

# GastanGO

Sistema de Gestión de Gastos Personales (PERN + Mobile)

Joel Stalin Tapia Pinta      Freddy Alexander Matailo Mora

2026-02-06

# Tabla de contenidos

<b>Resumen Ejecutivo</b>	<b>2</b>
<b>1. Introducción</b>	<b>3</b>
<b>2. Objetivos</b>	<b>4</b>
2.1. Objetivo General . . . . .	4
2.2. Objetivos Específicos . . . . .	4
<b>3. Alcance del Proyecto</b>	<b>5</b>
<b>4. Arquitectura del Sistema</b>	<b>6</b>
4.1. Diagramas C4 (Structurizr) . . . . .	6
4.1.1. Galería de Arquitectura . . . . .	6
4.2. Llaves de Arquitectura (Leyendas) . . . . .	6
4.2.1. Leyendas por Nivel de Arquitectura . . . . .	8
4.3. Stack Tecnológico . . . . .	8
<b>5. Diseño Móvil (Cross-Platform)</b>	<b>9</b>
5.1. Capturas de Pantalla de la Aplicación Móvil . . . . .	9
<b>6. Seguridad e Implementación</b>	<b>11</b>
<b>7. Documentación Multimedia</b>	<b>12</b>
7.1. Demostración Funcional . . . . .	12
<b>8. Manual de Usuario</b>	<b>13</b>
8.1. Instalación . . . . .	13
8.1.1. Requisitos . . . . .	13
8.1.2. Descarga de la Aplicación . . . . .	13
8.2. Uso Básico . . . . .	13
<b>9. Repositorio y Código Fuente</b>	<b>14</b>
<b>10. Conclusiones</b>	<b>15</b>

# Resumen Ejecutivo

## **i** Sobre el Proyecto

**GastanGO** es un ecosistema digital diseñado para la salud financiera. Combina la robustez de un backend empresarial con la flexibilidad de una experiencia móvil nativa.

GastanGO es un sistema multiplataforma orientado a la gestión y control de gastos, desarrollado bajo el stack **PERN (PostgreSQL, Express, React y Node.js)** e integrado con una **aplicación móvil cross-platform** construida con **Expo (React Native)**.

El proyecto busca ofrecer una solución accesible, escalable y segura para el registro, visualización y análisis de gastos personales.

---

# Capítulo 1

## Introducción

En la actualidad, la gestión eficiente de los recursos económicos personales se ha convertido en una necesidad fundamental. GastanGO surge como una propuesta tecnológica que integra tecnologías web modernas y desarrollo móvil multiplataforma.

Este documento presenta la **documentación técnica, arquitectónica y el manual de usuario**, describiendo su funcionamiento interno y guías de uso.

## Capítulo 2

# Objetivos

### 2.1. Objetivo General

Desarrollar un sistema multiplataforma para la gestión de gastos personales utilizando el stack **PERN** y una aplicación móvil basada en **Expo**.

### 2.2. Objetivos Específicos

- **Arquitectura:** Implementar un modelo cliente-servidor desacoplado y escalable.
- **Persistencia:** Diseñar e integrar una base de datos relacional robusta en PostgreSQL.
- **Movilidad:** Desarrollar una aplicación móvil nativa funcional para Android e iOS.
- **UX/UI:** Garantizar una experiencia de usuario intuitiva mediante interfaces modernas.

## Capítulo 3

# Alcance del Proyecto

Incluye	No Incluye
Registro y autenticación (JWT)	Integración bancaria en tiempo real
Gestión CRUD de gastos	Pagos directos desde la app
Visualización Web y Móvil	Inteligencia Artificial (Fase 2)
Sincronización vía API REST	

## Capítulo 4

# Arquitectura del Sistema

El sistema sigue un patrón de diseño **cliente-servidor** desacoplado. A continuación se detallan los niveles basados en el **Modelo C4**.

### 4.1. Diagramas C4 (Structurizr)

Para entender la arquitectura, utilizamos el modelo C4. Los siguientes diagramas están en formato vectorial (SVG) para una visualización de alta calidad sin pérdida de resolución.

 Guía: Cómo interpretar los diagramas Structurizr

#### Niveles de Zoom:

1. **C1 - Context:** Visión general del negocio y actores externos.
2. **C2 - Container:** Tecnología (Web, API, Mobile, DB).
3. **C3 - Component:** Estructura interna (Controladores, Servicios).
4. **C4 - Despliegue:** Infraestructura y configuración de despliegue.

#### 4.1.1. Galería de Arquitectura

[Explorar diagramas](#) [Ver Keys](#)

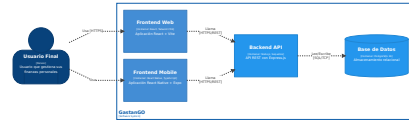
### 4.2. Llaves de Arquitectura (Leyendas)

Los diagramas de arquitectura utilizan una **notación estandarizada** basada en el modelo C4. A continuación se presentan las llaves o leyendas que explican los elementos visuales empleados en cada nivel arquitectónico:



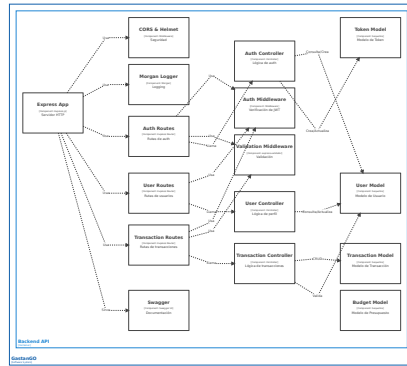
System Context View: GastanGO  
jueves, 5 de febrero de 2026, 21:41 hora estándar de Colombia

(a) Vista de Contexto (C1)



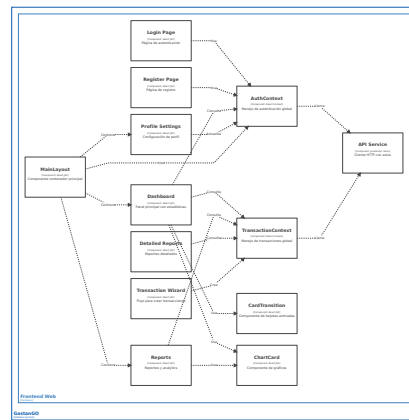
Container View: GastanGO  
jueves, 5 de febrero de 2026, 21:41 hora estándar de Colombia

(a) Vista de Contenedores (C2)



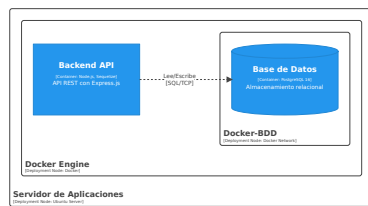
Component View: GastanGO - Backend API  
jueves, 5 de febrero de 2026, 21:41 hora estándar de Colombia

(a) Componentes Backend (C3)



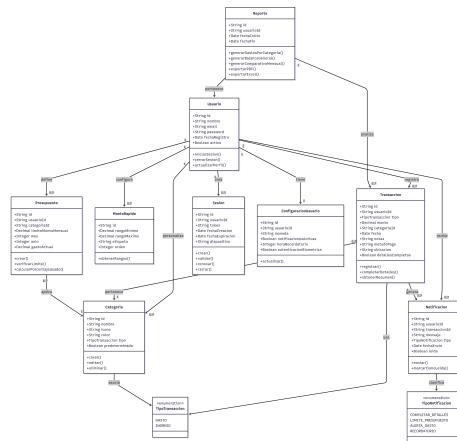
Component View: GastanGO - Frontend Web  
jueves, 5 de febrero de 2026, 21:41 hora estándar de Colombia

(a) Componentes Frontend (C3)

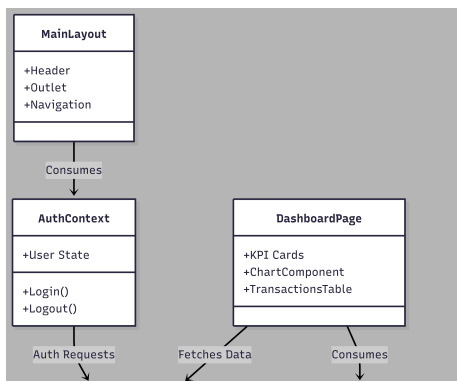


Deployment View: GastanGO - Production  
jueves, 5 de febrero de 2026, 21:41 hora estándar de Colombia

(a) Infraestructura de Despliegue



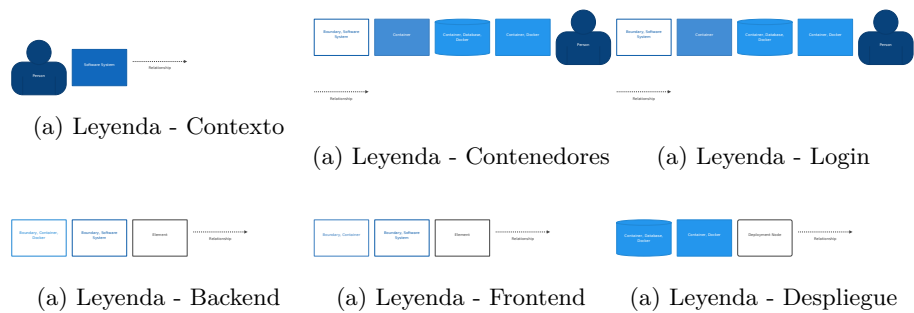
(a) Diagrama de código



### 💡 Importancia de las Llaves

Las llaves proporcionan una guía visual para interpretar correctamente cada diagrama, identificando los tipos de componentes, las relaciones entre ellos y las tecnologías empleadas.

#### 4.2.1. Leyendas por Nivel de Arquitectura



### ! Convenciones de Diseño

- **Rectángulos azules:** Sistemas externos (usuarios, servicios de terceros)
- **Rectángulos verdes:** Contenedores de la aplicación (Web, API, Mobile, DB)
- **Componentes internos:** Capas de controladores, servicios y modelos
- **Flechas continuas:** Comunicación sincrónica (HTTP/REST)
- **Flechas punteadas:** Dependencias o relaciones opcionales

#### 4.3. Stack Tecnológico

El sistema sigue una arquitectura **cliente-servidor** desacoplada:

- **Frontend Web:** React (Vite + Tailwind CSS)
- **Backend:** Node.js + Express (API REST)
- **Base de Datos:** PostgreSQL 16
- **Mobile App:** Expo (React Native)

## Capítulo 5

# Diseño Móvil (Cross-Platform)

La aplicación móvil fue desarrollada siguiendo el paradigma **cross-platform**, permitiendo su ejecución tanto en Android como en iOS a partir de una única base de código.

### Ventajas Clave:

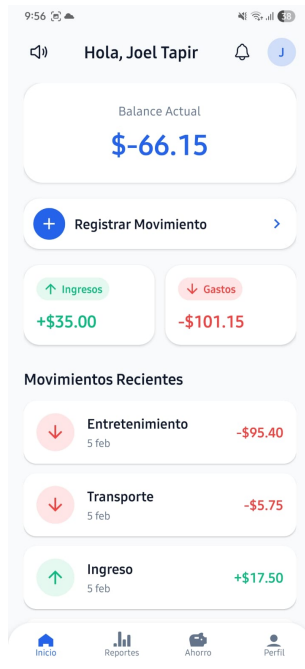
1. Reducción de costos de desarrollo.
2. Mantenimiento unificado.
3. Consistencia en la experiencia de usuario.

### 5.1. Capturas de Pantalla de la Aplicación Móvil

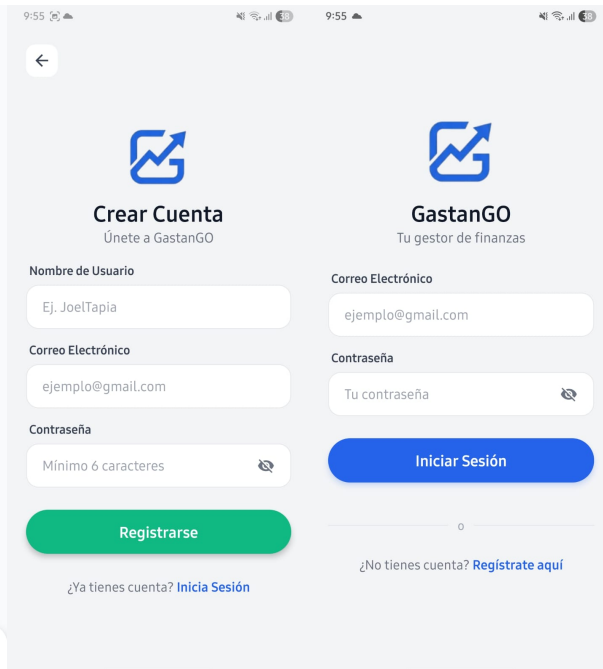
A continuación se presentan las principales pantallas de la aplicación móvil **GastanGO**, mostrando la interfaz de usuario y las funcionalidades implementadas:

#### Experiencia de Usuario

La aplicación mantiene una interfaz consistente y moderna, siguiendo los principios de **Material Design** para Android y **Human Interface Guidelines** para iOS, garantizando una experiencia nativa en ambas plataformas.

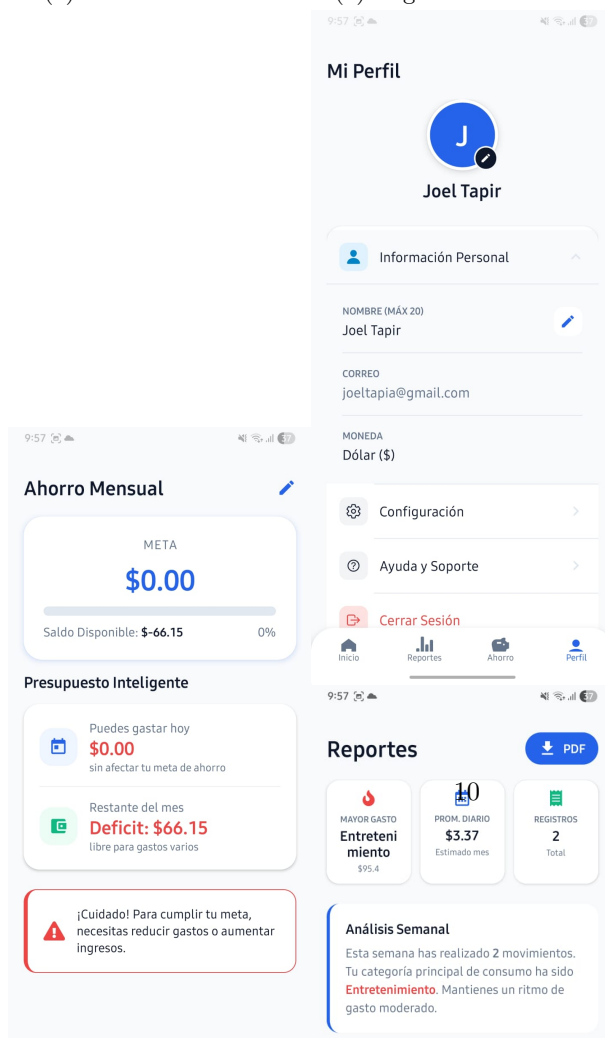


(a) Pantalla de Inicio



(a) Registro de Usuario

(a) Inicio de Sesión



## Capítulo 6

# Seguridad e Implementación

El sistema implementa mecanismos de defensa en profundidad:

- **Autenticación:** JWT (JSON Web Tokens).
  - **Validación de Datos:** Uso de middlewares en Express con librerías de validación (Zod/Express-validator).
  - **Cifrado:** Hasheo de credenciales en base de datos.
  - **Rutas Protegidas:** Guardias de navegación en Frontend y Mobile.
  - **Cross-Platform:** Código unificado mediante el SDK de Expo.
-

## Capítulo 7

# Documentación Multimedia

### 7.1. Demostración Funcional

A continuación, se presenta una demostración funcional del sistema en operación:

El video demostrativo del sistema está disponible en el siguiente enlace:

[Ver Video en YouTube](#)

---

## Capítulo 8

# Manual de Usuario

### 8.1. Instalación

#### 8.1.1. Requisitos

- Dispositivo Android (6.0+) o iOS (13+).
- Conexión a Internet activa.

#### 8.1.2. Descarga de la Aplicación

Puedes acceder a la última compilación de desarrollo a través de Expo:

[Descargar GastanGO Mobile \(Expo Build\)](#)

### 8.2. Uso Básico

1. **Registro:** Crea una cuenta con tu correo electrónico.
2. **Dashboard:** Visualiza tu saldo actual y resumen de transacciones.
3. **Nueva Transacción:** Pulsa el botón + para agregar un gasto o ingreso.
4. **Historial:** Revisa tus movimientos pasados en la pestaña de reportes.

## Capítulo 9

# Repositorio y Código Fuente

El desarrollo es de código abierto y está disponible para revisión y contribución.

<https://github.com/Jxel117/GastanGO>

[Ver Repositorio en GitHub](#)

## Capítulo 10

# Conclusiones

GastanGO demuestra la viabilidad de integrar tecnologías web modernas con desarrollo móvil multiplataforma, ofreciendo una solución práctica y eficiente para la gestión de finanzas personales.

El proyecto evidencia la aplicación de buenas prácticas en arquitectura de software, seguridad y experiencia de usuario.

---

### Desarrollado por:

- **Joel Stalin Tapia Pinta** (Jxel117)
- **Freddy Alexander Matailo Mora** (FreddyAlx1998Mora12)

<https://github.com/Jxel117/>

<https://github.com/FreddyAlx1998Mora12>

*Universidad Nacional de Loja - Ingeniería en Ciencias de la Computación - 2026*