Meli Prototype: Design Documentation and Development Report

Valerie Parra Cortés

August 26, 2025

Contents

1 Project Overview

The Meli Prototype is a full-stack e-commerce application designed to replicate item detail functionality. The catalog scope was intentionally reduced as much as possible in order to keep the project manageable within the limited timeframe available.

2 Design Choices

Initial research

I analyzed the Mercado Libre website to understand how this functionality currently works. I found that there are two main rendering approaches: one for top-selling items and another for regular items. Variants are only rendered for the top-selling items, while standard items have a much simpler visualization. I also studied how Mercado Libre structures its catalog and aimed to design a similar model for this challenge.

Backend technologies

The two options considered were Golang and Java. I personally have more experience with Golang, which led me to choose this technology. Additionally, Golang allows you to build an API very quickly with just a few files, making it a highly efficient choice for the project.

Database Design: PostgreSQL + GORM

I implemented a hybrid approach combining:

- **PostgreSQL**: Robust, ACID-compliant relational database
- **GORM**: Go ORM for simplified database operations and migrations
- **Database Migrations**: Using dbmate for version-controlled schema changes

The main reason for choosing these technologies was that, in the case of the catalog, a relational model was the most scalable option within the business domain. Entities such as relationships, families, and others are accessed from different areas of Mercado Libre, making this approach more suitable.

GORM was a new learning experience, but it proved extremely useful since the implemented model was complex and heavily relationship-based. With just a few attributes, GORM allowed us to manage all these relationships in a simplified way.

Regarding DB Mate, the goal was to completely separate database management from the application code. For this reason, DB Mate was integrated as an internal process within Docker, ensuring a clean and independent handling of schema initialization and migrations.

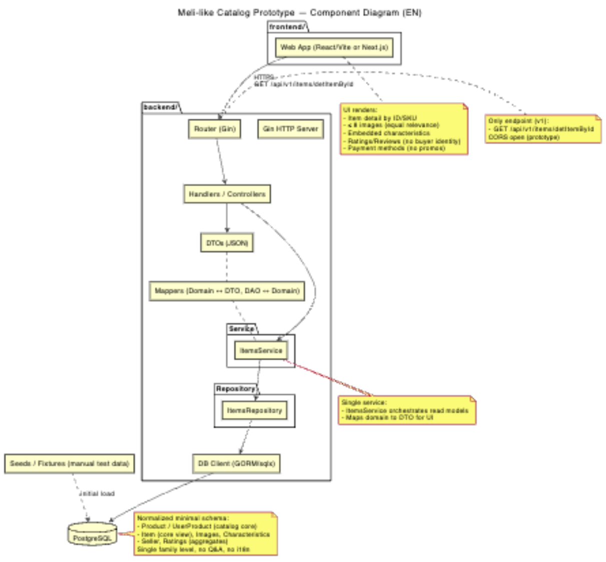
Architecture Pattern: Clean Architecture

The backend follows Clean Architecture principles with clear separation of concerns:

- **Domain Layer**: Core business entities and logic
- **Service Layer**: Business logic orchestration
- **Repository Layer**: Data access abstraction

- **HTTP Layer**: API endpoints and request handling

A visual representation of the architecture can be found here:


Meli-like Catalog Prototype — Component Diagram (EN)
frontend/
Web App (React/Vite or Next.js)
HTTPS
GET /api/v1/items/detItemById
backend/
Router (Gin)
Gin HTTP Server
Handlers / Controllers
DTOs (JSON)
Mappers (Domain ↔ DTO, DAO ↔ Domain)
Service
ItemsService
Repository
ItemsRepository
DB Client (GORM/sqlx)
Seeds / Fixtures (manual test data)
initial load
PostgreSQL
UI renders:
- Item detail by ID/SKU
- ≤ 8 images (equal relevance)
- Embedded characteristics
- Ratings/Reviews (no buyer identity)
- Payment methods (no promos)
Only endpoint (v1):
- GET /api/v1/items/detItemById
CORS open (prototype)
Single service:
- ItemsService orchestrates read models
- Maps domain to DTO for UI
Normalized minimal schema:
- Product / UserProduct (catalog core)
- Item (core view), Images, Characteristics
- Seller, Ratings (aggregates)
Single family level, no Q&A, no i18n

Frontend Architecture

Technology Stack Selection

I chose **React 19** with **TypeScript** for the frontend:

- **React 19**: Latest version with improved performance and concurrent features
- **TypeScript**: Strong typing for better code quality and developer experience
- **Vite**: Modern build tool with excellent development experience
- **Styled Components**: CSS-in-JS for component-scoped styling. This extraction of CSS also helps to the maintainability of the code

3 Challenges Faced and Solutions

Time

Only two days were available to complete this challenge, which made it quite difficult. The initial approach was to conduct thorough research to understand how the functionality worked and how the catalog could be managed. This first interaction with the AI can be found in the file first_architecture_draft.md.

The goal at the start was to design a solid model capable of supporting everything that a real catalog would require. However, it soon became clear that the scope could grow endlessly. As a result, the scope was gradually reduced, and in the end, only the rendering of the simplest Mercado Libre item was implemented.

Database Schema Design

Challenge: Complex E-commerce Relationships

Designing a database schema that accurately represents e-commerce relationships (products, items, sellers, reviews, questions) while maintaining data integrity and performance.

The final solution was implemented by discussing with ChatGPT in its version 5 about how variants were handled in the specific chaos for MELI, this particular chat is located in the repository.

The final database ERD of the app can be found in the following image:

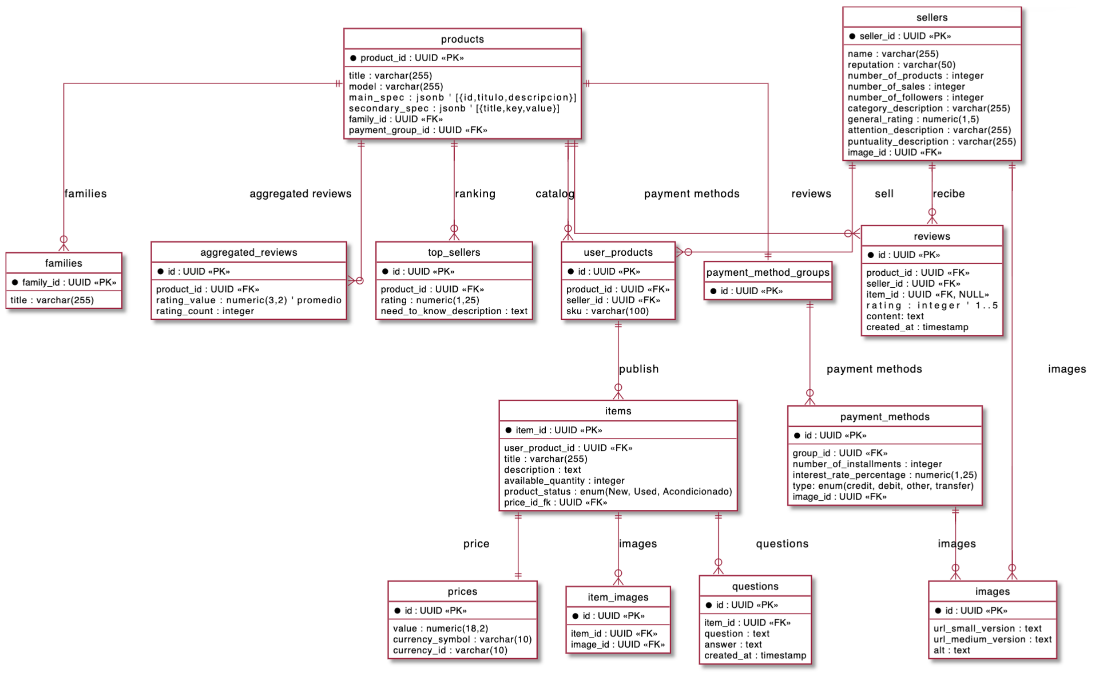


Figure 1: Enter Caption

Generative AI Usage

3.1 AI Models and Agents Used

Primary AI Assistant: Claude Sonnet 4

Why Chosen:

- **Code Quality**: Excellent understanding of Go and React best practices
- **Architecture Knowledge**: Strong grasp of software design patterns
- **Problem Solving**: Ability to analyze complex technical challenges
- **Personal experimentation**: In the month I've been testing the cursor, it's one of the models that generates code the best.

Secondary AI Assistant: ChatGPT 5

When it comes to generating text and explaining context, GPT-5 performs very well, explaining in detail and being able to evaluate the pros and cons of the discussions offered. It is also sometimes useful for troubleshooting specific code issues when Sonet in agent mode cannot.

UI components

Claude was heavily used to generate components. To improve code quality, I explicitly asked it to use styled-components, which forced a clear separation of CSS and TypeScript logic (since AI models tend to generate everything in a linear and tightly coupled manner by default).

Components were requested one by one, providing both screenshots and the expected props each component should receive. In the end, the user interface consisted of more than 30 graphical components. While this level of granularity provides scalability and maintainability, it may not have been the most suitable approach given the strict time constraints of the project. This process was slow and highly iterative, as it required careful validation of component props, colors, borders, and ensuring that the design was fully responsive. Although Cursor did not preserve the original chats, this document describes as faithfully as possible how the process was carried out.

3.2 AI Usage Patterns

Code Generation

- **Boilerplate Code**: Generating standard Go structs, interfaces, and methods
- **API Endpoints**: Creating RESTful endpoints with proper error handling
- **Database Models**: Generating GORM models with proper tags was a key step in the development process. A particularly useful tool in this context was PlantUML. While the model already had a good understanding of the overall architecture, it responds much better to structured text input. Being able to express the intended design through PlantUML diagrams provided valuable context and greatly improved the accuracy of the generated models.
- **React Components**: Creating reusable component templates

Problem Solving

- **Architecture Decisions**: Discussing trade-offs between different approaches
- **Performance Issues**: Analyzing and optimizing database queries
- **Code Review**: Identifying potential improvements and best practices
- **Testing Strategies**: Designing effective test cases and coverage

Documentation and Learning

- **Code Comments**: Generating meaningful documentation
- **README Files**: Creating comprehensive project documentation
- **API Documentation**: Explaining endpoint behavior and data structures
- **Best Practices**: Learning and implementing Go and React conventions

4 Development Methodology

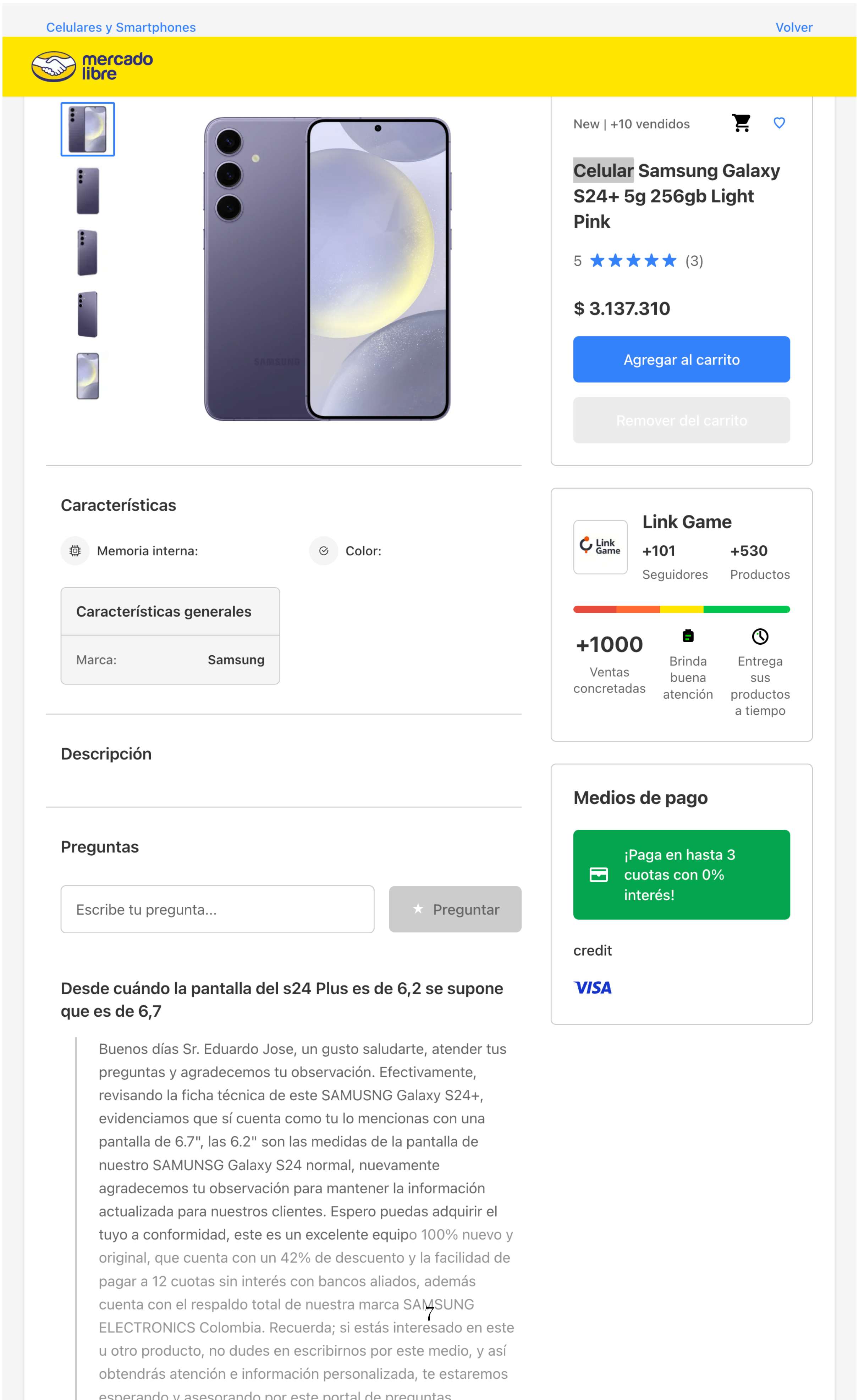
5 Testing Strategy

Most of the tests were auto-generated due to time constraints. The generator was asked to create tests that would cover 80% of the functionality while keeping it simple.

6 Screenshot - Final result

Desktop

A preview of the final result for Desktop and for mobile



New | +10 vendidos

Celular Samsung Galaxy S24+ 5g 256gb Light Pink

5 ★★★★★ (3)

$ 3.137.310

Link Game

+101	+530
Seguidores	Productos

+1000 Ventas concretadas

Brinda buena atención

Entrega sus productos a tiempo

Medios de pago

¡Paga en hasta 3 cuotas con 0% interés!

credit

VISA

Características

Memoria interna:

Color:

Características generales	
Marca:	Samsung

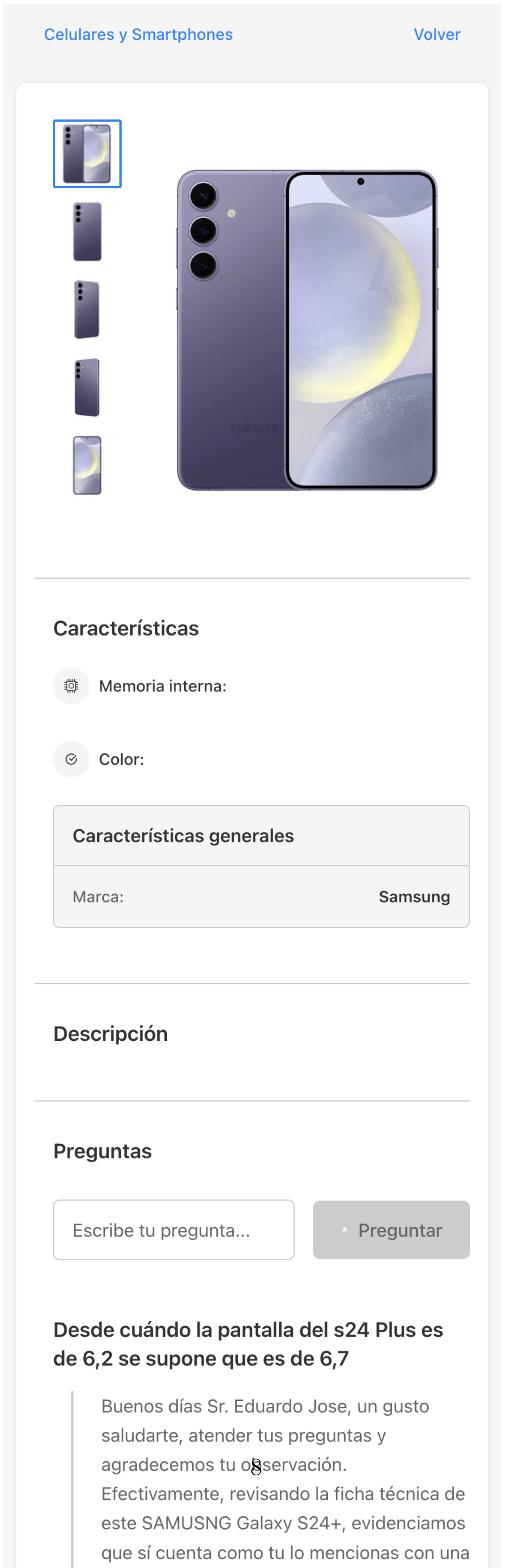
Descripción

Preguntas

Desde cuándo la pantalla del s24 Plus es de 6,2 se supone que es de 6,7

Buenos días Sr. Eduardo Jose, un gusto saludarte, atender tus preguntas y agradecemos tu observación. Efectivamente, revisando la ficha técnica de este SAMUSNG Galaxy S24+, evidenciamos que sí cuenta como tu lo mencionas con una pantalla de 6.7", las 6.2" son las medidas de la pantalla de nuestro SAMUNSG Galaxy S24 normal, nuevamente agradecemos tu observación para mantener la información actualizada para nuestros clientes. Espero puedas adquirir el tuyo a conformidad, este es un excelente equipo 100% nuevo y original, que cuenta con un 42% de descuento y la facilidad de pagar a 12 cuotas sin interés con bancos aliados, además cuenta con el respaldo total de nuestra marca SAMSUNG ELECTRONICS Colombia. Recuerda; si estás interesado en este u otro producto, no dudes en escribirnos por este medio, y así obtendrás atención e información personalizada, te estaremos esperando y asesorando por este portal de preguntas

Mobile



Celulares y Smartphones Volver

Características

Memoria interna:

Color:

Características generales	
Marca:	Samsung

Descripción

Preguntas

Desde cuándo la pantalla del s24 Plus es de 6,2 se supone que es de 6,7

Buenos días Sr. Eduardo Jose, un gusto saludarte, atender tus preguntas y agradecemos tu observación. Efectivamente, revisando la ficha técnica de este SAMUSNG Galaxy S24+, evidenciamos que sí cuenta como tu lo mencionas con una