Mercado Libre XS

Daniel Alonso Chavarro Chipatecua 20241020066
Universidad Francisco José de Caldas

Abstract

Mercado Libre XS is an e-commerce platform that allows users to buy and sell products online. The platform relies on a **FastAPI backend** that provides RESTful services to interact with a **MySQL database**. This poster details the **API services, database architecture, and system structure** that ensure scalability, security, and efficient e-commerce operations.

Project Structure

Mercado Libre XS is built using a **microservices-based architecture**, with a **FastAPI backend** and a **relational database**.

System Components

- **Frontend:** React.js for user interaction.
- **Backend:** FastAPI to handle requests and business logic.
- **Database:** MySQL with optimized entity relationships.
- **Security:** Authentication via JWT and password hashing.
- **Scalability:** Designed with modular APIs and cloud-ready deployment.

Key Features

- **User Management** – Account creation, authentication, and profile updates.
- **Product Listings** – CRUD operations for products, including filtering and searching.
- **Shopping Cart** – Add/remove products and manage purchases.
- **Secure Payments** – Integration with external payment gateways.
- **Order Tracking** – API services to manage deliveries and order statuses.

Database Design

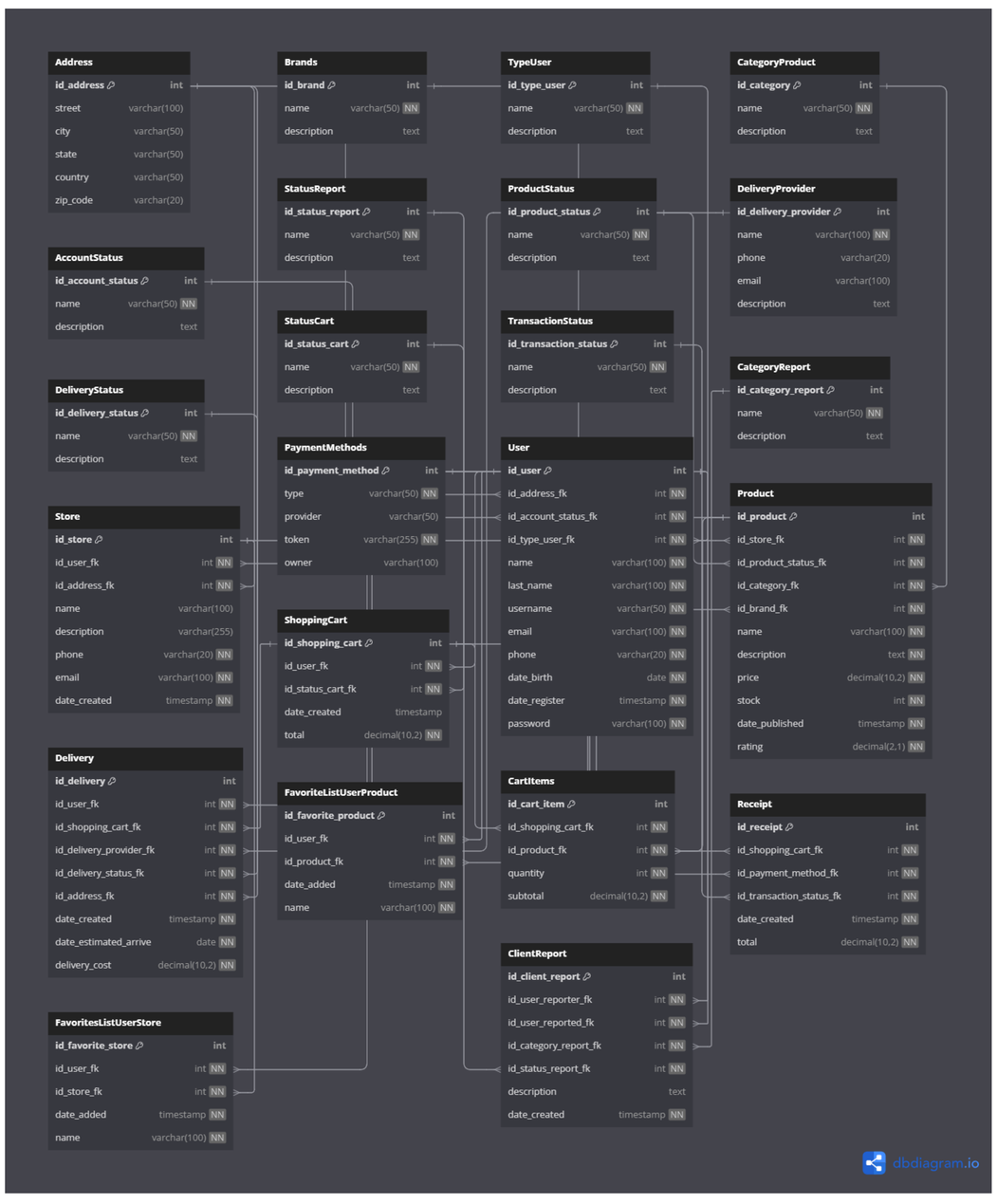
The relational database follows **normalization principles** to minimize redundancy and ensure consistency.

Main Entities

- **User** – Stores personal details and authentication info.
- **Product** – Contains product data such as name, price, and category.
- **Transaction** – Links users, products, and payment methods.
- **Shopping Cart** – Temporarily stores selected items before checkout.

Data Integrity and Constraints

- **Primary Keys** – Unique identifiers for each entity.
- **Foreign Keys** – Ensure relationships between tables.
- **Indexes** – Improve query performance for frequent searches.
- **Normalization** – Eliminates data redundancy and maintains consistency.



API Services and Database Queries

The backend exposes RESTful API endpoints to interact with the database. Below are the key services and their corresponding SQL queries.

User Management

Create a new user:

```
INSERT INTO Users (id_user,id_address_fk, id_account_status_fk, id_type_user_fk, name, last_name,
  username, email, phone, date_birth, date_register, password)
VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s);
```

Retrieve user by ID:

```
SELECT * FROM Users WHERE id_user = %s;
```

Product Management

Insert a new product:

```
INSERT INTO Product (id_store_fk, id_product_status_fk, id_category_fk, id_brand_fk, name,
  description, price, stock, date_published, rating)
VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s);
```

Get all products:

```
SELECT * FROM Products;
```

Filter products by category:

```
SELECT * FROM Products WHERE id_category_fk = %s;
```

Shopping Cart Service

Add item to cart:

```
INSERT INTO ShoppingCart (id_status_cart_fk, id_user_fk, date_created, total)
VALUES (%s, %s, %s, %s);
```

Get items in a user's cart:

```
SELECT p.name, p.price, c.quantity
FROM ShoppingCart c
JOIN Products p ON c.id_product = p.id_product
WHERE c.id_user = %s;
```

Backend Architecture

The Mercado Libre XS backend is built using **FastAPI**, a high-performance web framework for Python that enables **asynchronous operations, scalability, and efficient API handling**. It follows a **modular microservices approach** to ensure flexibility, maintainability, and easy integration with external services.

System Architecture

The backend is structured into multiple service layers:

- **API Layer:** Handles HTTP requests and responses.
- **Service Layer:** Implements business logic and processes requests.
- **Database Layer:** Communicates with MySQL to execute queries.
- **Security Layer:** Manages authentication, authorization, and data encryption.

Technology Stack

- **FastAPI** – Handles API requests and responses efficiently.
- **MySQL** – Stores structured e-commerce data.
- **SQLAlchemy ORM** – Facilitates interaction with the database.
- **JWT Authentication** – Secures user authentication.
- **Docker** – Enables containerized deployment.

API Interactions and Database Communication

FastAPI interacts with the MySQL database through **psycopg2-binary** for Posgress and **mysql** for MySql, allowing for **efficient CRUD operations** while maintaining integrity.

Request Handling Process

When a client (frontend or external service) makes an API request, the backend follows these steps:

1. The **API Layer** receives the request (e.g., fetching a product or creating a transaction).
2. The **Service Layer** processes the request and applies business rules.
3. The **Database Layer** interacts with MySQL to **retrieve or modify data**.
4. The response is returned in an object.

Database Connection Management

FastAPI manages database connections using psycopg or mysql. Each request opens a new database session, performs the operation, and then closes the session to **prevent memory leaks**.

Conclusion

The Mercado Libre XS platform leverages **FastAPI for API services** and **MySQL for structured data storage**. The integration of these technologies provides a **scalable, secure, and efficient** environment for e-commerce operations. With optimized **database design, API structure, and security practices**, the system ensures high performance and flexibility for future enhancements.