, aunque sin valor médico oficial. En caso de detección de patrones preocupantes, se enviarán notificaciones a los supervisores del usuario monitoreado.

Los usuarios podrán acceder a un historial de mediciones, así como eliminar o modificar datos según sus necesidades. La aplicación también permitirá la personalización de perfiles y la gestión de accesos según el rol del usuario.

La solución propuesta se ha desarrollado utilizando Vue.js para el frontend y C# (ASP.NET Core) para la lógica del servidor, mientras que Firebase se encargará del almacenamiento en tiempo real. La inteligencia artificial se implementará en Python, y la aplicación móvil, desarrollada en Kotlin o Flutter, utilizará Health Connect para recolectar datos de los dispositivos.

## Abstract

The development of the proposed tool will consist of creating a web application that will allow users to monitor their own health data and generate progress charts. In addition, it will offer the possibility for a user to supervise the health of other people, such as relatives or friends, which can be especially useful for older adults or people with little familiarity with technology.

The objective of this tool is to improve the tracking of health metrics through the use of smart devices and artificial intelligence algorithms, allowing the detection of patterns or potential risks. Health data will be collected through smartwatches and synchronized via Health Connect, then stored in Firebase for later analysis on the web platform.

The web application will enable users to register autonomously, providing them with a personal area where they can view their progress over time through interactive charts. They will also have the option to monitor third parties, with different access levels: observer, who can only view data and receive alerts, and administrator, who can configure parameters and manage the information.

The artificial intelligence system will analyze the collected data and provide a risk percentage or alerts in case it detects anomalies, although without official medical validity. In the event of concerning patterns, notifications will be sent to the supervisors of the monitored user.

Users will be able to access a history of measurements, as well as delete or modify data according to their needs. The application will also allow for profile customization and access management based on the user's role.

The proposed solution has been developed using Vue.js for the frontend and C# (ASP.NET Core) for the server logic, while Firebase will handle real-time storage. The artificial intelligence will be implemented in Python, and the mobile application, developed in Kotlin or Flutter, will use Health Connect to collect data from the devices.

## Definiciones y abreviaturas

- **Application Programming Interface (API):** Conjunto de funciones, protocolos y herramientas que permiten la comunicación entre diferentes sistemas de software, facilitando la integración y el intercambio de datos.
- **Backend:** Parte del sistema encargada del procesamiento de la lógica de negocio, la gestión de datos y la comunicación con la base de datos. Generalmente, se implementa en servidores y se accede a través de APIs.
- **Base de datos:** Conjunto estructurado de datos almacenados y gestionados mediante un sistema de gestión de bases de datos (SGBD), permitiendo consultas, actualizaciones y almacenamiento eficiente de información.
- **Cliente:** Dispositivo o aplicación que solicita servicios a un servidor, generalmente a través de una red, para obtener o enviar información. Puede ser una aplicación web, móvil o de escritorio.
- **Clean Architecture:** Es un enfoque de diseño de software que organiza el código en capas independientes, separando la lógica de negocio de los detalles de implementación. Esto facilita el mantenimiento y la escalabilidad del sistema.
- **Embedding:** Representación matemática de datos en un espacio vectorial de menor dimensión, utilizada en aprendizaje automático y procesamiento del lenguaje natural para mejorar la eficiencia y precisión de los modelos.
- **Entorno Integrado de Desarrollo (IDE):** Aplicación que proporciona herramientas como editores de código, depuradores y compiladores, facilitando el desarrollo, prueba y mantenimiento de software.
- **Firebase:** Es una plataforma de desarrollo de aplicaciones de Google que ofrece servicios en la nube, como bases de datos en tiempo real, almacenamiento, autenticación y análisis, simplificando la creación y gestión de aplicaciones web y móviles.
- **Framework:** Conjunto de bibliotecas, herramientas y convenciones que proporcionan una estructura predefinida para el desarrollo de software, agilizando la implementación y el mantenimiento de aplicaciones.
- **Frontend:** Parte del sistema con la que interactúa el usuario, generalmente ejecutada en el navegador o en una aplicación. Se encarga de la presentación de la información y la comunicación con el servidor.
- **HealthConnect:** Es una plataforma que centraliza datos de salud y bienestar, permitiendo a las aplicaciones recopilar y compartir información de diversos dispositivos y servicios de manera segura y unificada.
- **Inteligencia Artificial (IA):** Área de la informática que desarrolla algoritmos y modelos capaces de simular el comportamiento humano en tareas como el

reconocimiento de patrones, la toma de decisiones y el procesamiento del lenguaje natural.

- **Patrón de diseño:** Solución general reutilizable para problemas comunes en el desarrollo de software, proporcionando estructuras y prácticas recomendadas para mejorar la mantenibilidad y escalabilidad del código.
- **Servidor:** Máquina o sistema de software que gestiona y proporciona servicios a clientes en una red, procesando solicitudes y enviando respuestas a través de protocolos de comunicación como HTTP o TCP/IP.
- **SVM (Support Vector Machine o Máquina de Vectores de Soporte):** Algoritmo de aprendizaje supervisado que se utiliza tanto para clasificación como para regresión. Funciona encontrando un hiperplano en un espacio de alta dimensionalidad que separa de manera óptima los puntos de datos de diferentes clases, maximizando el margen entre ellos.
- **Wearables:** Dispositivos electrónicos que se llevan como accesorios, implantados en el cuerpo o incorporados en la ropa. Están equipados con sensores que recopilan datos sobre la actividad del usuario y su entorno, permitiendo monitorizar la salud, el estado físico y facilitar la comunicación.
- **VUE (Vue.js):** Framework progresivo de JavaScript de código abierto que se utiliza para construir interfaces de usuario y aplicaciones de una sola página (Single-Page Applications). Se centra en la capa de vista y es conocido por su facilidad de integración con otras bibliotecas y proyectos existentes.

## Agradecimientos

Me gustaría agradecer en primer lugar a mi familia que ha estado aguantándome todos los días cuando les preguntaba si les gustaba alguna cosa o la veían fácil, de usar, incluso sin entender mucho de tecnología ellos me ponían su mejor cara para que siguiera adelante y sacara tiempo de donde no lo había. Asimismo, quiero expresar mi sincero agradecimiento a todas las personas de GRIAL que han contribuido a mi formación, destacando especialmente a Iván y Alicia. A Iván, por su atención y por siempre buscar ideas innovadoras para mejorar mi proyecto; y a Alicia, por brindarme la oportunidad en uno de los momentos más estresantes de mi carrera como estudiante.

Finalmente, agradezco profundamente a mi novia, quien me ha ofrecido una valiosa perspectiva médica y se ha comprometido a consultar a sus profesores para enriquecer aún más este TFG.

# ÍNDICE CONTENIDO

Resumen .....	3
Abstract.....	4
Definiciones y abreviaturas .....	5
Agradecimientos .....	7
ÍNDICE CONTENIDO .....	8
ÍNDICE DE FIGURAS .....	11
ÍNDICE DE TABLAS .....	13
1    Introducción .....	14
2    Objetivos .....	16
2.1    Objetivos funcionales .....	16
2.2    Objetivos personales .....	19
3    Antecedentes .....	21
4    Descripción de la situación actual.....	23
4.1    Análisis del panorama en la monitorización de la salud cardiovascular y el cuidado asistido .....	23
4.2    Herramientas y enfoques existentes en la monitorización cardíaca: Foco en el individuo y limitaciones en el cuidado compartido .....	23
4.3    El Doble Desafío Crítico: Seguridad de Datos y Comunicación Efectiva en la Monitorización Asistida .....	24
4.4    Justificación y Propuesta de Valor Diferencial de BlutTruck: Hacia un Cuidado Cardiovascular Colaborativo y Preventivo .....	24
5    Normas y referencias.....	26
5.1    Métodos .....	26
5.2    Herramientas .....	27
5.2.1    Herramientas propias de la programación y desarrollo de la aplicación.....	27
5.2.2    Herramientas propias de la programación y desarrollo de la aplicación.....	30
5.2.3    Herramientas metodológicas.....	32
5.2.4    Herramientas para la elaboración del TFG.....	32



6	Modelos .....	33
6.1	Preparación y Limpieza de Datos .....	33
6.2	Análisis Exploratorio de Datos .....	34
6.3	Metodología y Construcción de Modelos.....	37
6.4	Resultados y Evaluación .....	37
6.5	Serialización y Despliegue Simulado.....	38
6.6	Análisis Técnico de Modelos Relevantes .....	38
6.6.1	<b>Fundamento de las Máquinas de Soporte Vectorial (SVM)</b> .....	39
6.6.2	<b>Componentes Clave del Modelo SVM</b> .....	39
6.6.3	<b>¿Cómo funciona el SVM para Regresión (SVR)?</b> .....	40
6.7	Prototipos .....	41
6.7.1	Prototipado de la Aplicación Móvil .....	41
6.7.2	Prototipado de la Página Web .....	42
6.8	Métricas .....	42
7	Requisitos iniciales .....	44
7.1	Diagrama de actores.....	44
7.2	Diagrama de casos de uso .....	45
7.2.1	Aplicación.....	46
7.2.2	Web.....	48
7.2.3	Diagrama de secuencia.....	51
8	Hipótesis y Restricciones / Alcance .....	53
9	Estudio de alternativas y viabilidad .....	56
9.1	Justificación de Elección de Herramientas .....	56
9.2	Viabilidad y Monetización .....	58
10	Descripción de la solución propuesta .....	60
10.1	Aplicación móvil.....	62
	<b>Gestión de Usuarios (Users)</b> .....	62
10.2	Aplicación web.....	72
10.3	Implementación de Componentes Clave .....	86
10.3.1	Puente de Comunicación Nativo (Kotlin-Dart) para Health Connect 86	
10.3.2	Gestión de Autenticación y Autorización en el Servidor .....	88

11	ANÁLISIS DE RIESGOS .....	91
12	ORGANIZACIÓN Y GESTIÓN DEL PROYECTO .....	94
12.1	Organización.....	94
12.1.1	Arquitectura del sistema .....	94
12.1.2	DIAGRAMA DE BASE DE DATOS .....	95
12.1.3	Desarrollo de la solución propuesta.....	97
12.2	Gestión del proyecto.....	99
12.2.1	<b>Calendario de trabajo.....</b>	<b>100</b>
12.2.2	<b>Diagrama de Gantt planificado.....</b>	<b>101</b>
12.2.3	<b>Diagrama de Gantt real .....</b>	<b>104</b>
13	CONCLUSIONES Y TRABAJO A FUTURO .....	108
13.1	Conclusiones.....	108
13.1.1	Logros Clave del Proyecto .....	108
13.1.2	Desafíos Superados y Lecciones Aprendidas.....	109
13.2	Impacto y Futuro de BlutTruck .....	109
14	BIBLIOGRAFÍA .....	110

## ÍNDICE DE FIGURAS

Ilustración 1 Datos CSV	33
Ilustración 2 Gráfica datos	34
Ilustración 3 Matriz correlación	35
Ilustración 4 Visualización por clase1	35
Ilustración 5 Presión vs peso	36
Ilustración 6 Visualización de datos 2	36
Ilustración 7 Resultado Modelo	38
Ilustración 8 Diagrama de actores	45
Ilustración 9 Diagrama de casos de uso	46
Ilustración 10 Ilustración 10 Paquetes de casos de uso 1 (Aplicación)	47
Ilustración 11 Paquetes de casos de uso 2 (Aplicación)	48
Ilustración 12 Paquetes de casos de uso 1 (Web)	49
Ilustración 13 Paquetes de casos de uso 2 (Web)	49
Ilustración 14 Paquetes de casos de uso 3 (Web)	50
Ilustración 15 Paquetes de casos de uso 4 (Web)	50
Ilustración 16 Sincronización Datos de Salud(HealthConnect)	51
Ilustración 17 Iniciar Sincronización Automática Datos de Salud	52
Ilustración 18 Estructura BlutTruck	60
Ilustración 19 Logo	62
Ilustración 20 Inicio Aplicación	63
Ilustración 21 Inicio Sesión App	64
Ilustración 22 Regístrate App	65
Ilustración 23 Inicio App	66
Ilustración 24 Perfil APP	67
Ilustración 25 Control de Servicio	68
Ilustración 26 Conexiones App	69
Ilustración 27 Dialog Conexiones App	70
Ilustración 28 QR Conexiones	71
Ilustración 29 Escaner QR	72
Ilustración 30 Landing Parte 1	72
Ilustración 31 Landing Parte 2	73
Ilustración 32 Iniciar Sesión Web	74
Ilustración 33 Registrarse Web	75
Ilustración 34 Estadísticas Usuario	76
Ilustración 35 Gráficas Usuario	77
Ilustración 36 Interfaz Gestion de Conexiones "Redes"	77
Ilustración 37 Panel Estadísticas Persona Monitoreada	78

Ilustración 38 Estadísticas Admin	79
Ilustración 39 Perfil Persona Monitorizada	79
Ilustración 40 Interfaz Predicción	80
Ilustración 41 Interfaz Perfil	81
Ilustración 42 Dialog Más opciones	82
Ilustración 43 Seleccionar Datos	83
Ilustración 44 Interfaz Guardar datos	84
Ilustración 45 Interfaz Quiz	85
Ilustración 46 Interfaz Guía	86
Ilustración 47 Código túnel parte1	87
Ilustración 48 Código Túnel parte 2	87
Ilustración 49 Json Config .NET	89
Ilustración 50 Conexion Firebase Admin	90
Ilustración 51 Imagen DAFO imagen de dynamicgc	91
Ilustración 52 Capas Servidor	94
Ilustración 53 Estructura sistema	95
Ilustración 54 Estructura base de datos	97
Ilustración 55 Calendario Project	100
Ilustración 56 Horas Laborables	101
Ilustración 57 Diagrama de Grantt planificado	101
Ilustración 58 Diagrama de Grantt planificado Hito1	102
Ilustración 59 Diagrama de Grantt planificado Hito 2	103
Ilustración 60 Diagrama de Grantt planificado Hito 3	103
Ilustración 61 Diagrama de Grantt planificado Hito 4	104
Ilustración 62 Diagrama de Grantt planificado Hito 5	104
Ilustración 63 Diagrama de Grantt real	105
Ilustración 64 Diagrama de Grantt real Hito 1	105
Ilustración 65 Diagrama de Grantt real Hito 2	106
Ilustración 66 Diagrama de Grantt real Hito 3	106
Ilustración 67 Diagrama de Grantt real Hito 4	106
Ilustración 68 Diagrama de Grantt real Hito 5	107

# ÍNDICE DE TABLAS

Tabla 1 Dart	27
Tabla 2 Flutter	27
Tabla 3 Kotlin	28
Tabla 4 C#	28
Tabla 5 .Net8	28
Tabla 6 Vue	28
Tabla 7 PrimeVue	28
Tabla 8 SakaiVue	29
Tabla 9 Tailwind CSS	29
Tabla 10 Pinia	29
Tabla 11 Health Connect	29
Tabla 12 Python	30
Tabla 13 Visual Studio Code	30
Tabla 14 Visual Studio	30
Tabla 15 Android Studio	30
Tabla 16 Git	31
Tabla 17 GitHub	31
Tabla 18 Firebase	31
Tabla 19 Anaconda	31
Tabla 20 Jupyter Notebook	31
Tabla 21 Microsoft Project	32
Tabla 22 Visusal Paradigm	32

# 1 Introducción

En el ámbito de la salud, la atención a los ataques cardíacos se ha convertido en una prioridad fundamental para la población general. El notable aumento en la incidencia de estos eventos ha puesto de manifiesto la imperiosa necesidad de educar y concienciar a la sociedad sobre la importancia de reconocer los síntomas y actuar de manera inmediata, lo que ha impulsado el desarrollo de programas de prevención y educación sanitaria.

Además, las técnicas actuales para monitorizar la salud cardíaca presentan desafíos significativos en términos de coste y accesibilidad. La implantación de dispositivos como los marcapasos no solo implica intervenciones quirúrgicas de alto riesgo, sino que también representa un gasto considerable para las familias y los sistemas de salud. Cabe destacar que, incluso las personas sin ninguna patología previa pueden verse afectadas por problemas cardíacos debido al elevado nivel de estrés que caracteriza a la sociedad actual. Por ello, se vuelve esencial invertir en el desarrollo de tecnologías preventivas y de seguimiento continuo que sean más asequibles y adaptadas a las necesidades de cada comunidad, garantizando así una atención más efectiva y oportuna.

De aquí surge la idea de desarrollar una solución accesible y fácil de utilizar tanto para monitorizar nuestra propia salud cardíaca como la de otras personas. El objetivo principal ha sido crear una herramienta de bajo coste que permita detectar posibles signos de riesgos de ataques cardíacos de manera sencilla, sin ser un servicio estrictamente médico, sino más bien una ayuda preventiva. Para ello, se ha diseñado una aplicación móvil y una página web que permiten realizar un seguimiento de los síntomas, registrar datos relevantes sobre la salud y monitorear las constantes de las personas, brindando alertas y recomendaciones basadas en los análisis realizados.

Esta herramienta se llevará a cabo como una aplicación básica desarrollada en Flutter, lo que permitirá a los usuarios utilizarla en dispositivos móviles de diversas plataformas. Los datos recogidos por Health Connect serán enviados a nuestra base de datos, garantizando un acceso seguro y eficiente. La plataforma contará con una página web creada en VUE, que proporcionará una interfaz accesible y fácil de usar. El servidor estará desarrollado en C#, lo que ofrecerá una solución robusta y escalable. Además, implementaremos inteligencia artificial en Python para un análisis más preciso y dinámico de los datos recopilados, mejorando la prevención y monitoreo de ataques cardíacos.

También ha sido necesario el uso de Firebase, concretamente mediante la

implementación de la base de datos Realtime Database para almacenar y sincronizar la información de los usuarios en tiempo real, junto con Firebase Storage para gestionar las imágenes de los mismos. Además, se ha integrado Firebase Authentication, lo que permite llevar a cabo un registro y autenticación seguros. Se ha optado por Firebase debido a su alta escalabilidad, su integración fluida con las demás herramientas y su eficacia en el manejo de datos en entornos de desarrollo móvil y web.

Esta memoria documenta de forma integral el desarrollo del proyecto de una aplicación destinada a la monitorización y detección de ataques cardíacos. En ella se detallan los objetivos perseguidos con la implementación de esta solución, así como los antecedentes que motivaron su creación. También se ofrece una descripción del contexto actual en torno al monitoreo de la salud cardíaca y el uso de tecnologías emergentes aplicadas a este ámbito.

El documento expone los métodos empleados durante el proceso de desarrollo, incluyendo las herramientas y tecnologías seleccionadas, así como la integración de inteligencia artificial para optimizar la detección de eventos cardíacos. Se aborda además el uso de prototipos en las distintas fases de diseño y prueba de la aplicación.

Se incluyen las métricas utilizadas para evaluar la efectividad del sistema, los requisitos iniciales del proyecto, las hipótesis planteadas y las restricciones consideradas. Asimismo, se delimita el alcance del proyecto y se realiza un estudio comparativo de alternativas, valorando la viabilidad técnica y operativa de la solución desarrollada.

La memoria presenta una descripción detallada de la propuesta final, acompañada de un análisis de riesgos asociados y una revisión de la organización y gestión llevadas a cabo durante todo el proceso. Finalmente, se exponen las conclusiones obtenidas y se plantea una proyección del trabajo futuro para continuar mejorando y ampliando las capacidades del sistema.

## 2 Objetivos

Para el desarrollo de la aplicación de monitorización y detección de riesgos cardíacos, BlutTruck, se han establecido una serie de objetivos que guían su diseño, implementación y futura evaluación. Estos se dividen en objetivos funcionales, que describen las capacidades y el comportamiento esperado del sistema, y objetivos personales, que reflejan el aprendizaje y desarrollo de competencias adquiridas durante la realización del proyecto.

### 2.1 Objetivos funcionales

El propósito fundamental de BlutTruck es desarrollar una solución tecnológica integral, accesible y segura para la monitorización proactiva de la salud cardíaca. Esta solución, materializada en una aplicación móvil (desarrollada en Flutter) y una plataforma web (desarrollada en Vue), utilizará datos de dispositivos *wearables* (a través de Health Connect) y la inteligencia artificial para la detección temprana de posibles anomalías o eventos cardíacos, proporcionando alertas y recomendaciones personalizadas con el fin de prevenir situaciones de riesgo y facilitar una posible intervención temprana.

Los principales objetivos funcionales del sistema son:

- **Seguridad Integral, Privacidad de Datos y Cumplimiento Normativo.** El sistema deberá manejar correctamente los datos de salud sensibles garantizando su seguridad, la privacidad de los usuarios y el cumplimiento de la legislación vigente. En particular:
  - Se asegurará la conformidad con la Ley Orgánica de Protección de Datos y Garantía de Derechos Digitales (LOPDGDD) y la Ley de Autonomía del Paciente.
  - La recopilación de datos a través de Health Connect y su posterior tratamiento se realizará con el consentimiento explícito del usuario.
  - Se garantizará la transmisión segura y encriptada de datos desde la aplicación móvil/web hacia la base de datos en Firebase.
  - Se implementarán medidas de seguridad robustas en Firebase (Realtime Database, Storage, Authentication) para proteger el acceso no autorizado.
  - Los usuarios tendrán control sobre sus datos, incluyendo el ejercicio de sus derechos ARCO (Acceso, Rectificación, Cancelación y Oposición).



- **Gestión de Interfaz de Usuario y Visualización de Datos.** El sistema deberá proporcionar interfaces de usuario intuitivas y accesibles tanto en la aplicación móvil como en la página web, permitiendo una interacción fluida y la correcta visualización de la información. Esto incluye:
  - Ofrecer un área personal para cada usuario donde pueda consultar sus datos de salud cardíaca registrados, el historial de monitorización y las tendencias.
  - Presentar de forma clara y comprensible las alertas y recomendaciones generadas por el sistema de IA.
  - Asegurar una experiencia de usuario consistente y fácil de usar en ambas plataformas.
  - Permitir la configuración de preferencias personales de monitorización y notificaciones.
  - Posibilitar la visualización de gráficas evolutivas (diarias, semanales, mensuales) y la exportación de datos o gráficas para su posible uso por profesionales médicos.
- **Gestión de Usuarios y Autenticación Segura.** El sistema deberá gestionar de manera correcta y segura todos los usuarios que utilicen la aplicación. Esto implica:
  - Permitir el registro de nuevos usuarios a través de un proceso sencillo.
  - Implementar un sistema de autenticación seguro para el inicio de sesión.
  - Permitir a los usuarios gestionar sus datos personales y de perfil, incluyendo la posibilidad de subir y gestionar imágenes de perfil.
  - Asegurar que cada usuario solo pueda acceder a su propia información de salud, excepto en los casos de monitorización delegada debidamente autorizada.
- **Monitorización de Salud Cardíaca y Recopilación de Datos.** El sistema deberá ser capaz de recopilar datos relevantes para la salud cardíaca de los usuarios de manera continua y eficiente. Esto comprende:
  - La integración con Health Connect para obtener datos de constantes vitales (ej. frecuencia cardíaca) medidos por dispositivos *wearables* compatibles.

- Permitir a los usuarios registrar manualmente síntomas o eventos relevantes para su salud cardíaca a través de la aplicación.
- Almacenar de forma segura los datos recopilados en Firebase Realtime Database, asociándolos al perfil de cada usuario.
- **Análisis Inteligente y Detección Temprana de Riesgos.** El sistema deberá ser capaz de analizar los datos de salud cardíaca recopilados para identificar patrones y detectar posibles riesgos de manera temprana. Esto supone:
  - Utilizar modelos de inteligencia artificial (implementados en Python y accesibles vía API) para procesar los datos históricos y en tiempo real del usuario.
  - Identificar anomalías o patrones preocupantes en las constantes vitales y síntomas registrados.
  - Generar alertas tempranas y recomendaciones personalizadas basadas en el análisis, dirigidas a la prevención.
  - Permitir la actualización dinámica de los parámetros de monitorización y de los modelos de IA conforme se disponga de nuevos datos o avances.
- **Monitorización Delegada y Seguimiento de Terceros.** El sistema deberá permitir que un usuario principal (ej. cuidador, familiar) pueda monitorizar la salud cardíaca de otros usuarios (ej. personas dependientes, familiares mayores) con el debido consentimiento y control. Esto implica:
  - Implementar un sistema de invitación y aceptación para que un usuario pueda otorgar permiso a otro para visualizar sus datos de salud cardíaca.
  - Permitir al usuario "cuidador" acceder al panel de datos, gráficas, alertas y recomendaciones del usuario "monitorizado" a través de su propia cuenta.
  - Asegurar una clara separación y gestión de los perfiles, manteniendo la privacidad y seguridad de los datos de cada individuo.
  - Permitir al usuario "monitorizado" revocar el acceso al "cuidador" en cualquier momento.
  - Configurar notificaciones específicas para el "cuidador" sobre eventos relevantes o alertas críticas del usuario "monitorizado".

- **Integración de Componentes y Escalabilidad del Sistema.** El sistema deberá asegurar la correcta integración y comunicación entre sus diversos componentes (aplicación móvil, página web, backend, IA y Firebase) y estar diseñado para ser escalable. Esto incluye:
  - Garantizar una comunicación fluida y eficiente entre el *frontend* (móvil y web) y el servidor.
  - Asegurar que la API de inteligencia artificial pueda ser consumida de manera fiable por el servidor para el análisis de datos.
  - Diseñar el backend en C# siguiendo principios de Clean Architecture para facilitar el mantenimiento, la testabilidad y la escalabilidad.
  - Estructurar la solución de manera que permita futuras ampliaciones de funcionalidad y el manejo de un creciente número de usuarios y volumen de datos.

## 2.2 Objetivos personales

El desarrollo de este Trabajo de Fin de Grado ha representado una oportunidad significativa para consolidar y expandir mis competencias técnicas en múltiples tecnologías y metodologías de vanguardia, esenciales en el desarrollo de software moderno:

- **Profundizar en Arquitecturas de Software Escalables:** Reforzar la comprensión y aplicación práctica de Clean Architecture en C#, con el fin de construir un backend robusto, mantenible y preparado para futuras evoluciones del sistema BlutTruck.
- **Dominio Avanzado de Desarrollo Móvil Multiplataforma:** Potenciar las habilidades en Flutter, especialmente en la configuración avanzada de aplicaciones móviles y la integración eficiente con servicios nativos y externos como Health Connect, superando mis conocimientos previos más limitados en este ámbito.
- **Iniciación y Desarrollo en Inteligencia Artificial Predictiva:** Adentrarme en el diseño e implementación de modelos de inteligencia artificial con Python para el análisis predictivo en el sector salud, un área con un impacto transformador en la detección temprana de patologías.
- **Consolidación de Experiencia en Desarrollo Web Moderno:** Reforzar la experiencia con frameworks de frontend como Vue.js, aplicándolo en la creación de una interfaz web interactiva y funcional para la plataforma BlutTruck.

- **Experiencia Práctica en Ecosistemas Cloud y APIs:** Adquirir una comprensión más profunda y práctica en la integración con servicios de Google Cloud Platform, específicamente Firebase (Authentication, Realtime Database, Storage) y la API de Health Connect, herramientas cruciales para la gestión de datos y la conectividad en soluciones tecnológicas actuales.

Estos objetivos personales no solo han guiado mi proceso de aprendizaje durante el proyecto, sino que también han contribuido directamente a la calidad, robustez y funcionalidad de la solución desarrollada.

### 3 Antecedentes

El desarrollo de esta herramienta se enmarca en el ámbito de la salud digital y la inteligencia artificial aplicada a la monitorización cardíaca. En los últimos años, la integración de la IA en la salud ha transformado la forma en que se previenen y gestionan los problemas cardiovasculares, ofreciendo el potencial necesario para detectar anomalías de forma temprana y personalizar las intervenciones según las necesidades individuales [1]. La tecnología ha permitido la creación de soluciones que facilitan la monitorización continua y en tiempo real mediante dispositivos conectados y plataformas digitales.

El auge de modelos predictivos y el análisis de datos en tiempo real ha hecho posible que tanto instituciones médicas como particulares se beneficien de herramientas innovadoras que optimizan la prevención de eventos cardíacos [2]. Además, la incorporación de tecnologías como los *smartwatch*, han redefinido los límites de la monitorización médica, haciéndola más accesible y eficaz.

Los modelos de *machine learning*, especialmente aquellos basados en algoritmos de regresión, son ampliamente utilizados para predecir eventos continuos o probabilidades de ocurrencia de eventos críticos. En el problema abordado se entrenan diversos modelos de regresión (como KNN, árboles de decisión, Random Forest, SVM, regresión lineal, entre otros) que evalúan su desempeño mediante métricas como el error cuadrático medio (MSE), el error absoluto medio (MAE) y el coeficiente de determinación ( $R^2$ ) (Machine Learning — Modelos de Regresión I, Brayan Buitrago).

Estos modelos permiten capturar relaciones complejas entre las variables de entrada y la variable respuesta, en este caso, un índice o probabilidad de ataque cardíaco. Métodos avanzados de *ensemble* y técnicas de *boosting* (por ejemplo, XGBoost, LightGBM o CatBoost) ayudan a mejorar la precisión y a reducir el riesgo de sobreajuste, aspecto crucial cuando se trabaja con datos clínicos donde cada error puede tener consecuencias significativas.

La metodología consiste en dividir el conjunto de datos en particiones de entrenamiento y prueba para asegurar que el modelo generalice correctamente a datos no vistos. La función implementada entrena el modelo, evalúa su rendimiento, lo guarda y luego lo recarga para realizar predicciones de ejemplo. Este proceso garantiza una adecuada validación interna y facilita la implementación del modelo en entornos reales, permitiendo la actualización continua de las predicciones conforme se disponga de nuevos datos.

Un sistema predictivo para ataques cardíacos puede transformar la práctica clínica al:

- Detectar de forma temprana a pacientes de alto riesgo, permitiendo intervenciones preventivas.
- Optimizar la asignación de recursos hospitalarios, focalizando la atención en aquellos casos con mayor probabilidad de un evento adverso.
- Personalizar los tratamientos y recomendaciones basadas en el perfil individual de cada paciente.

Estas aplicaciones no solo contribuyen a reducir la mortalidad asociada a eventos cardiovasculares, sino que también pueden mejorar la calidad de vida de los pacientes al posibilitar un manejo más efectivo y preventivo de su salud.

El desarrollo de esta herramienta se ha realizado en el grupo de investigación GRIAL (<https://grial.usal.es>) de la Universidad de Salamanca, donde he llevado a cabo mis prácticas y recibido formación para mejorar mi Trabajo de Fin de Grado. GRIAL es un equipo multidisciplinar que integra investigadores de diversas áreas, como la informática, la educación y las humanidades.

## 4 Descripción de la situación actual

### 4.1 Análisis del panorama en la monitorización de la salud cardiovascular y el cuidado asistido

El análisis del entorno actual es un proceso indispensable para evaluar las condiciones y soluciones existentes en el mercado relevantes para un producto como BlutTruck. Este se enfoca en la monitorización de la salud cardiovascular, un área de creciente preocupación debido al aumento en la incidencia de eventos cardíacos, tal como se destaca en la introducción de este proyecto. Identificar las fortalezas de las herramientas actuales, sus limitaciones –particularmente en términos de coste, accesibilidad y soporte a la monitorización de terceros– y las oportunidades de mejora es fundamental. El objetivo es asegurar que la aplicación a desarrollar esté alineada con las necesidades y expectativas de sus usuarios finales, que incluyen tanto a individuos interesados en su propia salud como a aquellos que desean supervisar el bienestar de otras personas (familiares, personas a cargo). Asimismo, se busca abordar eficazmente los desafíos del sector, especialmente en cuanto a usabilidad, personalización, la crucial seguridad de los datos y la provisión de alertas efectivas a cuidadores.

Para llevar a cabo este análisis, se han examinado diversas aplicaciones y plataformas que ofrecen funcionalidades relacionadas con el seguimiento cardíaco, la inteligencia artificial aplicada a la salud y la gestión de datos de *wearables*, considerando su idoneidad para un modelo de cuidado preventivo y asistido.

### 4.2 Herramientas y enfoques existentes en la monitorización cardíaca: Foco en el individuo y limitaciones en el cuidado compartido

En el ámbito de la monitorización de la salud cardiovascular, diversas soluciones han emergido, cada una con sus particularidades. Si bien han supuesto avances, muchas se centran predominantemente en el usuario individual, con funcionalidades limitadas para la monitorización activa por parte de terceros o la notificación a cuidadores:

- **Qardio:** Esta empresa es conocida por ofrecer productos que describe como "revolucionarios" para el monitoreo de la salud cardíaca, destacándose por su enfoque en el diseño y la tecnología de sus dispositivos para el uso personal.
- **Cardiogram:** Se trata de una aplicación que utiliza datos provenientes de dispositivos wearables para aplicar modelos predictivos. Su objetivo es la identificación de anomalías en el ritmo cardíaco del portador del dispositivo.

- **AliveCor:** Esta solución combina un dispositivo de electrocardiograma (ECG) portátil con una aplicación móvil. La inteligencia artificial juega un papel clave en su sistema, empleándose para detectar arritmias y otros problemas cardíacos en el usuario directo.

Si bien herramientas como las mencionadas se aproximan a algunos de los objetivos técnicos de BlutTruck (uso de wearables, IA para detección), un análisis más profundo revela que la **gestión de la monitorización de terceros y la notificación proactiva a cuidadores designados** no suele ser su principal propuesta de valor o presentan limitaciones significativas en este aspecto. La necesidad de soluciones de bajo coste y fácil acceso que permitan no solo la automonitorización sino también la supervisión remota y preventiva de seres queridos es una brecha importante, considerando los desafíos de coste y accesibilidad de métodos tradicionales mencionados en la introducción.

#### **4.3 El Doble Desafío Crítico: Seguridad de Datos y Comunicación Efectiva en la Monitorización Asistida**

Plataformas que envían información sensible a servidores externos sin las máximas garantías, o que no facilitan un sistema claro y seguro para el consentimiento y acceso por parte de cuidadores, presentan riesgos. En contraste, soluciones como **Health Connect de Android** (como plataforma de agregación de datos que BlutTruck utilizará) promueven un almacenamiento más controlado de la información, inicialmente en el dispositivo móvil, lo que sienta una base para una transferencia segura a la base de datos de BlutTruck. Este modelo es vital para construir la confianza necesaria en una aplicación que, como BlutTruck, no solo maneja datos personales sino que también los comparte selectivamente con cuidadores autorizados para la emisión de alertas. La integridad y la seguridad en esta comunicación son, por tanto, primordiales.

#### **4.4 Justificación y Propuesta de Valor Diferencial de BlutTruck: Hacia un Cuidado Cardiovascular Colaborativo y Preventivo**

El continuo avance de las tecnologías móviles, los sensores *wearables* y los modelos de inteligencia artificial en el sector de la salud ha estimulado significativamente la investigación y el desarrollo en la monitorización cardíaca. Como se expone en los objetivos de este proyecto, existe una clara necesidad de herramientas que faciliten una detección temprana y un seguimiento accesible. Sin embargo, más allá de la tecnología, persiste el reto de integrar soluciones que combinen:

- **Usabilidad:** Facilidad de uso tanto para la persona monitorizada como para el cuidador.



- **Personalización:** Adaptación a las necesidades individuales y umbrales de alerta.
- **Gestión Segura de Datos:** Garantía de privacidad y confidencialidad en un entorno de datos compartidos.
- **Soporte a la Monitorización Asistida y Alertas a Cuidadores:** Funcionalidad clave que permite a familiares o personas designadas recibir notificaciones ante posibles eventos de riesgo, facilitando una intervención temprana.

En este escenario, **BlutTruck** se proyecta como una solución innovadora que aborda de forma holística estos aspectos, con un énfasis particular en el **cuidado colaborativo**. La propuesta se distingue no solo por su capacidad para la automonitorización y la identificación de posibles anomalías cardíacas mediante IA, sino fundamentalmente por su funcionalidad que permite a **familiares o cuidadores designados ser notificados proactivamente**, tal como se concibió en la idea original del proyecto. Esta dualidad es un diferenciador clave, buscando ser una herramienta de bajo coste que actúe como una ayuda preventiva, no un sustituto médico, pero sí un facilitador de la atención oportuna.

En consecuencia, el desarrollo de BlutTruck se alinea con la tendencia hacia la digitalización y personalización en la salud, y responde a la necesidad imperiosa de herramientas accesibles que potencien tanto el autocuidado como el cuidado asistido por terceros, siempre con la protección y privacidad de los datos como pilares irrenunciables.

## 5 Normas y referencias

### 5.1 Métodos

Para el desarrollo de este proyecto, se ha implementado una metodología ágil, complementada con un enfoque pragmático en la definición de requisitos y el diseño funcional, asegurando que el producto final sea robusto y responda a las necesidades identificadas.

**Metodología Ágil (Scrum Adaptado) [3].** Esta elección ha facilitado una gestión flexible y un desarrollo incremental del proyecto, permitiendo una adaptación eficiente a los cambios y una entrega continua de valor. Concretamente, se ha empleado una versión adaptada de la metodología Scrum, que ha posibilitado el desarrollo del proyecto mediante incrementos, tras una definición inicial de los requisitos de la aplicación. La adaptación clave consistió en **modificar la dinámica y frecuencia de las reuniones de seguimiento y planificación de sprints**, ya que un intervalo semanal estándar resultaba insuficiente para revisar los avances de forma significativa y reorientar esfuerzos con la celeridad que el proyecto demandaba. Las fases principales seguidas han sido:

**Evaluación y Definición Inicial:** Se comenzó con la definición de los requisitos generales de la aplicación, estableciendo la visión y los objetivos a alcanzar.

**Planificación Iterativa:** Se definieron tareas y objetivos específicos para cada ciclo de desarrollo o incremento. Esto incluyó:

- El establecimiento del proceso para la recolección de datos a través de dispositivos wearables.
- El diseño y la subsiguiente implementación de los componentes de inteligencia artificial para su aplicación en el ámbito médico.

**Desarrollo Incremental y Revisión:** Durante cada iteración, se desarrollaron partes específicas del proyecto:

- Estructuración y creación de cada uno de los módulos que compondrían la aplicación.
- Elaboración del plan de desarrollo para el backend y el frontend de la página web.
- Integración de la aplicación móvil, poniendo especial énfasis en garantizar la seguridad en la gestión de la base de datos.

**Colaboración y Ajuste Continuo:** Se mantuvo una comunicación para revisar avances, identificar problemas y realizar los ajustes necesarios, optimizando el proceso de desarrollo mediante la retroalimentación constante.

**Implementación y Entrega:** Se implementaron las funcionalidades desarrolladas en cada incremento, avanzando hacia la solución completa.

Este enfoque ágil adaptado ha permitido que el proceso de entrega sea más eficiente y versátil frente a los desafíos y descubrimientos surgidos durante el desarrollo del proyecto, asegurando que la estructuración, creación de módulos y el desarrollo final de la aplicación web y móvil se realizaran de manera coherente y segura.

## 5.2 Herramientas

Este apartado se estructurará en tres secciones, según el tipo de herramientas utilizadas y su aplicación en el desarrollo del TFG: herramientas propias de la implementación, herramientas metodológicas y herramientas utilizadas en la elaboración del TFG.

### 5.2.1 Herramientas propias de la programación y desarrollo de la aplicación

- **Dart:** es un lenguaje de programación optimizado para clientes, desarrollado por Google, utilizado para construir aplicaciones móviles, de escritorio, de backend y web. En este proyecto, se ha utilizado como el lenguaje principal para el desarrollo de la aplicación móvil mediante el *framework* Flutter.

<b>Desarrollador</b>	Google
<b>Versión</b>	3.5.3
<b>URL</b>	<a href="https://dart.dev">https://dart.dev</a>
<b>Licencia</b>	BSD-style license

Tabla 1 Dart

- **Flutter:** es un *kit* de desarrollo de software (SDK) de interfaz de usuario de código abierto creado por Google. Se utiliza para desarrollar aplicaciones compiladas nativamente para móvil (iOS y Android), web y escritorio desde una única base de código escrita en Dart. En este TFG, Flutter ha sido el *framework* elegido para la implementación de la aplicación móvil.

<b>Desarrollador</b>	Google
<b>Versión</b>	3.24.3
<b>URL</b>	<a href="https://flutter.dev">https://flutter.dev</a>
<b>Licencia</b>	BSD-style license

Tabla 2 Flutter

- **Kotlin:** es un lenguaje de programación de tipado estático desarrollado por JetBrains. En este proyecto se utilizó Kotlin para realizar una parte de la configuración específica necesaria para establecer la conexión con Health Connect en la plataforma Android.

<b>Desarrollador</b>	JetBrains y Google
<b>Versión</b>	2.1.20
<b>URL</b>	<a href="https://kotlinlang.org/">https://kotlinlang.org/</a>
<b>Licencia</b>	Apache License 2.0

Tabla 3 Kotlin

- **C#:** es un lenguaje de programación moderno, orientado a objetos, desarrollado por Microsoft. En este proyecto, se optó por C# para el desarrollo del servidor, en combinación con el *framework* .NET.

<b>Desarrollador</b>	Microsoft
<b>Versión</b>	12
<b>URL</b>	<a href="https://docs.microsoft.com/en-us/dotnet/csharp/">https://docs.microsoft.com/en-us/dotnet/csharp/</a>
<b>Licencia</b>	El compilador Roslyn es open source (MIT License)

Tabla 4 C#

- **.NET 8:** es una versión con soporte a largo plazo (LTS) del *framework* de desarrollo de Microsoft. Es una plataforma de código abierto y multiplataforma. En este TFG, se ha utilizado .NET 8 para desarrollar el servidor, garantizando un soporte a largo plazo.

<b>Desarrollador</b>	Microsoft
<b>Versión</b>	.NET 8
<b>URL</b>	<a href="https://dotnet.microsoft.com">https://dotnet.microsoft.com</a>
<b>Licencia</b>	El compilador Roslyn es open source (MIT License)

Tabla 5 .Net8

- **Vue.js:** es un *framework* progresivo de JavaScript para construir interfaces de usuario. Para el *frontend* de la plataforma web de este proyecto se adoptó Vue.js.

<b>Desarrollador</b>	Evan You y el Vue Core Team
<b>Versión</b>	VITE v5.3.5
<b>URL</b>	<a href="https://vuejs.org">https://vuejs.org</a>
<b>Licencia</b>	MIT License

Tabla 6 Vue

- **PrimeVue:** es una biblioteca de componentes de interfaz de usuario para Vue.js. En este proyecto, se utilizó PrimeVue como apoyo a Vue.js para el desarrollo del frontend.

<b>Desarrollador</b>	PrimeTek
<b>URL</b>	<a href="https://primevue.org">https://primevue.org</a>
<b>Licencia</b>	MIT License

Tabla 7 PrimeVue

- **Sakai Vue:** es una plantilla de administración basada en Vue.js y PrimeVue. Se empleó la plantilla Sakai Vue como base para el frontend, la cual viene preinstalada con Tailwind CSS.

<b>Desarrollador</b>	PrimeTek
<b>Versión</b>	sakai-vue@4.2.0 dev
<b>URL</b>	<a href="https://www.primefaces.org/sakai-vue/">https://www.primefaces.org/sakai-vue/</a>
<b>Licencia</b>	MIT License

Tabla 8 SakaiVue

- **Tailwind CSS:** es un framework de CSS *utility-first*. En este proyecto, Tailwind CSS se utilizó a través de su integración en la plantilla Sakai Vue.

<b>Desarrollador</b>	Tailwind Labs
<b>URL</b>	<a href="https://tailwindcss.com">https://tailwindcss.com</a>
<b>Licencia</b>	MIT License

Tabla 9 Tailwind CSS

- **Pinia:** es una biblioteca de gestión de estado para Vue.js. En este proyecto, se integró Pinia para la gestión del estado global de la aplicación *frontend*.

<b>Desarrollador</b>	Eduardo San Martin Morote (Posva) y el Vue Core Team
<b>URL</b>	<a href="https://pinia.vuejs.org">https://pinia.vuejs.org</a>
<b>Licencia</b>	MIT License

Tabla 10 Pinia

- **Health Connect:** es una plataforma y API de Android para sincronizar datos de salud y *fitness*. En este proyecto, Health Connect es la herramienta empleada para acceder y recuperar los datos de salud adquiridos a través de dispositivos *wearables*.

<b>Desarrollador</b>	Google
<b>URL</b>	<a href="https://developer.android.com/health-and-fitness/guides/health-connect">https://developer.android.com/health-and-fitness/guides/health-connect</a>
<b>Licencia</b>	Parte del Android SDK

Tabla 11 Health Connect

- **Python:** es un lenguaje de programación interpretado y de alto nivel. En este proyecto, Python ha sido fundamental para el desarrollo del componente de inteligencia artificial, incluyendo la creación del modelo predictivo y el desarrollo de una API para su integración. Las principales librerías utilizadas en el contexto de la IA han sido: \*
- **Scikit-learn:** Para tareas generales de aprendizaje automático y evaluación de modelos. \*
- **XGBoost, LightGBM, CatBoost:** Para la implementación de algoritmos de Gradient Boosting. \*
- **Joblib:** Para la gestión y serialización (almacenamiento y carga) de los modelos entrenados.

<b>Desarrollador</b>	Python Software Foundation
<b>Versión</b>	Python 3.10.12
<b>URL</b>	<a href="https://www.python.org">https://www.python.org</a>
<b>Licencia</b>	Python Software Foundation License

Tabla 12 Python

### 5.2.2 Herramientas propias de la programación y desarrollo de la aplicación

- **Visual Studio Code (VS Code):** es un editor de código fuente ligero pero potente desarrollado por Microsoft. En este proyecto, el desarrollo del prototipo de la página web se llevó a cabo en Visual Studio Code, aprovechando su familiaridad y el conjunto de extensiones disponibles.

<b>Desarrollador</b>	Microsoft
<b>URL</b>	<a href="https://code.visualstudio.com/">https://code.visualstudio.com/</a>
<b>Licencia</b>	MIT License

Tabla 13 Visual Studio Code

- **Visual Studio:** Es un entorno de desarrollo integrado (IDE) completo y potente desarrollado por Microsoft. Visual Studio ofrece un amplio conjunto de herramientas para todo el ciclo de vida del desarrollo de software, desde la escritura y depuración de código hasta las pruebas y la implementación de aplicaciones en diversas plataformas. Es ampliamente utilizado por desarrolladores individuales y equipos de todos los tamaños para crear aplicaciones de escritorio, web, móviles, en la nube y de juegos.

<b>Desarrollador</b>	Microsoft
<b>URL</b>	<a href="https://visualstudio.microsoft.com/es/">https://visualstudio.microsoft.com/es/</a>
<b>Licencia</b>	Educativa

Tabla 14 Visual Studio

- **Android Studio:** es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones Android. Para la generación de un dispositivo móvil de prueba y complementar las pruebas en dispositivos reales, se utilizó Android Studio.

<b>Desarrollador</b>	Google y JetBrains
<b>URL</b>	<a href="https://developer.android.com/studio">https://developer.android.com/studio</a>
<b>Licencia</b>	Apache License 2.0

Tabla 15 Android Studio

- **Git y Github:**
  - Git: es un sistema de control de versiones distribuido, que facilita el desarrollo de un proyecto entre varios programadores. Se basa en una copia local del proyecto que se sincroniza subiendo y bajando actualizaciones a un servidor git.

- **Github:** es una plataforma para llevar a cabo el control de versiones durante todo el desarrollo de un proyecto. Se almacenan los repositorios de código en la nube para que varios programadores puedan trabajar en un mismo proyecto y ver las ediciones de los demás en tiempo real. Además, se integra perfectamente y cómodamente con Intelligent IDEA, editor de código que se ha usado para el desarrollo.

<b>Desarrollador</b>	Git
<b>URL</b>	<a href="https://git-scm.com/">https://git-scm.com/</a>
<b>Licencia</b>	GNU General Public License v2.0. Código abierto

Tabla 16 Git

<b>Desarrollador</b>	GitHub, Inc.
<b>URL</b>	<a href="https://github.com/">https://github.com/</a>
<b>Licencia</b>	Gratuita

Tabla 17 GitHub

- **Firebase:** es una plataforma de desarrollo de aplicaciones móviles y web de Google. En este proyecto, se utilizó Firebase Authentication para la autenticación de usuarios, Firebase Realtime Database para la gestión de la base de datos en tiempo real, y Firebase Storage para el almacenamiento de imágenes de perfil.

<b>Desarrollador</b>	Google
<b>URL</b>	<a href="https://firebase.google.com">https://firebase.google.com</a>
<b>Licencia</b>	Servicio freemium

Tabla 18 Firebase

- **Anaconda:** es una distribución de los lenguajes Python y R para computación científica. En este proyecto, se utilizó Anaconda para la gestión del entorno de desarrollo de Python, facilitando la instalación y manejo de las librerías necesarias para la IA.

<b>Desarrollador</b>	Anaconda, Inc.
<b>URL</b>	<a href="https://www.anaconda.com">https://www.anaconda.com</a>

Tabla 19 Anaconda

- **Jupyter Notebook:** es una aplicación web de código abierto que permite crear y compartir documentos con código en vivo, ecuaciones, visualizaciones y texto narrativo. En este TFG, se hizo uso de *notebooks* de Jupyter para la creación, experimentación y validación del modelo de inteligencia artificial.

<b>Desarrollador</b>	Project Jupyter
<b>URL</b>	<a href="https://jupyter.org">https://jupyter.org</a>
<b>Licencia</b>	BSD-style license

Tabla 20 Jupyter Notebook

### 5.2.3 Herramientas metodológicas

Para obtener una estimación precisa del esfuerzo requerido y elaborar una planificación temporal detallada, se han empleado dos herramientas clave: EZEstimate y Microsoft Project.

EZEstimate ha facilitado la aproximación de los meses/hombre necesarios, considerando la complejidad y el alcance del proyecto. Esta herramienta, basándose en datos históricos y métricas específicas del desarrollo, ha permitido obtener una valoración realista del esfuerzo, ayudando a definir de forma inicial los recursos humanos indispensables para la ejecución.

Por otro lado, Microsoft Project se ha utilizado para diseñar un diagrama de Gantt que ilustra el proceso de desarrollo. Este diagrama no solo muestra las distintas fases y tareas, sino que también identifica las dependencias y establece hitos clave. Así, se ha logrado una organización eficiente del cronograma, permitiendo un seguimiento riguroso del progreso y la posibilidad de realizar ajustes en tiempo real conforme evolucionan las actividades del proyecto.

La combinación de ambas herramientas ha sido fundamental para la toma de decisiones, optimizando la asignación de recursos y facilitando una gestión integral del proyecto desde su inicio hasta su finalización.

<b>Desarrollador</b>	Microsoft
<b>URL</b>	<a href="https://www.microsoft.com/es-es/microsoft-365/project/project-management-software">https://www.microsoft.com/es-es/microsoft-365/project/project-management-software</a> ]( <a href="https://www.microsoft.com/es-es/microsoft-365/project/project-management-software">https://www.microsoft.com/es-es/microsoft-365/project/project-management-software</a>
<b>Licencia</b>	Educativa, aplicación para <i>Windows</i>

Tabla 21 Microsoft Project

### 5.2.4 Herramientas para la elaboración del TFG

Durante el desarrollo del TFG, se ha empleado la herramienta Visual Paradigm para la elaboración de diversos diagramas, incluyendo diagramas de secuencia, diagramas de casos de uso y diagramas de clases.

Estos diagramas han desempeñado un papel fundamental en la definición de los requisitos del proyecto, proporcionando una base estructurada antes de iniciar su desarrollo. Además, han sido esenciales en la construcción del modelo de dominio de la aplicación, permitiendo visualizar la interacción entre los diferentes componentes del sistema y garantizar una arquitectura bien definida y coherente.

<b>Desarrollador</b>	Visual Paradigm International Ltd.
<b>URL</b>	<a href="https://www.visual-paradigm.com/">https://www.visual-paradigm.com/</a>
<b>Licencia</b>	Licencia Educativa, aplicación para <i>Windows</i>

Tabla 22 Visual Paradigm



## 6 Modelos

Para el desarrollo del proyecto, se eligió el lenguaje de programación Python, dada su robustez y el extenso ecosistema de librerías especializadas en ciencia de datos y aprendizaje automático. Con el fin de gestionar eficazmente las dependencias y garantizar un entorno de desarrollo estable y reproducible, se optó por utilizar la distribución Anaconda. El trabajo se articuló mediante cuadernos de Jupyter, una herramienta que facilita un desarrollo interactivo e iterativo, ideal para la exploración de datos, la experimentación con modelos y la visualización de resultados, permitiendo una documentación clara y secuencial de todo el proceso.

### 6.1 Preparación y Limpieza de Datos

El fundamento de cualquier modelo predictivo es la calidad de los datos. En esta fase, se abordaron las siguientes tareas críticas utilizando la librería **Pandas**:

- **Carga y Unificación:** Se trabajaron múltiples ficheros de datos (heart\_data.csv, datos\_train.csv), estandarizando la nomenclatura de las columnas a un formato coherente en español (ej. age a edad, weight a peso\_kg), como se observa en la Ilustración 1

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol
1	0	18393	2	168	62.0	110	80	1
2	1	20228	1	156	85.0	140	90	3
3	2	18857	1	165	64.0	130	70	3
4	3	17623	2	169	82.0	150	100	1
5	4	17474	1	156	56.0	100	60	1
6	8	21914	1	151	67.0	120	80	2
7	9	22113	1	157	93.0	130	80	3
8	12	22584	2	178	95.0	130	90	3
9	13	17668	1	158	71.0	110	70	1
10	14	19834	1	164	68.0	110	60	1
11	15	22530	1	169	80.0	120	80	1
12	16	18815	2	173	60.0	120	80	1
13	18	14791	2	165	60.0	120	80	1
14	21	19809	1	158	78.0	110	70	1
15	23	14532	2	181	95.0	130	90	1
16	24	16782	2	172	112.0	120	80	1
17	25	21706	1	170	75.0	120	70	1

Ilustración 1 Datos CSV

- **Limpieza de Inconsistencias:** Se aplicaron filtros para eliminar valores atípicos o no realistas en variables como la altura (acm) y el peso (kg).
- **Normalización de Datos Clínicos:** Se acotaron los valores de presión arterial (presion\_sistolica y presion\_diastolica) dentro de rangos clínicamente plausibles para evitar distorsiones en el modelo.
- **Transformación de Variables:** Se convirtió la variable edad, originalmente en días, a años para una mejor interpretabilidad. Además, se transformaron

variables categóricas (ej. 'YES'/'NO' a 1/0) para su uso en algoritmos numéricos.

**Manejo de Columnas Faltantes:** Se prepararon los conjuntos de datos para su posible fusión, creando columnas ausentes y asignándoles valores nulos (None) como paso previo a una estrategia de imputación.

## 6.2 Análisis Exploratorio de Datos

Para comprender la naturaleza de los datos, se emplearon las librerías **Matplotlib** y **Seaborn** para generar una serie de visualizaciones clave:

- **Distribución de Variables:** En la Ilustración 2 se observa cómo se crearon histogramas para analizar la distribución de todas las variables numéricas, identificando sesgos y rangos de valores.

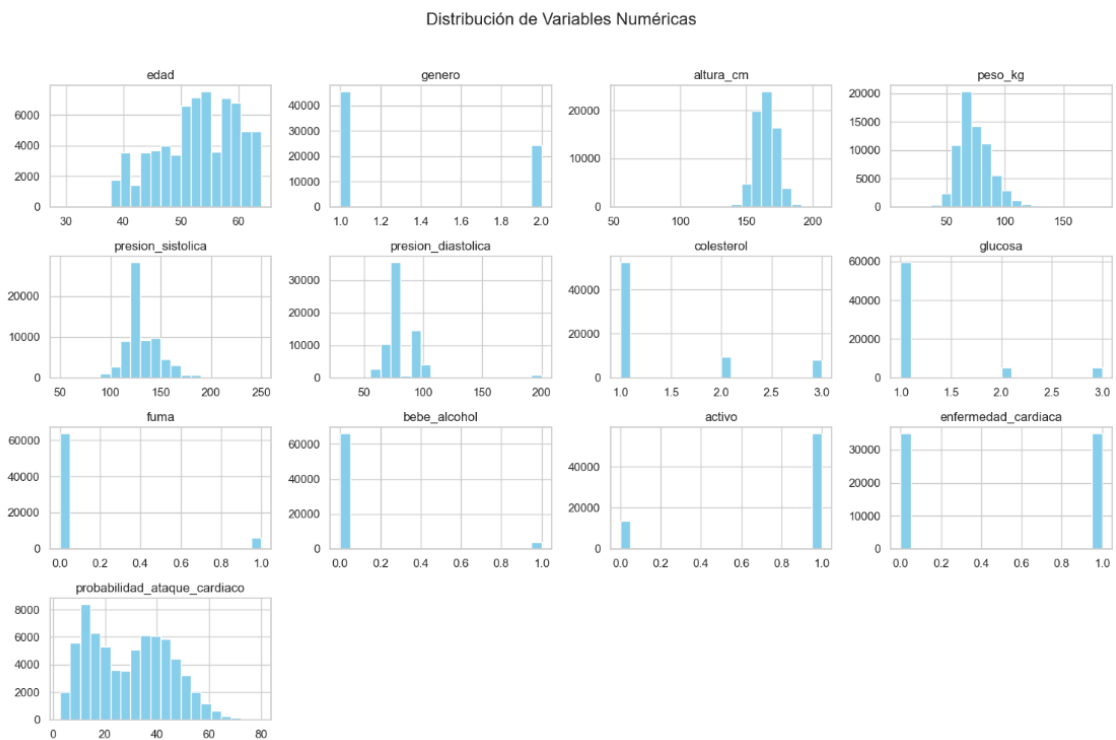


Ilustración 2 Gráfica datos

- **Análisis de Correlación:** Una matriz de correlación de la Ilustración 3 con *heatmap* permitió visualizar la relación lineal entre las variables, destacando la fuerte correlación entre el peso, la altura y las presiones arteriales.

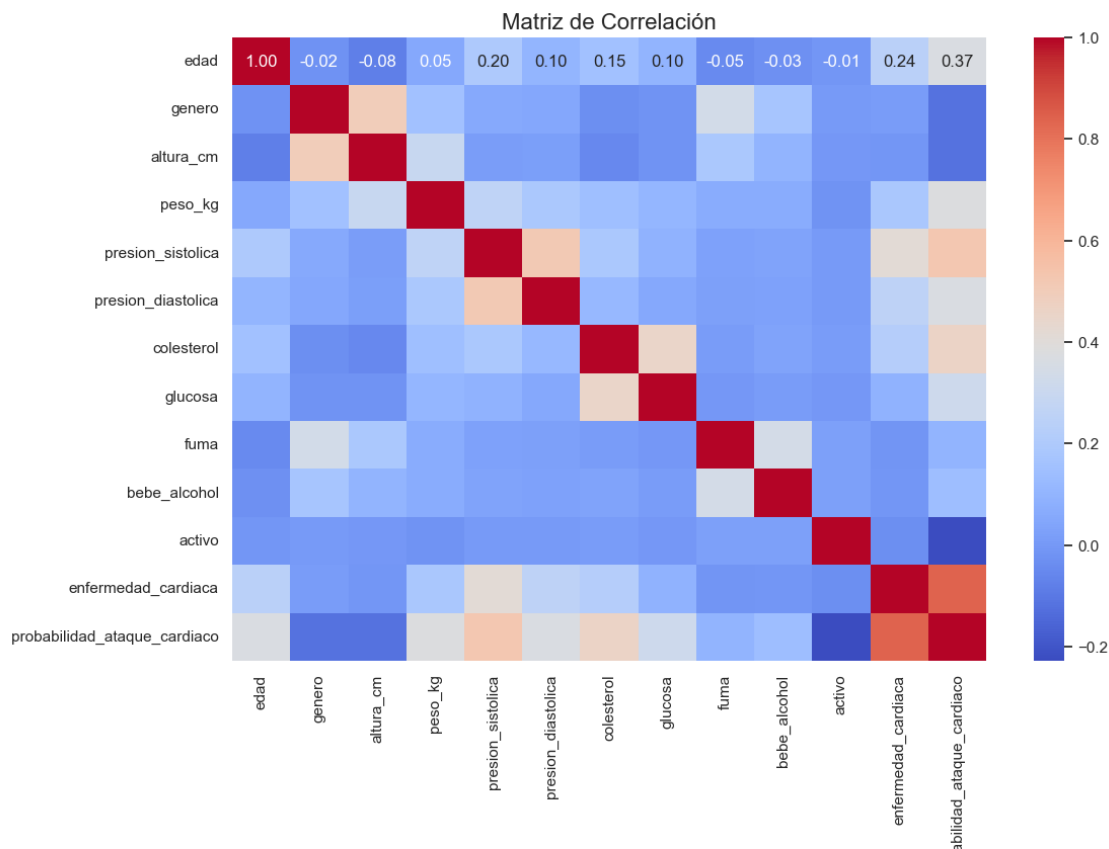


Ilustración 3 Matriz correlación

- **Visualización por Clase:** Se utilizaron gráficos de barras y diagramas de caja, como en Ilustración 4, para comparar la distribución de variables clave en función de la presencia o ausencia de enfermedad cardíaca.

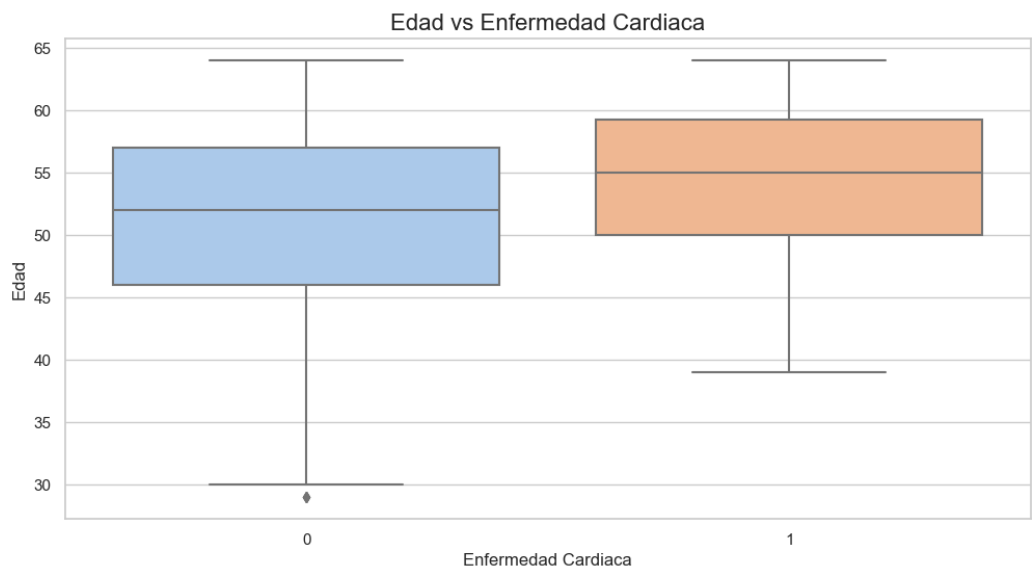


Ilustración 4 Visualización por clase1

- **Relaciones Bivariadas:** Gráficos de dispersión ayudaron a explorar la relación entre pares de variables importantes , segmentando los datos por la variable objetivo dos ejemplos son la Ilustración 5 y la Ilustración 6.

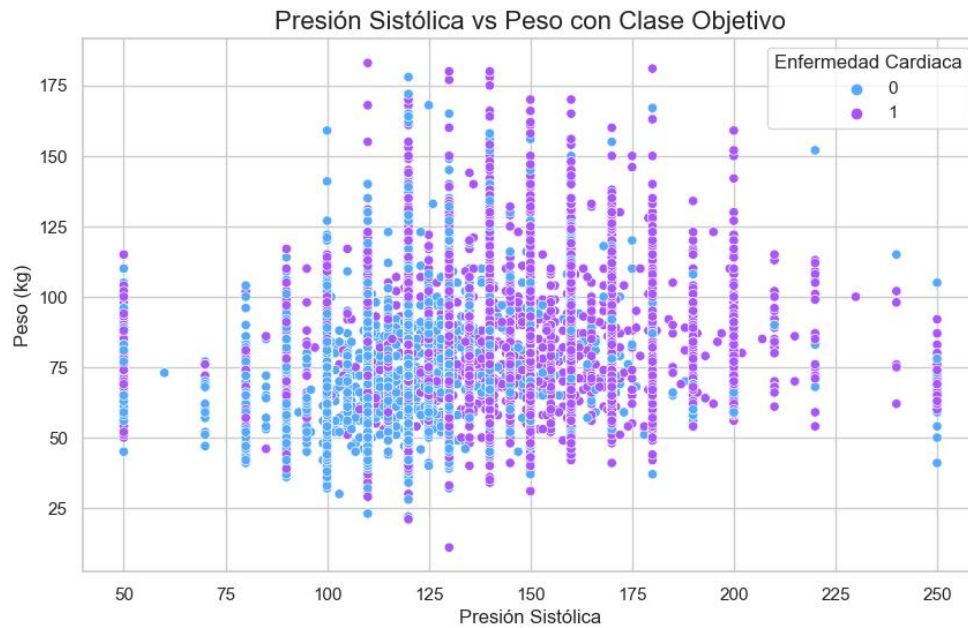


Ilustración 5 Presión vs peso

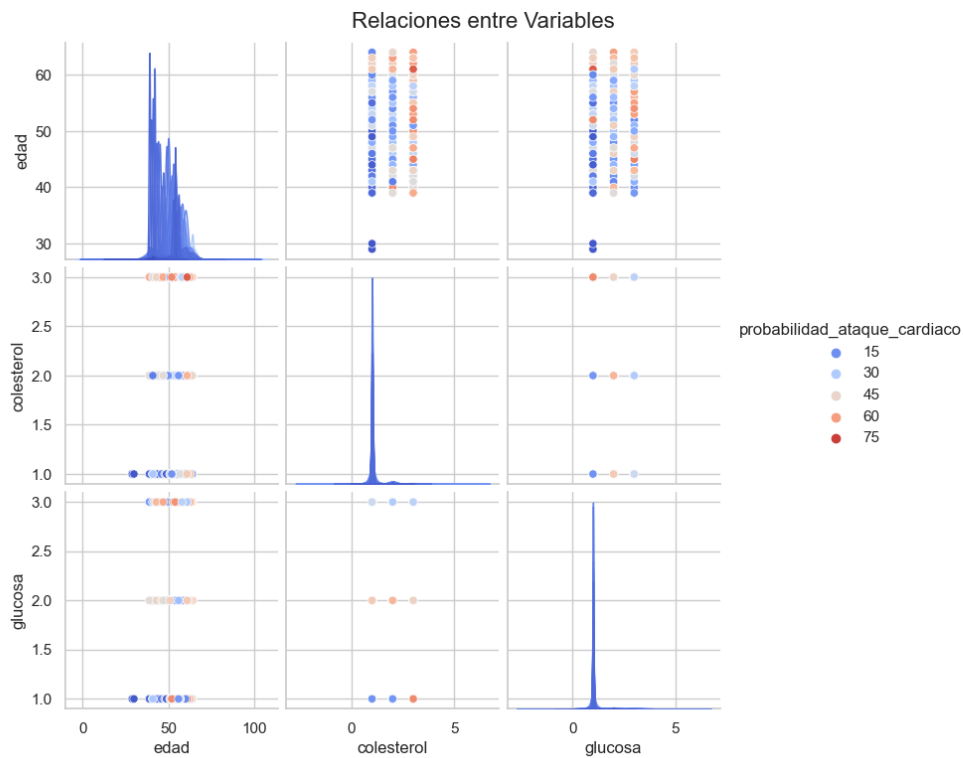


Ilustración 6 Visualización de datos 2

### 6.3 Metodología y Construcción de Modelos

Esta es la fase central del proyecto, donde se aplicaron técnicas de aprendizaje automático con **Scikit-learn**, **XGBoost**, **LightGBM** y **CatBoost**.

- **Preprocesamiento:** Se dividieron los datos en conjuntos de entrenamiento y prueba. A continuación, se aplicó un escalado de características mediante **StandardScaler**, un paso fundamental para normalizar los datos y mejorar el rendimiento de muchos algoritmos.
- **Selección y Entrenamiento Comparativo de Modelos:** Se entrenó y evaluó un amplio espectro de modelos de regresión para encontrar el de mayor poder predictivo. Los modelos probados incluyen:
  - **Modelos Lineales:** **LinearRegression**.
  - **Modelos Basados en Distancia:** **KNeighborsRegressor**.
  - **Máquinas de Soporte Vectorial:** **SVR**.
  - **Modelos de Árbol:** **DecisionTreeRegressor**.
  - **Métodos de Ensamblado (Ensemble):** **RandomForestRegressor**, **GradientBoostingRegressor**, **AdaBoostRegressor**, **ExtraTreesRegressor**.
  - **Modelos de Gradient Boosting Avanzados:** **XGBRegressor**, **LGBMRegressor**, **CatBoostRegressor**.
  - **Redes Neuronales:** **MLPRegressor**.
- **Evaluación:** El rendimiento de cada modelo se midió utilizando métricas estándar de regresión: **Error Cuadrático Medio (MSE)**, **Error Absoluto Medio (MAE)** y el **Coeficiente de Determinación ( $R^2$ )**.

### 6.4 Resultados y Evaluación

Tras la evaluación, se observó un rendimiento sobresaliente en varios de los modelos avanzados, con un  **$R^2$  Score de 1.00** o muy cercano en los siguientes algoritmos, indicando una capacidad de predicción casi perfecta en el conjunto de prueba:

- **Red Neuronal (MLPRegressor):** MSE de 0.01.
- **CatBoost Regressor:** MSE de 0.02.
- **XGBoost Regressor:** MSE de 0.54.

- **Extra Trees Regressor:** MSE de 0.99.

Modelos como **Random Forest** y **SVM** también mostraron un rendimiento excelente ( $R^2$  de 0.99). En contraste, el modelo **AdaBoost** tuvo el rendimiento más bajo ( $R^2$  de 0.91 y MSE de 22.12), siendo descartado como opción viable. Así que después de unas pruebas se pudo observar que el mejor modelo era SVM.

En la Ilustración 7 se puede ver el resultado dado:

```
Evaluación del modelo: svm_model
Mean Squared Error (MSE): 1.46
Mean Absolute Error (MAE): 0.49
R2 Score: 0.99
Modelo svm_model guardado como 'modelos\svm_model.pkl'.

Modelo svm_model cargado con éxito. Ejemplo de predicción:
[52.16174182 22.46237327 36.15634259 50.17511763 11.38132739 16.96214727
12.58215141 36.05619829 18.05798727 22.189523 ]
```

Ilustración 7 Resultado Modelo

## 6.5 Serialización y Despliegue Simulado

Para asegurar la usabilidad futura de los modelos, se implementó un proceso de guardado y carga:

- **Serialización:** Cada uno de los doce modelos entrenados, junto con el objeto *StandardScaler*, se guardó en disco utilizando la librería *joblib*. Esto permite que puedan ser cargados posteriormente sin necesidad de re-entrenamiento.
- **Simulación de API:** Se realizó una prueba cargando uno de los modelos guardados (*svm\_model.pkl*) y utilizándolo para predecir la probabilidad de ataque cardíaco para un nuevo conjunto de datos hipotético, simulando así su integración en una aplicación real.

## 6.6 Análisis Técnico de Modelos Relevantes

Tras la fase de evaluación comparativa, se identificaron varios modelos con un rendimiento predictivo sobresaliente. De entre ellos, se seleccionó el modelo de **Máquina de Soporte Vectorial para Regresión (SVR)** como la implementación final, debido a su excelente balance entre precisión ( $R^2$  Score de 0.99) y robustez teórica.

A continuación, se detalla el fundamento técnico de este algoritmo para comprender su capacidad de modelar relaciones complejas en los datos de salud cardiovascular.

### 6.6.1 Fundamento de las Máquinas de Soporte Vectorial (SVM)

El SVM es un algoritmo de aprendizaje supervisado que, a diferencia de otros modelos, no se basa en probabilidades ni en estructuras de árbol, sino en principios de **geometría y optimización matemática**.

#### El Objetivo Principal: Encontrar el Hiperplano Óptimo

Aunque en este proyecto se usó para regresión (SVR), el concepto fundamental del SVM nace en la clasificación. Para entenderlo de forma sencilla, imaginemos que tenemos que separar dos clases de puntos (por ejemplo, "pacientes con riesgo" y "pacientes sin riesgo").

- El SVM no busca simplemente una línea que separe los dos grupos. Su objetivo es encontrar la **"calle" o "margen" más ancho posible** que mantenga los dos grupos de datos perfectamente separados.
- La línea que se encuentra justo en el centro de esta calle es lo que se conoce como el **hiperplano óptimo de separación**.

Este principio de maximizar el margen es lo que dota al SVM de una gran capacidad de generalización, haciéndolo menos propenso a cometer errores con datos nuevos.

### 6.6.2 Componentes Clave del Modelo SVM

1. **El Hiperplano:** Es el plano de decisión. En un espacio de 2 dimensiones (con 2 variables), es una línea. En 3 dimensiones, es un plano. Con más de 3 dimensiones, como en nuestro caso de 12 variables, se le denomina *hiperplano*.
2. **Los Vectores de Soporte (Support Vectors):** Este es el concepto que da nombre al algoritmo. Los SVM son eficientes porque no necesitan todos los puntos de datos para definir el *hiperplano*.
  - Solo utilizan los puntos más cercanos a la frontera de decisión.
  - Estos puntos críticos se llaman **vectores de soporte** porque son los que "sostienen" o definen la posición del hiperplano. Si moviéramos cualquiera de estos puntos, el hiperplano óptimo cambiaría. Si moviéramos cualquier otro punto (que no sea un vector de soporte), no tendría ningún efecto.
3. **El Truco del Kernel (Kernel Trick) para Datos No Lineales:** En la mayoría de los problemas del mundo real, como el nuestro, los datos no se pueden separar con una línea recta (no son linealmente separables). Aquí es donde el SVM demuestra su mayor potencia.

- **El Problema:** Las relaciones entre la edad, el colesterol, la presión arterial y el riesgo cardíaco no son simples ni lineales.
- **La Solución:** El "**truco del kernel**" es una función matemática que transforma los datos de entrada a un espacio de características de mayor dimensión. La idea es que, en este nuevo espacio de más dimensiones, los datos que antes no se podían separar linealmente, ahora sí puedan serlo.
- **Kernel Utilizado (RBF):** En este proyecto, el SVR de Scikit-learn utiliza por defecto el *kernel* de **Función de Base Radial (RBF)**. Este es un *kernel* muy potente y popular porque puede manejar relaciones extremadamente complejas y no lineales, mapeando los datos a un espacio de dimensiones infinitas.

### 6.6.3 ¿Cómo funciona el SVM para Regresión (SVR)?

El modelo que hemos utilizado es un **SVR (Support Vector Regression)**, que adapta los conceptos anteriores para predecir un valor continuo (la probabilidad de ataque cardíaco).

- **De la "Calle" al "Tubo":** En lugar de encontrar una "calle" que separe dos clases, el SVR intenta encontrar un "tubo" o "corredor" que **contenga la mayor cantidad posible de puntos de datos**.
- **Margen de Tolerancia (Épsilon):** El ancho de este tubo se define por un hiperparámetro llamado **épsilon ( $\epsilon$ )**. A diferencia de otros modelos de regresión que penalizan cualquier error, el SVR no considera un error si el punto de dato real se encuentra dentro de este tubo. La penalización solo ocurre para los puntos que están fuera.
- **Vectores de Soporte en SVR:** Al igual que en clasificación, los vectores de soporte son los puntos que definen el modelo. En este caso, son los puntos que se encuentran **en los bordes o fuera del tubo de épsilon**.

El modelo SVR que hemos implementado funciona encontrando la mejor "línea" de tendencia que logra que la mayoría de nuestros datos de pacientes queden dentro de un margen de error predefinido, basándose en los casos más "difíciles" (los vectores de soporte) para definir su predicción. Gracias al uso del kernel RBF, es capaz de hacerlo incluso cuando las relaciones entre las variables son altamente complejas y no lineales.



## 6.7 Prototipos

La fase de prototipado fue un pilar fundamental en el desarrollo de "BlutTruck", el momento en que las ideas conceptuales se transformaron en soluciones tangibles y evaluables. Siguiendo rigurosamente la metodología de **Diseño Centrado en el Usuario (DCU)**, el objetivo fue posicionar al usuario en el epicentro de todas las decisiones, buscando crear una solución no solo funcional, sino también altamente usable e intuitiva. El proceso se abordó de manera diferenciada para la identidad visual, la aplicación móvil y la plataforma web. Identidad Visual y Prototipado Inicial

El proyecto comenzó con la definición de la identidad visual y la conceptualización del producto.

- **Nombre:** La búsqueda de una denominación memorable, concisa y semánticamente relevante culminó en la elección de "**BlutTruck**". Este nombre es una adaptación fonética del término alemán *Blutdruck* ("presión arterial"), aludiendo directamente a la función principal de la aplicación y creando al mismo tiempo una identidad de marca única.
- **Identidad Visual:** Se definió una paleta de colores estratégica basada en el **azul celeste (#5996FF)** y el **blanco puro (#FFFFFF)**. En el ámbito sanitario, el azul se asocia con la tranquilidad y la seguridad. Se optó por una tonalidad celeste, moderna y vibrante, para proyectar innovación y reducir la posible ansiedad asociada a la monitorización de la salud. El blanco, como lienzo de fondo, maximiza la claridad y la legibilidad. A partir de esta paleta, se diseñó un logotipo minimalista que fusiona dos conceptos clave: una **línea de pulso cardíaco** (símbolo universal de la salud) con el contorno de un **marcador de seguimiento** (simbolizando la monitorización continua y el cuidado asistido). El resultado es un logo que proyecta una imagen tecnológica, clara y de confianza.

Para el desarrollo, se crearon prototipos funcionales iniciales con objetivos técnicos específicos:

### 6.7.1 Prototipado de la Aplicación Móvil

1. **Integración con Firebase:** Se realizó un primer prototipo para conectar la app con Firebase. Al ser una tecnología nueva para el equipo, este paso fue clave para familiarizarse con la autenticación y la base de datos en tiempo real.
2. **Conexión con Health Connect:** Un segundo prototipo se centró en integrar la aplicación con Health Connect para la recogida de datos. Se realizaron pruebas con usuarios que revelaron que, si bien la app

era intuitiva, el proceso de configuración inicial de permisos con Health Connect presentaba cierta complejidad. Se concluyó que esta dificultad era un paso técnico ineludible y externo a la lógica de la aplicación.

### 6.7.2 Prototipado de la Página Web

Para el desarrollo de la página web, se tomó una decisión estratégica: en lugar de utilizar herramientas de diseño visual como Figma, **se optó por crear directamente un prototipo de alta fidelidad utilizando Vue.js**. Esta elección se basó en dos motivos principales:

1. **Evaluación de Viabilidad Técnica:** Permitió evaluar de manera práctica si las funcionalidades de interfaz deseadas podían implementarse con Vue.js, mitigando riesgos técnicos en una fase temprana.
2. **Eficiencia en el Desarrollo:** Este "prototipo en código", aunque utilizaba datos estáticos, sirvió como base para el código final del *frontend*, optimizando significativamente los tiempos de desarrollo.

## 6.8 Métricas

Para la evaluación del diseño, se realizaron **pruebas de usabilidad** sobre los prototipos digitales de alta fidelidad, tanto de la aplicación móvil como de la plataforma web. El objetivo principal era medir la simplicidad, claridad y facilidad de uso de las interfaces propuestas.

- **Metodología:** Se llevaron a cabo sesiones con un grupo reducido de usuarios (familiares y amigos), quienes interactuaron libremente con las funcionalidades clave del sistema.
- **Resultados:** La retroalimentación general fue mayoritariamente positiva. Los participantes destacaron la **simplicidad de la navegación** y la facilidad para comprender las funciones principales. La paleta de colores y la identidad visual también recibieron una valoración favorable.
- **Áreas de Fricción Identificadas:** El único punto de fricción relevante fue el **proceso de configuración inicial y sincronización con Health Connect**. Los usuarios manifestaron que este paso resultaba algo complejo.
- **Conclusión de la Evaluación:** Tras analizar los resultados, se determinó que la dificultad percibida en la configuración de Health Connect es un requisito técnico ineludible y externo a la aplicación "BlutTruck". Dado que el resto de la navegación y la interacción fueron validadas positivamente, se concluyó que los prototipos eran suficientemente robustos y usables. Por

tanto, se procedió a la fase de desarrollo completo sin realizar modificaciones significativas en la estructura visual o de navegación.

## 7 Requisitos iniciales

### 7.1 Diagrama de actores

Una manera eficaz de visualizar los actores que forman parte del sistema es mediante un diagrama de actores, el diagrama de actores creados se observa en la Ilustración 8.

1. **Anónimo.** Usuario no autenticado que interactúa con las partes públicas de la aplicación. Puede visualizar información general sobre BlutTruck (funcionalidades, beneficios) y acceder a las opciones de inicio de sesión o registro en la plataforma.
2. **Usuario Registrado.** Es la base para cualquier usuario autenticado en el sistema. Se encarga de la gestión de su perfil personal (modificación de datos como nombre, correo, contraseña), configuración de preferencias de la cuenta y tiene la capacidad de solicitar la descarga de todos sus datos almacenados o la eliminación completa de su cuenta y información asociada de la aplicación.
3. **Usuario de Monitoreo Personal.** Además de todas las funcionalidades del Usuario Registrado, este actor es el foco principal de la monitorización. Sincroniza datos de salud desde Health Connect, registra manualmente síntomas o eventos relevantes, visualiza su panel personal con datos históricos, gráficas de evolución, recibe y consulta las alertas y recomendaciones personalizadas generadas por el Motor de Análisis y Alertas. Es quien consiente (o revoca el consentimiento) para ser monitorizado por un Cuidador.
4. **Cuidador.** Junto con las funcionalidades del Usuario Registrado, este actor tiene la capacidad de solicitar y gestionar el permiso para monitorizar la salud cardíaca de uno o más Usuarios de Monitoreo Personal. Una vez autorizado, puede acceder a una vista del panel de datos, gráficas, alertas y recomendaciones de los UMP a su cargo, facilitando así un seguimiento preventivo y permitiéndole actuar ante notificaciones importantes.
5. **Motor de Análisis y Alertas.** Se trata de un actor automatizado del sistema, no humano. Su función principal es procesar continuamente los datos de salud recopilados de los Usuarios de Monitoreo Personal. Aplica los modelos de inteligencia artificial para analizar patrones, detectar posibles riesgos cardíacos de manera temprana, generar recomendaciones personalizadas y disparar las alertas correspondientes que se enviarán tanto al UMP como a sus Cuidadores designados.
6. **Administrador del Sistema.** Posee el nivel más alto de acceso y control sobre la plataforma BlutTruck. Sus responsabilidades incluyen la gestión integral de todas las cuentas de usuario (altas, bajas, modificación de perfiles y roles, resolución de problemas de acceso), la configuración de parámetros generales

del sistema, la supervisión del rendimiento y la salud de la aplicación (incluyendo el Motor de Análisis y Alertas), la gestión de la API de IA (si requiere intervención manual para actualizaciones o ajustes) y la implementación de nuevas versiones o mejoras en la plataforma.

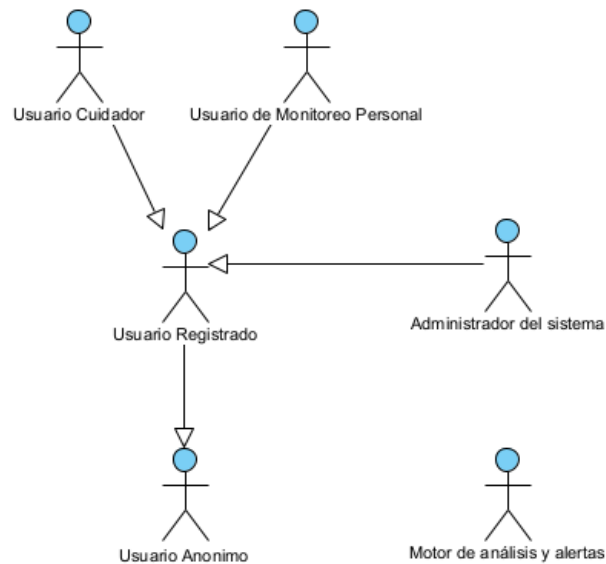


Ilustración 8 Diagrama de actores

## 7.2 Diagrama de casos de uso

El diagrama de casos de uso constituye una herramienta de modelado visual que describe de forma gráfica las funcionalidades de un sistema de software, enfocándose en la interacción desde el punto de vista del usuario [4]. Permite identificar y representar claramente los siguientes elementos clave:

- **Casos de Uso:** Son descripciones de secuencias de acciones que el sistema ejecuta para producir un resultado observable y de valor para un actor particular. Cada caso de uso representa una funcionalidad específica del sistema.
- **Actores:** Definen los roles que desempeñan los usuarios, otros sistemas o entidades externas cuando interactúan con el sistema. Un actor participa en la ejecución de uno o más casos de uso para alcanzar sus metas.
- **Interrelaciones (o Relaciones):** Representan las conexiones y dependencias entre los elementos del diagrama. Primordialmente, muestran qué actores participan en qué casos de uso, pero también

pueden ilustrar relaciones entre los propios casos de uso (como las de inclusión «include» o extensión «extend») o entre actores (generalización).

El diagrama completo se presenta en la Ilustración 9. Para facilitar su análisis y mejorar la claridad de su visualización, este se ha desglosado en la Ilustración 10, la Ilustración 11, la Ilustración 12, la Ilustración 13, la Ilustración 14, la Ilustración 15, donde se exponen sus diferentes secciones de manera individualizada.

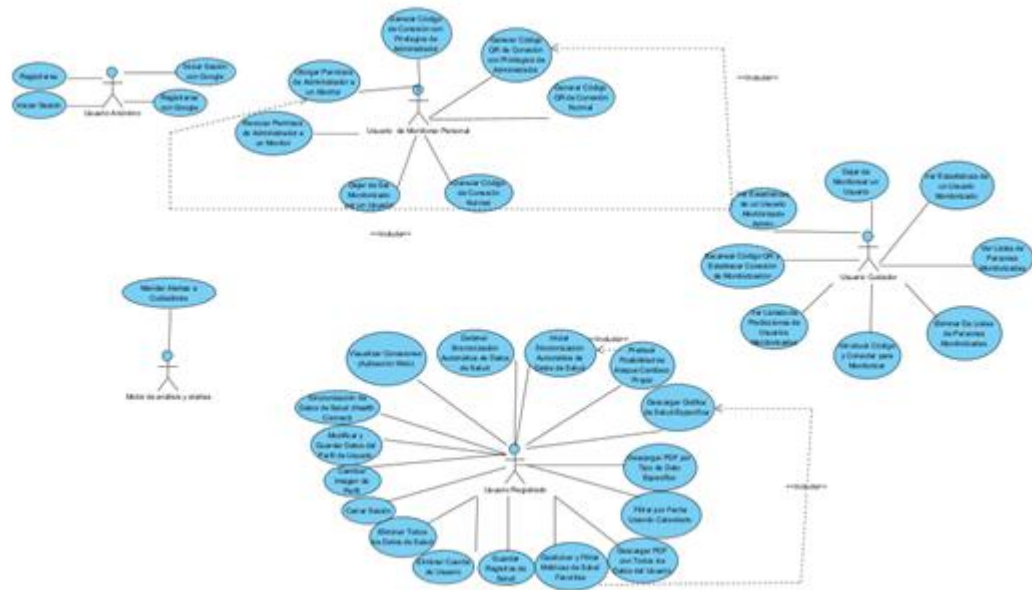


Ilustración 9 Diagrama de casos de uso

### 7.2.1 Aplicación

La aplicación móvil desarrollada se estructura en cinco módulos principales, cada uno con responsabilidades y funcionalidades específicas:

1. **Módulo de Autenticación y Gestión de Acceso.** Este módulo gestiona los procesos de identificación y registro de usuarios en el sistema. Ofrece la funcionalidad de inicio de sesión y creación de nuevas cuentas, permitiendo la autenticación tanto a través de credenciales de Google como mediante el uso de una dirección de correo electrónico y contraseña.
2. **Módulo de Integración con Health Connect.** Responsable de la interacción con la plataforma Health Connect de Android. Permite al usuario configurar los permisos necesarios y establecer la conexión para la posterior extracción, procesamiento de los datos de salud y actividad física almacenados en dicha plataforma.
3. **Módulo de Gestión de Perfil de Usuario.** Proporciona al usuario registrado las herramientas necesarias para la administración de su

información personal dentro de la aplicación. Incluye funcionalidades para la modificación de datos del perfil, la actualización de la fotografía asociada a la cuenta y el cierre de la sesión activa en el dispositivo.

4. **Módulo de Sincronización en Primer Plano (Foreground).** Este módulo habilita la configuración de un servicio de monitorización continua que opera en primer plano. Su finalidad es permitir que la aplicación realice un seguimiento o recolección de datos de manera persistente, según las preferencias establecidas por el usuario, incluso cuando la interfaz principal de la aplicación no esté visible.
5. **Módulo de Conexiones Sociales.** Facilita la interacción y el establecimiento de vínculos entre los usuarios de la plataforma. Permite a un usuario escanear o crear QR para facilitar la conexión, y visualizar una lista de las conexiones ya establecidas.

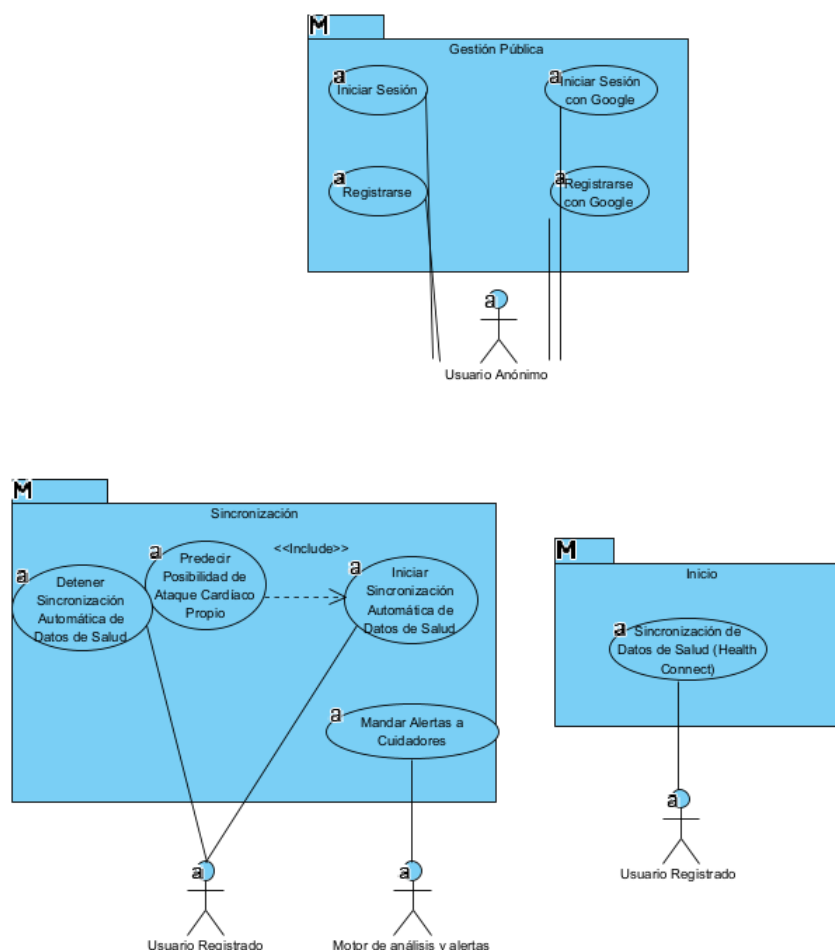


Ilustración 10 Ilustración 10 Paquetes de casos de uso 1 (Aplicación)

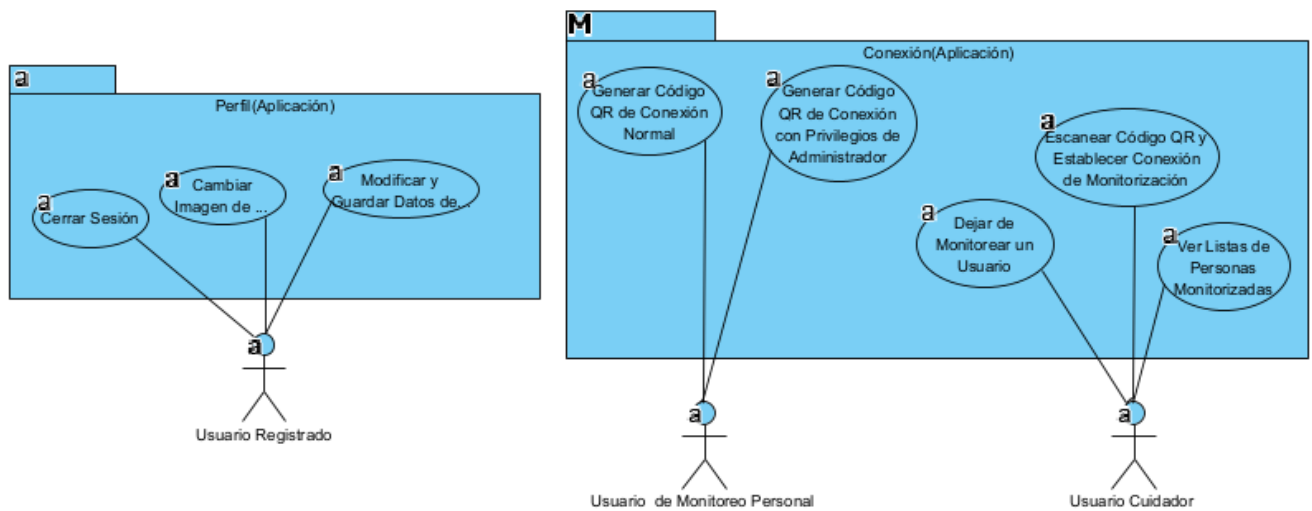


Ilustración 11 Paquetes de casos de uso 2 (Aplicación)

### 7.2.2 Web

La plataforma web se estructura en cinco módulos principales, orientados a la gestión de la salud y la interconexión de usuarios:

1. **Módulo de Autenticación y Acceso Público.** Gestiona el acceso y registro de usuarios ("Usuario Anónimo") mediante credenciales de Google o correo electrónico y contraseña.
2. **Módulo de Gestión de Perfil de Usuario.** Permite al "Usuario Registrado" administrar su información personal, configuración de cuenta, y gestionar sus datos (descarga en PDF, eliminación de datos de salud o cuenta completa).
3. **Módulo de Redes.** Facilita las interacciones y la monitorización entre diferentes roles de usuarios. Los usuarios ("Usuario Registrado" / "Usuario de Monitoreo Personal") pueden gestionar permisos y códigos para ser monitorizados, mientras que los "Usuarios Cuidadores" pueden monitorizar a otros mediante códigos y acceder a sus estadísticas.
4. **Módulo de Gestión de Datos y Estadísticas de Salud.** Posibilita al "Usuario Registrado" la introducción de registros de salud, así como la visualización, filtrado y exportación (PDF, gráficas) de sus métricas y estadísticas.
5. **Módulo de Predicciones de Salud.** Ofrece información prospectiva. El "Usuario Registrado" puede obtener predicciones propias (ej. riesgo cardíaco), y el "Usuario Cuidador" visualizar las predicciones de los usuarios que monitoriza.



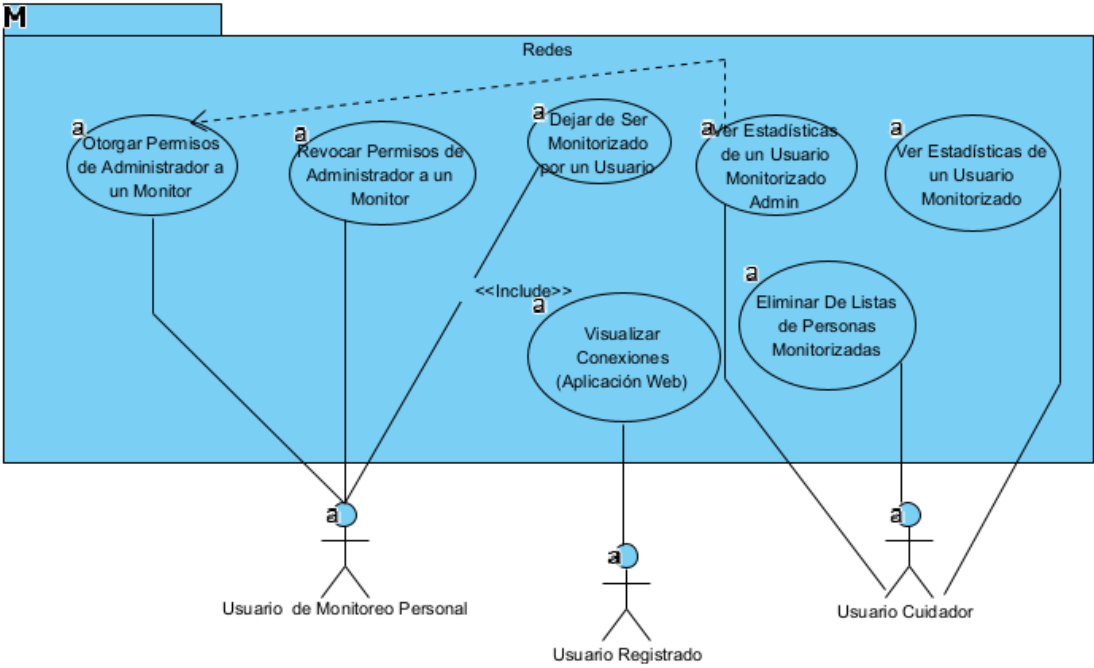


Ilustración 12 Paquetes de casos de uso 1 (Web)

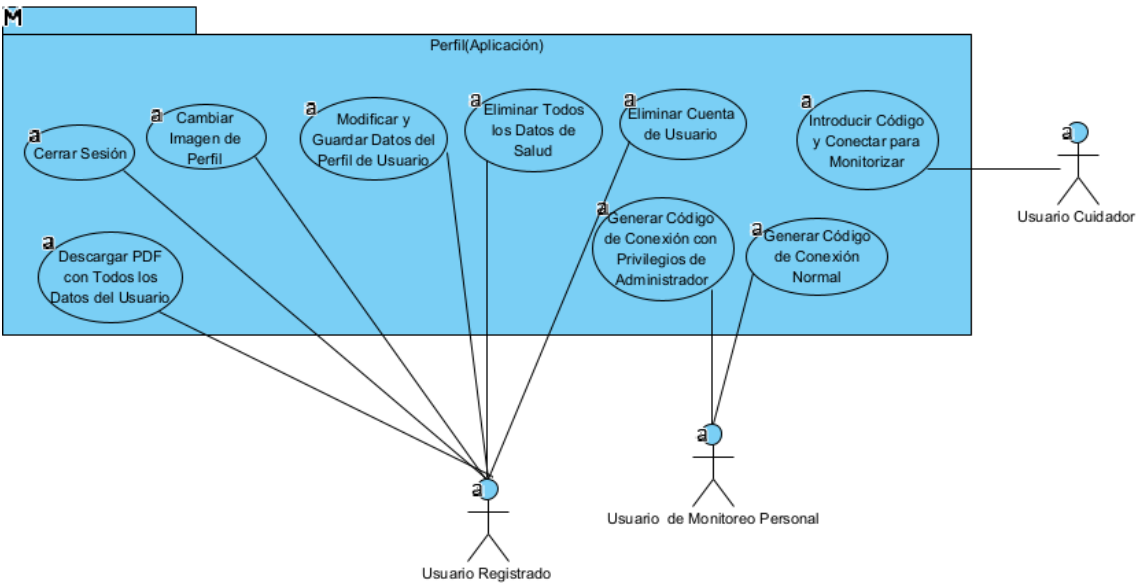


Ilustración 13 Paquetes de casos de uso 2 (Web)

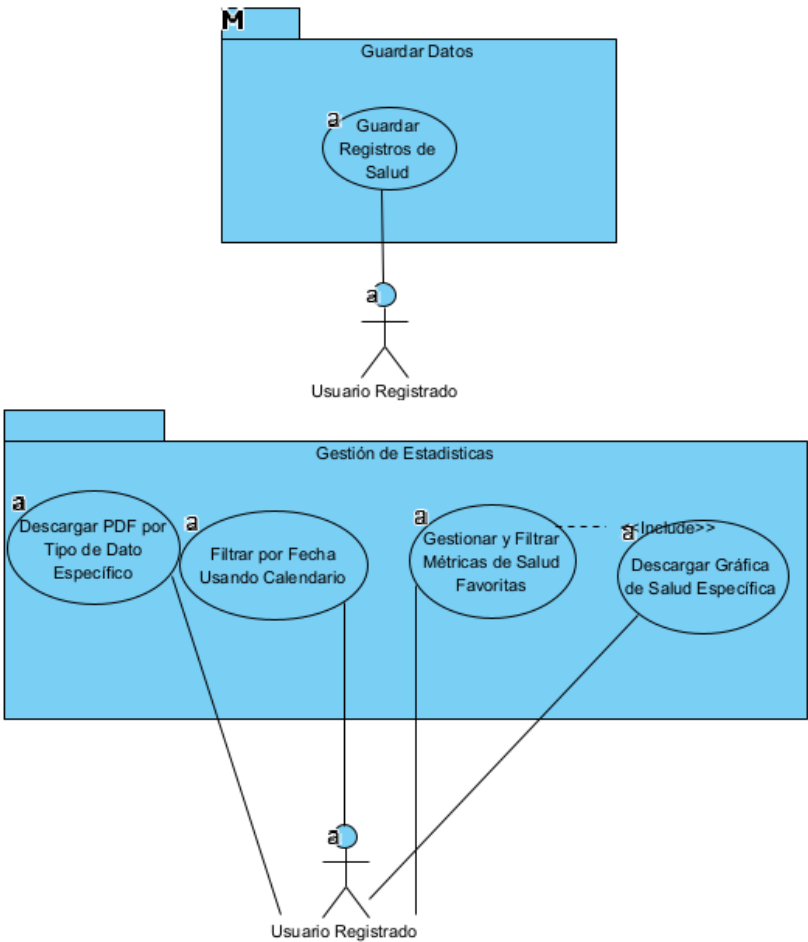


Ilustración 14 Paquetes de casos de uso 3 (Web)

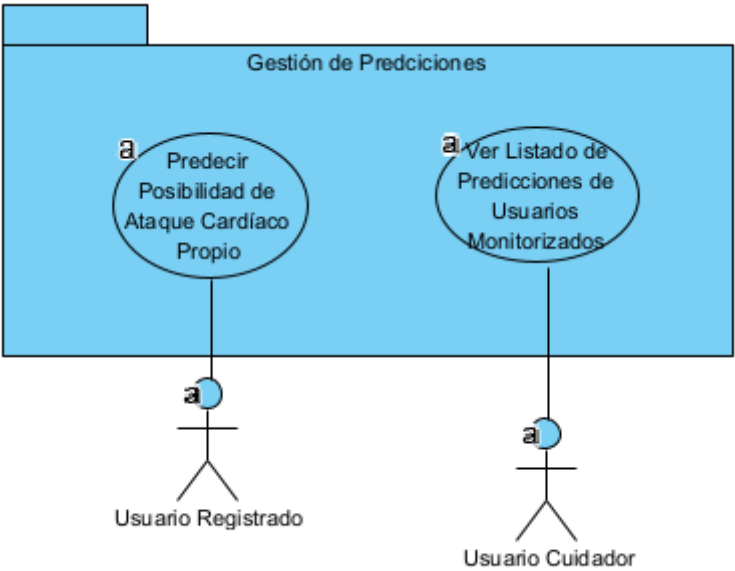


Ilustración 15 Paquetes de casos de uso 4 (Web)

### 7.2.3 Diagrama de secuencia

Un diagrama de secuencia [5] es una representación gráfica que muestra la interacción de los distintos objetos dentro de un sistema a lo largo del tiempo, así como la secuencia y el orden de los mensajes intercambiados en un caso de uso. A continuación, se exponen diagramas de secuencia para aquellas funcionalidades cuyas interacciones se consideraron de mayor complejidad o resultan cruciales para una clara comprensión del comportamiento dinámico del sistema.

La Ilustración 16, muestra la funcionalidad de sincronización de datos de salud a través de "Health Connect". El proceso comienza cuando un **Usuario Registrado** intenta subir sus datos. El sistema primero **verifica los permisos** necesarios y si la **conexión está activa** con Health Connect. Si los permisos y la conexión son correctos, se **recopilan los datos** de salud y se **almacenan** en la base de datos de "Datos Diarios" a través de un "ServicioDatosSalud". Se manejan explícitamente los escenarios donde los permisos son incorrectos o la conexión no está activa, informando al usuario en cada caso.

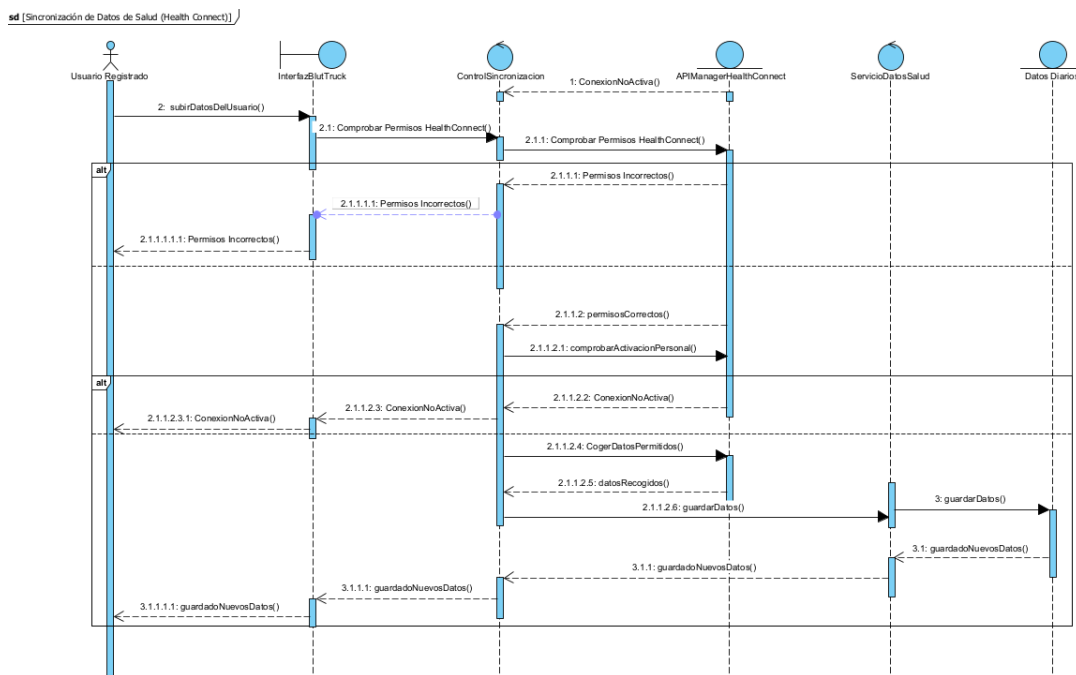


Ilustración 16 Sincronización Datos de Salud(HealthConnect)

Ilustración 17 describe la **sincronización automática de datos de salud**. El sistema recolecta datos de Health Connect, los guarda, y luego un módulo de IA los **analiza para predecir posibles ataques cardíacos**. Si se detectan "datos peligrosos", el sistema envía una **notificación urgente al usuario**. También gestiona si la sincronización en segundo plano está permitida y comunica al usuario si los datos son estables o si se ha enviado una alerta.

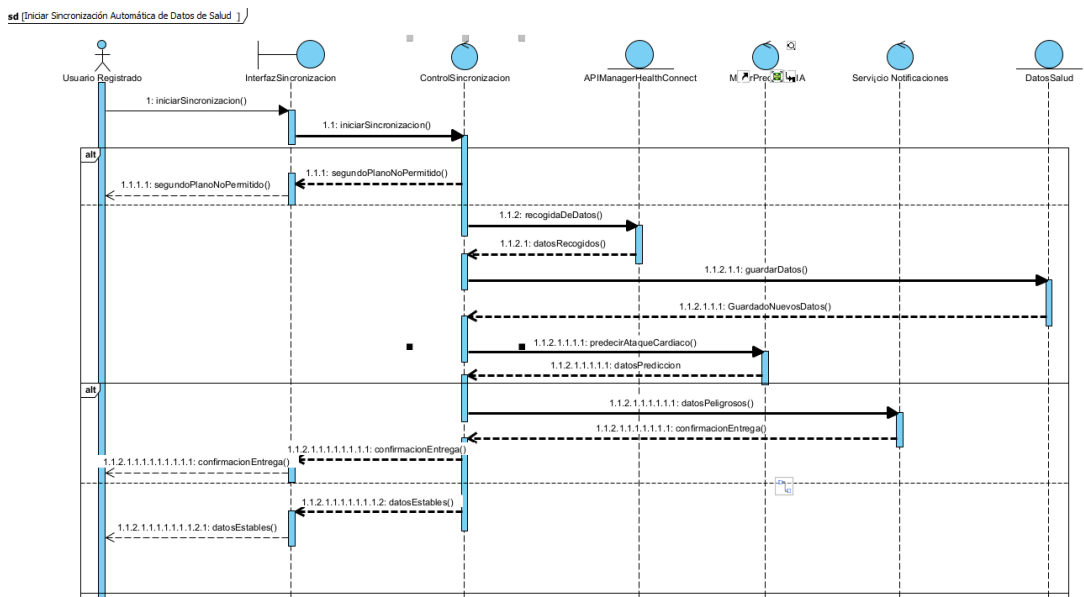


Ilustración 17 Iniciar Sincronización Automática Datos de Salud

## 8 Hipótesis y Restricciones / Alcance

El desarrollo de la aplicación para la predicción de ataques cardíacos se enfrenta a diversas consideraciones técnicas y normativas. No obstante, se espera que la implementación de esta solución genere un impacto significativo y positivo en el ámbito de la salud digital.

### 6.1. Restricciones del Proyecto

Las principales restricciones del proyecto están intrínsecamente ligadas a los requisitos no funcionales, siendo crucial el cumplimiento de las normativas de seguridad y la garantía de operatividad del sistema en diversos dispositivos. A continuación, se detallan las restricciones identificadas:

- **Privacidad y Protección de Datos:** La naturaleza sensible de la información médica manejada por la aplicación exige un cumplimiento estricto de normativas como el Reglamento General de Protección de Datos (GDPR). Para asegurar la confidencialidad e integridad de los datos de los usuarios:
  - Los datos personales y biométricos se almacenarán localmente en el dispositivo del usuario o en servidores seguros designados, y no serán compartidos con terceros sin el consentimiento explícito del usuario.
  - Se implementarán mecanismos de cifrado robustos para la protección de toda la información intercambiada y almacenada.
- **Limitaciones del Modelo de Inteligencia Artificial (IA):** A pesar de la utilización de múltiples algoritmos de aprendizaje automático con el fin de optimizar la precisión en la predicción de ataques cardíacos, el rendimiento del modelo predictivo puede verse condicionado por:
  - La cantidad y, fundamentalmente, la calidad de los datos disponibles para el entrenamiento del modelo.
  - La posibilidad de que ciertos modelos introduzcan sesgos si los conjuntos de datos de entrenamiento no son suficientemente diversos o representativos de la población de usuarios.
- **Compatibilidad y Recursos del Dispositivo:** La integración con Health Connect, si bien estratégica, presenta ciertas limitaciones inherentes a su fase temprana de desarrollo y a las políticas de los sistemas operativos:
  - Actualmente, Health Connect no ofrece soporte para algunos datos biométricos avanzados, como los electrocardiogramas (ECG), aunque ciertos dispositivos *wearables* son capaces de capturarlos.

- La compatibilidad no se extiende a todas las aplicaciones de *wearables* existentes, ya que, en algunos casos, los datos deben transitar primero por Google Fit antes de ser accesibles para Health Connect.
- Las restricciones impuestas por el sistema operativo Android dificultan la ejecución continua en segundo plano *background*, lo que podría obligar a mantener la aplicación en primer plano *foreground* para una correcta sincronización, con el consecuente impacto en el consumo de batería del dispositivo.
- **Fiabilidad y Disponibilidad del Sistema:** Es imperativo que la aplicación opere de manera continua y fiable, asegurando la disponibilidad constante de las predicciones y la integridad de los datos almacenados. Para ello:
  - Se implementarán mecanismos de respaldo *backup* y recuperación de datos para mitigar el impacto de posibles fallos en la base de datos o en el sistema.
  - Se optimizarán las consultas a la base de datos y la arquitectura general del sistema para minimizar los tiempos de inactividad y garantizar una respuesta ágil.

## 6.2. Impacto Esperado y Beneficios

La implementación y adopción de esta herramienta para la predicción de ataques cardíacos se proyecta con una serie de ventajas y mejoras significativas tanto para los usuarios finales como para el sector de la salud en general. Estos beneficios se fundamentan en la capacidad de la tecnología para ofrecer soluciones proactivas y personalizadas:

- **Mejora en la prevención de enfermedades cardíacas:** La aplicación capacitará a los usuarios para conocer su riesgo estimado de sufrir un ataque cardíaco prácticamente en tiempo real. Esta información puede facilitar la adopción temprana de medidas preventivas y fomentar un mayor control y concienciación sobre su salud cardiovascular.
- **Accesibilidad y facilidad de uso:** El diseño de la aplicación se centrará en ofrecer una interfaz intuitiva y accesible para un amplio espectro de usuarios, independientemente de sus conocimientos médicos previos, promoviendo así su adopción y uso continuado.
- **Optimización del diagnóstico temprano y seguimiento:** Los datos recopilados, agregados y analizados por la aplicación pueden constituir una fuente de información valiosa para médicos y especialistas, contribuyendo

a mejorar el seguimiento personalizado de sus pacientes y la detección precoz de posibles riesgos.

- **Impulso de la salud digital y la inteligencia artificial en medicina:** Este proyecto sirve como demostración práctica del potencial de la Inteligencia Artificial aplicada a la medicina para mejorar la calidad de vida de los usuarios. Además, se espera que contribuya a abrir nuevas vías y oportunidades en el desarrollo de soluciones tecnológicas innovadoras para el sector salud.

## 9 Estudio de alternativas y viabilidad

En el proceso de desarrollo de esta aplicación, se llevó a cabo una evaluación de distintas tecnologías, herramientas y plataformas con el objetivo primordial de seleccionar aquellas que resultaran más adecuadas para cumplir con los requisitos del proyecto en términos de compatibilidad, eficiencia, seguridad y escalabilidad. La elección de cada componente se fundamentó en criterios como la experiencia previa del equipo de desarrollo, el soporte a largo plazo ofrecido por la tecnología, la facilidad de implementación y la robustez en la gestión y seguridad de los datos.

### 9.1 Justificación de Elección de Herramientas

A continuación, se describe la justificación de la elección de las principales herramientas y tecnologías empleadas:

1. **Framework de Desarrollo Móvil:** Se consideraron diversas opciones para el desarrollo de la aplicación móvil. Finalmente, se optó por **Flutter**.
  - *Alternativas Consideradas:* Desarrollo nativo (Java/Kotlin para Android, Swift/Objective-C para iOS), React Native, Xamarin.
  - *Justificación de Flutter:* Su capacidad multiplataforma fue un factor decisivo; aunque la fase inicial del proyecto se centra en Android, Flutter facilita una futura expansión a iOS con una reutilización significativa del código base. Además, ofrece un alto rendimiento al compilar a código nativo y cuenta con una amplia comunidad de desarrolladores y un sólido respaldo por parte de Google, lo que asegura su evolución y mantenimiento a largo plazo.
2. **Gestión de Datos Biométricos:** La integración con fuentes de datos biométricos fue un aspecto crucial. Se seleccionó **Health Connect**.
  - *Alternativas Consideradas:* Google Fit API, Apple HealthKit.
  - *Justificación de Health Connect:* A pesar de que Google Fit API ofrecía compatibilidad con Android e iOS, su anunciado abandono para mayo de 2025 la descartó como opción viable a largo plazo. Apple HealthKit, siendo una solución nativa para iOS, no se alineaba con el enfoque inicial en Android y la disponibilidad de dispositivos para pruebas. Health Connect fue elegida por su enfoque en la seguridad y privacidad, permitiendo una gestión centralizada y encriptada de los datos biométricos en el dispositivo. Aunque actualmente presenta limitaciones como la no admisión de datos avanzados (ECG) y una compatibilidad con aplicaciones de *wearables* aún en



desarrollo (requiriendo en ocasiones el paso previo por Google Fit), se considera la opción con mayor proyección y seguridad futura.

3. **Servidor:** Para la arquitectura del servidor y la lógica de negocio, se eligió **C# y .NET 8**.

- *Alternativas Consideradas:* Node.js con Express, Python con Django/Flask, Java con Spring Boot.
- *Justificación de C# y .NET 8:* La sólida experiencia previa del equipo con este lenguaje y su ecosistema fue un factor determinante, agilizando el desarrollo y el mantenimiento. .NET es reconocido por su potencia, escalabilidad y robustez, siendo ampliamente utilizado en la industria para aplicaciones de alto rendimiento. Se implementó una arquitectura por capas (Controladores, Servicios, Repositorios) para mejorar la mantenibilidad y facilitar futuras ampliaciones.

4. **API de Inteligencia Artificial:** Para el procesamiento de datos y la entrega de predicciones, se desarrolló una API en **Python utilizando Flask**.

- *Alternativas Consideradas:* FastAPI, Django (para la API de IA), o integrar la lógica de IA directamente en el backend .NET.
- *Justificación de Python con Flask:* Se priorizó la simplicidad y la eficiencia para este componente específico. La facilidad de uso de Flask y la experiencia previa en su implementación permitieron un desarrollo rápido. Si bien se reconoce que *frameworks* como FastAPI o Django podrían ofrecer características de seguridad más robustas, Flask se consideró suficiente para las necesidades de manejo de solicitudes en la fase actual del proyecto, con la posibilidad de evaluar una migración futura si la aplicación escala significativamente.

5. **Frontend (Interfaz de Usuario Web/Dashboard):** Para la interfaz de usuario accesible vía web, se optó por **Vue.js**.

- *Alternativas Consideradas:* Angular, React.
- *Justificación de Vue.js:* La experiencia previa del equipo con Vue.js redujo la curva de aprendizaje y agilizó la implementación. Su simplicidad, estructura basada en componentes reutilizables y su rendimiento optimizado, que ofrece un buen equilibrio con un consumo eficiente de recursos, fueron factores clave para su selección.

6. **Plantilla de Interfaz de Usuario:** Para acelerar el desarrollo del *frontend* y asegurar una estética profesional, se utilizó la plantilla **Sakai Vue**.
  - *Justificación de Sakai Vue:* Se eligió por ser una opción gratuita, completa y estéticamente atractiva, compatible con Vue 3. Su diseño moderno y profesional, junto con componentes personalizables, permitió desarrollar una interfaz intuitiva sin invertir tiempo excesivo en diseño desde cero.
7. **Plataforma como Servicio (BaaS):** Para la gestión de almacenamiento, base de datos y autenticación de usuarios, se seleccionó **Firebase**.
  - *Alternativas Consideradas:* AWS Amplify, Azure Mobile Apps, Parse, desarrollo propio de backend para autenticación y almacenamiento.
  - *Justificación de Firebase:*
    - **Motivos generales:** Su integración sencilla con Flutter y Vue.js, los precios asequibles y escalables (especialmente el almacenamiento gratuito de imágenes en Firebase Storage para las etapas iniciales), y la seguridad integrada fueron determinantes.
8. **Base de Datos (Firebase Realtime Database):** Dentro de las opciones de Firebase (Realtime Database vs. Firestore), se optó por Realtime Database debido a su optimización para la sincronización en tiempo real, menor latencia (crucial para datos de salud) y costos más asequibles para volúmenes de datos pequeños o medianos, adecuados para la fase actual.
9. **Autenticación (Firebase Authentication):** Se utilizó para la creación y autenticación de usuarios (correo/contraseña e inicio de sesión con Google) por su robustez, seguridad y facilidad de implementación, evitando el desarrollo de una infraestructura propia.

## 9.2 Viabilidad y Monetización

El uso estratégico de tecnologías gratuitas o con modelos de precios escalables (como Firebase en sus niveles iniciales) ha permitido minimizar los costos de desarrollo iniciales. No obstante, se identifican los siguientes costos potenciales a medida que la aplicación gane tracción y escale:

- **Almacenamiento:** Aunque inicialmente cubiertos por niveles gratuitos o de bajo costo de Firebase, estos costos aumentarán con el incremento en el número de usuarios y el volumen de datos gestionados.
- **Servicios de IA:** Aunque actualmente no se necesiten muchos recursos cuanto mejor, más rápida y precisa sea la IA más costos supondrá.

- **Mantenimiento y Actualizaciones:** La evolución continua de la aplicación requerirá inversión en mejoras de seguridad, optimización de rendimiento, y adaptación a nuevas versiones de sistemas operativos y librerías.

### **Modelo de Monetización**

Para asegurar la sostenibilidad económica y el crecimiento futuro del proyecto, se han analizado diversas estrategias de monetización, que podrían implementarse de forma individual o combinada:

#### **Versión Freemium:**

Ofrecer un conjunto de funcionalidades básicas de forma gratuita para atraer a una amplia base de usuarios.

Proporcionar acceso a características avanzadas (por ejemplo, predicciones más detalladas, análisis de tendencias a largo plazo, almacenamiento extendido de historial de datos) mediante una suscripción premium.

#### **Suscripción Mensual o Anual:**

Establecer un plan de pago recurrente que permita a los usuarios acceder a la totalidad de las funcionalidades, incluyendo informes de salud personalizados, comparativas anónimas con perfiles similares, y un seguimiento avanzado de sus datos biométricos.

#### **Colaboraciones con Empresas de Salud y Seguros:**

Explorar integraciones con compañías aseguradoras, clínicas, hospitales o empresas de dispositivos médicos que puedan ofrecer la aplicación como un valor añadido dentro de sus servicios de monitoreo de salud o programas de bienestar.

#### **Publicidad y Patrocinios:**

Implementar anuncios no intrusivos dentro de la versión gratuita de la aplicación.

Establecer colaboraciones con empresas del sector salud (farmacéuticas, fabricantes de *wearables*, etc.) para recomendar dispositivos compatibles o servicios complementarios, siempre de forma transparente para el usuario.

#### **Venta de Datos Agregados y Anónimos:**

Previa anonimización y agregación de los datos, y siempre en estricto cumplimiento con las normativas de privacidad (GDPR), se podría ofrecer análisis de tendencias de salud poblacional a instituciones de investigación médica, organismos de salud pública o empresas de tecnología en salud. Este enfoque requeriría una especial atención a los aspectos éticos y legales.

## 10 Descripción de la solución propuesta

El objetivo principal del proyecto ha sido desarrollar una herramienta accesible y fácil de usar que permita a cualquier persona monitorear la salud cardiovascular de sus seres queridos de manera sencilla y asequible. Para ello, se ha diseñado una aplicación intuitiva, enfocada en ofrecer solo las funcionalidades esenciales, garantizando una experiencia de usuario fluida y adaptada a cualquier nivel de conocimiento tecnológico. La prioridad ha sido crear una solución eficiente y ligera, sin perder de vista la seguridad y la privacidad de los datos del usuario.

En la **Ilustración 18** se presenta un diagrama de la arquitectura del sistema propuesto, detallando los componentes clave y los flujos de comunicación entre ellos. El diseño se basa en un modelo cliente-servidor centralizado, donde el servidor actúa como núcleo para orquestar el intercambio de información entre las distintas capas del sistema.



Ilustración 18 Estructura BlutTruck

A continuación, se describe cada uno de los componentes y sus interacciones:

- **Dispositivo Wearable (Smartband):** Es el punto de partida para la adquisición de datos biométricos del usuario. Su función es exclusivamente la de capturar y transmitir información de manera unidireccional hacia la aplicación móvil a través de una conexión Bluetooth.
- **Aplicación Móvil ("BlutTruck"):** Actúa como el cliente principal y la interfaz directa con el usuario. Establece una comunicación bidireccional con el servidor y cumple con las siguientes funciones:

- **Envío de datos:** Remite al servidor la información recopilada por la *Smartband*, así como datos generados directamente por la interacción del usuario, tales como modificaciones en su perfil o el establecimiento de nuevas conexiones mediante lectura de códigos QR.
- **Recepción de datos:** Solicita y recibe del servidor la información del perfil del usuario, su historial y sus conexiones para mostrarla en la interfaz.
- **Servidor (Backend):** Constituye el núcleo central del sistema, funcionando como una API que gestiona toda la lógica de negocio. Es el único componente que se comunica con todas las demás partes del ecosistema, centralizando las operaciones para:
  - Autenticar y autorizar a los usuarios.
  - Recibir, procesar y validar los datos provenientes de los clientes (aplicación móvil y web).
  - Interactuar con la base de datos para almacenar y recuperar información.
  - Suministrar los parámetros necesarios al módulo de inteligencia artificial y recibir los resultados.
  - Exponer los *endpoints* necesarios para el funcionamiento tanto de la aplicación móvil como de la plataforma web.
- **Base de Datos (Firebase):** Para la persistencia de la información, se utiliza la plataforma Firebase de Google. La arquitectura de datos se divide en dos servicios principales:
  - **Realtime Database:** Almacena la información estructurada de los usuarios, como perfiles, configuraciones y registros de datos en tiempo real.
  - **Firebase Storage:** Se emplea para el almacenamiento de archivos binarios, principalmente las imágenes de perfil de los usuarios.
- **Módulo de Inteligencia Artificial Predictiva:** Este componente opera como un servicio especializado que recibe un conjunto de parámetros específicos desde el servidor. Tras procesar estos datos mediante sus modelos predictivos, devuelve al servidor una inferencia o predicción, que puede ser utilizada para ofrecer funcionalidades avanzadas al usuario.

- **Plataforma Web:** Proporciona una interfaz de usuario secundaria, accesible a través de un navegador. Al igual que la aplicación móvil, su funcionalidad depende completamente de la comunicación bidireccional con el servidor. Comparte la mayoría de los *endpoints* de la API con la aplicación, lo que garantiza la coherencia de los datos y la lógica de negocio en ambas plataformas.

## 10.1 Aplicación móvil

Nada más descargar la aplicación, se habrá que clicar en el logo de la App que será como el de la Ilustración 19, al clicar, el usuario es recibido con una pantalla de bienvenida que muestra el logotipo personalizado de la aplicación. Esta interfaz inicial refuerza la identidad visual del proyecto y proporciona una primera impresión clara y profesional.

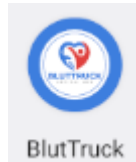


Ilustración 19 Logo

### Gestión de Usuarios (Users)

La aplicación incorpora un sistema de gestión de usuarios que permite un acceso seguro y personalizado a cada usuario. Las principales funcionalidades implementadas son las siguientes:

- **Registro:** Almacena los datos de un nuevo usuario, validando que sean correctos antes de proceder con la creación de la cuenta.
- **Inicio de sesión (Login):** Verifica la existencia del usuario en la base de datos y valida las credenciales introducidas. Si son correctas, se procede con el inicio de sesión.
- **Gestión del perfil:** Permite a los usuarios modificar su información personal, como la foto de perfil y otros datos relevantes, asegurando que los cambios se reflejen correctamente en la base de datos.

Este sistema garantiza una experiencia fluida y segura para cada usuario, manteniendo la integridad y privacidad de sus datos.

Para ello se crearon las siguientes interfaces:

Se puede ver en la Ilustración 20, la primera vista que el usuario encontrará al abrir la aplicación. En esta pantalla, se presenta el nombre y el logotipo de la aplicación,

lo que refuerza la identidad visual. La interfaz dos botones para iniciar sesión o registrarse según desee el usuario:



Ilustración 20 Inicio Aplicación

En la Ilustración 20, al pulsar "Iniciar Sesión, se presenta la Ilustración 21, destaca en la parte superior el logotipo de la aplicación. La interfaz solicita el "Correo electrónico" y la "Contraseña" en campos claramente delimitados.

Adicionalmente, ofrece funcionalidades comunes como una casilla de "Recuérdame" para mantener la sesión activa y un enlace de "¿Olvidaste contraseña?" para la recuperación de acceso. El botón principal de acción es "Iniciar sesión", con un diseño prominente. Como alternativa de acceso rápido, se

incluye la opción "Continuar con Google", y finalmente, un enlace "¿No tienes cuenta? Regístrate" dirige a los nuevos usuarios al proceso de creación de cuenta.

---



### Iniciar Sesión

☐ Recuérdame [¿Olvidaste contraseña?](#)

---

 [Continuar con Google](#)


[¿No tienes cuenta? Regístrate](#)

Ilustración 21 Inicio Sesión App

La Ilustración 22 muestra la pantalla de "Crear Cuenta", accesible para usuarios nuevos. Al igual que en la pantalla de inicio de sesión, el logotipo de "BLUTTRUCK MEDICAL WEB" se sitúa en la cabecera. Se solicitan los campos necesarios para el registro: "Nombre", "Correo electrónico", "Contraseña" y "Confirmar contraseña". Es requisito indispensable la aceptación de los "términos y condiciones" mediante una casilla de verificación antes de poder completar el registro con el botón "Registrarse". Similar a la pantalla de inicio de sesión, se ofrece la opción de "Continuar con Google" para agilizar el proceso de alta.



---



### Crear Cuenta

☐ Acepto los [términos y condiciones](#)

---


 Continuar con Google

Ilustración 22 Registrase App

Una vez que el usuario ha accedido, se presenta la pantalla principal o panel de control, como se observa en la Ilustración 23. Esta vista está personalizada con un mensaje de bienvenida ("Bienvenido Carlos León Arjona"). El logotipo de la aplicación vuelve a ocupar una posición central destacada. Un botón de acción principal, "Obtener datos de hoy", con la función de obtener los datos del día actual de Health Connect. Debajo, se encuentran botones para la gestión de "Ajustes Health Connect" y "Conexión Health Connect", indicando si la conexión está permitida o no. La navegación principal de la aplicación se estructura mediante una barra inferior con iconos y texto para las secciones: "Inicio" (actualmente seleccionada), "Perfil", "Salud" y "Conexión".



Ilustración 23 Inicio App

La Ilustración 24 detalla la sección de "Perfil" del usuario. Esta pantalla muestra en la parte superior el nombre del usuario ("Carlos León Arjona") y permite la gestión de una imagen de perfil, con la opción de "Eliminar imagen". Se presentan campos para visualizar y editar datos personales relevantes como "Peso (kg)", "Altura (cm)" y "Fecha de nacimiento". Se incluye también una opción para seleccionar el género ("Hombre" o "Mujer") mediante botones de radio. También, aunque no se aprecia en la foto está la opción de indicar tu colesterol, y otro tipo de información relevante para la aplicación.

Ilustración 24 Perfil APP

En la Ilustración 25 se muestra la pantalla "Control de Servicio" que permite al usuario manejar la función principal de monitorización de salud de la app.

- Una **imagen central** ilustra el seguimiento de la salud, enfocada a un adulto mayor.
- El botón "**Iniciar servicio**" activa un servicio en segundo plano (foreground service). Este servicio:
  - Se conecta a Health Connect.
  - Recoge datos de salud cada 20 minutos.
  - Usa estos datos para predecir la posibilidad de un ataque cardíaco.
  - Muestra una notificación persistente mientras está activo.
- El botón "**Detener servicio**" para esta recolección y predicción.
- El botón "**Permisos**" es clave para que el usuario autorice a la app el acceso a sus datos de Health Connect y el funcionamiento del servicio.

En esencia, esta pantalla es el interruptor principal para que la app vigile la salud cardíaca del usuario de forma continua y gestione los permisos necesarios.

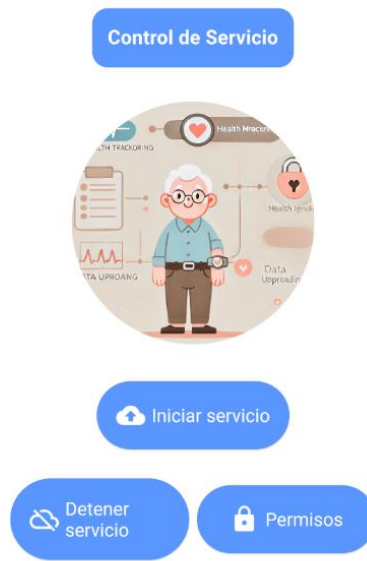


Ilustración 25 Control de Servicio

Ilustración 26, **Visualización de Conexiones:** En la parte superior, bajo el título "Conexiones", se muestra una lista de las personas que el usuario actual está monitorizando. En el ejemplo, vemos a "Juanito Suárez" con un icono de información.

**Acciones Principales:** En la parte inferior de la pantalla, dos botones prominentes ofrecen las funcionalidades clave:

- **"Escanear QR":** Esta opción permite al usuario monitorizar a otra persona. Al seleccionarla, se activará la cámara para escanear el código QR generado por el otro usuario.
- **"Generar QR":** Esta opción permite que el usuario actual sea monitorizado por otra persona. Al seleccionarla, la aplicación generará un código QR que otro usuario podrá escanear.



Ilustración 26 Conexiones App

Al pulsar "Generar QR" en la pantalla anterior, se presenta un diálogo modal.

Aparece la Ilustración 27 con la pregunta de privilegios: El diálogo pregunta: "¿Deseas generar un QR con privilegios de admin o normal?". Esto indica que existen diferentes niveles de acceso o control que se pueden otorgar a la persona que va a monitorizar.

**Opciones:** Se ofrecen dos botones: "Normal" y "Admin", para que el usuario elija el tipo de privilegio asociado al QR que se va a generar.

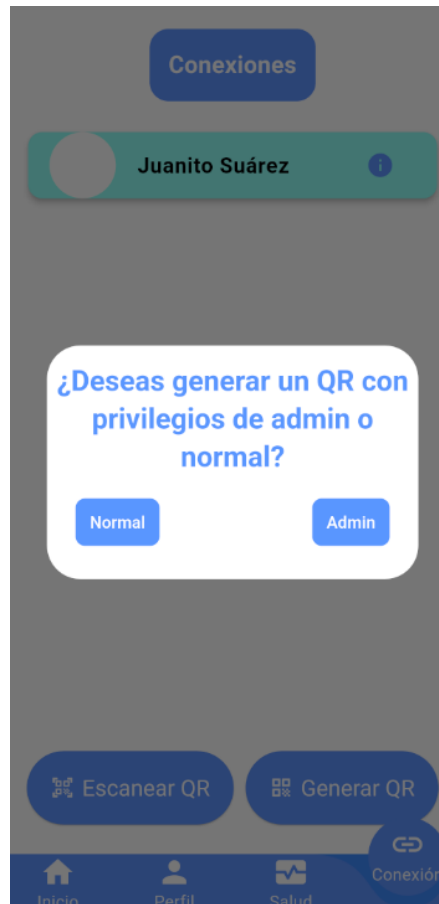


Ilustración 27 Dialog Conexiones App

Una vez seleccionado el nivel de privilegio (Normal o Admin), la aplicación muestra el código QR generado en un nuevo diálogo modal, como se observa en la Ilustración 28.

**Instrucción:** Un mensaje como "Conecta con tus familiares" contextualiza el uso de esta función, sugiriendo que está orientada a permitir que personas cercanas puedan monitorizar al usuario.

**Código QR:** El código QR se muestra de forma clara y grande para facilitar su escaneo por parte de otra persona.

**Cerrar:** Un botón "Cerrar" permite al usuario descartar la visualización del QR una vez que ha sido compartido o escaneado.



Ilustración 28 QR Conexiones

Al seleccionar "Escanear QR" en la Ilustración 26, se accede a la interfaz del "Escáner de Código QR", Ilustración 29.

**Vista de Cámara:** La pantalla utiliza la cámara del dispositivo, mostrando un recuadro guía donde el usuario debe enfocar el código QR de la persona a la que desea monitorizar.

**Instrucción:** Un texto en la parte inferior, "Escanea un código QR para conectarte con otra persona", junto con un icono de escaneo QR, refuerza la acción a realizar.

**Navegación:** Una flecha de retroceso en la esquina superior izquierda permite volver a la pantalla anterior.

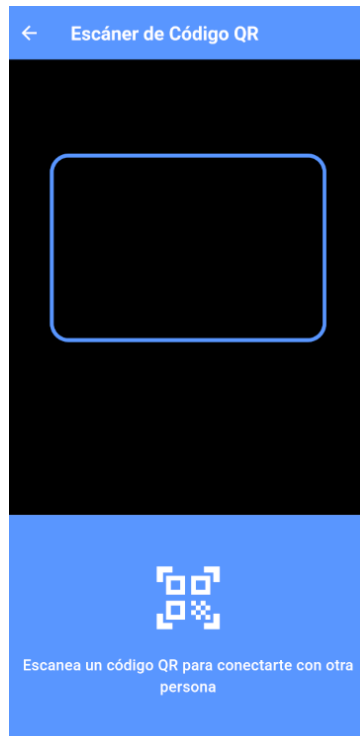


Ilustración 29 Escaner QR

## 10.2 Aplicación web

La página de inicio del sitio web de BlutTruck (Ilustración 30 y Ilustración 31) está diseñada para presentar la propuesta de valor de la aplicación y las formas de acceder a sus servicios.

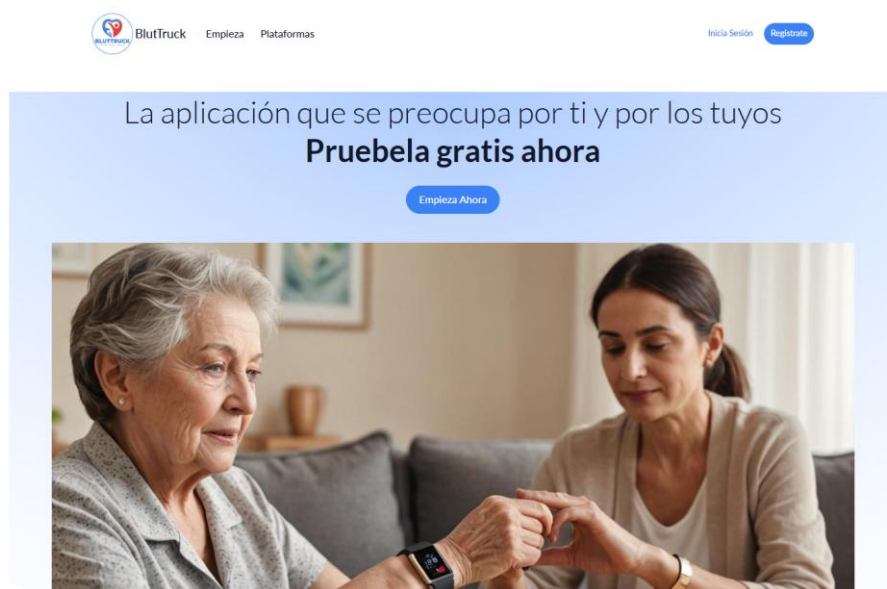


Ilustración 30 Landing Parte 1

La sección principal (Ilustración 30) capta la atención con el eslogan "La aplicación que se preocupa por ti y por los tuyos" y una clara llamada a la acción para



"Pruébala gratis ahora". Esto se acompaña de una imagen emotiva que refuerza el enfoque en el cuidado de la salud, especialmente para personas mayores, mediante el uso de tecnología como smartwatches. La cabecera incluye el logotipo y opciones de navegación esenciales como "Inicia Sesión" y "Regístrate".

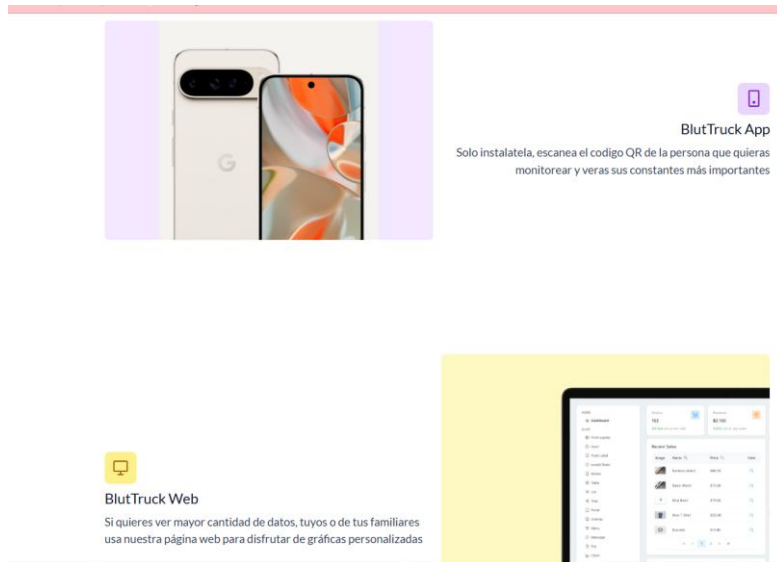


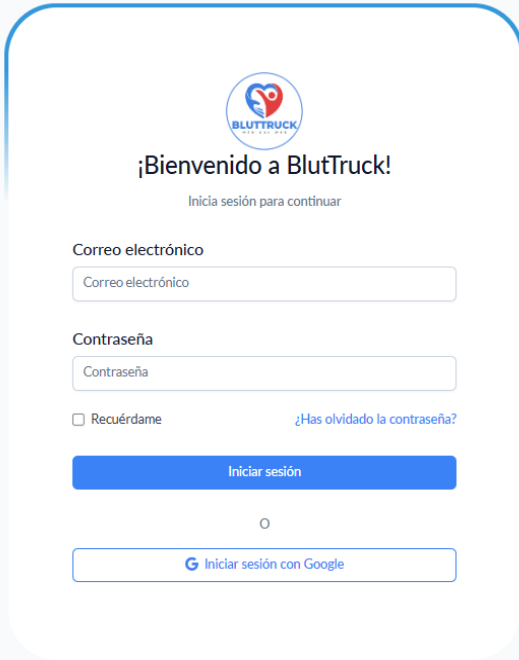
Ilustración 31 Landing Parte 2

La Ilustración 31, detalla las dos plataformas principales a través de las cuales los usuarios pueden interactuar con Blutruck:

- La "**Blutruck App**" para móviles, que se destaca por su facilidad para monitorear constantes vitales de otras personas mediante el escaneo de códigos QR.
- "**Blutruck Web**", una plataforma accesible vía navegador, orientada a la visualización de una mayor cantidad de datos y al análisis a través de gráficas personalizadas, tanto para datos propios como de familiares.

En conjunto, la página de inicio busca comunicar de manera efectiva el propósito de Blutruck como una solución integral para la monitorización de la salud, accesible tanto desde dispositivos móviles para un seguimiento ágil como desde la

web para un análisis más profundo, siempre con un enfoque en el cuidado personal y familiar.



¡Bienvenido a BlutTruck!

Inicia sesión para continuar

Correo electrónico

Contraseña

☐ Recuérdame [¿Has olvidado la contraseña?](#)

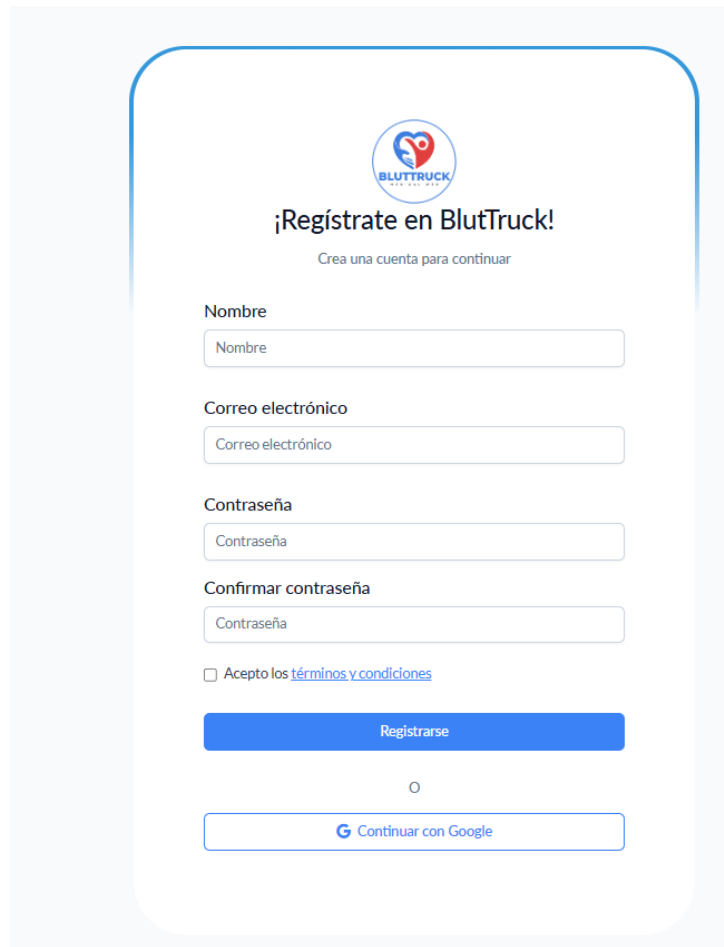
Iniciar sesión

O

[Iniciar sesión con Google](#)

Ilustración 32 Iniciar Sesión Web

La Ilustración 32 muestra la interfaz destinada al inicio de sesión de usuarios en el sitio web de BlutTruck. Este formulario se presenta dentro de un contenedor visualmente definido por bordes redondeados, encabezado por el logotipo distintivo de BlutTruck, lo que asegura la consistencia de la marca. Su propósito es permitir a los usuarios ya registrados acceder a sus cuentas personales. Para ello, se solicitan las credenciales estándar: "Correo electrónico" y "Contraseña". El formulario se complementa con funcionalidades de conveniencia, como la opción "Recuérdame" para mantener la sesión activa y un enlace de "¿Has olvidado la contraseña?" para facilitar la recuperación del acceso. La acción principal se efectúa mediante el botón "Iniciar sesión". Adicionalmente, se ofrece como alternativa el método de "Iniciar sesión con Google", que agiliza el proceso de ingreso utilizando cuentas de Google preexistentes.



The image shows a web registration form for 'BlutTruck'. At the top is the BlutTruck logo, which consists of a red heart with a white truck silhouette inside, and the text 'BLUTTRUCK' below it. Below the logo is the heading '¡Regístrate en BlutTruck!' followed by the subtext 'Crea una cuenta para continuar'. The form contains four input fields: 'Nombre', 'Correo electrónico', 'Contraseña', and 'Confirmar contraseña'. Each field has a placeholder text matching its label. Below these fields is a checkbox labeled 'Acepto los términos y condiciones'. At the bottom of the form is a blue button labeled 'Registrarse'. Below the button is a small 'O' separator, followed by a button with the Google logo and the text 'Continuar con Google'.

Ilustración 33 Registrarse Web

De forma análoga en cuanto a diseño y estructura, la Ilustración 33 corresponde al formulario de registro del sitio web de BlutTruck. Este también se encuentra encapsulado en un contenedor con bordes suavizados y exhibe el logotipo de BlutTruck en su parte superior. Su finalidad es guiar a los nuevos usuarios en el proceso de creación de una cuenta para utilizar los servicios de la plataforma. El formulario requiere la introducción de datos personales básicos para el alta, incluyendo "Nombre", "Correo electrónico", "Contraseña" y un campo para "Confirmar contraseña", asegurando la correcta introducción de esta última. Un paso indispensable es la aceptación de los "términos y condiciones", indicada mediante una casilla de verificación. La culminación del proceso se realiza a través del botón "Registrarse". De manera similar al formulario de inicio de sesión, se proporciona la opción de "Continuar con Google", ofreciendo una vía rápida y simplificada para el registro mediante una cuenta de Google existente.

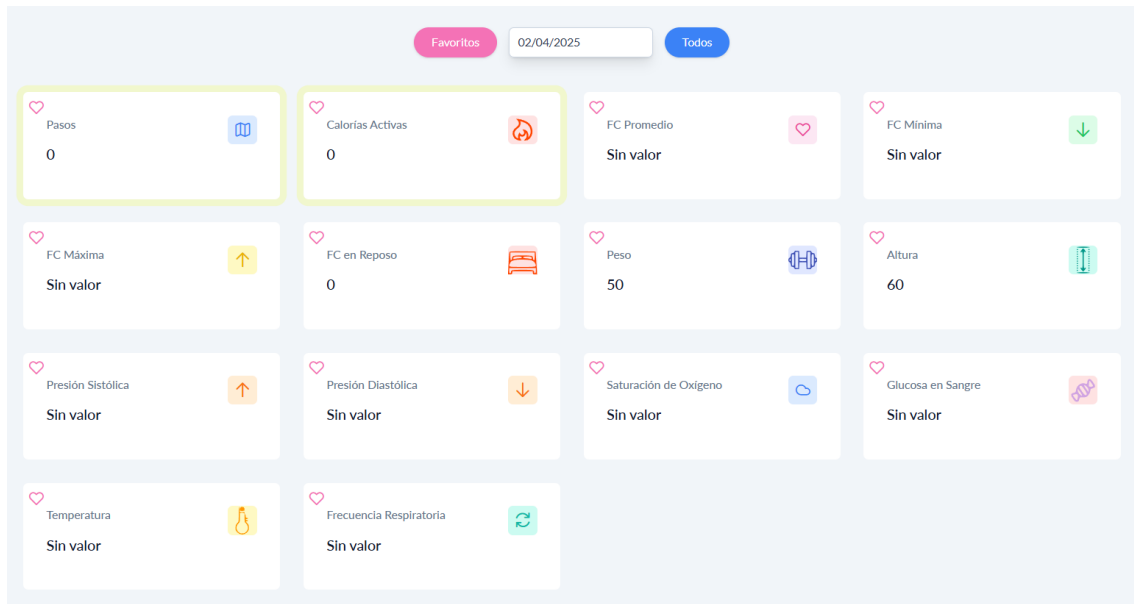


Ilustración 34 Estadísticas Usuario

Al iniciar sesión, el usuario es dirigido a un panel de control principal (Ilustración 34). Esta interfaz muestra un resumen de sus estadísticas de salud más recientes mediante un sistema de tarjetas individuales. Cada tarjeta presenta una métrica específica, como pasos, calorías activas, diversos parámetros de frecuencia cardíaca (promedio, mínima, máxima, en reposo), peso, altura, presión sistólica y diastólica, saturación de oxígeno, glucosa en sangre, temperatura y frecuencia respiratoria.

Para personalizar la visualización de estos datos, el panel ofrece dos filtros principales:

1. Un **selector de fecha** (ej. "10/05/2025") que permite al usuario consultar las estadísticas correspondientes a un día específico.
2. Un sistema de **"Favoritos"**: el usuario puede marcar con un icono de corazón las tarjetas de estadísticas que considera más importantes, para luego filtrarlas y visualizar solo su selección.

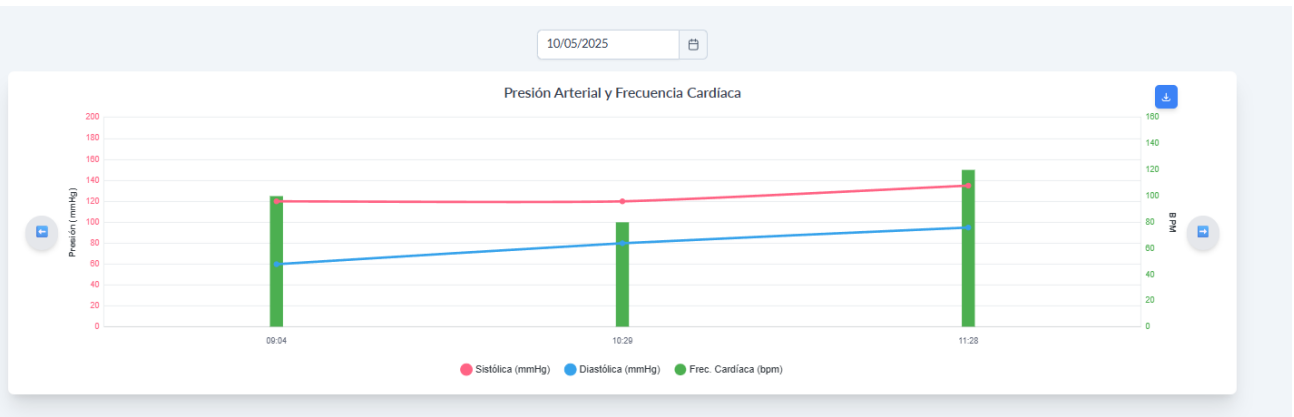


Ilustración 35 Gráficas Usuario

La plataforma también permite profundizar en los datos a través de vistas gráficas detalladas, Ilustración 35. Un ejemplo es el gráfico de "Presión Arterial y Frecuencia Cardíaca", que representa la evolución de la presión sistólica (mmHg), diastólica (mmHg) y la frecuencia cardíaca (bpm) para la fecha seleccionada. Una funcionalidad clave en estas vistas detalladas es un **botón de descarga**, situado en la parte superior derecha del gráfico, que facilita la exportación de la información gráfica o los datos correspondientes.

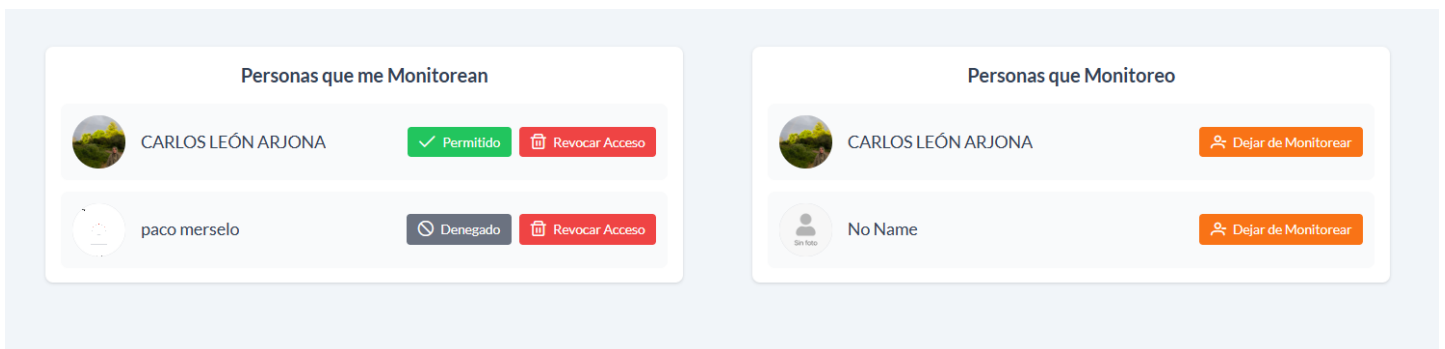


Ilustración 36 Interfaz Gestión de Conexiones "Redes"

La Ilustración 36 presenta la sección de "Redes", que es fundamental para administrar las relaciones de monitorización entre usuarios. Esta interfaz se organiza en dos áreas claramente diferenciadas:

1. **"Personas que me Monitorean"**: En esta lista, el usuario puede ver quiénes tienen acceso para monitorizar sus datos de salud. Para cada persona, se indica el estado del permiso. Desde aquí, el usuario tiene control para gestionar estos accesos, pudiendo "Revocar Acceso" o, como se indica, conceder o retirar permisos de administrador sobre su propia cuenta a quienes le monitorizan.
2. **"Personas que Monitorio"**: Aquí se muestran los usuarios a los cuales el usuario actual está monitorizando. La acción principal disponible es "Dejar

de Monitorizar" para finalizar el seguimiento de una persona. Crucialmente, al interactuar con un usuario de esta lista, el monitor puede acceder a sus estadísticas de salud y, si cuenta con privilegios de administrador sobre la cuenta de la persona monitorizada, también podrá visualizar y gestionar su perfil detallado.

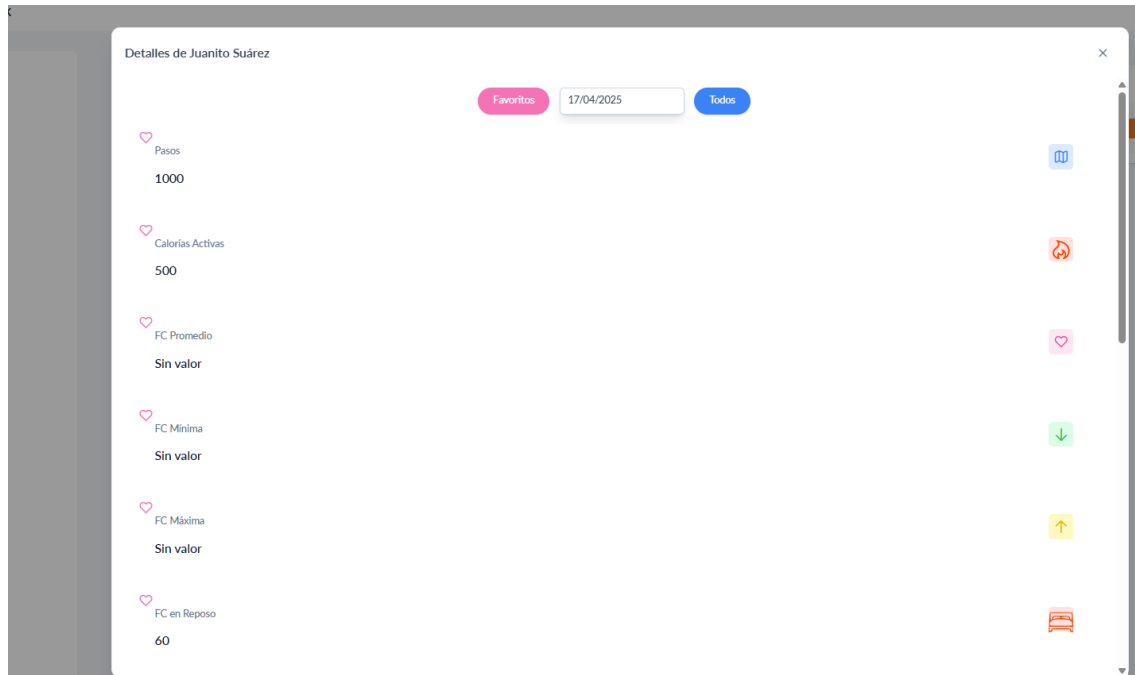


Ilustración 37 Panel Estadísticas Persona Monitoreada

La Ilustración 37 ejemplifica la pantalla a la que accede un usuario cuando visualiza las estadísticas de una persona a la que monitoriza. Esta interfaz es similar al panel de estadísticas personales del propio usuario, mostrando diversas métricas como "Pasos", "Calorías Activas", y diferentes mediciones de frecuencia cardíaca para una fecha específica. Se mantienen los controles de filtrado por "Favoritos", selector de fecha y "Todos", permitiendo al monitor personalizar la vista de los datos de la persona seguida.

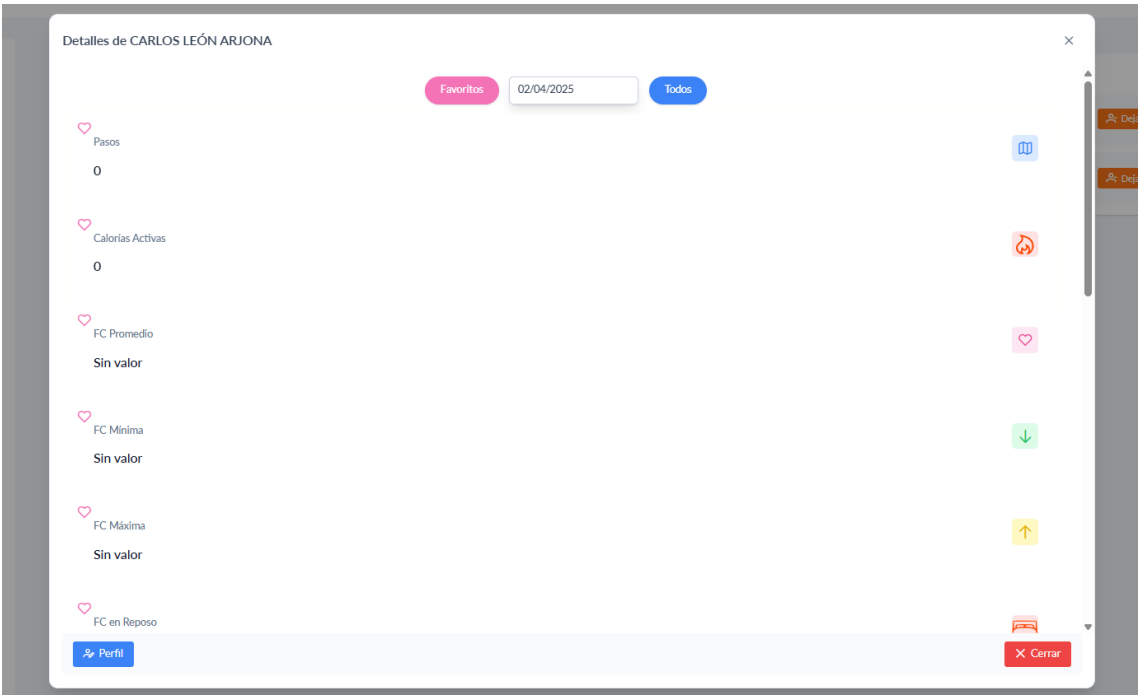


Ilustración 38 Estadísticas Admin

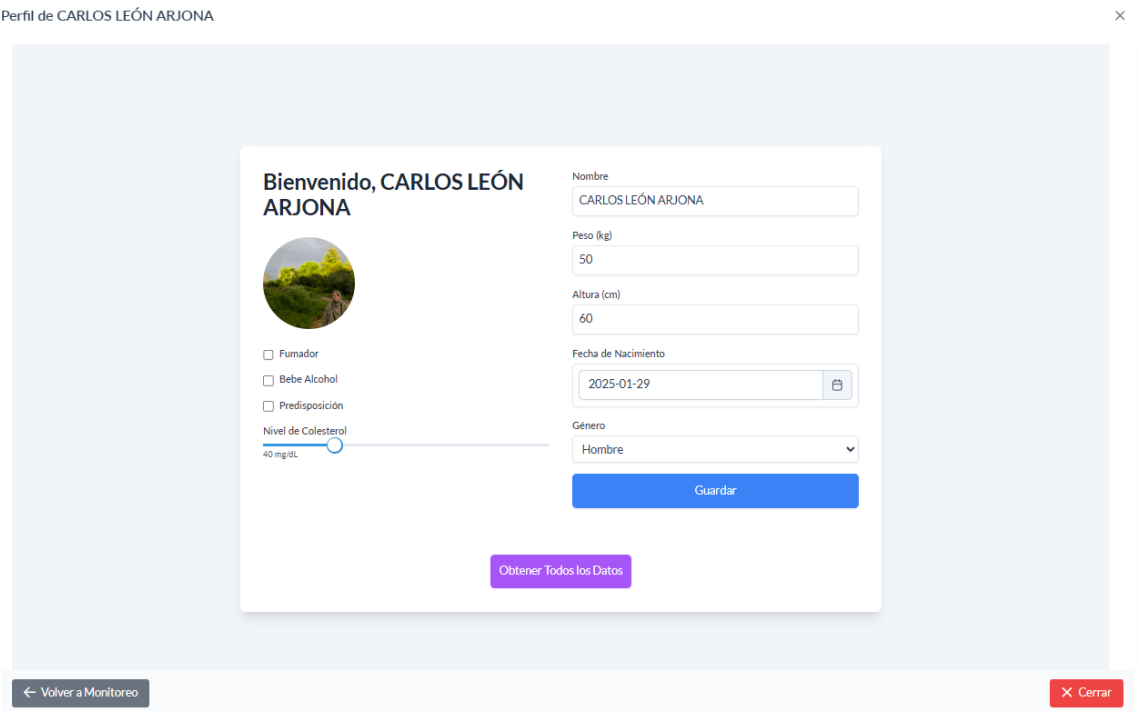


Ilustración 39 Perfil Persona Monitorizada

En la Ilustración 38 podemos ver como la opción de ver el perfil se activa por ser Administradores de esa persona y la Ilustración 39 ofrece una visión más

completa de este perfil, añadiendo campos como "Fecha de Nacimiento", "Género", e información relevante para la salud como "Fumador", "Bebe Alcohol", "Predisposición" y un control deslizante para el "Nivel de Colesterol". La presencia de un botón "Guardar" confirma la capacidad del administrador para modificar estos datos. El botón "Obtener Todos los Datos" Habilita la función de descargar todos los datos del usuario. Los botones inferiores "Volver a Monitoreo" y "Cerrar" facilitan la navegación.

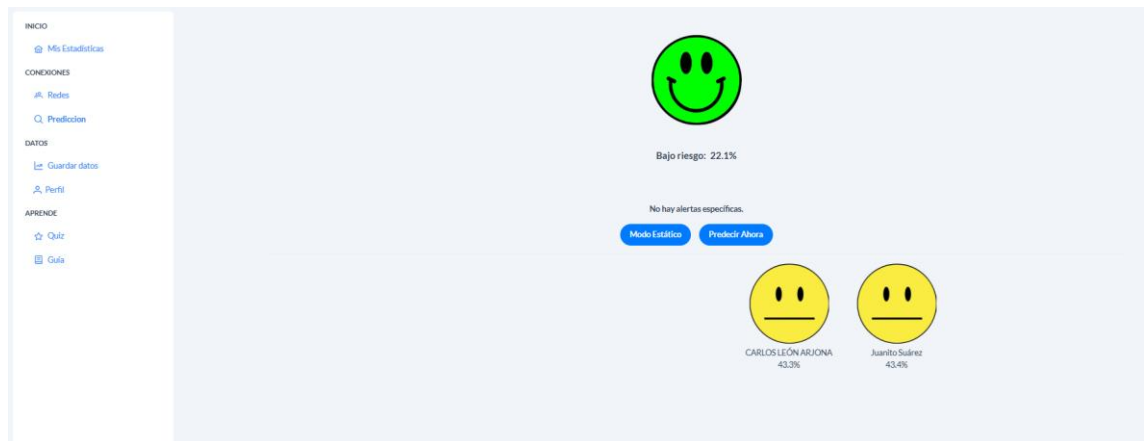


Ilustración 40 Interfaz Predicción

La Ilustración 40 corresponde a la sección "Predicción" de BlutTruck, diseñada para mostrar la **probabilidad estimada de sufrir un ataque cardíaco**.

- Para el **usuario principal**, se presenta un icono de una cara cuyo color (verde, amarillo o rojo) y expresión varían según el nivel de riesgo. En el ejemplo, una cara verde indica "Bajo riesgo: 22.1%", y se añade que "No hay alertas específicas".
- Los botones "Modo Estático" y "Predecir Ahora" permiten interactuar con la lista de personas monitoreadas y dan una visión estática o en movimiento de las predicciones.
- La pantalla también muestra las **predicciones para las personas que el usuario monitorea** (ej., "CARLOS LEÓN ARJONA" con 40.3% y "Juanito Suárez" con 43.4%). Cada uno se representa con un icono de cara similar, indicando visualmente su respectivo nivel de probabilidad de ataque cardíaco.



Ilustración 41 Interfaz Perfil

La Ilustración 41 muestra la pantalla principal del perfil del usuario en la plataforma BlutTruck. Esta interfaz permite una gestión integral de la información personal y las conexiones de monitorización.

En la sección superior, el usuario puede visualizar y **editar sus datos personales y de salud**. Se incluyen campos como "Nombre", "Peso (kg)", "Altura (cm)", "Fecha de Nacimiento" y "Género". Adicionalmente, hay opciones para indicar hábitos o condiciones como "Fumador", "Bebe Alcohol", "Predisposición Genética" y ajustar el "Nivel de Colesterol" mediante un control deslizante. Todos los cambios realizados en esta sección se pueden **guardar** mediante el botón "Guardar Cambios".

Más abajo, la pantalla se divide en dos funcionalidades clave para la gestión de conexiones:

1. **"Generar Código de Conexión"**: Esta sección permite al usuario crear un código para que otras personas puedan monitorizarlo. Se ofrecen dos opciones diferenciadas por botones: "Generar Código Admin" (para otorgar permisos de administrador) y "Generar Código Normal" (para permisos estándar).
2. **"Conectar con Otra Persona"**: Aquí, el usuario puede introducir un código de conexión que le haya proporcionado otra persona y, al pulsar "Conectar", establecerá una conexión para comenzar a monitorizar a ese individuo.

Finalmente, en la parte inferior de la pantalla, se encuentran los botones para **"Cerrar Sesión"** y acceder a **"Más Opciones"**, que la funcionalidad se verá en la Ilustración 42.

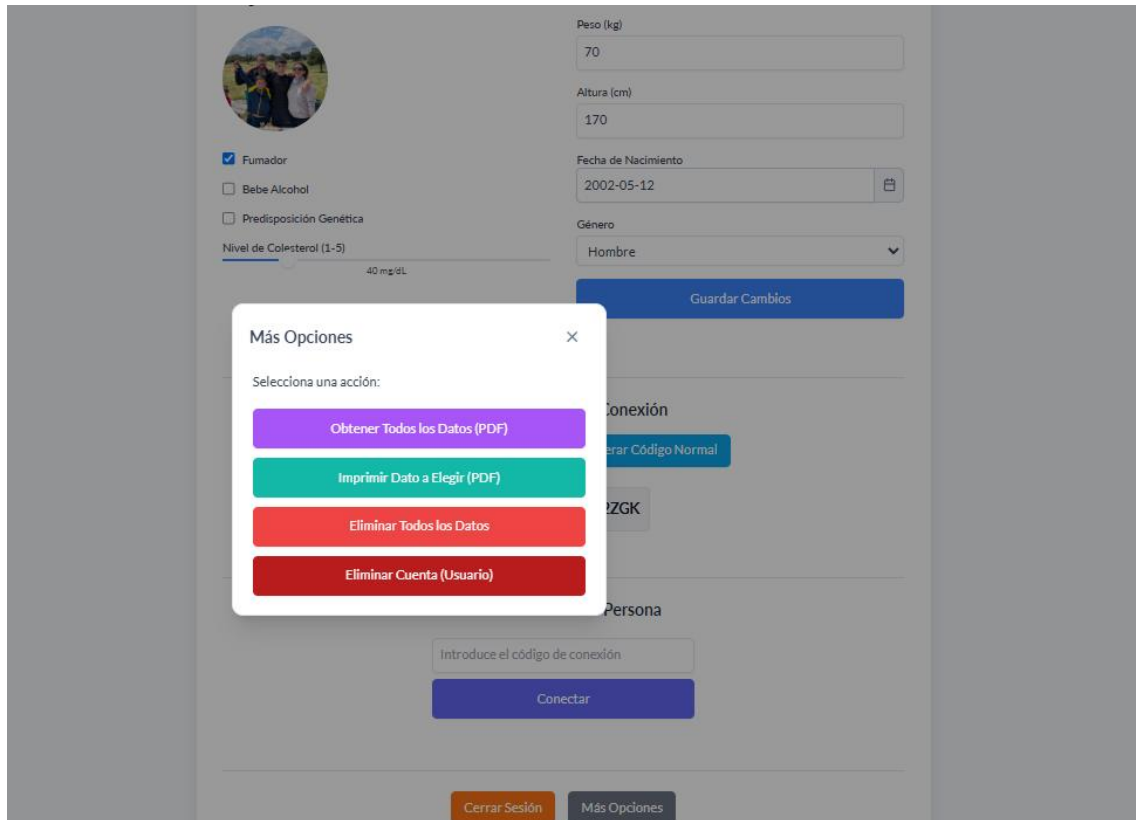


Ilustración 42 Dialog Más opciones

En la Ilustración 42, más pequeño y titulado "Más Opciones", también se superpone a la misma interfaz de perfil de usuario. Ofrece al usuario una selección de acciones generales.

Contiene cuatro botones con diferentes colores y funciones:

- **Obtener Todos los Datos (PDF):** Sugiere la descarga de toda la información del usuario en un archivo PDF.
- **Imprimir Dato a Elegir (PDF):** Implica que el usuario podrá seleccionar datos específicos para imprimir en PDF, abriendo la **Ilustración 34**.
- **Eliminar Todos los Datos:** Indica una opción para borrar toda la información registrada del usuario.
- **Eliminar Cuenta (Usuario):** Representa la acción de borrar permanentemente la cuenta del usuario.

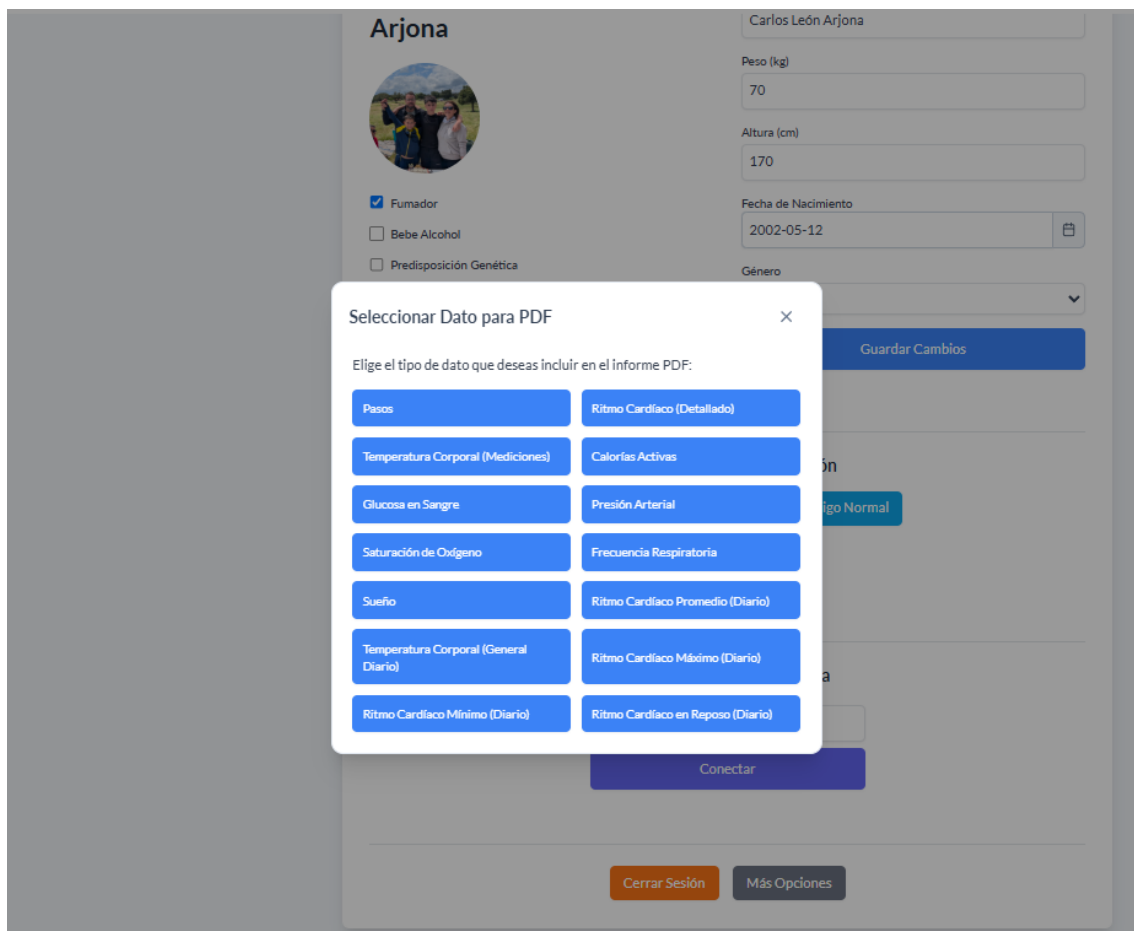
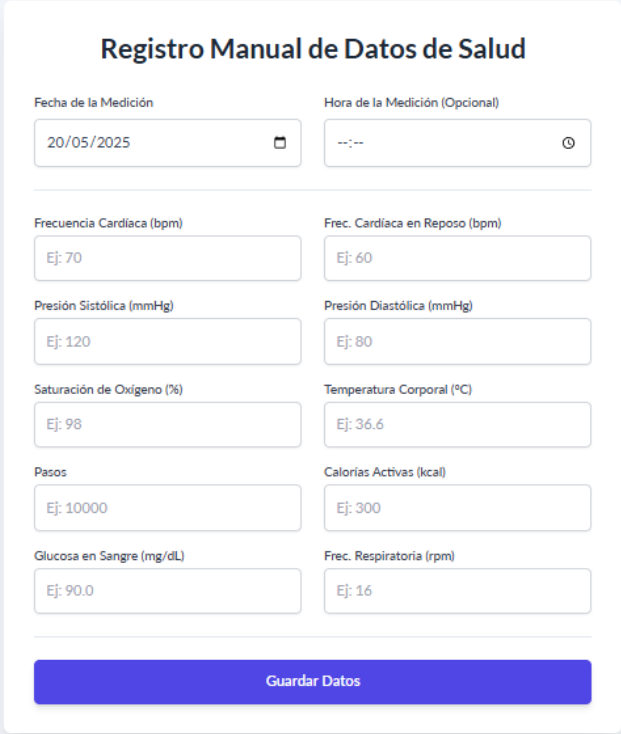


Ilustración 43 Seleccionar Datos

La Ilustración 43, muestra un dialog para elegir que PDF quiere el usuario.



The image shows a web form titled "Registro Manual de Datos de Salud". It contains several input fields for health data, arranged in two columns. At the top, there are fields for "Fecha de la Medición" (with a date picker showing 20/05/2025) and "Hora de la Medición (Opcional)" (with a time picker showing --:--). Below these are ten fields for various health metrics, each with a placeholder example value: "Frecuencia Cardíaca (bpm)" (Ej: 70), "Frec. Cardíaca en Reposo (bpm)" (Ej: 60), "Presión Sistólica (mmHg)" (Ej: 120), "Presión Diastólica (mmHg)" (Ej: 80), "Saturación de Oxígeno (%)" (Ej: 98), "Temperatura Corporal (°C)" (Ej: 36.6), "Pasos" (Ej: 10000), "Calorías Activas (kcal)" (Ej: 300), "Glucosa en Sangre (mg/dL)" (Ej: 90.0), and "Frec. Respiratoria (rpm)" (Ej: 16). At the bottom of the form is a large blue button labeled "Guardar Datos".

Ilustración 44 Interfaz Guardar datos

La Ilustración 44 muestra un formulario web titulado "**Registro Manual de Datos de Salud**". Este formulario está diseñado para que un usuario pueda introducir manualmente diversas métricas de salud.

El formulario presenta los siguientes campos de entrada:

- **Fecha de la Medición:** Un campo para introducir la fecha y tiene la posibilidad de generar un calendario y elegir una fecha.
- **Hora de la Medición (Opcional):** Un campo para la hora, si se pulsa se podrá elegir una hora y minuto.
- **Frecuencia Cardíaca (bpm):** Campo para la frecuencia cardíaca en latidos por minuto.

- **Frec. Cardíaca en Reposo (bpm):** Campo para la frecuencia cardíaca en reposo.
- **Presión Sistólica (mmHg):** Campo para la presión arterial sistólica en milímetros de mercurio.
- **Presión Diastólica (mmHg):** Campo para la presión arterial diastólica.
- **Saturación de Oxígeno (%):** Campo para el nivel de saturación de oxígeno en porcentaje.
- **Temperatura Corporal (°C):** Campo para la temperatura corporal en grados Celsius.
- **Pasos:** Campo para el número de pasos.
- **Calorías Activas (kcal):** Campo para las calorías activas quemadas en kilocalorías.
- **Glucosa en Sangre (mg/dL):** Campo para el nivel de glucosa en sangre en miligramos por decilitro.
- **Frec. Respiratoria (rpm):** Campo para la frecuencia respiratoria en respiraciones por minuto.

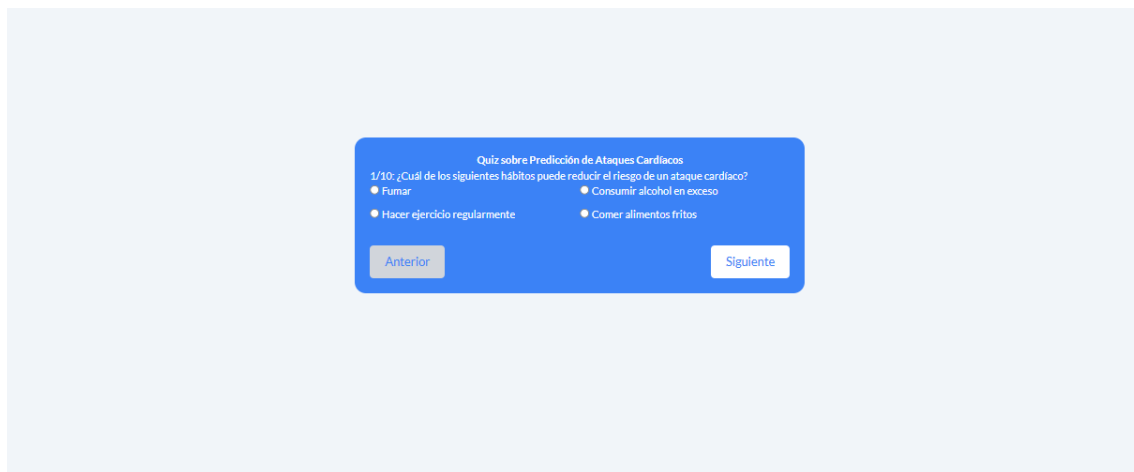


Ilustración 45 Interfaz Quiz

La Ilustración 45, muestra la interfaz de un *quiz* interactivo diseñado para educar al usuario sobre la predicción y prevención de ataques cardíacos.



Ilustración 46 Interfaz Guía

La Ilustración 46 presenta una guía informativa titulada "**Guía Completa: Ataque Cardíaco**". La guía está dividida en varias secciones que ofrecen información detallada sobre diferentes aspectos de los ataques cardíacos.

## 10.3 Implementación de Componentes Clave

En esta sección se detallan algunos de los componentes de software más relevantes y técnicamente interesantes desarrollados para el proyecto BlutTruck. Estos fragmentos de código han sido seleccionados por su importancia en la arquitectura de la solución, ya que resuelven desafíos específicos de interoperabilidad entre tecnologías o demuestran la implementación de funcionalidades cruciales para la aplicación.

### 10.3.1 Puente de Comunicación Nativo (Kotlin-Dart) para Health Connect

Uno de los mayores desafíos técnicos del proyecto fue la integración con **Health Connect**, la plataforma de Android para la gestión de datos de salud. Dado que BlutTruck es una aplicación desarrollada en Flutter (Dart), un framework multiplataforma, fue necesario crear un puente de comunicación (conocido como *method channel*) para interactuar con las APIs nativas de Android, escritas en Kotlin.

Este puente es fundamental para manejar correctamente el flujo de permisos de Health Connect. Cuando el sistema operativo Android muestra el diálogo de permisos y el usuario pulsa en el enlace de la "política de privacidad", la aplicación

debe ser capaz de interceptar esta acción y mostrar la pantalla correspondiente dentro de la propia aplicación Flutter. Esta lógica no puede ser gestionada directamente desde Dart, por lo que se requiere una implementación nativa.

A continuación, se presenta el código de la clase MainActivity.kt, que actúa como el punto de entrada de la aplicación en Android y gestiona esta comunicación, como se puede ver en la Ilustración 47 y la Ilustración 48 :

```
class MainActivity : FlutterFragmentActivity() {
    companion object {
        private const val CHANNEL = "app.channel.flutter.health.connect.show.privacy.policy"
        private const val FUNCTION = "checkIfShouldShowPrivacyPolicy"
    }

    private var shouldShowPrivacyPolicy: Boolean = false

    // If your application's android:launchMode is set to "standard" or "singleTop" in
    // your AndroidManifest then "onCreate" will be called when a user taps
    // the "privacy policy" button in Google Health's permission dialog.
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        handleIntent(intent)
    }

    // If your application's android:launchMode is set to "singleTask" or "singleInstance" or "singleInstancePerTask" in
    // your AndroidManifest then "onNewIntent" will be called when a user taps
    // the "privacy policy" button in Google Health's permission dialog.
    override fun onNewIntent(intent: Intent) {
        handleIntent(intent)
        super.onNewIntent(intent)
    }
}
```

Ilustración 47 Código túnel parte1

```
private fun handleIntent(intent: Intent) {
    val action = intent.action
    if (Intent.ACTION_VIEW_PERMISSION_USAGE == action || "androidx.health.ACTION_SHOW_PERMISSIONS_RATIONALE" == action) {
        shouldShowPrivacyPolicy = true
    }
}

override fun configureFlutterEngine(flutterEngine: FlutterEngine) {
    GeneratedPluginRegistrant.registerWith(flutterEngine)

    MethodChannel(flutterEngine.dartExecutor.binaryMessenger, CHANNEL)
        .setMethodCallHandler { call: MethodCall, result: MethodChannel.Result ->
            if (call.method.contentEquals(FUNCTION)) {
                result.success(shouldShowPrivacyPolicy)
                shouldShowPrivacyPolicy = false
            }
        }
}
```

Ilustración 48 Código Túnel parte 2

**Análisis del funcionamiento:**

1. **MethodChannel:** Se define un canal de comunicación con un nombre único (CHANNEL). Este canal actúa como un túnel entre el código Dart y el código nativo Kotlin.
2. **Ciclo de Vida de Android (onCreate, onNewIntent):** La aplicación intercepta los Intent del sistema operativo. Si la acción del Intent corresponde a la visualización de los permisos de Health Connect (ACTION\_VIEW\_PERMISSION\_USAGE), se activa una bandera booleana (shouldShowPrivacyPolicy).
3. **Manejador de Llamadas (setMethodCallHandler):** En el lado nativo, se establece un "oyente" que espera llamadas desde Dart a través del canal.
4. **Invocación y Respuesta:**
  - Desde la lógica de la aplicación en Dart, se invoca periódicamente la función checkIfShouldShowPrivacyPolicy a través del canal.
  - El código Kotlin recibe esta llamada, comprueba el valor de la bandera shouldShowPrivacyPolicy y devuelve el resultado (true o false) a Dart.
  - Inmediatamente después de enviar la respuesta, la bandera se resetea a false para evitar que la pantalla de privacidad se muestre repetidamente.

Esta implementación es un claro ejemplo de cómo extender las capacidades de un framework multiplataforma como Flutter para integrarse de manera profunda y robusta con las funcionalidades específicas del sistema operativo subyacente, garantizando una experiencia de usuario fluida y coherente.

### 10.3.2 Gestión de Autenticación y Autorización en el Servidor

Para garantizar la seguridad, integridad y el correcto control de acceso a los datos en la plataforma BlutTruck, el servidor backend implementa un robusto sistema de autenticación y autorización. Este sistema se encarga de dos tareas fundamentales:

1. **Autenticar al propio servidor** ante los servicios de Google y Firebase para realizar tareas administrativas.



2. **Verificar la identidad de cada usuario** que realiza una solicitud y autorizar la operación basándose en sus permisos específicos (usuario normal o administrador/cuidador).

#### 10.3.2.1 Autenticación del Servidor con Credenciales de Cuenta de Servicio

Para que el backend (desarrollado en .NET) pueda interactuar con las APIs de Firebase a un nivel administrativo (por ejemplo, para crear tokens personalizados o gestionar usuarios), debe autenticarse utilizando una **cuenta de servicio**. Esto se logra mediante un fichero de configuración JSON privado que contiene las credenciales de servicio.

Este fichero, que nunca debe ser expuesto públicamente ni incluido en el código de las aplicaciones cliente (móvil o web), se carga en el servidor en el momento del arranque para inicializar el SDK de FirebaseAdmin.

#### Estructura del fichero de credenciales (Ilustración 49):

```
{
  "type": "service_account",
  "project_id": "proyectocsharp-tfg",
  "private_key_id": "[OCULTO POR SEGURIDAD]",
  "private_key": "-----BEGIN PRIVATE KEY-----\n[CLAVE PRIVADA OCULTA POR SEGURIDAD]",
  "client_email": "firebase-adminsdk-fbsvc@proyectocsharp-tfg.iam.gserviceaccount.",
  "client_id": "[OCULTO POR SEGURIDAD]",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/...",
  "universe_domain": "googleapis.com"
}
```

Ilustración 49 Json Config .NET

#### 10.3.2.2 Validación de Tokens de Usuario y Control de Acceso por Roles

Cada solicitud que llega al servidor desde la aplicación web o móvil debe incluir un **ID Token** de Firebase. Este token es la prueba de que el usuario ha iniciado sesión correctamente. El repositorio de datos (HealthDataRepository) es el encargado de validar este token y determinar los permisos del usuario para la acción solicitada.

El proceso de validación y autorización se implementa de forma consistente en todos los métodos del repositorio, como se muestra en el siguiente fragmento del método WriteDataWebAsync, como se ve en Ilustración 50:

```

public async Task WriteDataAsync(WriteDataInputDTO request)
{
    var decodedToken = await FirebaseAuth.DefaultInstance.VerifyIdTokenAsync(request.Credentials.IdToken);
    bool isTokenFromAdmin = decodedToken.Claims.TryGetValue("is_admin", out var isAdminClaim) && (bool)isAdminClaim;

    FirebaseClient client;

    // Si es admin, creamos un cliente autenticado con su propio token de admin.
    if (isTokenFromAdmin)
    {
        Console.WriteLine($"Acción de administrador para usuario: {request.Credentials.UserId}");
        client = new FirebaseClient(_databaseUrl, new FirebaseOptions { AuthTokenAsyncFactory = () => Task.FromResult(request.Credentials.IdToken) });
    }
    // Si es un usuario normal, verificamos seguridad y usamos su token.
    else
    {
        if (decodedToken.Uid != request.Credentials.UserId)
        {
            throw new Exception($"Error de seguridad: Un usuario intentó escribir en la cuenta de otro.");
        }

        // Comprobación de conexión que ya tenías
        var status = await GetConnectionStatusAsync(new GetConnectionStatusInputDTO { Credentials = request.Credentials });
        if (status.ConnectionStatus != 1)
        {
            Console.WriteLine("La conexión del usuario no está activa. No se escriben los datos.");
            return; // No se escriben los datos si el estado no es 1
        }

        client = new FirebaseClient(_databaseUrl, new FirebaseOptions { AuthTokenAsyncFactory = () => Task.FromResult(request.Credentials.IdToken) });
    }
}

```

Ilustración 50 Conexión Firebase Admin

1. **Verificación del Token:** `VerifyIdTokenAsync` valida la firma y caducidad del token con Google, asegurando su autenticidad. Si el token no es válido, el método lanza una excepción y la operación se detiene.
2. **Autorización por Rol (`is_admin`):** Se inspeccionan los *claims* (información adicional) dentro del token. La presencia del claim personalizado `is_admin` permite diferenciar a un cuidador con privilegios de un usuario estándar.
3. **Prevención de Suplantación:** La comprobación `decodedToken.Uid != request.Credentials.UserId` es una capa de seguridad esencial. Garantiza que un usuario autenticado no pueda enviar una solicitud para modificar los datos de otro usuario simplemente cambiando el `UserId` en el cuerpo de la petición. El identificador del token verificado siempre tiene la última palabra.
4. **Ejecución:** Finalmente, se crea una instancia del cliente de base de datos (`FirebaseClient`) que operará con los permisos del token validado, y se realiza la operación de escritura de forma segura.

Este patrón de **verificación de token y comprobación de roles y propietario** se repite en toda la capa de acceso a datos (`HealthDataRepository`), constituyendo la columna vertebral de la seguridad de la aplicación.

## 11 ANÁLISIS DE RIESGOS

El **análisis DAFO** (conocido también como FODA) es una herramienta de planificación estratégica que permite a una persona u organización evaluar su situación actual y sus perspectivas de futuro [6]. Su nombre es un acrónimo de los cuatro factores que analiza: **D**ebilidades, **A**menazas, **F**ortalezas y **O**portunidades.

Consiste en una matriz que se divide en dos análisis principales:

**Análisis Interno:** Examina los factores que la propia organización puede controlar.

**Fortalezas:** Son los atributos y recursos internos que le dan una ventaja competitiva. Por ejemplo, una marca fuerte, una base de clientes leales o una tecnología propia.

**Debilidades:** Son las áreas internas que la perjudican o limitan. Por ejemplo, la falta de capital, una mala ubicación o una tecnología obsoleta.

**Análisis Externo:** Evalúa los factores del entorno que escapan a su control.

**Oportunidades:** Son las condiciones favorables en el entorno que podrían ser aprovechadas. Por ejemplo, un nuevo mercado, cambios en la legislación o avances tecnológicos.

**Amenazas:** Son las corrientes desfavorables en el entorno que podrían perjudicar su desempeño. Por ejemplo, la entrada de nuevos competidores, una crisis económica o cambios en los gustos de los consumidores.

Se puede ver en la Ilustración 51 un ejemplo más visual.



Ilustración 51 Imagen DAFO imagen de [dynamicgc](#)

**Fortalezas:**

**Seguridad:** Se han implementado rigurosas medidas de seguridad para garantizar la confidencialidad, integridad y disponibilidad de la información, un aspecto crítico al manejar datos sensibles del sector salud. Los detalles técnicos de la implementación se especifican en el **Anexo IV**.

**Accesibilidad:** La aplicación ha sido diseñada con un enfoque en la usabilidad, ofreciendo una interfaz intuitiva que permite a los usuarios realizar el seguimiento de personas de manera eficiente y sin una curva de aprendizaje pronunciada, optimizando la experiencia de usuario (UX).

**Selección Tecnológica Estratégica:** La elección de tecnologías consolidadas como **.NET** para el backend y **Vue** para el frontend permite un desarrollo ágil. La reutilización de componentes y la madurez de estos ecosistemas facilitan la implementación rápida de nuevas funcionalidades y aseguran la mantenibilidad del código a largo plazo.

**Arquitectura Backend Escalable:** El servidor se ha construido sobre una arquitectura sólida y modular, diseñada no solo para las necesidades actuales, sino también para soportar una futura evolución y un crecimiento en la carga de trabajo, garantizando la escalabilidad del sistema.

**Debilidades:**

**Aprendizaje de las tecnologías desde cero:** Las tecnologías utilizadas para el proyecto se han aprendido desde prácticamente cero, excepto **.Net**, utilizando recursos disponibles en internet, como tutoriales, documentación oficial y foros de la comunidad.

**Limitaciones de tiempo:** El periodo de desarrollo se ha realizado en un espacio de tiempo reducido, limitando los hitos y reduciendo el alcance del proyecto, llevando una implementación menos completa de las funcionalidades deseadas.

**Limitaciones para conocer las necesidades reales dada la complejidad:** La complejidad del ámbito sanitario, dificulta conocer en profundidad todas las necesidades reales de los usuarios finales.

**Creación de IA propia:** aunque más segura ya que los datos no salen del sistema, la calidad de la IA depende de la cantidad y calidad de los datos usados en el entrenamiento.

### **Oportunidades:**

**Potencial de expansión:** La constante evolución de la plataforma Health Connect de Google representa una oportunidad estratégica clave para el crecimiento futuro del proyecto. La posible incorporación de nuevos tipos de datos biométricos, métricas de mayor precisión o APIs más avanzadas por parte de Google abriría la puerta al desarrollo de modelos predictivos más sofisticados y a una notable mejora en la monitorización de los usuarios. Esto permitiría expandir el alcance de la aplicación a nuevos ámbitos de la salud y el bienestar, ofreciendo un seguimiento más holístico e integral de la persona.

**Ampliación al Ecosistema de Apple (iOS):** Existe un considerable potencial de mercado en los usuarios de dispositivos Apple. La arquitectura actual y el uso de tecnologías modernas podrían facilitar una adaptación a iOS mediante frameworks de desarrollo multiplataforma, lo que permitiría duplicar la base de usuarios potenciales sin necesidad de reconstruir la aplicación desde cero.

### **Amenazas:**

**Incertidumbre Regulatoria:** El sector de la salud está sujeto a una estricta y cambiante regulación (ej. normativas de protección de datos, certificación de software médico). Futuras modificaciones legislativas podrían exigir adaptaciones complejas o incluso comprometer la viabilidad de la aplicación si no se gestionan adecuadamente.

**Evolución de la medicina:** La constante evolución de técnicas y procedimientos médicos puede implicar que algunos datos o funcionalidades no estén contemplados en la aplicación.

**Dependencia de Infraestructura Cloud (Firebase):** La arquitectura se apoya en servicios en la nube como Firebase. Cualquier interrupción del servicio, cambio en las políticas de uso o en el modelo de precios por parte del proveedor (Google) representa un riesgo directo para la disponibilidad, operatividad y costes del proyecto.

## 12 ORGANIZACIÓN Y GESTIÓN DEL PROYECTO

### 12.1 Organización

#### 12.1.1 Arquitectura del sistema

BlutTruck utiliza una arquitectura escalable y modular con comunicación basada en **APIs** y un **servidor por capas**. Sus componentes principales son:

- **Cliente Web:** Desarrollado con **Vue3 (SakaiVue)**, proporciona la interfaz de usuario y se comunica con el backend vía **API REST**.
- **Aplicación Móvil:** Creada con **Flutter**, permite la recolección de datos y la interacción del usuario, comunicándose también con el servidor.
- **Servidor de Aplicación Principal (servidor):** El núcleo del sistema, gestiona la lógica de negocio y el acceso a datos. Implementado en **capas (Controller, Service, Repository)**, utiliza Data Transfer Objects(**DTOs**) para la transferencia de datos.

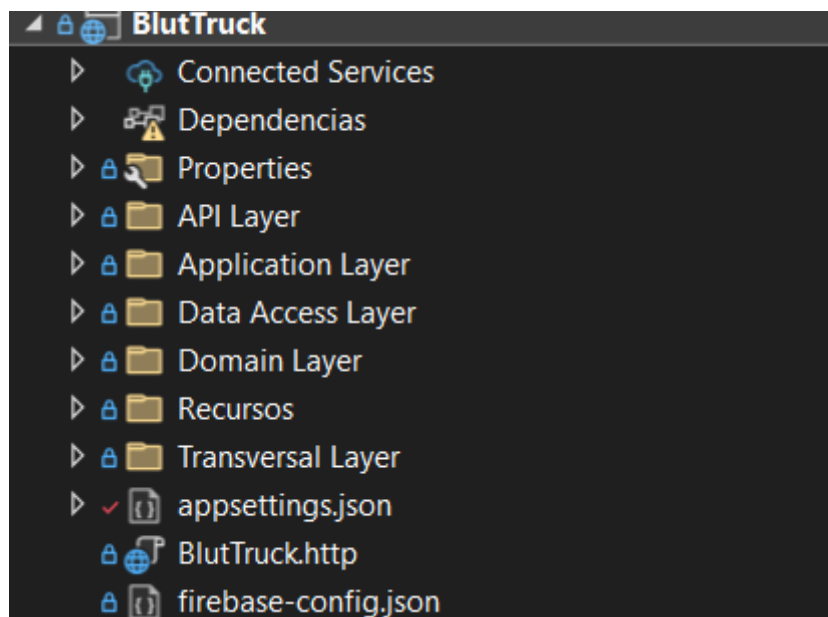


Ilustración 52 Capas Servidor

Ilustración 52 se puede ver las Capas, API Layer(Controller), Application Layer (Services), Data Acces Layer (Repositories), además de otras capas que sirven de ayuda para guardar imágenes, configuraciones, etc

- **Base de Datos Principal: Firebase Realtime Database (NoSQL)** almacena y sincroniza datos en tiempo real. La capa de Repositorio del backend interactúa directamente con ella.

- **Servidor de Inteligencia Artificial (Flask):** Un microservicio especializado en **Python (Flask)** que expone un endpoint para predecir la probabilidad de un ataque cardíaco. El Servidor de Aplicación Principal consume este endpoint para obtener las predicciones.

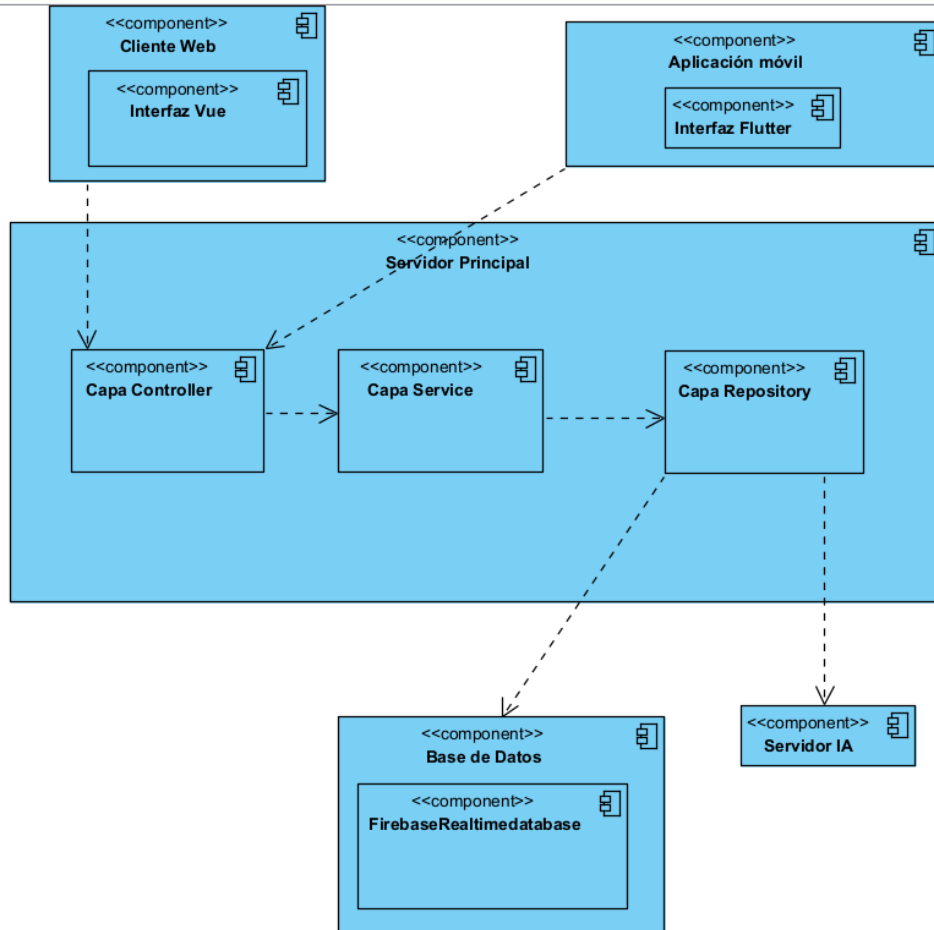


Ilustración 53 Estructura sistema

La conexión entre Cliente y servidor y entre servidores se hacen a través de json tanto en la web como en la aplicación se crean variables con la misma estructura del DTO que espera recibir el Servidor. En la Ilustración 53, puede ver como está formado el sistema y que estructuras tiene.

### 12.1.2 DIAGRAMA DE BASE DE DATOS

Para la gestión de datos del proyecto, se ha optado por utilizar **Firestore Realtime Database**, una solución de base de datos NoSQL alojada en la nube. Esta elección se fundamenta en varios factores clave que se alinean con los requisitos del proyecto, a pesar de implicar consideraciones distintas a las bases de datos relacionales tradicionales alojadas localmente, como posibles costes asociados al uso y una configuración específica.

La principal ventaja de Firebase Realtime Database radica en su capacidad para ofrecer **acceso constante y ubicuo a los datos**, al estar alojada en la nube. Esto facilita la colaboración y el acceso desde múltiples dispositivos y ubicaciones.

Firebase Realtime Database se basa en un **modelo de datos NoSQL orientado a documentos** (específicamente, un gran árbol JSON). Este enfoque permite una **mayor flexibilidad en la estructura de los datos** en comparación con los esquemas rígidos de las bases de datos SQL. Además, la recuperación de grandes conjuntos de datos o de estructuras de datos complejas anidadas puede ser más directa, ya que los datos relacionados suelen almacenarse juntos en el mismo documento JSON.

Otro aspecto relevante es la **simplicidad en las operaciones de lectura y escritura** y la **sincronización en tiempo real**. Los datos se actualizan automáticamente en todos los clientes conectados cuando se produce un cambio, lo que resulta en una experiencia de usuario fluida y reactiva. Esto es especialmente beneficioso para aplicaciones que requieren una alta interactividad y datos actualizados al instante.

Debido a estas características y a la naturaleza del modelo NoSQL de Firebase Realtime Database, la representación de la estructura de la base de datos no se realizará mediante un diagrama Entidad-Relación tradicional, que es propio de las bases de datos relacionales. En su lugar, la estructura se documentará describiendo la jerarquía de los nodos JSON, sus campos y las relaciones implícitas entre los datos, reflejando cómo la información es almacenada y accedida por la aplicación.

La base de datos se organiza a partir de nodos raíz que contienen la información categorizada, como se observa en la Ilustración 54:



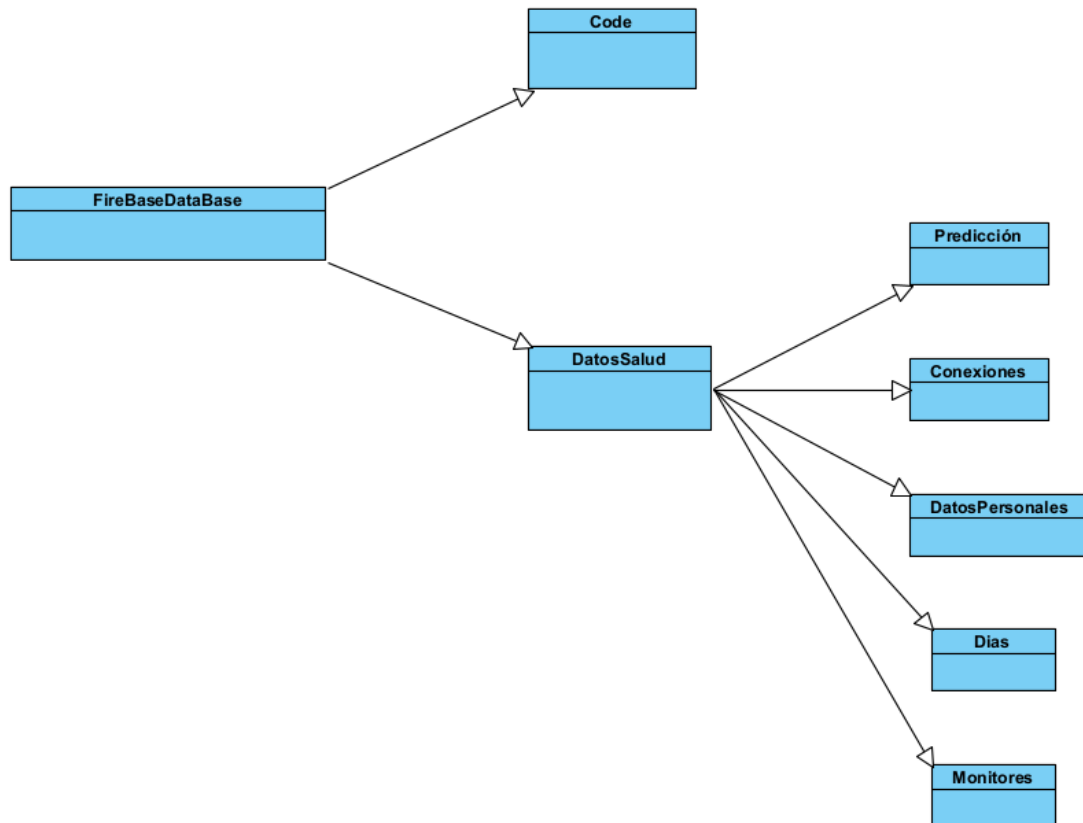


Ilustración 54 Estructura base de datos

### 12.1.3 Desarrollo de la solución propuesta

Tal y como se describe en el apartado **5.1. Métodos**, el desarrollo de la aplicación BlutTruck se ha guiado por una metodología ágil, específicamente una adaptación de Scrum, y un diseño centrado en el usuario. Este enfoque ha implicado una atención constante a las necesidades y expectativas de los usuarios finales. Desde el inicio del proyecto, se ha recogido y evaluado de manera informal la retroalimentación de potenciales usuarios y del tutor, lo que ha permitido mejorar la aplicación de forma iterativa.

Inicialmente, la propuesta se centró en desarrollar una solución integral para la monitorización de la salud, con especial énfasis en la prevención de riesgos cardíacos. El objetivo era que la aplicación no solo permitiera al usuario registrar y visualizar sus propios datos, sino que también ofreciera la innovadora capacidad de supervisar la salud de terceros, como familiares. La evolución del proyecto se puede desglosar en las siguientes fases:

#### 12.1.3.1 Análisis del panorama actual

Para estructurar el proyecto y validar la propuesta de valor, se llevó a cabo un análisis de las soluciones existentes en el mercado de la monitorización de salud. Se concluyó que, si bien existían herramientas centradas en el usuario individual,

había una brecha importante en el cuidado colaborativo, con funcionalidades limitadas para la monitorización por parte de terceros y la notificación proactiva a cuidadores.

#### 12.1.3.2 Definición de funcionalidades

A diferencia de un proceso formal de entrevistas, la definición de funcionalidades se nutrió de la investigación inicial y de consultas informales con el entorno cercano, incluyendo la valiosa perspectiva médica mencionada en los agradecimientos. Este proceso ayudó a validar las necesidades reales y a consolidar las características clave de BlutTruck, que se diferenciaba por su enfoque en la monitorización asistida. Se identificaron las siguientes funcionalidades esenciales para la aplicación:

- Monitorización de datos de salud propios y generación de gráficos de progreso.
- Recolección de datos a través de *smartwatches* y su sincronización mediante Health Connect.
- Supervisión de la salud de terceros con diferentes roles (observador y administrador).
- Implementación de un sistema de inteligencia artificial para analizar los datos, detectar patrones de riesgo y generar alertas.
- Envío de notificaciones a los supervisores en caso de detectar anomalías.
- Gestión de perfiles de usuario, con un área personal para la visualización de datos históricos.
- Registro y autenticación de usuarios de forma autónoma y segura.

#### 12.1.3.3 Prototipado

Con un conjunto de ideas ya establecidas, se procedió a la fase de prototipado. En lugar de utilizar herramientas de diseño visual como Figma para la aplicación web, se optó por un enfoque más práctico, creando un prototipo de alta fidelidad directamente con Vue.js. Este "prototipo en código" no solo sirvió para validar la viabilidad técnica, sino que también constituyó la base del *frontend* final, optimizando los tiempos de desarrollo. Para la aplicación móvil, se crearon prototipos funcionales específicos para asegurar la correcta integración con Firebase y Health Connect.

#### 12.1.3.4 Programación

Una vez validados los prototipos, se procedió a la programación de la solución completa, desarrollando los distintos componentes de la arquitectura: la aplicación móvil en Flutter, la plataforma web en Vue.js, el servidor principal en C# (.NET 8) y el microservicio de inteligencia artificial en Python.

#### 12.1.3.5 Pruebas de usuario con los prototipos

Se realizaron pruebas de usabilidad con un grupo reducido de usuarios (familiares y amigos) para verificar que la aplicación cumpliera con las expectativas de simplicidad y facilidad de uso.

#### 12.1.3.6 Validación y ajustes menores

La retroalimentación de las pruebas fue mayoritariamente positiva, validando la simplicidad de la navegación y la claridad de las funciones. Aunque se detectó que el proceso de configuración inicial de Health Connect presentaba cierta complejidad para los usuarios, se concluyó que esta era una fricción técnica ineludible y externa a la lógica de la aplicación. Por tanto, al haber sido validado el diseño, se procedió con el desarrollo final sin necesidad de implementar modificaciones significativas en la estructura visual o de navegación.

## 12.2 Gestión del proyecto

La **planificación temporal** es un componente esencial en la gestión de proyectos, actuando como un indicador clave del progreso y permitiendo evaluar el impacto de posibles desviaciones sobre la fecha de finalización. Un seguimiento riguroso del cronograma es fundamental para asegurar la consecución de los objetivos dentro de los plazos establecidos.

Para el desarrollo de este proyecto, se ha adoptado una **metodología ágil**, caracterizada por su enfoque iterativo y adaptativo. Esta metodología se estructura en **hitos** bien definidos, cada uno de ellos coincidiendo con una reunión de seguimiento con el tutor. Tras la finalización de cada hito, se procederá a la **documentación** detallada de los avances en los anexos y la memoria del proyecto. Asimismo, se realizará un **ajuste y redefinición de las tareas futuras** basándose en el *feedback* recibido, garantizando así la continua mejora y adaptación del plan de trabajo a lo largo del desarrollo del proyecto. Este enfoque permite una mayor flexibilidad y capacidad de respuesta ante los desafíos que puedan surgir, optimizando la eficiencia del proceso.

A continuación, se presentarán dos diagramas de Gantt para ilustrar la gestión temporal del proyecto:

- **Diagrama de Planificación:** Este diagrama representa la **línea base inicial** del proyecto, detallando las previsiones de tiempo estimadas para cada tarea. Sirve como una guía estratégica para la ejecución, reflejando las expectativas originales del desarrollo.
- **Diagrama Real:** Este diagrama, elaborado una vez finalizado el proyecto, muestra el **tiempo real invertido** en la realización de cada una de las tareas. Su comparación con el diagrama de planificación permitirá un análisis

retrospectivo de la eficiencia en la gestión del tiempo y la identificación de las posibles desviaciones, proporcionando valiosas lecciones aprendidas para futuros proyectos.

### 12.2.1      Calendario de trabajo

El calendario de trabajo, esencial para el desarrollo del proyecto, se detalla en la Ilustración 55. Se han establecido dos tipos de horarios distintos debido a la necesidad de compaginar el proyecto con las asignaturas de cuarto FSI y SSI, además de trabajar todos los días de 7 a 15 de lunes a viernes.

Por tanto, se ha hecho un calendario de 15 a 20:00, Ilustración 56, de lunes a viernes quitado la época de exámenes en Navidad que se usó para estudiar, además los fines de semanas como se empezó el proyecto en octubre se usaron para descansar.

Cambiar calendario laboral

Para calendario: Estándar (Calendario del proyecto)

Crear calendario...

El calendario 'Estándar' es un calendario base.

Leyenda:

Laborable

No laborable

31 Horas laborables modificadas

En este calendario:

31 Día de excepción

31 Semana laboral no predeterminada

Haga clic en un día para ver sus periodos laborables:

25 diciembre 2024 es no laborable.

diciembre 2024

L	M	M	J	V	S	D
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Basado en:

Excepción 'Periodo estudio P...' en el calendario 'Estándar'.

Excepciones   Semanas laborales

	Nombre	Comienzo	Fin
1	Periodo estudio Primera convocatoria y exámenes	25/12/2024	12/01/2025

Dgtalles...

Eliminar

Ayuda

Opciones...

Aceptar

Cancelar

Ilustración 55 Calendario Project

100

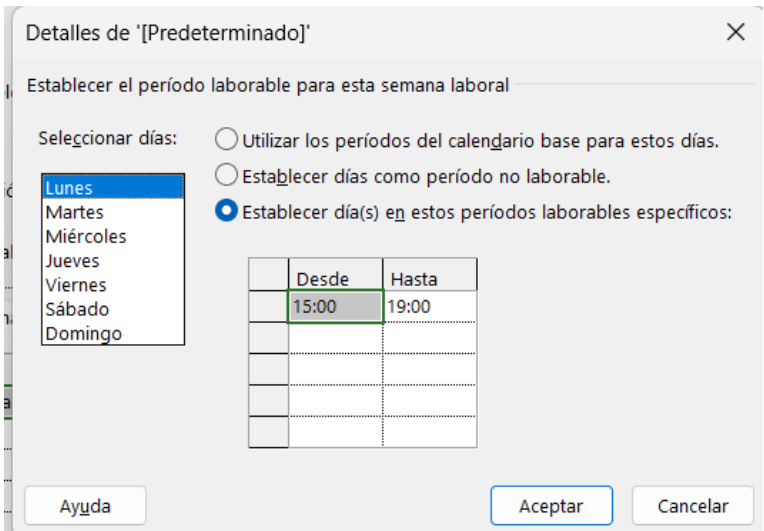


Ilustración 56 Horas Laborables

12.2.2 Diagrama de Gantt planificado

El diagrama de Gantt resultante de la planificación refleja las previsiones establecidas para la ejecución del proyecto. La duración asignada a las tareas es una estimación, un hito no comienza antes de que finalice el anterior, el resumen de los hitos se puede ver en la Ilustración 57.

	Planificación personal	162,5 días	mar 01/10/24	mar 03/06/25
	HITO 1	26 días	mar 01/10/24	mié 06/11/24
	HITO 2	31 días	mié 06/11/24	jue 19/12/24
	HITO 3 Desarrollo backend	40,5 días	jue 19/12/24	mar 04/03/25
	HITO 4 Desarrollo Frontend	27 días	mar 04/03/25	jue 10/04/25
	HITO 5 Terminar documentación y mejoras	38 días	jue 10/04/25	mar 03/06/25

Ilustración 57 Diagrama de Grantt planificado

**Hito 1: Identificación y Conceptualización Inicial.** Este hito, con una duración de 26 días, se enfoca en las etapas iniciales del proyecto. Comienza con la identificación de un problema y la búsqueda de las primeras ideas, seguido de una reunión clave con Iván Álvarez Navia. Posteriormente, se revisan las ideas y se investigan métodos para la recolección de datos mediante pulseras. El hito concluye con la investigación sobre las mejores tecnologías y posibles arquitecturas, y la elaboración de la primera parte de la documentación. Las tareas están secuenciadas para asegurar una progresión lógica desde la conceptualización hasta la base técnica inicial y la documentación preliminar, como se observa en la Ilustración 58.









	Planificación personal	162,5 días	mar 01/10/24	mar 03/06/25
	HITO 1	26 días	mar 01/10/24	mié 06/11/24
	Identificación de un problema y búsqueda de primeras ideas	14,5 días	mar 01/10/24	lun 21/10/24
	Reunión con Iván Álvarez Navia	0,5 días	mar 22/10/24	mar 22/10/24
	Revisión de ideas	1,5 días	mié 23/10/24	jue 24/10/24
	Búsqueda de maneras para recoger datos pulsera	2 días	jue 24/10/24	lun 28/10/24
	Investigación sobre mejores tecnología y posibles arquitectura	3 días	lun 28/10/24	jue 31/10/24
	Primeras parte de documentación	4 días	jue 31/10/24	mié 06/11/24

Ilustración 58 Diagrama de Grantt planificado Hito1

**Hito 2: Desarrollo de Conectividad y Funcionalidades Clave** Este hito, que abarca un total de 31 días, se centra en la implementación de la **infraestructura de conectividad** y el desarrollo de **funcionalidades esenciales de la aplicación móvil**. Se inicia con una segunda reunión con Iván, seguida rápidamente por la configuración de la conexión con **Firestore**. Posteriormente, se establecerá la conexión con **HealthConnect**, un paso crucial para la integración de datos de salud. Las fases siguientes se dedicarán a la funcionalidad de escaneo, la conexión con usuarios y la configuración de perfiles, así como a la implementación del método Foreground. El hito concluye con tareas destinadas al aumento de la seguridad y cambios en la base de datos, asegurando la robustez y protección del sistema, como se detalla en la Ilustración 59.








	HITO 2	31 días	mié 06/11/24	jue 19/12/24
	Segunda Reunión con Iván	0 días	mié 06/11/24	mié 06/11/24
	Conexión con Firestore	2,5 días	mié 06/11/24	vie 08/11/24
	Conexión con HealthConnect	6 días	lun 11/11/24	lun 18/11/24
	Funcionlidad escane, conexión con usuarios y perfil	9 días	mar 19/11/24	vie 29/11/24
	Método Foreground	5,5 días	lun 02/12/24	lun 09/12/24
	Aumento seguridad y cambio en base de datos	8 días	lun 09/12/24	jue 19/12/24

Ilustración 59 Diagrama de Grantt planificado Hito 2

**Hito 3: Desarrollo servidor.** Este hito, con una duración total de 40.5 días, se dedica por completo a la **construcción de la infraestructura backend** tanto para la aplicación móvil como para la web. Comienza con una tercera reunión con Iván, seguida de las configuraciones iniciales del **servidor .NET**. Un componente clave de este hito es el **entrenamiento de la inteligencia artificial** y la **creación del servidor Flask**.

Posteriormente, se procederá a la creación de los **endpoints principales** para las operaciones CRUD y de perfil, y se continuará con la creación de endpoints dedicados a guardar datos y realizar predicciones. El hito culmina con la creación de los endpoints restantes, completando así la funcionalidad del backend, como se detalla en la Ilustración 60.

	4 HITO 3 Desarrollo backend	40,5 días	jue 19/12/24	mar 04/03/25
	Tercera Reunión con Iván	0 días	jue 19/12/24	jue 19/12/24
	Configuraciones iniciales del servidor .NET	2 días	vie 20/12/24	lun 23/12/24
	Entrenamiento de IA y creación del servidor Flask	10 días	mar 24/12/24	jue 23/01/25
	Creación de endpoints principales de CRUD y perfil	3,5 días	vie 24/01/25	mié 29/01/25
	Creación de endpoints de guardar datos y predecir	4,5 días	mié 29/01/25	mar 04/02/25
	Creación de endpoints restantes	20 días	mié 05/02/25	mar 04/03/25

Ilustración 60 Diagrama de Grantt planificado Hito 3

**Hito 4: Desarrollo Frontend.** Este hito, con una duración de 27 días, se enfoca en la **construcción de la interfaz de usuario de la aplicación**. Comienza con una cuarta reunión con Iván, seguida de la búsqueda de plantillas adecuadas para el diseño. Las tareas clave incluyen la creación de las **vistas CRUD y del Dashboard**, así como la **vista para guardar datos** y la **creación de la vista de redes**. El hito concluye con la implementación de las vistas restantes, completando así la experiencia visual y la interactividad del usuario en la aplicación, como se muestra en la Ilustración 61.

	<b>HITO 4 Desarrollo Frontend</b>	<b>27 días</b>	<b>mar 04/03/25</b>	<b>jue 10/04/25</b>
	Cuarta Reunión Iván	0 días	mar 04/03/25	mar 04/03/25
	Búsqueda de plantil	2 días	mié 05/03/25	jue 06/03/25
	Creación Vistas CRUD y Dashboard	6 días	vie 07/03/25	vie 14/03/25
	Creación vista guardar datos y	3 días	lun 17/03/25	mié 19/03/25
	Creación Vista de redes y	8 días	jue 20/03/25	lun 31/03/25
	Creación de Vistas restantes	8 días	mar 01/04/25	jue 10/04/25

Ilustración 61 Diagrama de Grantt planificado Hito 4

**Hito 5: Finalización de Documentación y Mejoras.** Este hito, con una duración total de 38 días, marca la etapa final del proyecto, centrándose en la **consolidación de la documentación** y la **implementación de mejoras**. Inicia con la última reunión con Iván. Posteriormente, se dedicará tiempo a la elaboración de los **Anexos I, II, III y IV**, cada uno con una duración de cinco días. La redacción de la **Memoria del proyecto**, una pieza central de la documentación, se llevará a cabo durante ocho días. Finalmente, el hito culmina con un período de diez días dedicado a **mejoras tanto en la documentación existente como en el código de la aplicación**, asegurando la calidad y completitud del entregable final, como se detalla en la Ilustración 62.









	<b>HITO 5 Terminar documentación y mejoras</b>	<b>38 días</b>	<b>jue 10/04/25</b>	<b>mar 03/06/25</b>
	Última reunión Iván	0 días	jue 10/04/25	jue 10/04/25
	Anexo I	5 días	vie 11/04/25	jue 17/04/25
	Anexo II	5 días	vie 18/04/25	jue 24/04/25
	Anexo III	5 días	vie 25/04/25	jue 01/05/25
	AnexoIV	5 días	vie 02/05/25	jue 08/05/25
	Memoria del proyecto	8 días	vie 09/05/25	mar 20/05/25
	Mejoras de documentación y código	10 días	mié 21/05/25	mar 03/06/25

Ilustración 62 Diagrama de Grantt planificado Hito 5

El diagrama de planificación finaliza con un tiempo de trabajo de 815 horas, en comparación con las 942 horas estimadas por EZEstimate, lo cual es considerablemente menor.

### 12.2.3 Diagrama de Gantt real

El diagrama de Gantt elaborado al finalizar el trabajo muestra el tiempo realmente invertido en cada tarea. Si bien los hitos permanecen inalterados, las estimaciones de



tiempo han cambiado, resultando en una finalización 27 días más tarde de lo previsto (Ilustración 63, Ilustración 64, Ilustración 65, Ilustración 66, Ilustración 67, Ilustración 68).

	▣ <b>Planificación personal</b>	<b>181,5 días</b>	<b>mar 01/10/24</b>	<b>lun 30/06/25</b>
	▷ <b>HITO 1</b>	<b>27,5 días</b>	<b>mar 01/10/24</b>	<b>jue 07/11/24</b>
	▷ <b>HITO 2</b>	<b>41,5 días</b>	<b>jue 07/11/24</b>	<b>jue 23/01/25</b>
	▷ <b>HITO 3 Desarrollo Backend</b>	<b>47 días</b>	<b>jue 23/01/25</b>	<b>lun 31/03/25</b>
	▷ <b>HITO 4 Desarrollo Frontend</b>	<b>29,5 días</b>	<b>lun 31/03/25</b>	<b>sáb 10/05/25</b>
	▷ <b>HITO 5 Terminar documentación y mejoras</b>	<b>36 días</b>	<b>sáb 10/05/25</b>	<b>lun 30/06/25</b>

Ilustración 63 Diagrama de Grantt real








	▣ <b>HITO 1</b>	<b>27,5 días</b>	<b>mar 01/10/24</b>	<b>jue 07/11/24</b>	
	Identificación de un problema y búsqueda de primeras ideas	14,5 días	mar 01/10/24	lun 21/10/24	
	Reunión con Iván Álvarez Navia	0,5 días	mar 22/10/24	mar 22/10/24	3
	Revisión de ideas	1,5 días	mar 22/10/24	mié 23/10/24	4
	Búsqueda de maneras para recoger datos pulsera	2 días	jue 24/10/24	vie 25/10/24	5
	Investigación sobre mejores tecnología y posibles arquitectura	3 días	lun 28/10/24	jue 31/10/24	6
	Primeras parte de documentación	5,5 días	jue 31/10/24	jue 07/11/24	7

Ilustración 64 Diagrama de Grantt real Hito 1

	<b>HITO 2</b>	<b>41 días</b>	<b>jue 07/11/24</b>	<b>mié 22/01/25</b>	
	Segunda Reunión con Iván	0 días	jue 07/11/24	jue 07/11/24	2
	Conexión con Firebase	3,5 días	vie 08/11/24	mié 13/11/24	10
	Conexión con HealthConnect	6,5 días	mié 13/11/24	jue 21/11/24	11
	Funcionlidad escane, conexión con usuarios y perfil	9 días	vie 22/11/24	mié 04/12/24	12
	Método Foreground	8 días	jue 05/12/24	lun 16/12/24	13
	Aumento seguridad y cambio en base de datos	14 días	mar 17/12/24	mié 22/01/25	14

Ilustración 65 Diagrama de Grantt real Hito 2








	<b>HITO 3 Desarrollo Backend</b>	<b>50 días</b>	<b>vie 17/01/25</b>	<b>vie 28/03/25</b>	
	Tercera Reunión con Iván	0 días	vie 17/01/25	vie 17/01/25	9
	Configuraciones iniciales del servidor .NET	2 días	lun 20/01/25	mar 21/01/25	9
	Entrenamiento de IA y creación del servidor Flask	10,5 días	mié 22/01/25	mié 05/02/25	18
	Creación de endpoints principales de CRUD y perfil	3,5 días	mié 05/02/25	lun 10/02/25	19
	Creación de endpoints de guardar datos y predecir	4,5 días	mar 11/02/25	lun 17/02/25	20
	Creación de endpoints restantes	26,5 días	jue 20/02/25	vie 28/03/25	21

Ilustración 66 Diagrama de Grantt real Hito 3

	<b>HITO 4 Desarrollo Frontend</b>	<b>30 días</b>	<b>vie 28/03/25</b>	<b>sáb 10/05/25</b>	
	Cuarta Reunión Iván	0 días	vie 28/03/25	vie 28/03/25	16
	Búsqueda de plantilla	2,5 días	mar 01/04/25	jue 03/04/25	16
	Creación Vistas CRUD y Dashboard	5 días	lun 07/04/25	vie 11/04/25	
	Creación vista guardar datos y perfil	3,5 días	lun 14/04/25	jue 17/04/25	26
	Creación Vista de redes y predicciones	9,5 días	vie 18/04/25	jue 01/05/25	27
	Creación de Vistas restantes	6,5 días	jue 01/05/25	sáb 10/05/25	28

Ilustración 67 Diagrama de Grantt real Hito 4









	<b>HITO 5 Terminar documentación y mejoras</b>	<b>36 días</b>	<b>sáb 10/05/25</b>	<b>lun 30/06/25</b>	<b>29</b>
	Última reunión Iván	0 días	sáb 10/05/25	sáb 10/05/25	23
	Anexo I	4 días	mié 14/05/25	lun 19/05/25	23
	Anexo II	3,5 días	mar 20/05/25	vie 23/05/25	32
	Anexo III	3 días	lun 26/05/25	jue 29/05/25	33
	Anexo IV	4 días	jue 29/05/25	mié 04/06/25	34
	Memoria del proyecto	5 días	jue 12/06/25	mié 18/06/25	35
	Mejoras de documentación y código	5 días	mar 24/06/25	lun 30/06/25	36

Ilustración 68 Diagrama de Grantt real Hito 5

Se puede observar cómo el proyecto se retrasó 19 días más de lo previsto y con ello nos dio 910 horas reales, lo cual se aproxima más a las 942 horas estimadas por ZEstimate, evidenciando una mayor precisión en la estimación del tiempo.

## 13 CONCLUSIONES Y TRABAJO A FUTURO

### 13.1 Conclusiones

El proyecto BlutTruck representa un avance significativo en la **monitorización de la salud cardíaca** mediante el uso de tecnologías innovadoras. Su objetivo principal, crear una **herramienta accesible y de bajo costo** para la detección temprana de posibles riesgos de ataques cardíacos, se ha logrado satisfactoriamente, ofreciendo una solución integral que va más allá de la automonitorización, al incluir la **supervisión de terceros**.

#### 13.1.1 Logros Clave del Proyecto

El desarrollo de BlutTruck ha cumplido con sus ambiciosos **objetivos funcionales** y ha permitido alcanzar importantes **metas personales** para el autor:

- **Enfoque en la Prevención:** La aplicación prioriza la detección temprana de anomalías, capacitando a los usuarios para conocer su riesgo estimado en tiempo real y fomentar la adopción de medidas preventivas. Aunque no ofrece un valor médico oficial, actúa como una **ayuda preventiva** crucial.
- **Diseño Centrado en el Usuario y Accesibilidad:** Se ha desarrollado una interfaz **intuitiva y accesible** tanto en la aplicación móvil como en la plataforma web, facilitando su uso a un amplio espectro de usuarios, incluyendo a personas mayores o con poca familiaridad tecnológica. La simplicidad del diseño, validada por prototipos y pruebas de usuario, es un pilar fundamental.
- **Monitorización Delegada y Colaborativa:** Una de las funcionalidades más destacadas es la capacidad de **supervisar la salud de otras personas** (familiares o amigos) con diferentes niveles de acceso (observador y administrador), siempre con el consentimiento explícito. Esto facilita el **cuidado asistido** y permite a los cuidadores recibir alertas ante patrones preocupantes.
- **Integración Tecnológica Avanzada:** El proyecto ha combinado de forma estratégica diversas tecnologías de vanguardia:
  - **Frontend: Vue.js** para la plataforma web y **Flutter/Dart** para la aplicación móvil, garantizando interfaces interactivas y multiplataforma.
  - **Backend: C# (ASP.NET Core)** y **.NET 8** para una lógica de servidor robusta y escalable, siguiendo principios de Clean Architecture.

- **Inteligencia Artificial: Python** con librerías como Scikit-learn, XGBoost, LightGBM y CatBoost para desarrollar un **modelo predictivo** preciso, accesible vía API (Flask), que analiza datos y genera alertas. El uso de **SVM** como modelo definitivo destaca por su alta fiabilidad.
- **Gestión de Datos: Firebase** (Realtime Database, Storage y Authentication) para un almacenamiento seguro y sincronización en tiempo real, lo que es crucial para los datos de salud.
- **Recopilación de Datos: Health Connect** (con configuración en Kotlin) para integrar de manera segura los datos de salud de smartwatches y otros dispositivos *wearables*.
- **Seguridad y Privacidad de Datos:** Se ha puesto un énfasis primordial en la **seguridad integral** y la **privacidad** de los datos sensibles de salud, cumpliendo con la LOPDGDD y el RGPD, y asegurando el control del usuario sobre su información.

### 13.1.2 Desafíos Superados y Lecciones Aprendidas

El proyecto ha sido un campo de aprendizaje intensivo, especialmente al abordar **tecnologías desde cero** y trabajar con las **limitaciones de tiempo** inherentes a un Trabajo de Fin de Grado. La **metodología ágil** (Scrum adaptado) ha sido clave para gestionar estos retos, permitiendo una adaptación flexible y una mejora continua a través de iteraciones y retroalimentación constante con el tutor.

La creación de una **IA propia** para el análisis de riesgos, aunque más segura al mantener los datos dentro del sistema, subraya la dependencia de la cantidad y calidad de los datos de entrenamiento para su efectividad, un desafío intrínseco en este tipo de desarrollos.

## 13.2 Impacto y Futuro de BlutTruck

BlutTruck no solo mejora la prevención individual, sino que también contribuye al **impulso de la salud digital y la inteligencia artificial en medicina**. Su diseño facilita el diagnóstico temprano y un seguimiento optimizado para profesionales médicos.

Mirando hacia el futuro, el proyecto tiene un **gran potencial de expansión**. Las líneas futuras podrían incluir la adaptación al ecosistema de Apple (iOS), la ampliación de tipos de datos biométricos soportados por Health Connect (como ECGs), y la continua mejora de la inteligencia artificial con más datos, lo que podría aumentar aún más la precisión y el valor de la aplicación en la prevención de eventos cardíacos.

## 14 BIBLIOGRAFÍA

[1] Dilsizian, S. E. y Siegel, E. L., 2014, “Artificial intelligence in medicine and cardiac imaging: harnessing big data and advanced computing to provide personalized medical diagnosis and treatment,” *Current Cardiology Reports*, vol. 16, no. 1, p. 441. [En línea]. Disponible en: <https://doi.org/10.1007/s11886-013-0441-8>.

[2] European Society of Cardiology, 25 Aug. 2021, “2021 ESC Guidelines on cardiovascular disease prevention in clinical practice,” *European Heart Journal*, vol. 42, no. 34, pp. 3227–3337. [Online].

Available: <https://doi.org/10.1093/eurheartj/ehab484>.

[3] Proyectos Ágiles, s.f., “Qué es Scrum”. [En línea]. Disponible en: <https://proyectosagiles.org/que-es-scrum/>.

[4] Lucidchart Content Team, n.d., “DIAGRAMAS DE CASOS DE USO,” *Lucidchart*. [Online]. Available: <https://lucid.co/es/plantillas/diagrama-de-caso-de-uso>

[5] Lucidchart Content Team, n.d., “Qué es un diagrama de secuencia,” *Lucidchart*. [Online].

Available: <https://www.lucidchart.com/pages/es/diagrama-de-secuencia>.

[6] Asana, n.d., “Análisis SWOT: qué es y para qué sirve (con ejemplos),” *Asana*. [Online].

Available: <https://asana.com/es/resources/swot-analysis>.

### Referencias de Herramientas y Tecnologías:

Google. (s.f.). **Dart**. Recuperado de <https://dart.dev>

Google. (s.f.). **Flutter**. Recuperado de <https://flutter.dev>

JetBrains y Google. (s.f.). **Kotlin**. Recuperado de <https://kotlinlang.org/>

Microsoft. (s.f.). **C#**. Recuperado de <https://docs.microsoft.com/en-us/dotnet/csharp/>

Microsoft. (s.f.). **.NET 8**. Recuperado de <https://dotnet.microsoft.com>

You, E. y Vue Core Team. (s.f.). **Vue.js**. Recuperado de <https://vuejs.org>

PrimeTek. (s.f.). **PrimeVue**. Recuperado de <https://primevue.org>

PrimeTek. (s.f.). **Sakai Vue**. Recuperado de <https://www.primefaces.org/sakai-vue/>

Tailwind Labs. (s.f.). **Tailwind CSS**. Recuperado de <https://tailwindcss.com>

San Martin Morote, E. & Vue Core Team. (s.f.). **Pinia**. Recuperado de <https://pinia.vuejs.org>

Google. (s.f.). **Health Connect**. Recuperado de

<https://developer.android.com/health-and-fitness/guides/health-connect>

Python Software Foundation. (s.f.). **Python**. Recuperado de <https://www.python.org>

\* (Para librerías de Python, aunque se mencionan en el texto, generalmente no se citan individualmente en la bibliografía a menos que sean un foco principal del trabajo o se refieran a un estudio específico que las use de forma particular. Sin embargo, para un TFG, podrías citar las más relevantes si hay un documento o publicación oficial.) \* **Scikit-learn**: Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830. \* **XGBoost**: Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794. \* **LightGBM**: Ke, G., et al. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, 314s9-3157. \* **CatBoost**: Prokhorenkova, L., et al. (2017). CatBoost: unbiased boosting with categorical features. *arXiv preprint arXiv:1706.09516*. \* **Joblib**: The joblib developers. (s.f.). *joblib*. Recuperado de <https://github.com/joblib/joblib> (o DOI: <https://doi.org/10.5281/zenodo.14915601>)

Microsoft. (s.f.). **Visual Studio Code (VS Code)**. Recuperado de <https://code.visualstudio.com/>

Microsoft. (s.f.). **Visual Studio**. Recuperado de

<https://visualstudio.microsoft.com/es/>

Google y JetBrains. (s.f.). **Android Studio**. Recuperado de

<https://developer.android.com/studio>

Git. (s.f.). **Git**. Recuperado de <https://git-scm.com/>

GitHub, Inc. (s.f.). **GitHub**. Recuperado de <https://github.com/>

Google. (s.f.). **Firebase**. Recuperado de <https://firebase.google.com>

Anaconda, Inc. (s.f.). **Anaconda**. Recuperado de <https://www.anaconda.com>

Project Jupyter. (s.f.). **Jupyter Notebook**. Recuperado de <https://jupyter.org>

Microsoft. (s.f.). **Microsoft Project**. Recuperado de <https://www.microsoft.com/es-es/microsoft-365/project/project-management-software>

Visual Paradigm International Ltd. (s.f.). **Visual Paradigm**. Recuperado de <https://www.visual-paradigm.com/>

Página web de BlutTruck: <https://bluttruck.grial.eu/>

Descarga APK: <https://drive.google.com/file/d/1tQHDJZnTuWvE59biR-ZVCVwrcH3yRasf/view?usp=sharing>