

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA E INFORMÁTICA
CONCEPCIÓN – CHILE



**“APLICACIÓN MÓVIL PARA EL REGISTRO Y GESTIÓN
DE DONACIONES DE HOGARES DE MENORES”**

NICOLE MELANIE GALLARDO FIERRO

**MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERA EN INFORMÁTICA**

**Profesor Guía:
Cristian Antonio Lara Valenzuela**

Agosto - 2025



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): Memoria o trabajo de título; Tesis de Postgrado;

Título del trabajo: Aplicación móvil para el registro y gestión de donaciones de hogares de menores.

Nombre del candidato(a): Nicole Melanie Gallardo Fierro

Carrera / Grado: Ingeniería en Informática

Campus: Concepcion ; Departamento: Electronica e Informatica

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Cristian Antonio Lara Valenzuela, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

El trabajo **NO** contiene información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

6 meses; 12 meses; 2 años; 3 años; 5 años; 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis: **CRISTIAN LARA VALENZUELA** Firmado digitalmente por CRISTIAN LARA VALENZUELA
Fecha: 2025.07.31 00:31:42 -04'00'
Fecha: _____ ; Firma: _____

Estudiante o Candidato(a):

Fecha: 30/07/2025 ; Firma: _____

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

DEDICATORIA

A mi padre y madre, que me han apoyado incondicionalmente durante toda mi carrera. A mis amigos, quienes siempre estuvieron para alentarme, y a los profesores orientación constante y motivación.

AGRADECIMIENTOS

Quisiera expresar mi más sincero agradecimiento a la Universidad Técnica Federico Santa María por brindarme las herramientas necesarias para mi desarrollo académico. A mi compañero de proyecto por el esfuerzo y compromiso. Y a mi familia por su paciencia y apoyo incondicional.

RESUMEN

La presente tesis desarrolla el diseño e implementación de una aplicación móvil para Android orientada a transformar el sistema de donaciones a niños residentes en hogares de menores en Chile.

Este proyecto surge tras observar las dificultades actuales en el proceso de donación: falta de transparencia, complejidad en la gestión y limitaciones de los sistemas manuales tradicionales. Como respuesta, se propone una plataforma digital que busca mejorar significativamente la calidad de vida de los menores beneficiarios mediante un sistema más eficiente y accesible.

La plataforma integra diversas funcionalidades esenciales: un sistema diferenciado de registro para administradores de hogares y potenciales donantes, perfiles personalizados de los niños donde se detallan sus necesidades específicas, una galería interactiva que permite visualizar las solicitudes activas y un módulo de donaciones monetarias con pasarelas de pago seguras. La arquitectura tecnológica se basa en una autenticación mediante Firebase, y la utilización de servicios de AWS para garantizar escalabilidad y seguridad.

MySQL en el backend, todo soportado por servicios AWS que garantizan un funcionamiento continuo, seguro y escalable según la demanda.

En los objetivos que se espera alcanzar destacan: la optimización sustancial de los tiempos y recursos invertidos en el proceso de donación, la creación de una experiencia más transparente y accesible para quienes desean colaborar, y la generación de un impacto tangible y medible en los hogares participantes. Este trabajo constituye un aporte significativo al sector social chileno, al combinar innovación tecnológica con un claro enfoque de impacto comunitario, demostrando cómo las soluciones digitales pueden abordar problemáticas sociales complejas.

Palabras clave: Donaciones infantiles; Trazabilidad; Infancia vulnerable; Plataforma móvil; Firebase; MySQL; AWS.

ABSTRACT

This thesis develops the design and implementation of an Android mobile application aimed at transforming the donation system for children residing in children's homes in Chile.

This project arose after observing the current difficulties in the donation process: lack of transparency, management complexity, and limitations of traditional manual systems. In response, a digital platform is proposed that seeks to significantly improve the quality of life of child beneficiaries through a more efficient and accessible system.

The platform integrates several essential functionalities: a differentiated registration system for home administrators and potential donors, personalized child profiles detailing their specific needs, an interactive gallery that allows viewing active requests, and a monetary donation module with secure payment gateways. The technological architecture is based on authentication through Firebase and the use of AWS services to ensure scalability and security.

MySQL is used in the backend, all supported by AWS services that guarantee continuous, secure, and scalable operation based on demand. The expected objectives include: substantially optimizing the time and resources invested in the donation process, creating a more transparent and accessible experience for those wishing to contribute, and generating a tangible and measurable impact on participating households. This work constitutes a significant contribution to the Chilean social sector, combining technological innovation with a clear focus on community impact, demonstrating how digital solutions can address complex social problems.

Keywords: Child donations; Traceability; Vulnerable childhood; Mobile platform; Firebase; MySQL; AWS.

GLOSARIO

AWS (Amazon Web Services): Plataforma de servicios en la nube que proporciona soluciones de computación, almacenamiento, bases de datos, monitoreo y seguridad. En este proyecto se utiliza para hospedar la infraestructura, gestionar la base de datos (RDS), almacenar archivos (S3) y monitorear el rendimiento (CloudWatch).

API REST (Application Programming Interface - Representational State Transfer): Conjunto de reglas que permite la comunicación entre sistemas a través de peticiones HTTP. En esta tesis, se utiliza para conectar la aplicación móvil con el backend y acceder a la base de datos de forma eficiente.

Android: Sistema operativo basado en Linux para dispositivos móviles. Es el sistema sobre el cual se ejecuta la aplicación desarrollada en este proyecto.

Android Studio: Entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones Android. Utilizado para compilar y desplegar la aplicación móvil de este proyecto.

Autenticación: Proceso de verificación de identidad de los usuarios. En esta aplicación se implementa mediante Firebase Authentication, que permite el registro e inicio de sesión mediante correo electrónico y contraseña.

Base de Datos MySQL: Sistema de gestión de bases de datos relacional, utilizado en este proyecto para almacenar la información estructurada del sistema como usuarios, perfiles de niños, hogares, donaciones, entre otros.

CloudWatch (AWS): Servicio de monitoreo y observabilidad que permite recolectar y visualizar métricas de los recursos y servicios desplegados en AWS.

Donación: Acción voluntaria de entregar recursos materiales o monetarios a una causa social. En esta plataforma, puede realizarse en especie o mediante transferencias electrónicas.

Donante: Usuario de la plataforma que contribuye con recursos a las necesidades de los niños en hogares de menores.

EC2 (Elastic Compute Cloud): Servicio de AWS que permite lanzar y administrar instancias virtuales (servidores) en la nube. Utilizado para hospedar el backend de la aplicación.

Firebase Authentication: Servicio de autenticación provisto por Google que permite gestionar el registro, login y sesión de usuarios de manera segura. Implementado para facilitar el acceso a la plataforma tanto a donantes como administradores.

Firebase Storage: Servicio de almacenamiento de archivos en la nube ofrecido por Firebase. Utilizado en este proyecto para guardar imágenes asociadas a perfiles de niños y solicitudes.

Frontend: Parte visual de una aplicación con la que interactúan los usuarios. En este proyecto está desarrollado con React Native para dispositivos móviles.

Galería de niños: Sección de la aplicación que muestra los perfiles de los menores con sus respectivas necesidades. Permite a los donantes seleccionar a quién desean apoyar.

Hogar de menores: Institución que acoge a niños en situación de vulnerabilidad social, y que gestiona sus necesidades a través de la plataforma.

Infante: Niño o niña que reside en un hogar de menores y es beneficiario de las donaciones gestionadas a través de la aplicación.

Mapa interactivo: Funcionalidad que permite visualizar geográficamente la ubicación de los hogares de menores registrados en la plataforma.

Modelo de datos relacional: Diseño estructurado de la base de datos que organiza la información en tablas interrelacionadas, garantizando integridad y eficiencia en las consultas.

Node.js: Entorno de ejecución para JavaScript del lado del servidor, utilizado para desarrollar el backend de la plataforma.

Pasarela de pagos: Servicio que permite realizar transacciones monetarias de forma segura. Aunque no se implementó en esta versión, está contemplado como mejora futura.

Perfil de niño: Representación digital del infante dentro de la aplicación, incluyendo información como nombre, edad, necesidades específicas e imágenes.

Plataforma móvil: Aplicación desarrollada para dispositivos Android, que permite la interacción entre donantes y hogares de menores.

React Native: Framework de desarrollo de aplicaciones móviles multiplataforma basado en JavaScript. Utilizado para construir la interfaz de usuario de esta solución.

RDS (Relational Database Service): Servicio de base de datos administrado de AWS. Permite alojar, respaldar y escalar la base de datos MySQL usada en el proyecto.

Registro de usuarios: Proceso de creación de una cuenta en la plataforma. Está diferenciado por tipo de usuario: donante o administrador.

Seguridad: Conjunto de mecanismos implementados para proteger los datos y el acceso a la plataforma, incluyendo autenticación, permisos y cifrado de información.

Trazabilidad: Capacidad del sistema para registrar y mostrar el flujo completo de una donación, desde su ingreso hasta su entrega, promoviendo la transparencia.

Transparencia: Principio fundamental del proyecto, referido a la visibilidad y claridad con que los usuarios pueden verificar el uso de los recursos donados.

Usuario administrador: Representa a los responsables del hogar de menores. Tienen acceso a funcionalidades de gestión como registrar perfiles de niños y solicitudes de ayuda.

Webpay: Pasarela de pagos chilena utilizada para procesar transacciones electrónicas de forma segura. Se menciona en el flujo de donaciones.

Retrofit: Librería cliente HTTP utilizada en Android para consumir APIs REST de forma sencilla y segura.

GsonConverterFactory: Componente de Retrofit que permite convertir respuestas JSON en objetos Kotlin.

Google Maps SDK: Kit de desarrollo utilizado para integrar mapas y marcadores interactivos en la aplicación móvil.

Token JWT: Tipo de token seguro utilizado en procesos de autenticación para validar las sesiones de usuario (relacionado a Firebase).

Mockup: Representación gráfica de la interfaz de usuario de una aplicación, utilizada para mostrar diseño y funcionalidad antes del desarrollo real.

Caja Negra (Black-Box Testing): Tipo de prueba funcional que evalúa el comportamiento del sistema sin conocer su lógica interna, centrada en entradas y salidas esperadas.

Activity: Componente de Android que representa una pantalla con interfaz gráfica donde el usuario interactúa con la aplicación.

SharedPreferences: Sistema de almacenamiento clave-valor en Android utilizado para guardar datos ligeros como tokens de sesión o configuraciones.

S3 (Amazon Simple Storage Service): Servicio de AWS para almacenamiento de archivos e imágenes, usado para las galerías de infantes.

Material Design: Conjunto de directrices de diseño visual de Google que se aplican en los componentes de la aplicación para asegurar una interfaz moderna y coherente.

I. TABLA DE CONTENIDO

RESUMEN	5
ABSTRACT	6
GLOSARIO	7
I. Tabla de contenido	11
II. INDICE DE FIGURAS	13
III. INDICE DE TABLAS	14
INTRODUCCIÓN	15
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	16
1.1 Contextualización	16
1.2 Descripción del problema	16
1.3 Actores Involucrados	17
1.4 Consecuencias de No Abordar el Problema	17
1.5 Motivación para la Solución Propuesta	17
1.6 Objetivo general y específicos del proyecto	18
CAPÍTULO 2: MARCO CONCEPTUAL	19
2.1 Definición los procesos de donaciones de hogares de menores	19
2.2 Plataformas de Gestión y Registro de Donaciones	20
2.3 Metodologías	21
2.4 Modelos de arquitectura	21
2.5 Modelo de clases	22
2.6 Modelo de datos Relacional	23
2.7 Tecnologías para el desarrollo de software	23
2.8 Planes de pruebas funcionales	26
CAPÍTULO 3: PROPUESTA DE SOLUCION	27
3.1 Casos de Uso: Procesos de registro y gestión de donaciones	27
3.2. Modelo de arquitectura de software	41

3.3 Modelo de clases	42
3.4 Modelo Entidad-Relación (MER).....	45
3.5 Modelo de datos relacional.....	47
3.6 Ambiente de desarrollo	51
3.7 Códigos relevantes	53
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN	57
4.1 Plan de pruebas funcional del proyecto.	57
4.2 Validación de Funcionalidad Técnica.....	57
CAPÍTULO 5: CONCLUSIONES	60

II. INDICE DE FIGURAS

Ilustración 1: Diagrama de requerimientos de Registro de infante	28
Ilustración 2: Interfaz de registro de infante	28
Ilustración 3: Diagrama de requerimientos de Registro de Donante	29
Ilustración 4: Interfaz de creación de cuenta del Donante	30
Ilustración 5: Diagrama de requerimientos de Visualizar el perfil del Infante	31
Ilustración 6: Interfaces del proceso de visualización del infante.....	31
Ilustración 7: Diagrama de requerimiento de registrar donación. Fuente: Elaboración propia.	33
Ilustración 8: Interfaz de Registro de donación	34
Ilustración 9: Diagrama de requerimientos de realizar apadrinamiento Fuente: elaboración propia.	35
Ilustración 10: Interfaz de registrar Apadrinamiento. Fuente: Elaboración Propia.....	36
Ilustración 11: Diagrama de requerimientos de revisar pagos. Fuente: Elaboración propia.....	37
Ilustración 12: Interfaz de Revisión de pagos	38
Ilustración 13:Diagrama de requerimientos Revisar mapa. Fuente: elaboración propia.....	39
Ilustración 14:Interfaz Revisar Mapa. Fuente: Elaboración propia.	40
Ilustración 15: Diagrama de Arquitectura de Seguridad y Autenticación	41
Ilustración 16: Modelo de clases	42
Ilustración 17: Modelo Entidad-Relación. Fuente: Elaboración propia.	46
Ilustración 18: Modelo físico de datos. Fuente: elaboración Propia.	48
Ilustración 19: Configuración de Retrofit. Fuente: Elaboración Propia.	53
Ilustración 20: Endpoint de la API REST.	54
Ilustración 21: Código de seguridad (Autenticación) Fuente: Elaboración propia.	55
Ilustración 22: código Geolocalización de hogares beneficiarios. Fuente: Elaboración Propia.	55

III. INDICE DE TABLAS

Tabla 1: Pruebas unitarias y de integración. Fuente: Elaboración propia.	57
Tabla 2: Feedback De Usuarios. Fuente: elaboración propia.....	59

INTRODUCCIÓN

El presente trabajo aborda el desarrollo de una plataforma móvil orientada a cambiar la forma en que se realizan las donaciones de elementos esenciales a niños de hogares de menores. En el contexto actual, los hogares de menores enfrentan dificultades significativas para satisfacer las necesidades básicas de los niños, debido a la falta un sistema que conecte a los donantes con dichas necesidades de manera clara, accesible y transparente.

A pesar de la disposición de muchas personas y organizaciones para donar, los procesos manuales y la falta de información estructurada generan desconfianza, limitando el impacto de estas contribuciones. Además, los donantes suelen carecer de visibilidad sobre cómo se utilizan sus aportes, lo que afecta negativamente la continuidad de su apoyo.

Para abordar estos desafíos, se propone una plataforma tecnológica que no solo facilita las donaciones mediante contribuciones monetarias, sino que también garantiza la transparencia y eficiencia del proceso. La solución está diseñada para permitir a los administradores gestionar perfiles de los niños y sus solicitudes de artículos esenciales, mientras que los donantes podrán interactuar con la plataforma a través de una galería de publicaciones.

Este documento está estructurado en cinco capítulos. En el Capítulo 1, se define el problema, contextualizando la situación actual de los hogares de menores y las barreras para realizar donaciones efectivas. El Capítulo 2 presenta el marco conceptual, incluyendo las tecnologías y metodologías utilizadas en el desarrollo de la solución. En el Capítulo 3, se detalla la propuesta de solución, explicando los aspectos técnicos y funcionales del proyecto. El Capítulo 4 valida la solución mediante pruebas y análisis que demuestran su viabilidad y efectividad. Finalmente, el Capítulo 5 concluye con las principales contribuciones y recomendaciones del proyecto, destacando su impacto potencial en la comunidad.

Este proyecto no solo busca resolver una problemática específica, sino también sentar las bases para un modelo replicable en otros contextos sociales, utilizando la tecnología como herramienta para el cambio positivo. La integración futura de la pasarela de pagos es un área clave de mejora que se priorizará en versiones posteriores.

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

1.1 Contextualización

En Chile, más de 7.000 niños, niñas y adolescentes viven en hogares de menores pertenecientes al sistema de protección especializado, según datos del Servicio Nacional de Protección Especializada a la Niñez y Adolescencia (2023). Estos hogares acogen a menores que han sido vulnerados en sus derechos y requieren atención integral para satisfacer sus necesidades básicas, tales como vestimenta, higiene, alimentación, salud y educación.

La mayoría de estos hogares depende en gran medida de donaciones externas para complementar el financiamiento estatal, el cual frecuentemente es insuficiente para cubrir todos los requerimientos. Sin embargo, la forma en que se gestionan actualmente estas donaciones presenta múltiples deficiencias. Las solicitudes se realizan mediante redes sociales, correos electrónicos o llamados telefónicos, sin un sistema digital que permita formalizar, organizar o hacer seguimiento del proceso.

Un informe de Fundación Colunga (2022) revela que el 63% de las instituciones colaboradoras en Chile no cuenta con plataformas digitales especializadas para la gestión de donaciones, y un 72% de los hogares señala dificultades para sistematizar las necesidades de sus residentes. Esta situación no solo afecta la eficiencia en la entrega de ayudas, sino que también genera desconfianza en los donantes, quienes muchas veces no obtienen información clara sobre el destino de sus aportes.

Además, una encuesta exploratoria realizada como parte de este proyecto identificó que el 78% de los potenciales donantes estaría más dispuesto a colaborar si existiera un sistema que mostrara con claridad el destino de sus contribuciones, y el 65% considera que la falta de trazabilidad desincentiva su participación constante. Estos datos reflejan una oportunidad concreta para aplicar tecnologías que fortalezcan la transparencia, la eficiencia y la conexión directa entre quienes quieren ayudar y quienes lo necesitan.

1.2 Descripción del problema

Actualmente, los hogares de menores enfrentan varios desafíos:

Falta de un sistema eficiente para expresar necesidades: Los administradores de los hogares no cuentan con una estructura con plataforma para comunicar sus requerimientos específicos, lo que dificulta que los donantes puedan contribuir de manera precisa y oportuna.

Transparencia limitada: Muchos donantes no tienen claridad sobre cómo se utilizan sus aportes, lo que genera desconfianza y afecta su disposición a colaborar de manera constante.

Procesos de donación engorrosos: La ausencia de un sistema digital eficiente para coordinar las entregas de artículos o contribuciones monetarias genera fricción en el proceso, desincentivando la participación de posibles donantes.

1.3 Actores Involucrados

Administradores de hogares de menores: Encargados de identificar las necesidades de los niños y gestionar las donaciones.

Donantes o organizaciones: Personas o organizaciones, que buscan contribuir con recursos o contribución monetaria.

Niños en hogares: Beneficiarios directos de las donaciones.

1.4 Consecuencias de No Abordar el Problema

Si no se implementa una solución efectiva, los hogares de menores seguirán enfrentando limitaciones en la provisión de bienes esenciales. Perpetuando las condiciones precarias de los niños. Además, la desmotivación de los donantes podría aumentarse debido a la falta de confianza en los procesos actuales.

1.5 Motivación para la Solución Propuesta

Existen plataformas de donación en otros contextos, pero suelen enfocarse en modelos generalizados que no abordan las necesidades específicas de los niños en hogares de menores. Por ello, se requiere una solución personalizada y transparente que conecte directamente a los donantes con los beneficiarios.

La memoria tiene como objetivo desarrollar una plataforma móvil que resuelva estas problemáticas al permitir que los hogares gestionen perfiles de los niños y sus necesidades específicas. Además, se busca crear un espacio digital donde los donantes puedan confiar en la gestión y el impacto de sus contribuciones, fomentando un ciclo sostenible de apoyo.

En resumen, este proyecto no solo responde a una necesidad urgente en el contexto chileno, sino que también busca generar un modelo replicable que pueda ser adaptado a otras realidades sociales.

1.6 Objetivo general y específicos del proyecto

Objetivo General

Desarrollar una aplicación móvil para el registro y gestión de donaciones dirigidas a hogares de menores, con el fin de mejorar la eficiencia, trazabilidad y transparencia del proceso, mediante la aplicación de tecnologías móviles y servicios en la nube.

Objetivos específicos

1. Identificar y analizar los procesos actuales relacionados con la gestión de donaciones en hogares de menores, con el objetivo de detectar necesidades, limitaciones y oportunidades de mejora.
2. Diseñar un modelo funcional y técnico que incluya interfaces de usuario, flujos de procesos, estructura de almacenamiento de datos y una arquitectura de software escalable, en base a los requerimientos levantados.
3. Desarrollar e implementar una aplicación móvil funcional que permita el registro, visualización, seguimiento y gestión de donaciones, cumpliendo con las especificaciones técnicas definidas en la etapa de diseño.
4. Realizar pruebas de funcionalidad que permitan validar el correcto funcionamiento de la aplicación móvil, evaluando su rendimiento, usabilidad, conectividad y cumplimiento de los objetivos definidos.

CAPÍTULO 2: MARCO CONCEPTUAL

En este capítulo se describirá fundamentos teóricos y prácticos que sustentan el desarrollo de la aplicación móvil de ayuda a la infancia

2.1 Definición los procesos de donaciones de hogares de menores

Los procesos de donación en hogares de menores corresponden a un conjunto de acciones administrativas y operativas que permiten a estas instituciones recibir, gestionar y distribuir aportes materiales o económicos provenientes de personas naturales, empresas u organizaciones de la sociedad civil.

Estos procesos suelen desarrollarse en etapas informales, debido a la carencia de herramientas tecnológicas que estructuren la gestión de manera eficiente. En general, los pasos comunes observados en los hogares son los siguientes:

Identificación de necesidades:

Cada hogar realiza un levantamiento interno de las necesidades de los niños, niñas y adolescentes bajo su cuidado, considerando aspectos como vestuario, artículos de higiene, útiles escolares, juguetes, medicamentos o alimentos.

Divulgación de requerimientos:

Las necesidades detectadas se comunican a través de redes sociales, correos electrónicos o mensajes informales, sin un formato estandarizado ni trazabilidad sobre el estado de la solicitud.

Recepción de donaciones:

Los donantes responden a los llamados mediante entregas directas o transferencias bancarias, muchas veces sin claridad sobre el destino específico del aporte o sin recibir confirmación de su recepción.

Distribución y uso de recursos:

Una vez recibido el aporte, los hogares destinan los bienes o recursos según prioridad, pero no siempre existe un registro digital ni evidencia documentada que permita validar el cumplimiento del objetivo de la donación.

Falta de retroalimentación al donante

En la mayoría de los casos no se establece un canal de comunicación posterior con el donante, lo cual limita el sentido de transparencia y reduce la probabilidad de una donación recurrente.

2.2 Plataformas de Gestión y Registro de Donaciones

Las herramientas de gestión y registro de donaciones son plataformas tecnológicas que facilitan el proceso de recibir, organizar y distribuir aportes para causas sociales. Estas herramientas permiten a las instituciones beneficiarias gestionar las solicitudes de donaciones, realizar el seguimiento de los aportes recibidos, y mantener la comunicación con los donantes de manera eficiente. Algunas de las organizaciones más destacadas en Chile, en la recaudación de donaciones son: UNICEF, Abrazame, Las Crèches, y Aldeas Infantiles SOS. Sin embargo, a pesar de su funcionalidad, presentan limitaciones importantes en términos de personalización de las solicitudes y trazabilidad de los aportes, lo que dificulta una gestión eficiente de las donaciones y reduce la confianza de los donantes en el proceso.

Limitaciones de las herramientas existentes:

Falta de personalización en las solicitudes: Las plataformas actuales generalmente no permiten que las necesidades de los niños y hogares se presenten de manera individualizada y detallada, lo que limita la conexión emocional y el compromiso de los donantes con causas específicas.

Escasa trazabilidad: Muchas de estas plataformas no proporcionan un seguimiento claro sobre el destino final de las donaciones, lo que genera desconfianza y falta de transparencia.

Falta de integración con los procesos internos de los hogares: Las plataformas existentes no permiten una integración efectiva con los sistemas administrativos internos de los hogares de menores, lo que hace que la gestión de las donaciones sea aún más fragmentada.

2.3 Metodologías

En el desarrollo de software, es crucial seleccionar una metodología adecuada para la gestión de requerimientos. Algunas de las metodologías más relevantes que se pueden practicar son:

Scrum (Metodología ágil)

Descripción: Scrum es una metodología ágil que promueve la entrega incremental de software en ciclos cortos llamados sprints. En cada sprint se desarrollan pequeñas partes del sistema, asegurando que el equipo pueda adaptarse rápidamente a cambios y feedback constante.

Rational Unified Process (RUP)

Descripción: RUP es un marco de desarrollo orientado a objetos que se enfoca en el ciclo de vida completo del software, dividido en fases (inicio, elaboración, construcción, transición). Utiliza artefactos como casos de uso, diagramas de clases y flujos de procesos.

Unified Process (UP)

Descripción: Similar a RUP, pero con un enfoque más flexible y adaptable para proyectos más pequeños. Facilita la integración de técnicas ágiles y modelos tradicionales.

Metodología elegida:

Scrum será la metodología adoptada para el desarrollo de este proyecto debido a su enfoque ágil y su capacidad para responder rápidamente a cambios y retroalimentación. Los sprints de dos semanas permitieron gestionar y evaluar eficientemente el progreso del desarrollo de la plataforma, y facilitaron la colaboración continua entre el equipo y las partes interesadas.

2.4 Modelos de arquitectura

Existen varias opciones a considerar para aplicaciones de este tipo. Las principales arquitecturas que se pueden aplicar en este tipo de plataformas son:

Arquitectura Monolítica: La aplicación y la base de datos están integradas en un solo bloque. Aunque fácil de implementar, no es escalable y dificulta las actualizaciones en módulos independientes.

Arquitectura de Microservicios: Cada componente del sistema (como la gestión de donantes, los perfiles de los niños, las solicitudes de donaciones) se desarrolla como un

servicio independiente. Esto facilita la escalabilidad, mantenimiento, y la integración de nuevas tecnologías sin afectar el resto del sistema.

Arquitectura elegida:

Se optará por una arquitectura de **microservicios** debido a su escalabilidad, flexibilidad y la facilidad con que permite realizar actualizaciones independientes de componentes del sistema.

2.5 Modelo de clases

El modelo de clases hace referencia a la representación de las entidades en el sistema y cómo estas se interrelacionan. Las principales entidades y clases dentro de este sistema son las siguientes:

Donante: Representa a los individuos o entidades que hacen donaciones. Contiene atributos como nombre, dirección, email y los detalles sobre el tipo de donación realizada.

Hogar: Esta clase contiene la información sobre los hogares de menores, incluyendo, nombre, dirección, comuna y detalles de los niños a su cuidado. Cada hogar tiene un registro relacionado con las necesidades de los infantes que alberga.

Infante: Representa a los niños que residen en los hogares. Contiene el rut infante, nombre, fecha de nacimiento y la relación con las necesidades específicas que deben ser atendidas.

Donación: Esta clase guarda la información relacionada con las contribuciones de los donantes, incluyendo el id de la donación, monto, fecha de donación, tipo de donación (y se relaciona con el hogar destinatario) y el estado de la donación.

Razón para usar este modelo:

Modelo orientado a objetos: Utilizar un modelo orientado a objetos permite organizar la lógica del sistema de manera estructurada, representando entidades claves como donantes, hogares e infantes como clases con atributos y métodos. Las relaciones entre estas clases están claramente definidas, facilitando la creación de consultas eficientes y de fácil mantenimiento.

2.6 Modelo de datos Relacional

El modelo de datos relacional es una estructura lógica que organiza la información en forma de tablas (también llamadas relaciones), donde cada tabla representa una entidad del sistema y cada fila corresponde a una instancia específica de dicha entidad. Esta estructura se basa en una teoría matemática que utiliza álgebra relacional para la manipulación de datos y es ampliamente utilizada en los sistemas de gestión de bases de datos (SGBD) modernos, como PostgreSQL.

Dentro del contexto de plataformas digitales orientadas a la gestión de procesos sociales, como los sistemas de donaciones y apadrinamiento desarrollados en este proyecto, el modelo relacional resulta particularmente adecuado por su capacidad de:

- Garantizar la integridad de los datos a través de claves primarias y foráneas.
- Facilitar la trazabilidad de las acciones de los usuarios (como donaciones, pagos o registros).
- Permitir relaciones complejas entre múltiples entidades (por ejemplo, donante–apadrinamiento–infante).
- Optimizar las operaciones de lectura y escritura mediante estructuras indexadas y normalizadas.

El uso de un modelo relacional en esta plataforma permite establecer una base sólida para representar con precisión a los diferentes actores (donantes, administradores, infantes), sus vínculos y las transacciones entre ellos. Además, facilita la implementación de reglas de negocio, como la validación de que un mismo infante no sea apadrinado por el mismo donante más de una vez, o la actualización automática del estado de una solicitud tras recibir una donación.

2.7 Tecnologías para el desarrollo de software

Framework para el desarrollo móvil:

Opción 1: React Native

React Native es un framework basado en JavaScript que permite desarrollar aplicaciones móviles multiplataforma (iOS y Android) con una única base de código. Se utiliza ampliamente en la industria por su rendimiento y facilidad de integración con otras tecnologías. A través de componentes nativos, React Native ofrece una experiencia de usuario cercana a la de una app nativa, lo que mejora la fluidez de la aplicación.

Ventajas de React Native:

- Desarrollo multiplataforma: Una sola base de código para iOS y Android.
- Comunidad activa: Amplio soporte de la comunidad, con una extensa biblioteca de paquetes y soluciones.
- Rendimiento optimizado: Interfaz rápida gracias a los componentes nativos.

Opción 2: Flutter

Flutter es un framework de desarrollo de aplicaciones móviles creado por Google. Usa el lenguaje Dart para crear aplicaciones con un alto rendimiento y un diseño altamente personalizado. Al igual que React Native, permite crear aplicaciones para iOS y Android desde una sola base de código.

Ventajas de Flutter:

- Rendimiento cercano al nativo: Utiliza un motor gráfico propio (Skia) que permite renderizar aplicaciones con animaciones y transiciones fluidas.
- Personalización: Mayor control sobre la apariencia visual y personalización de los elementos de la interfaz.
- Desarrollo rápido: El uso de "*Hot Reload*" permite realizar cambios en tiempo real durante el desarrollo.

Opción 3: Android SDK con Kotlin

Android SDK es el framework oficial de Google para el desarrollo de aplicaciones móviles nativas para Android. Se utiliza junto con el entorno Android Studio y el lenguaje Kotlin, que ofrece una sintaxis moderna, segura y eficiente. Esta opción permite un control total sobre el rendimiento, la interfaz y la integración con funciones nativas del sistema operativo Android.

Ventajas de Android SDK con Kotlin:

- Rendimiento nativo: Acceso directo a componentes del sistema operativo sin capas intermedias.
- Integración total: Compatible con servicios nativos como Firebase, Google Maps y APIs RESTful.
- Comunidad oficial y soporte extendido: Respaldo directo de Google y documentación actualizada.
- Herramientas avanzadas: Uso de Android Studio, emuladores, depuración, testing y perfiles de rendimiento integrados.

Decisión:

Si bien se analizaron alternativas multiplataforma como React Native y Flutter por su capacidad de reutilizar código entre sistemas operativos, el proyecto se desarrolló finalmente utilizando **Android SDK** con Kotlin, dado que:

- Se requería alto rendimiento y control total sobre los recursos del sistema Android.
- La aplicación solo está dirigida a dispositivos Android, por lo que no se requería compatibilidad con iOS.
- La integración con servicios como Firebase Authentication, Google Maps SDK, Retrofit, y la arquitectura con APIs REST desplegadas en AWS EC2 se facilita de forma directa en un entorno nativo.
- Se utilizó Android Studio como entorno de desarrollo, lo cual permitió una mejor productividad, herramientas de testing integradas y seguimiento detallado del rendimiento.
- Por estas razones, Android SDK con Kotlin se presentó como la mejor opción técnica para cumplir con los objetivos funcionales, técnicos y de escalabilidad.

Motor de bases de datos:

Opción 1: MySQL

MySQL es un sistema de gestión de bases de datos relacional muy popular y ampliamente utilizado en la industria. Es ideal para aplicaciones que requieren consultas rápidas y transacciones seguras. MySQL es muy eficaz en la gestión de grandes volúmenes de datos estructurados y ofrece alta disponibilidad y seguridad.

Ventajas de MySQL:

- **Rendimiento**: Capacidad de realizar consultas rápidas en grandes volúmenes de datos.
- **Escalabilidad**: Fácil de escalar para manejar más usuarios y datos sin perder rendimiento.
- **Compatibilidad con AWS**: Se integra perfectamente con Amazon RDS, lo que asegura una alta disponibilidad y seguridad.

Opción 2: PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional (SGBDR) de código abierto. Es conocido por su fiabilidad, flexibilidad y cumplimiento de estándares, y se utiliza para almacenar y gestionar datos de forma segura en diversas aplicaciones.

Ventajas de PostgreSQL

- Soporte para transacciones complejas: Es ideal para aplicaciones que necesitan realizar consultas complejas o que manejan datos no estructurados.
- Extensibilidad: Permite agregar funcionalidades personalizadas mediante extensiones.
- Integridad referencial: Mayor énfasis en la consistencia de los datos y la integridad referencial.

Decisión:

Para este proyecto, se ha elegido MySQL como motor de base de datos debido a su alto rendimiento en consultas simples, su facilidad de uso y su compatibilidad con AWS RDS, lo que garantiza escalabilidad, disponibilidad y seguridad en el manejo de datos. Además, la naturaleza del proyecto no requiere consultas altamente complejas o de gran volumen que justificarían el uso de PostgreSQL.

2.8 Planes de pruebas funcionales

Las pruebas funcionales constituyen una fase fundamental en el ciclo de desarrollo de software, cuyo objetivo principal es verificar que el sistema cumpla con los requerimientos funcionales definidos. Estas pruebas se centran en evaluar el comportamiento del sistema desde la perspectiva del usuario, asegurando que cada funcionalidad —como el registro de usuarios, la creación de donaciones, el apadrinamiento o la visualización del mapa— se ejecute de forma correcta, sin errores y respondiendo a los flujos esperados.

Para este proyecto, se optará por la estrategia de prueba de tipo caja negra (black-box testing), dado que se evaluará la salida del sistema sin considerar la lógica interna del código fuente. Esta estrategia permite que las pruebas sean diseñadas desde los requisitos del usuario, garantizando que la plataforma funcione como se espera bajo distintas condiciones de entrada.

El plan de pruebas funcionales incluirá los siguientes elementos:

- Identificación de funcionalidades clave: cada caso de uso documentado, como; registrar infantes, gestionar solicitudes, realizar pagos o visualizar ubicación en el mapa. será cubierto por al menos un conjunto de pruebas.
- Definición de escenarios y entradas: se establecerán condiciones válidas, inválidas y límites para comprobar la robustez del sistema.
- Evaluación de resultados esperados: se compararán los resultados obtenidos frente a los esperados para cada prueba.
- Registro de resultados: los datos recopilados serán sistematizados para su posterior análisis en el **capítulo 4**, donde se detallarán los resultados de las pruebas ejecutadas.

CAPÍTULO 3: PROPUESTA DE SOLUCION

3.1 Casos de Uso: Procesos de registro y gestión de donaciones

En esta sección se describen los procesos clave que permiten registrar y gestionar las donaciones en la plataforma. Para cada uno de estos procesos, se presentarán los casos de uso correspondientes, acompañados de sus respectivos diagramas de requerimientos (diagramas de secuencia) y modelos de interfaz.

Caso de uso 1: Registrar infante

Nombre: Registrar infante

Actor principal: Administrador del hogar

Descripción: Este caso de uso permite al administrador registrar en la plataforma a un nuevo niño o niña que ha sido incorporado al hogar. El registro incluye información personal básica y archivos multimedia relacionados, como su fotografía y galería de imágenes.

Precondición: El administrador ha iniciado sesión en la plataforma.

Postcondición: El infante queda registrado en la base de datos y vinculado al hogar correspondiente.

Flujo principal:

1. El administrador selecciona la opción "Registrar infante".
2. Ingresar los datos del infante: RUT, nombres, apellidos, fecha de nacimiento.
3. Carga la foto de perfil y la galería de imágenes del infante.
4. Confirma el registro.
5. El sistema valida los datos y guarda el registro en la base de datos.
6. El sistema muestra un mensaje de confirmación.

Diagrama de requerimientos:

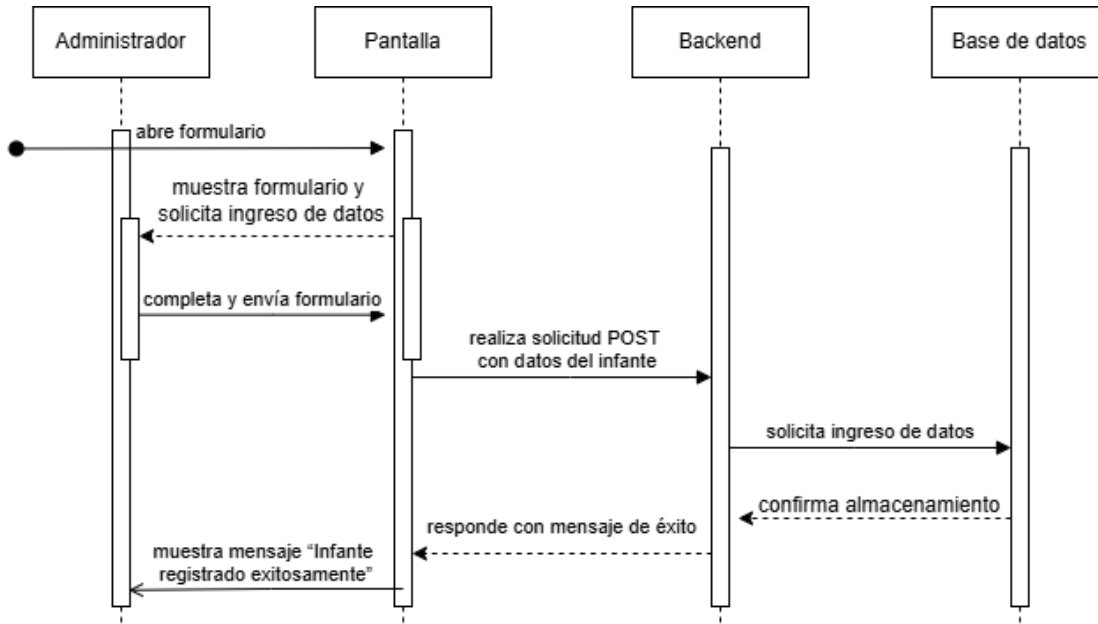


Ilustración 1: Diagrama de requerimientos de Registro de infante
Fuente: Elaboración propia

Interfaz:

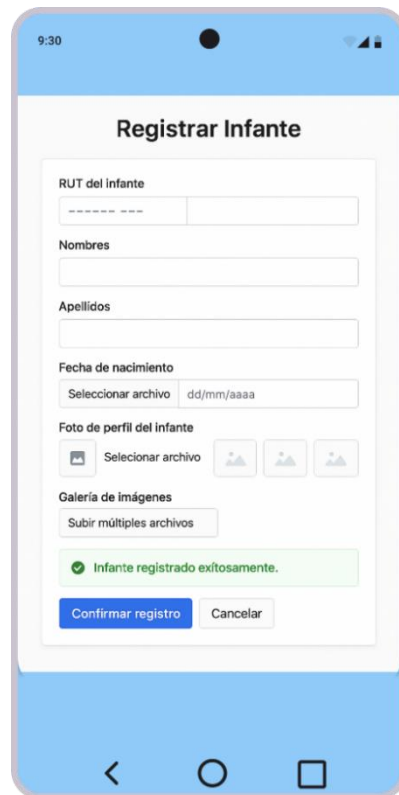


Ilustración 2: Interfaz de registro de infante
Fuente: Elaboración propia

Caso de uso 2: Registro de donante

Nombre: Registrar donante

Actor principal: Donante (usuario externo)

Descripción: Este caso de uso permite a cualquier persona registrarse como donante en la plataforma, proporcionando información básica como nombre completo, RUT, correo electrónico y dirección. Este paso es necesario para poder realizar donaciones o apadrinar a un niño o niña.

Precondición: El donante ha accedido a la sección de registro en la aplicación móvil o web.

Postcondición: El sistema almacena los datos del donante y crea un perfil vinculado a su RUT.

Flujo principal:

1. El donante selecciona la opción “Crear cuenta”.
2. Ingresa su RUT, nombres, apellidos y correo electrónico.
3. Crea una contraseña segura.
4. Confirma el formulario.
5. El sistema valida los datos y verifica que no exista otro usuario con ese RUT.
6. El sistema almacena la información en la base de datos.
7. El sistema muestra un mensaje de bienvenida y redirige al inicio de sesión.

Diagrama de secuencia:

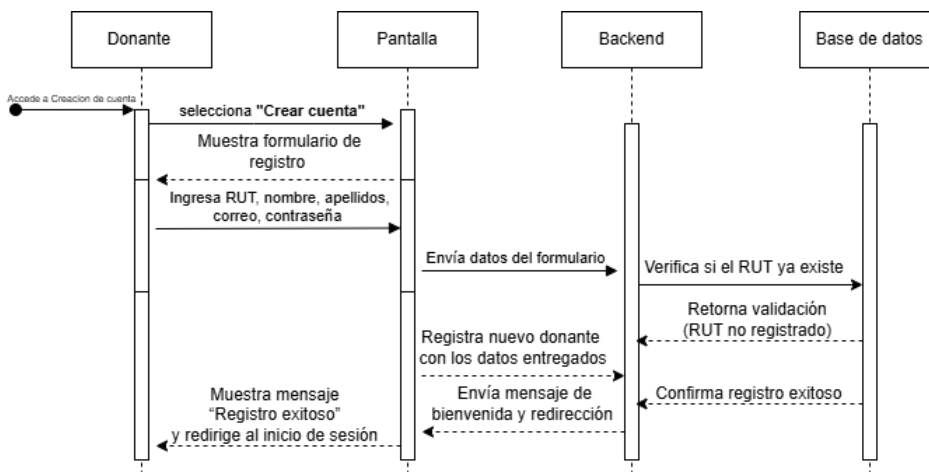


Ilustración 3: Diagrama de requerimientos de Registro de Donante

Fuente: Elaboración propia



Ilustración 4: Interfaz de creación de cuenta del Donante
Fuente: Elaboración propia

Caso de uso 3: Visualizar perfil de infante

Nombre: Visualizar perfil de infante

Actor principal: Donante (usuario registrado)

Descripción: Este caso de uso permite al donante acceder al perfil completo de un niño o niña registrado en el sistema. Desde esta vista, el donante podrá ver la información básica del infante, su galería de fotos, necesidades actuales y solicitudes vigentes del hogar al que pertenece. Esto permite tomar decisiones informadas para realizar donaciones o apadrinamientos.

Precondición: El donante debe haber iniciado sesión correctamente en la aplicación.

Postcondición: Se carga en pantalla el perfil completo del infante seleccionado.

Flujo principal:

1. El donante inicia sesión en la aplicación.
2. Accede a la sección "Galería de Infantes".

3. Selecciona a un infante en particular.
4. El sistema consulta en la base de datos la información personal del infante, su galería de imágenes y las solicitudes del hogar vinculadas a ese infante.
5. El sistema despliega el perfil completo con todos los datos e imágenes.

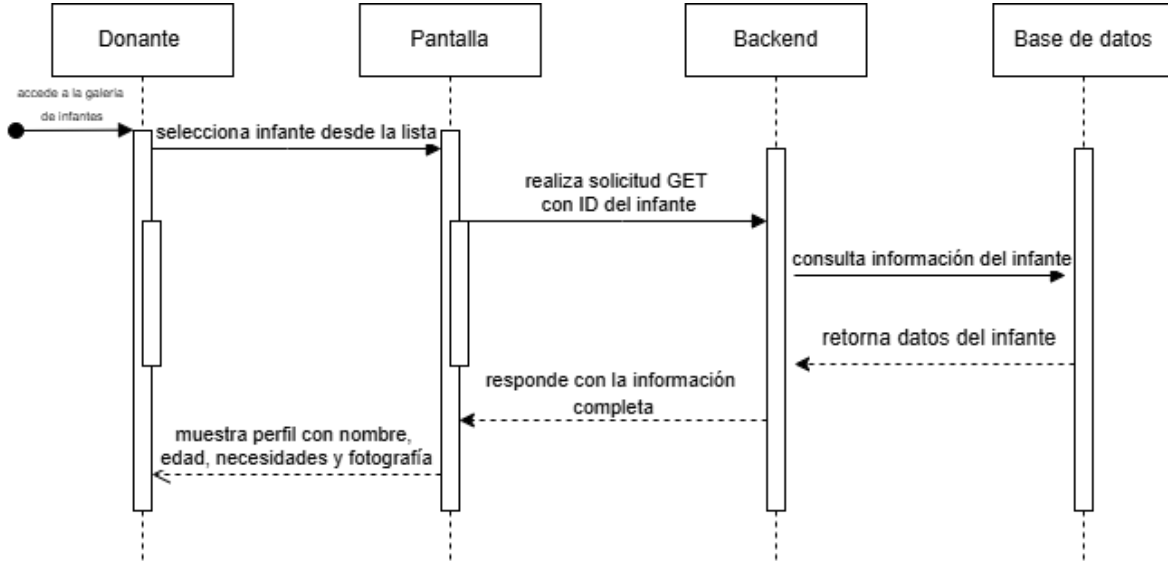


Ilustración 5: Diagrama de requerimientos de Visualizar el perfil del Infante

Fuente: Elaboración propia

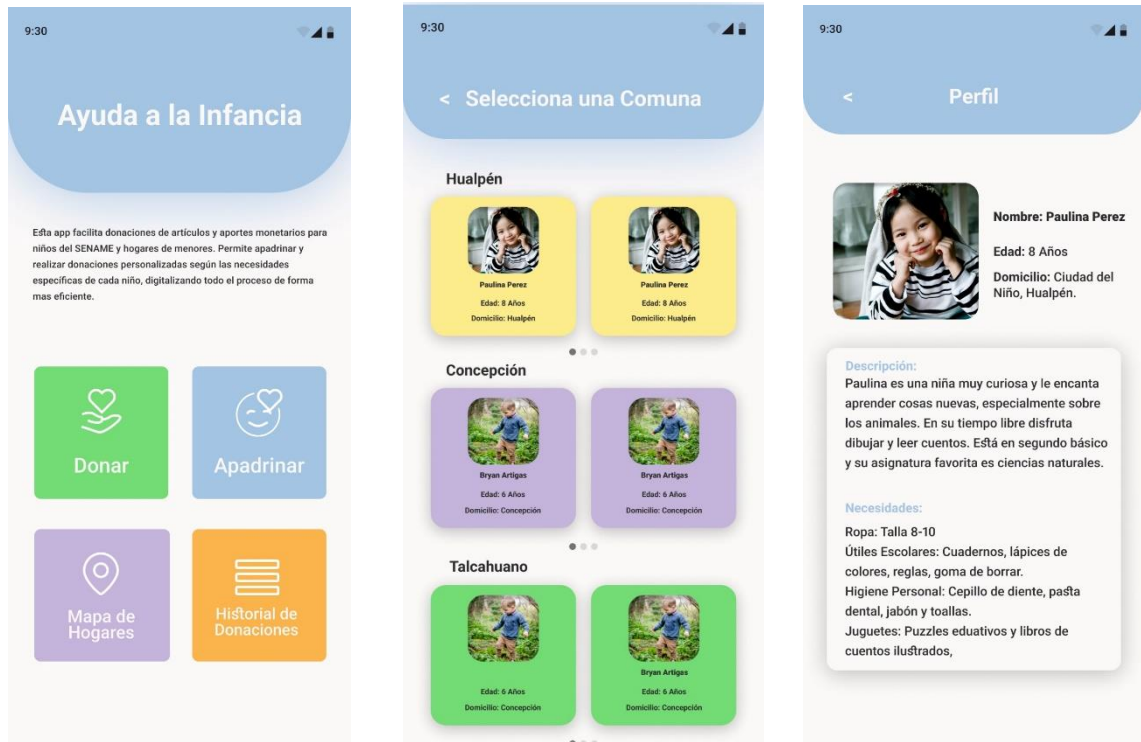


Ilustración 6: Interfaces del proceso de visualización del infante

Fuente: Elaboración propia

Caso de uso 4: Registrar donación

Nombre: Registrar donación

Actor principal: Donante

Descripción: Este caso de uso permite al donante realizar una donación económica en respuesta a una solicitud previamente publicada por un hogar. El proceso contempla la selección de una solicitud, la elección del monto (personalizado o sugerido), la opción de añadir un mensaje o realizar la donación de forma anónima, y la confirmación del aporte a través del sistema de pagos Webpay. Una vez validada la transacción, la donación queda registrada en la base de datos, vinculada al donante y a la solicitud correspondiente, y se informa al usuario el éxito de la operación.

Precondición:

El donante debe estar registrado e iniciar sesión en la plataforma.

Debe existir al menos una solicitud publicada.

Postcondición:

La donación queda registrada en la base de datos.

Se vincula al donante y a la solicitud correspondiente.

El donante recibe confirmación de su aporte.

Flujo principal:

1. El donante inicia sesión en la aplicación.
2. Accede a la sección de solicitudes de donación.
3. Visualiza las solicitudes publicadas y selecciona una para donar.
4. La aplicación muestra el detalle de la solicitud seleccionada, incluyendo descripción, monto sugerido, cantidad y fecha.
5. El donante ingresa el monto de su donación o selecciona un monto predefinido.
6. (Opcional) El donante agrega un mensaje personalizado y elige si desea que su donación sea anónima.
7. Confirma la donación.
8. La app redirige al sistema de pago Webpay, donde el donante ingresa los datos de su tarjeta para realizar la transacción.
9. Tras recibir la confirmación del pago exitoso, la app guarda la información de la donación en la base de datos, la vincula al donante y actualiza el estado de la solicitud.
10. Se genera un identificador único para la transacción.
11. El sistema muestra al donante un mensaje de agradecimiento, indicando que la donación fue registrada con éxito.

Flujo alternativo (A1): Error en la transacción

Si la validación falla o hay error en la transacción, el sistema muestra un mensaje de error y solicita corregir los datos o reintentar.

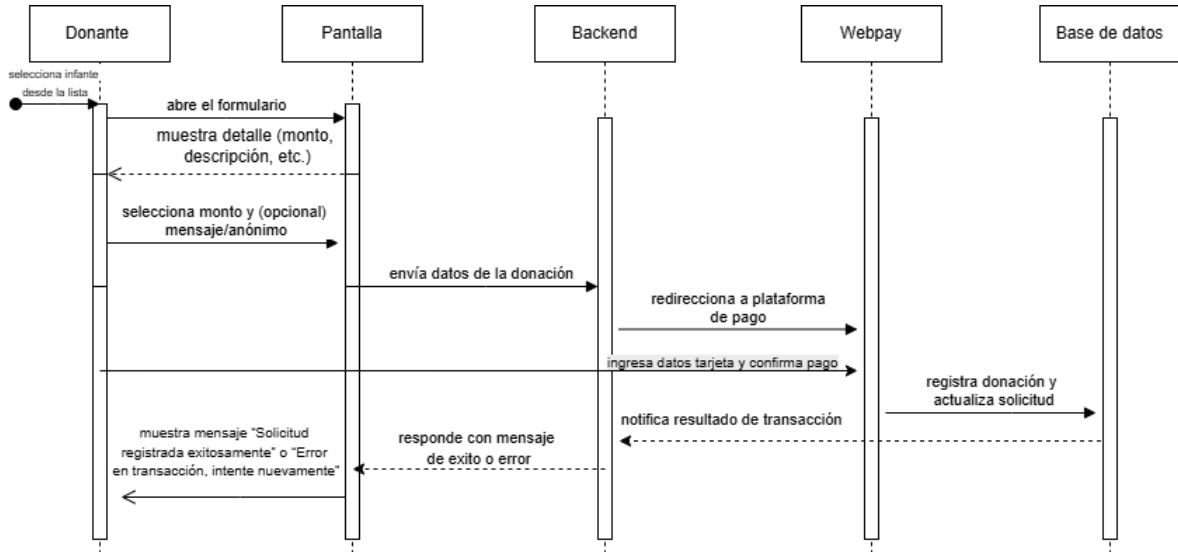


Ilustración 7: Diagrama de requerimiento de registrar donación.

Fuente: Elaboración propia.



Ilustración 8: Interfaz de Registro de donación

Fuente: Elaboración propia

Caso de uso 5: Registrar apadrinamiento

Nombre: Registrar apadrinamiento

Actor principal: Donante

Descripción: Este caso de uso permite al donante registrarse como padrino o madrina de un niño o niña de un hogar específico. Incluye la selección del infante, el ingreso del monto mensual de apadrinamiento y la confirmación del compromiso mensual en la plataforma.

Precondición:

- El donante debe estar registrado e iniciar sesión en la plataforma.
- El infante debe estar previamente registrado en el sistema.

Postcondición:

- El apadrinamiento queda registrado en la base de datos.
- Se genera el vínculo entre donante e infante, con el monto mensual establecido.
- El sistema muestra confirmación al donante.

Flujo principal:

1. El donante inicia sesión en la aplicación.
2. Accede a la sección “Infantes disponibles para apadrinamiento”.
3. Visualiza la lista de infantes y selecciona uno para apadrinar.
4. La aplicación muestra el perfil del infante con sus datos básicos y galería.
5. El donante selecciona la opción “Apadrinar”.
6. Ingresa el monto mensual de su aporte.
7. Confirma el apadrinamiento.
8. El sistema valida los datos ingresados.
9. El sistema registra el apadrinamiento en la base de datos con su identificador único, vinculando al donante con el infante seleccionado.
10. El sistema muestra un mensaje de confirmación al donante, indicando que el apadrinamiento fue registrado exitosamente.

Flujo alternativo (A1):

- Cancelación del proceso
- Si el donante decide cancelar antes de confirmar, el sistema descarta los datos ingresados y regresa a la lista de infantes.

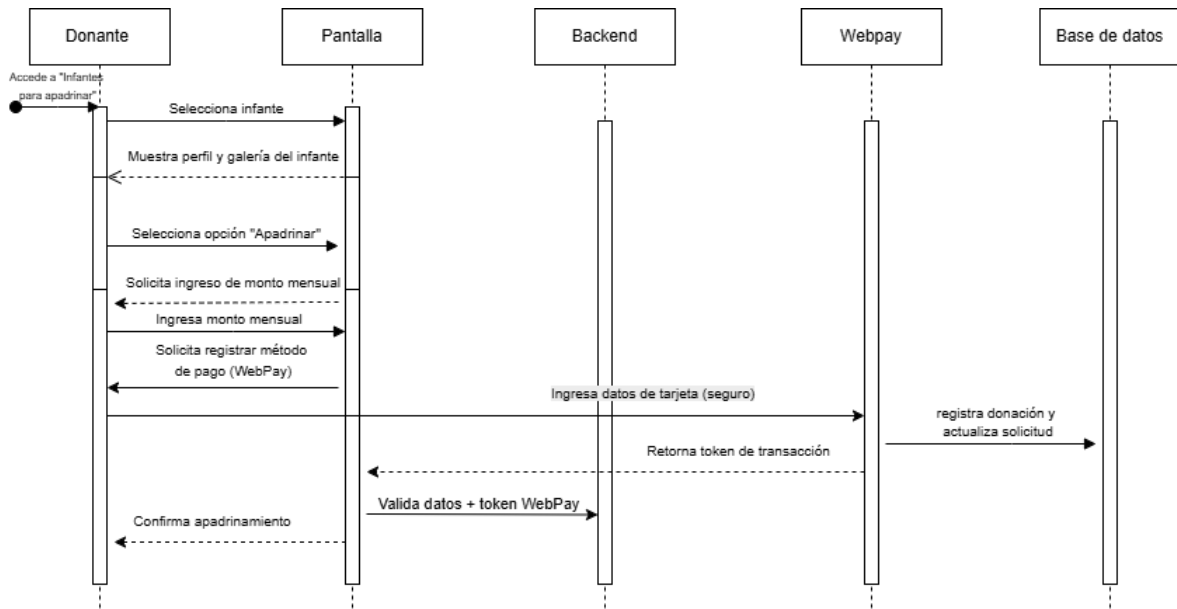


Ilustración 9: Diagrama de requerimientos de realizar apadrinamiento

Fuente: elaboración propia.



Ilustración 10: Interfaz de registrar Apadrinamiento.

Fuente: Elaboración Propia.

Caso de uso 6: Revisar pagos

Nombre: Revisar pagos de apadrinamientos y donaciones

Actor principal: Donante

Descripción: Este caso de uso permite al donante realizar el pago mensual correspondiente al apadrinamiento de un infante. Incluye la selección del apadrinamiento activo, el ingreso del monto (si no es automático), y la confirmación de la transacción.

Precondición:

- El donante debe estar registrado e iniciar sesión en la plataforma.
- Debe existir un apadrinamiento activo registrado en el sistema.

Postcondición:

- El pago queda registrado en la base de datos, asociado al apadrinamiento correspondiente.
- El sistema muestra confirmación al donante y actualiza el historial de pagos.

Flujo principal:

1. El donante inicia sesión en la aplicación.
2. Accede a la sección “Historial de donaciones”.
3. Visualiza la lista de apadrinamientos activos y el historial de donaciones que ha tenido.
4. Selecciona el apadrinamiento para el cual desea realizar el pago.
5. Ingresa el monto a pagar (si no es predefinido).
6. Confirma el pago.
7. El sistema valida los datos ingresados.
8. El sistema registra el pago en la base de datos con su identificador único, asociado al apadrinamiento correspondiente.
9. El sistema muestra un mensaje de confirmación al donante, indicando que el pago fue registrado exitosamente.

Flujo alternativo (A1): Error en la transacción

- Si la validación falla o hay error en la transacción, el sistema muestra un mensaje de error y solicita corregir los datos o reintentar.

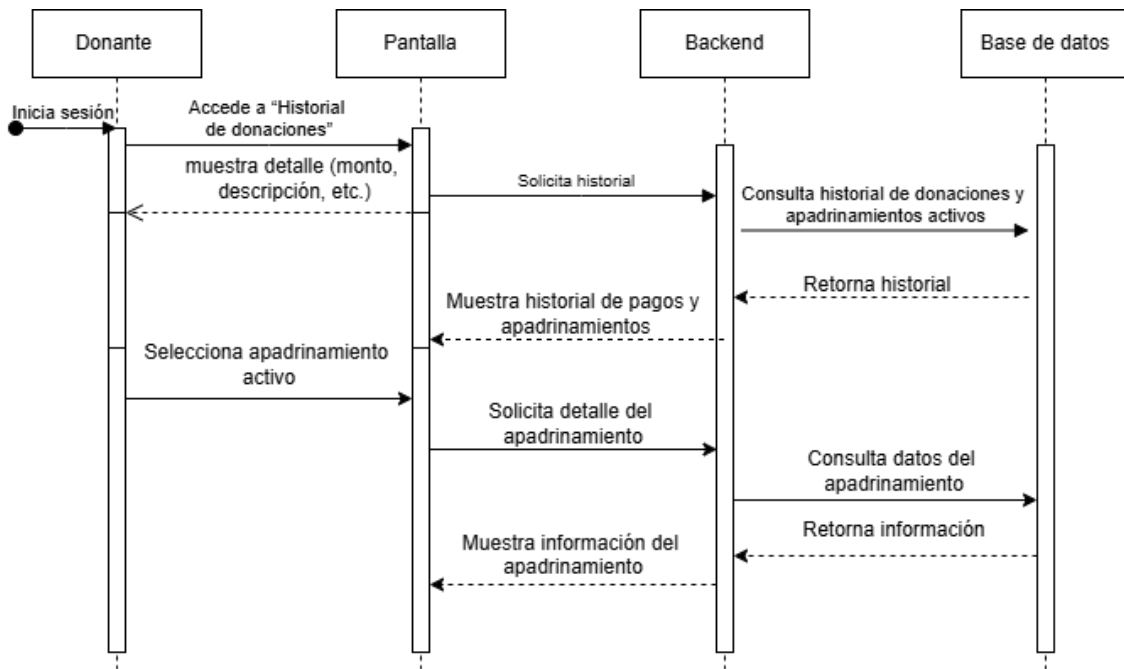


Ilustración 11: Diagrama de requerimientos de revisar pagos.

Fuente: Elaboración propia.

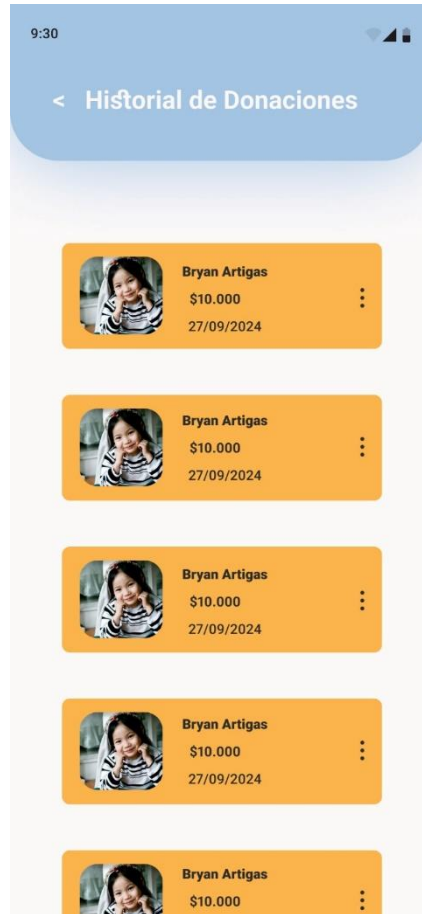


Ilustración 12: Interfaz de Revisión de pagos

Caso de uso 7: Visualizar mapa

Nombre: Visualizar mapa

Actor principal: Donante

Descripción:

Este caso de uso permite al donante acceder a un mapa interactivo donde puede visualizar los hogares de niños registrados en la plataforma. Cada hogar se encuentra geolocalizado con información relevante. El mapa facilita la exploración territorial para tomar decisiones informadas sobre donaciones o apadrinamientos.

Precondiciones:

- El donante debe estar registrado e iniciar sesión en la plataforma.
- Debe haber hogares registrados con geolocalización en el sistema.

Postcondiciones:

- El donante visualiza un mapa actualizado con la ubicación de hogares.

- El sistema puede registrar la interacción del usuario con el mapa (por ejemplo, clics o selecciones sobre un hogar para análisis posterior).

Flujo principal:

1. El donante inicia sesión en la aplicación.
2. Accede a la sección “Mapa de hogares”.
3. El sistema solicita los datos geográficos de los hogares disponibles.
4. El backend consulta la base de datos con la información de ubicación de cada hogar.
5. El sistema renderiza un mapa interactivo con marcadores para cada hogar.
6. El donante visualiza el mapa y puede interactuar con los marcadores.
7. (Opcional) Al seleccionar un marcador, se despliega información básica del hogar, incluyendo dirección, nombre, y cantidad de infantes registrados.

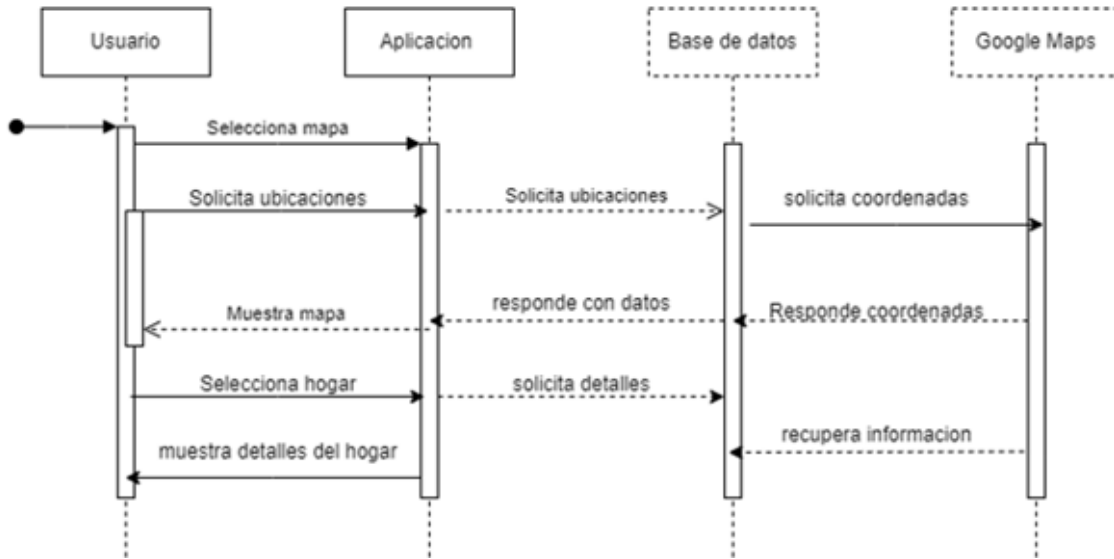


Ilustración 13:Diagrama de requerimientos Revisar mapa.

Fuente: elaboración propia.

3.2. Modelo de arquitectura de software

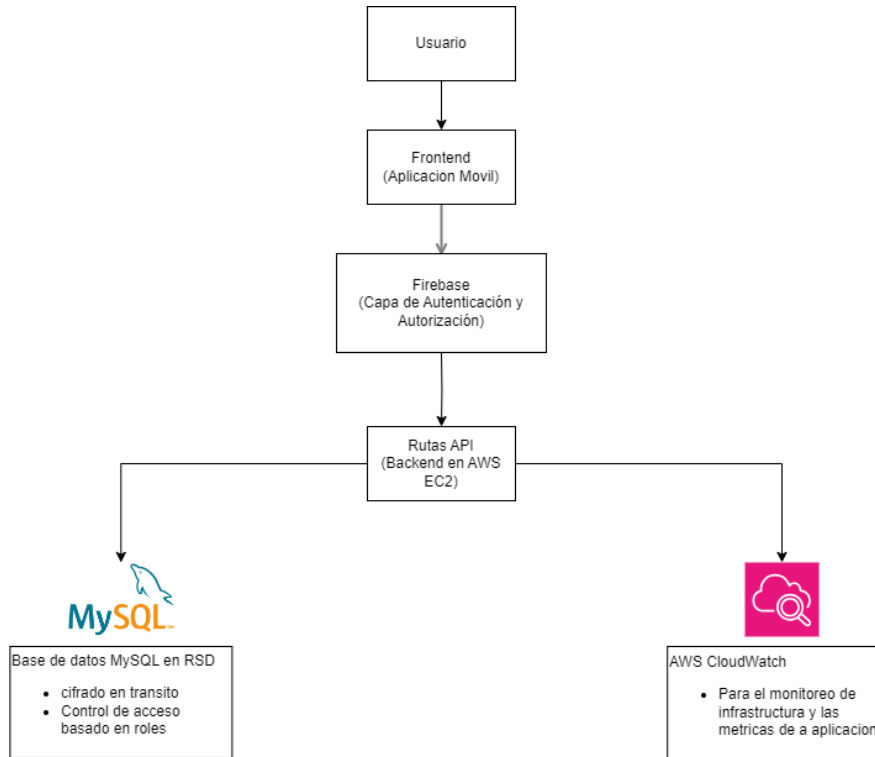


Ilustración 15: Diagrama de Arquitectura de Seguridad y Autenticación

Fuente: Elaboración propia

La solución fue diseñada bajo un modelo cliente-servidor distribuido, haciendo uso de servicios cloud para garantizar escalabilidad, seguridad y disponibilidad.

La arquitectura está compuesta por los siguientes módulos:

Frontend: Aplicación móvil

La aplicación móvil actúa como interfaz principal para los usuarios. Está desarrollada para facilitar la interacción intuitiva con las funcionalidades del sistema, incluyendo el apadrinamiento, visualización de solicitudes y registro de donaciones.

Capa de autenticación y autorización: Firebase

Se utiliza Firebase Authentication para gestionar el acceso seguro de los usuarios a la plataforma. Esta capa permite registrar e iniciar sesión con validación en tiempo real, controlando también los permisos por tipo de usuario (donante, administrador, etc.).

Backend: APIs RESTful en AWS EC2

Las funcionalidades principales de negocio se encuentran implementadas en un backend desplegado sobre una instancia EC2 en AWS. El backend expone una serie de rutas API que permiten la gestión de usuarios, infantes, solicitudes de donación, apadrinamientos y reportes.

Base de Datos: Amazon RDS con MySQL

El almacenamiento de la información se realiza en una base de datos relacional gestionada mediante Amazon RDS. Se utiliza MySQL como motor de base de datos, con medidas de seguridad como cifrado en tránsito y control de acceso basado en roles.

Monitoreo: AWS CloudWatch

Se utiliza AWS CloudWatch para supervisar métricas del sistema, comportamiento de la aplicación, consumo de recursos, y fallos críticos. Esto permite mantener la operatividad de la solución y responder a errores de forma proactiva.

3.3 Modelo de clases

El modelo de clases representa la estructura del sistema desde una perspectiva orientada a objetos, describiendo las clases, sus atributos, métodos y las asociaciones entre ellas en un instante de tiempo determinado. A diferencia del modelo entidad-relación, que refleja relaciones históricas a lo largo del tiempo, el modelo de clases permite observar el estado actual del sistema, es decir, cómo interactúan las clases en un momento dado.

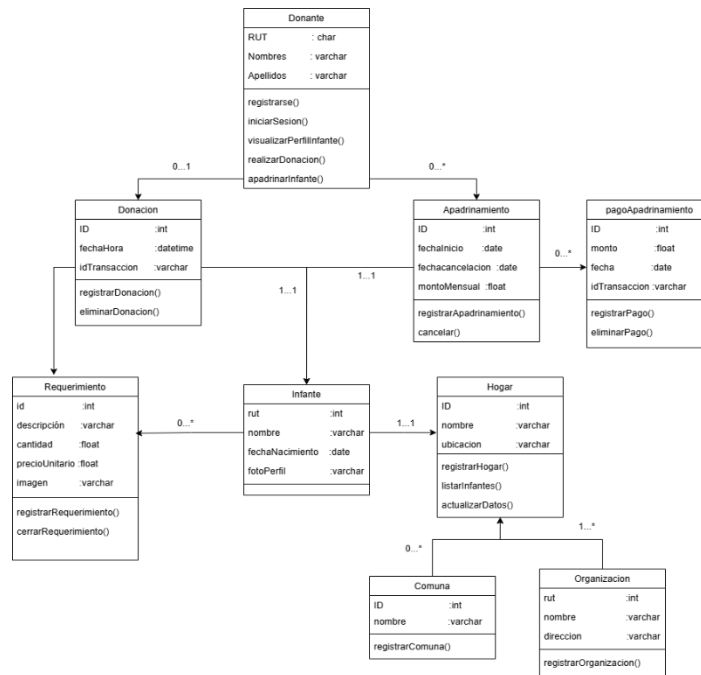


Ilustración 16: Modelo de clases

Fuente: Elaboración propia

Clases principales y sus relaciones

1. Clase: Donante

Atributos: RUT, Nombres, Apellido Paterno, Apellido Materno

Métodos: registrarDonante(), iniciarSesion(), actualizarDonante(), eliminarDonante().

Asociaciones:

- En un momento dado, un donante puede estar realizando 0 o 1 donación → **0..1**
- Puede tener 0 o muchos apadrinamientos activos → **0..***

2. Clase: Donación

Atributos: id, fechaHora, monto, idTransaccion.

Métodos: registrarDonacion(), eliminarDonacion().

Asociaciones:

- Toda donación está asociada a 1 donante → **1..1**
- Toda donación está dirigida a 1 hogar o a 1 infante → **1..1**
- Puede estar asociada a 0 o 1 solicitud de requerimiento → **0..1**

3. Clase: Hogar

Atributos: id, nombre, ubicacion.

Métodos: registrarHogar(), listarInfantes(), actualizarDatos().

Asociaciones:

- En un momento dado, puede alojar 1 o más infantes → **1..***
- Puede tener 0 o más donaciones en curso → **0..***
- Puede tener 0 o más solicitudes activas → **0..***

4. Clase: Infante

Atributos: rut, nombre, fechaNacimiento, fotoPerfil.

Asociaciones:

- En un momento dado, un infante está alojado en 1 hogar → **1..1**
- Puede tener 0 o más solicitudes de requerimiento activas → **0..***
- Puede estar siendo apadrinado por 0 o 1 donante → **0..1**

5. Clase: Requerimiento

Atributos: id, descripcion, cantidad, precioUnitario, imagen.

Métodos: registrarRequerimiento(), cerrarRequerimiento().

Asociaciones:

- En un momento dado, cada solicitud está asociada a 1 infante → **1..1**
- Puede estar financiada por 0 o 1 donaciones → **0..1**

6. Clase: Apadrinamiento:

Atributos: id, fechaInicio, montoMensual.

Métodos: registrarApadrinamiento(), cancelar().

Asociaciones:

- En un momento dado, relaciona 1 donante con 1 infante → **1..1**
- Puede tener 0 o más pagos activos o en espera → **0..***

7. Clase: PagoApadrinamiento

Atributos: id, fecha, monto, idTransaccion.

Métodos: registrarPago(), eliminarPago().

Asociaciones:

- Pertenece a 1 apadrinamiento → **1..1**

8. Clase: Organización

Atributos: rut, nombre, direccion.

Métodos: registrarOrganizacion(), editarDatos().

Asociaciones:

- Puede estar administrando 1 o más hogares → **1..***

9. Clase: Comuna

Atributos: id, nombre.

Métodos: registrarComuna().

Asociaciones:

- Puede contener 0 o más hogares → **0..***

Justificación del enfoque

Este modelo permite entender con precisión las restricciones instantáneas del sistema, ayudando a evitar inconsistencias y a definir mejor las reglas de negocio.

3.4 Modelo Entidad-Relación (MER)

El modelo entidad-relación propuesto para la plataforma de donaciones y apadrinamiento está orientado a garantizar la trazabilidad de las acciones realizadas por los donantes, la gestión de infantes y hogares, y el seguimiento de los aportes económicos y materiales dentro del sistema. Se diseñó con un enfoque normalizado, asegurando integridad referencial, bajo acoplamiento entre entidades y soporte a funcionalidades críticas del sistema, el apadrinamiento, la recepción de donaciones y la gestión de requerimientos.

Describe de manera conceptual la estructura de la base de datos del sistema. Este modelo incluye:

- **Las entidades** principales que intervienen en el proceso (donantes, infantes, hogares, organizaciones, etc.).
- **Los atributos** relevantes para cada entidad, que permiten almacenar la información necesaria para la operación del sistema.
- **Las relaciones** entre entidades, que permiten modelar el comportamiento real del sistema, como la asignación de donaciones a hogares, el seguimiento de estadías de los infantes y la gestión de pagos de apadrinamiento.
- **La definición de llaves primarias y foráneas**, que aseguran la integridad referencial de los datos.

Este modelo constituye la base para el diseño relacional, permitiendo traducir las necesidades funcionales del sistema en una estructura de datos coherente, normalizada y alineada con los principios de una base de datos relacional robusta.

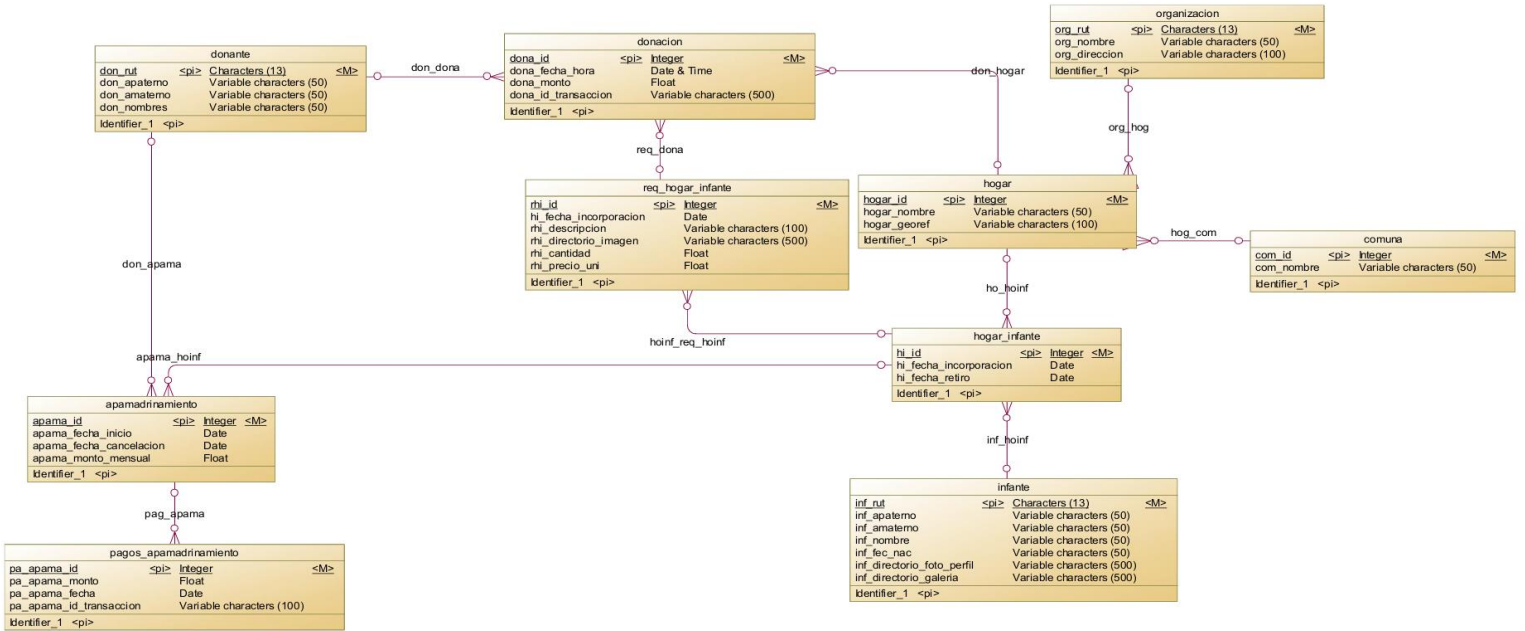


Ilustración 17: Modelo Entidad-Relación.

Fuente: Elaboración propia.

Entidades principales

- **Donante:** Representa a la persona que realiza aportes a la plataforma. Está identificada por su RUT y contiene atributos personales como nombres y apellidos. Puede realizar donaciones puntuales o comprometerse con un apadrinamiento mensual.
- **Donación:** Registra un aporte económico o material realizado por un donante. Cada donación contiene la fecha y hora de realización, el monto donado y un identificador de transacción. Está vinculada directamente a un hogar, y opcionalmente, a un requerimiento específico publicado para un infante.
- **Apadrinamiento:** Representa el compromiso recurrente de un donante hacia un infante, a través de una cuota mensual. Se asocia a un infante dentro de un hogar específico y permite mantener un historial de pagos periódicos realizados.
- **Pagos de Apadrinamiento:** Tabla transaccional que almacena cada uno de los pagos mensuales asociados a un apadrinamiento. Registra la fecha, el monto abonado y el código de transacción para efectos de trazabilidad financiera.

- **Infante:** Entidad que representa a los niños o niñas beneficiarios del sistema. Incluye datos personales como nombre, fecha de nacimiento y rutas a imágenes para su perfil y galería. Cada infante puede haber estado en distintos hogares a lo largo del tiempo.
- **Hogar:** Representa una residencia de acogida infantil. Contiene información sobre su nombre y localización geográfica. Cada hogar está vinculado a una organización que lo administra y a una comuna donde se encuentra ubicado.
- **Organización:** Entidad que agrupa a uno o más hogares. Corresponde a la institución legalmente responsable de los mismos.
- **Comuna:** Entidad de referencia territorial que normaliza el lugar de ubicación de cada hogar, permitiendo generar reportes geográficos y análisis territoriales.
- **Hogar_Infante:** Tabla intermedia que registra el historial de estadía de los infantes en los hogares, incluyendo fechas de ingreso y salida. Permite mantener trazabilidad de los movimientos de cada niño o niña en el tiempo.
- **Requerimientos del Hogar para el Infante:** Permite a los hogares publicar necesidades específicas asociadas a infantes bajo su cuidado, como útiles escolares, ropa o medicamentos. Cada requerimiento puede estar vinculado a una donación que lo financie total o parcialmente.

Relaciones clave

- Un donante puede realizar múltiples donaciones y apadrinar a varios infantes.
- Un infante puede residir en distintos hogares, lo que se registra mediante la entidad hogar_infante, permitiendo conservar el historial completo de su estadía.
- Cada apadrinamiento está vinculado a un infante en un hogar específico y genera múltiples pagos a lo largo del tiempo.
- Los requerimientos están ligados a un niño/a específico dentro de un hogar en un período concreto, y pueden ser cubiertos mediante donaciones.
- Cada hogar pertenece a una organización y a una comuna, lo cual permite segmentar la información territorial y administrativamente.

3.5 Modelo de datos relacional

El modelo físico de datos (MFD) representa la implementación concreta del modelo entidad-relación en un sistema gestor de bases de datos relacional. El modelo fue implementado utilizando MySQL como motor de base de datos. El modelo define con precisión los tipos de datos, las claves primarias y foráneas, así como las restricciones necesarias para garantizar la integridad y eficiencia del almacenamiento de la información.

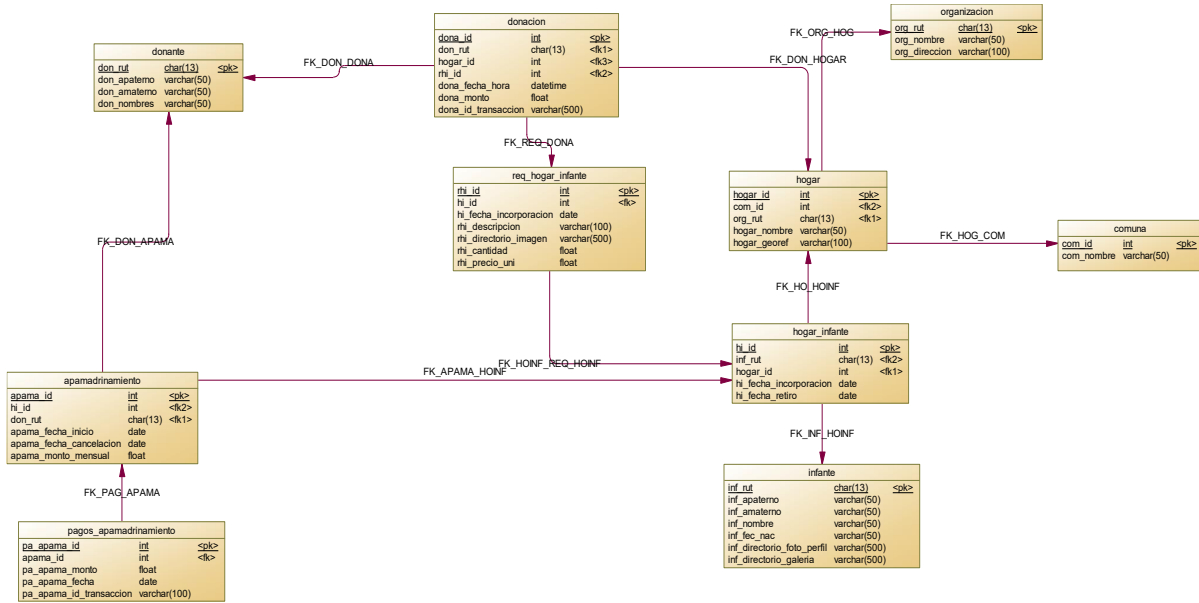


Ilustración 18: Modelo físico de datos.

Fuente: elaboración Propia.

1. Tabla: Donante

El donante es la persona o entidad que contribuye con recursos (dinero o bienes) a los hogares de menores. Esta tabla contiene información clave como:

- don_rut: Identificador único del donante (RUT).
- don_apaterno y don_amaterno: Apellidos del donante.
- don_nombres: Nombres del donante.

Relaciones:

Está relacionada con la tabla donación a través de don_rut (el donante hace una donación).
 Está relacionada con la tabla apadrinamiento a través de don_rut (el donante puede apadrinar a un niño de manera mensual).

Ejemplo:

Un donante llamado "Juan Pérez" realiza una donación monetaria a un hogar específico. Su RUT será registrado en la tabla donante, y cada donación futura se asociará con su identificador único.

2. Tabla: Donación

Cada vez que un donante realiza una donación, esta acción se registra en la tabla donación, que incluye:

dona_id: Identificador único de la donación.

don_rut: Relaciona la donación con el donante.

hogar_id: Relaciona la donación con el hogar beneficiado.

rhi_id: Relaciona la donación con la solicitud de necesidades (si es una donación específica).

dona_fecha_hora: Fecha y hora en que se realiza la donación.

dona_monto: Monto de la donación realizada.

dona_id_transaccion: Identificador único de la transacción.

Relaciones:

Está relacionada con donante (don_rut), hogar (hogar_id) y req_hogar_infante (rhi_id).

Ejemplo:

Juan Pérez decide donar \$50.000 a la Fundación Hogar de Niños. Su donación se registra con un dona_id único, asociando los datos del donante (RUT), el hogar beneficiado, y el monto donado.

3. Tabla: Requerimientos de Hogar e Infante (req_hogar_infante)

Cada hogar tiene necesidades específicas relacionadas con los niños que alberga. Estas necesidades se registran en esta tabla:

rhi_id: Identificador único de la solicitud.

hi_id: Identificador del niño asociado (FK a hogar_infante).

rhi_fecha_incorporacion: Fecha de incorporación de la solicitud.

rhi_descripcion: Descripción de la necesidad.

rhi_cantidad: Cantidad de artículos solicitados.

rhi_precio_uni: Precio unitario estimado para cubrir la necesidad.

Relaciones:

Está relacionada con hogar_infante (hi_id) y donación (rhi_id).

Ejemplo:

El hogar recibe la solicitud de un niño llamado "Pedro", quien necesita ropa nueva. La solicitud tiene un rhi_id único y se asocia con el infante y la descripción de la necesidad.

4. Tabla: Hogar

El hogar es la institución que alberga a los niños y recibe las donaciones. La tabla contiene:

hogar_id: Identificador único del hogar.

com_id: Identificador de la comuna asociada (FK).

org_rut: RUT de la organización asociada (FK).

hogar_nombre: Nombre del hogar.

hogar_georef: Información de geolocalización del hogar.

Relaciones:

Está relacionada con comuna (com_id), organización (org_rut), hogar_infante (hogar_id), y donación (hogar_id).

Ejemplo:

La Fundación "Hogar San Juan" tiene un hogar_id único. Esta fundación recibe diversas donaciones que se registran en la plataforma, y su ubicación geográfica es asociada a la comuna correspondiente.

5. Tabla: Hogar-Infante (hogar_infante)

Esta tabla vincula a los niños con los hogares en los que residen. Contiene:

hi_id: Identificador único de la relación hogar-infante.

inf_rut: Identificador único del niño (FK).

hogar_id: Identificador del hogar asociado (FK).

hi_fecha_incorporacion: Fecha de incorporación del niño al hogar.

hi_fecha_retiro: Fecha de retiro del niño del hogar.

Relaciones:

Está relacionada con infante (inf_rut), hogar (hogar_id), y req_hogar_infante (hi_id).

Ejemplo:

El niño Juanito ingresa al Hogar San Juan, y se crea un registro en la tabla hogar_infante para vincularlo al hogar específico.

6. Tabla: Apadrinamiento

El apadrinamiento es un proceso donde un donante se compromete a aportar regularmente a un niño específico. Esta tabla incluye:

apama_id: Identificador único del apadrinamiento.

hi_id: Identificador del niño asociado (FK).

don_rut: Identificador del donante (FK).

apama_fecha_inicio: Fecha de inicio del apadrinamiento.

apama_monto_mensual: Monto mensual del apadrinamiento.

Relaciones:

Está relacionada con hogar_infante (hi_id), donante (don_rut), y pagos_apadrinamiento (apama_id).

Ejemplo:

Juan Pérez decide apadrinar a Juanito con un monto mensual de \$30.000. Se crea un registro en la tabla apadrinamiento vinculando a Juan Pérez (donante) con Juanito (niño).

Conclusión:

Este modelo de datos refleja un sistema robusto y flexible para gestionar donaciones y apadrinamientos. Cada entidad está vinculada para reflejar el ciclo completo de una donación o apadrinamiento, desde la solicitud de necesidades por parte de los hogares hasta la confirmación de la donación y el apadrinamiento continuado.

3.6 Ambiente de desarrollo

Para el desarrollo de la plataforma móvil de registro y gestión de donaciones y apadrinamientos, se implementó un ambiente tecnológico que garantiza escalabilidad, seguridad y eficiencia en el desarrollo e integración de sus módulos.

- **Framework y Lenguajes Utilizados**

Android SDK (Android Framework)

El proyecto se implementó como aplicación móvil nativa para Android, utilizando Android Studio como IDE y el framework oficial de Android con soporte para Kotlin. Este enfoque permitió desarrollar interfaces y lógicas optimizadas con compatibilidad total en dispositivos Android, aprovechando librerías modernas y acceso directo al hardware del dispositivo cuando sea necesario.

Kotlin

Lenguaje principal para la lógica de la aplicación, elegido por su integración nativa con Android, su sintaxis clara, manejo seguro de nulos (null safety) y sus capacidades de programación funcional, orientada a objetos y reactiva, fundamentales para la conexión con APIs y la gestión de vistas dinámicas.

XML

Utilizado para diseñar la estructura visual de cada pantalla (Activity/Fragment), permitiendo separar completamente la capa de presentación de la lógica de negocio y asegurando interfaces responsivas y consistentes.

SDKs y Librerías Implementadas

Firebase Authentication (SDK)

Gestiona el registro e inicio de sesión de usuarios mediante correo electrónico y contraseña. La app conecta con los servidores de Firebase enviando las credenciales, recibe tokens seguros de autenticación y los utiliza para acceder a las funcionalidades protegidas de la plataforma.

Google Maps SDK (SDK)

Permite integrar mapas interactivos y agregar marcadores dinámicos que geolocalizan los hogares de menores beneficiarios. Las coordenadas se obtienen desde la base de datos mediante APIs y se representan con iconos personalizados.

Retrofit (Librería)

Cliente HTTP que consume APIs RESTful del backend. Su integración permite crear interfaces de servicios de manera tipada y clara, realizar peticiones asincrónicas y manejar respuestas JSON eficientemente.

Glide (Librería)

Utilizada para la carga rápida y eficiente de imágenes de los infantes y solicitudes, aplicando caché automático para mejorar tiempos de carga y disminuir uso de datos.

AndroidX y Material Design Components (Librerías)

Proveen compatibilidad extendida con versiones antiguas de Android y componentes visuales modernos, alineados a las guías de diseño de Google para apps intuitivas y accesibles.

Motor de Base de Datos y Conexiones

MySQL en Amazon RDS

Aunque la app no almacena datos de manera local (SQLite), se conecta a un backend desarrollado en Node.js con Express.js, que a su vez interactúa con una base de datos MySQL alojada en Amazon RDS.

Esta arquitectura desacopla el cliente móvil de la base de datos, mejorando la seguridad (no expone credenciales), escalabilidad (maneja múltiples conexiones concurrentes) y mantenimiento (las modificaciones de la base no afectan directamente la app).

Flujo de conexión general:

1. La app realiza la petición HTTP (Retrofit) →
2. Llega al endpoint de la API REST en el servidor Node.js →
3. El servidor procesa la lógica y consulta la BD MySQL en RDS →
4. Retorna la respuesta JSON a la app, que la renderiza en tiempo real.

3.7 Códigos relevantes

I. Configuración de Retrofit

```

val retrofit = Retrofit.Builder()
    .baseUrl("http://ec2-3-235-13-166.compute-1.amazonaws.com:3000/")
    .addConverterFactory(GsonConverterFactory.create())
    .build()

val service = retrofit.create(ApiService::class.java)
val call = service.getChildren() // Obtener todos los niños

call.enqueue(object : Callback<List<Child>> {

```

Ilustración 19: Configuración de Retrofit.

Fuente: Elaboración Propia.

Retrofit es una librería de Android utilizada como cliente HTTP para consumir APIs RESTful. Su principal ventaja es que permite convertir automáticamente las respuestas JSON del servidor en objetos de Kotlin/Java definidos en la aplicación, simplificando la comunicación entre el frontend móvil y el backend.

Explicación general del código:

- La app realiza la petición HTTP a la instancia EC2.
- La instancia EC2 está configurada en AWS para aceptar tráfico en el puerto 3000.
- El backend (Node.js + Express.js) recibe la petición, consulta la base de datos MySQL en Amazon RDS y devuelve la respuesta en JSON.
- Retrofit recibe la respuesta y la convierte en un objeto Kotlin (por ejemplo, List<Child>), mostrando los datos en la interfaz de usuario.

II. Endpoints de la API REST

```
import retrofit2.Call
import retrofit2.http.GET
import retrofit2.http.Query

interface ApiService {
    @GET("children/") // Endpoint para obtener la lista de niños
    fun getChildren(): Call<List<Child>>

    @GET("items/") // Endpoint para obtener los ítems filtrados por childId
    fun getItemsByChildId(@Query("childId") childId: Int): Call<List<Item>>
}
```

Ilustración 20: Endpoints de la API REST.

Fuente: Elaboración Propia.

Explicación general del código:

- El método `getChildren()` permite realizar una solicitud HTTP GET al endpoint `/children/`, obteniendo como respuesta una lista de objetos de tipo `Child`, que representan a los niños registrados en el sistema.
- El método `getItemsByChildId(childId: Int)` realiza una solicitud GET al endpoint `/items/`, filtrando los ítems por el identificador del niño (`childId`) recibido como parámetro. El resultado se retorna como una lista de objetos `Item`.

Integración con Retrofit:

Retrofit convierte automáticamente las respuestas JSON obtenidas desde el servidor en listas de objetos tipados (`List<Child>`, `List<Item>`), gracias al uso de `GsonConverterFactory`. Esto permite una comunicación fluida, segura y desacoplada entre la aplicación móvil y el backend remoto desplegado en la instancia EC2.

III. Autenticación de Usuarios con Firebase

Para implementar el sistema de autenticación, se utilizó Firebase Authentication, que permite gestionar el registro e inicio de sesión mediante correo electrónico y contraseña. Esta integración facilita la administración de usuarios, garantizando un inicio de sesión seguro y controlado.

```
private fun setup() {
    title= "Autenticación"

    signUpButton.setOnClickListener {
        if (emailEditText.text.isNotEmpty() && passwordEditText.text.isNotEmpty()){

            FirebaseAuth.getInstance()
                .createUserWithEmailAndPassword(emailEditText.text.toString(),
                    passwordEditText.text.toString()).addOnCompleteListener {
                    if (it.isSuccessful) {
                        showHome( email: it.result?.user?.email ?: "", ProviderType.BASIC)
                    } else {
                        showAlert()
                    }
                }
        }
    }
}
```

Ilustración 21: Código de seguridad (Autenticación)

Fuente: Elaboración propia.

Para el registro de nuevos usuarios en la plataforma móvil se utilizó *Firebase Authentication*, permitiendo una autenticación segura sin manejar credenciales directamente. El método *createUserWithEmailAndPassword* toma el correo y la contraseña desde los campos de entrada, y si el registro es exitoso, redirige al usuario a la pantalla principal mediante *showHome*; en caso de error, se despliega una alerta. Esta integración asegura cifrado, validación en la nube y una gestión escalable de usuarios, cumpliendo con buenas prácticas en aplicaciones móviles.

IV. Geolocalización de hogares beneficiarios:

```
override fun onMapReady(googleMap: GoogleMap) {
    mMap = googleMap

    // Marcamos los hogares de menores en la comuna de Concepción
    val concepcion1 = LatLng( latitude: -36.821761169941695, longitude: -73.03241929293863)
    val concepcion2 = LatLng( latitude: -36.79802040125925, longitude: -73.03256437210109)
    val concepcion3 = LatLng( latitude: -36.83074307803708, longitude: -73.03928455570899)
    val concepcion4 = LatLng( latitude: -36.812628677290526, longitude: -73.05166403329291)

    // Marcamos los hogares de menores en la comuna de Hualpén
    val hualpen = LatLng( latitude: -36.790288749883736, longitude: -73.08512071150456)

    //Concepción
    mMap.addMarker(MarkerOptions().position(concepcion1).title("Hogar de Niños Arnulfo Concepción"))
    mMap.addMarker(MarkerOptions().position(concepcion2).title("Hogar de Menores Club de Leones Andalien"))
    mMap.addMarker(MarkerOptions().position(concepcion3).title("Hogar de Menores Madre Teresa Bacq"))
    mMap.addMarker(MarkerOptions().position(concepcion4).title("Hogar de Menores Santa Rosa de Lima"))

    //Hualpén
    mMap.addMarker(MarkerOptions().position(hualpen).title("Hogar de Menores Santa Rosa de Lima"))
}
```

Ilustración 22: código Geolocalización de hogares beneficiarios.

Fuente: Elaboración Propia.

La aplicación integra Google Maps SDK para mostrar visualmente la ubicación de los hogares infantiles. Al iniciar la actividad `MapActivity.kt`, se define un marcador con coordenadas específicas (`LatLng`) y se renderiza con `mMap.addMarker()`. Esta función posiciona al instante la dirección del hogar en el mapa, permitiendo al usuario ubicar geográficamente a los beneficiarios y facilitando decisiones informadas al momento de donar o apadrinar. La visualización es dinámica y permite escalar la funcionalidad para agregar múltiples marcadores a futuro.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN

4.1 Plan de pruebas funcional del proyecto.

Este capítulo presenta el proceso de validación técnica y de usuario aplicado a la plataforma móvil desarrollada para la gestión de donaciones y apadrinamientos. El objetivo fue comprobar que el sistema cumple con los requerimientos funcionales y no funcionales definidos, validando su funcionamiento desde una perspectiva técnica, así como su usabilidad y pertinencia para los usuarios finales.

Se llevaron a cabo dos tipos de validación complementaria:

- Validación funcional (pruebas técnicas).
- Validación de usuarios (experiencia real de uso y nivel de satisfacción).

4.2 Validación de Funcionalidad Técnica

4.2.1 Plan de Pruebas Funcionales

Se llevaron a cabo pruebas unitarias y de integración enfocadas en las funcionalidades críticas de la plataforma. Estas pruebas permitieron verificar que cada módulo cumpliera con los requerimientos establecidos y funcionara correctamente al ser interconectado con otros componentes del sistema.

Tabla 1: Pruebas unitarias y de integración.

Fuente: Elaboración propia.

ID prueba	Funcionabilidad	Descripción	Resultado esperado	Resultado obtenido
PF-01	Registro de Usuario	El usuario debe poder registrarse correctamente.	El sistema registra los datos y crea una cuenta.	Éxito
PF-02	Autenticación de usuario	Validar inicio de sesión con credenciales válidas.	Acceso permitido/denegado según credenciales.	Éxito

PF-03	Publicación de necesidades	El hogar debe poder publicar artículos requeridos.	Publicación visible en la galería.	Éxito
PF-04	Notificaciones	Validar el envío automático de notificaciones.	El sistema envía la notificación correspondiente.	Éxito
PF-05	Pasarela de donaciones	Simular el flujo de donaciones monetarias.	Visualizar flujo hasta el punto previo al pago.	Pendiente

4.2.2 Pruebas de Integración

Se validó la correcta integración entre los diferentes módulos del sistema, considerando:

Base de Datos MySQL (Amazon RDS): Verificación de conexión, almacenamiento y recuperación de datos.

Rutas API (Next.js): Validación de operaciones CRUD y comunicación entre frontend-backend.

Gestión de imágenes (Cloudinary): Pruebas exitosas de carga y visualización de imágenes asociadas a perfiles.

4.3 Validación con Usuarios Finales

Objetivo

La validación con usuarios tuvo como propósito evaluar la usabilidad, comprensión del flujo y percepción de valor de la aplicación móvil por parte de personas reales. Se realizó con un grupo reducido de usuarios pertenecientes al entorno cercano al hogar, quienes probaron la aplicación como si fueran usuarios reales.

Pruebas de Aceptación

Las siguientes pruebas reflejan el grado de satisfacción de los usuarios tras interactuar con la aplicación en su versión funcional actual:

Tabla 2: Feedback De Usuarios.

Fuente: elaboración propia.

ID Prueba	Escenario	Feedback del Usuario	Nivel de satisfacción
PF-01	Registro de Usuario e Inicio de sesión	“El proceso es simple y rápido.”	95%
PF-02	Búsqueda de necesidades	“Me gusta poder filtrar por ubicación y artículos específicos.”	90%
PF-03	Publicación de necesidades	“La interfaz es clara y fácil de usar.”	85%
PF-04	Proceso de donación	“Simulación del proceso funciona, aunque falta la pasarela de pago.”	85%

4.4 Análisis de Resultados

Los resultados obtenidos en ambas perspectivas permiten concluir lo siguiente:

- La plataforma es funcional y cumple con los requerimientos técnicos principales definidos en la propuesta inicial.
- El proceso de registro, autenticación y publicación de necesidades funciona de manera estable y satisfactoria.
- Los usuarios reportaron altos niveles de comprensión e intuición en los flujos de la aplicación.
- El módulo de donaciones fue probado de forma parcial, pendiente de integración con un sistema real de pago en futuras etapas.

CAPÍTULO 5: CONCLUSIONES

Este proyecto abordó un problema social crítico: la falta de un sistema eficiente, transparente y accesible para gestionar donaciones destinadas a niños en hogares de menores en Chile. Según el Servicio Nacional de Menores (SENAME), actualmente más de 7,000 niños residen en hogares de menores en el país, enfrentando condiciones de vulnerabilidad que afectan directamente su calidad de vida. Este contexto refleja la urgencia de implementar soluciones tecnológicas que optimicen la gestión de recursos y fomenten la participación de donantes.

A lo largo del desarrollo, se identificaron y superaron importantes desafíos. Los métodos tradicionales para gestionar donaciones en hogares de menores suelen ser manuales y carecen de transparencia, lo que genera desconfianza entre los donantes. Una encuesta realizada como parte del proceso de validación mostró que el 78% de los usuarios considera esencial tener acceso a información detallada sobre el destino de sus aportes, mientras que el 65% indicó que los métodos actuales desincentivan su participación por falta de confianza. En respuesta, la plataforma implementó funcionalidades clave como perfiles detallados de los niños, una galería interactiva, un sistema seguro de registro y autenticación mediante Firebase, y la utilización de servicios de AWS para garantizar escalabilidad y seguridad.

La plataforma propuesta permitió superar estas limitaciones mediante el diseño de funcionalidades, como perfiles detallados de los niños con sus necesidades específicas, un sistema de donaciones interactivo y un mapa geográfico que visibiliza la ubicación de los hogares beneficiarios. Si bien se logró una mejora en la digitalización del proceso, la ausencia de una pasarela de pagos en esta versión inicial limitó el alcance total de la solución. Esta implementación será prioritaria en futuras iteraciones para garantizar un sistema completo y funcional.

Es importante destacar que la solución desarrollada no solo digitalizó un sistema previamente manual, sino que también lo transformó en un modelo eficiente, confiable y replicable. Sin embargo, como toda iniciativa tecnológica, existen áreas de mejora que podrían abordarse en versiones futuras. Además, la integración de herramientas de análisis de datos podría proporcionar información estratégica para optimizar la asignación de recursos y priorizar las necesidades de los niños de manera más efectiva.

Desde una perspectiva técnica, este proyecto reafirma la importancia de la planificación estratégica y la selección adecuada de tecnologías para garantizar una solución escalable y sostenible. La utilización de AWS para la infraestructura en la nube, Firebase para la autenticación y Android SDK para el desarrollo móvil demuestra que las herramientas

tecnológicas modernas pueden ser utilizadas de manera efectiva para resolver problemas sociales. Asimismo, el enfoque centrado en el usuario final fue clave para garantizar que la solución respondiera a las expectativas y necesidades tanto de los donantes como de los administradores de los hogares.

En términos de impacto social, este proyecto representa una contribución tangible y significativa. La plataforma no solo facilita el proceso de donación, sino que también fomenta la confianza y la participación de la comunidad en la mejora de las condiciones de vida de los niños en situación de vulnerabilidad. Además, su potencial para ser escalado o adaptado a otros contextos amplía su alcance, posicionándolo como una solución versátil y de alto impacto.

Adicionalmente, se destaca que se cumplieron satisfactoriamente todos los objetivos específicos planteados en el Capítulo 1:

- Se investigaron y analizaron los procesos actuales de donación, identificando brechas y oportunidades de mejora.
- Se diseñó e implementó una plataforma digital adaptada a las necesidades tanto de donantes como de administradores de hogares.
- Se integraron funcionalidades clave como el registro, la visualización de infantes, la realización de donaciones y el apadrinamiento, junto con visualización geográfica.
- Se validó el funcionamiento del sistema mediante pruebas funcionales alineadas con los casos de uso definidos.
- Se propusieron mejoras proyectivas para una futura implementación robusta y escalable.

Todo lo anterior permitió alcanzar el objetivo general del proyecto, que consistía en desarrollar una plataforma tecnológica que facilite, optimice y sea transparente el proceso de gestión de donaciones a hogares de menores, integrando tecnología, usabilidad y escalabilidad como pilares fundamentales.

En conclusión, este proyecto es una prueba de cómo la ingeniería informática puede ser una herramienta poderosa para generar cambios positivos en la sociedad. Más allá de resolver una problemática específica, este trabajo demuestra que la tecnología, cuando se diseña con un enfoque centrado en las personas, tiene el potencial de transformar vidas. Espero que esta plataforma inspire a otros a continuar desarrollando soluciones tecnológicas con un enfoque social, promoviendo una sociedad más justa, inclusiva y solidaria.