LiftDrop

Mobile App for Deliveries

Gonçalo Morais, n.º 49502, e-mail: a49502@alunos.isel.pt, tel.: 927468061
João Ramos, n.º 49424, e-mail: a49424@alunos.isel.pt, tel.: 919222551

Orientador: Miguel Gamboa, e-mail: miguel.gamboa@isel.pt

Co-Orientador: Diogo Silva, e-mail: diogo.silva@lyzer.tech

Março de 2025

Introduction

Contextualization

The rise of the **gig economy** has transformed various industries, particularly in the food delivery sector. This economic model is characterized by short-term, flexible work arrangements where individuals take on temporary jobs, often facilitated through digital platforms. Companies such as **Uber Eats, Glovo, and Bolt Food** have leveraged this framework to revolutionize urban logistics, enabling customers to order food on demand while providing couriers with independent work opportunities.

By prioritizing **customer convenience**, these platforms have successfully streamlined food delivery processes through real-time technology, ensuring efficiency and reliability at scale. However, as demand for delivery services continues to grow, the gig economy also presents operational challenges, particularly affecting the couriers who sustain this system.

Project Goals

To address these challenges, this project aims to develop a delivery-focused mobile application that enhances the working experience of couriers by:

- **Enhancing decision-making** by providing couriers with **real-time demand insights** through heat maps that highlight high-order density areas.
- **Increasing courier safety** by implementing a **safety rating system** for neighborhoods and delivery areas.
- **Enabling earnings tracking** (Optional) to help couriers monitor income and performance trends.

Research

Developing a mobile application focused on delivery presents challenges, particularly in real-time data handling and order assignment. Since multiple couriers interact with the platform simultaneously, the system must manage updates effectively while ensuring a smooth user experience. After some research and consideration, we identified two primary approaches to handling real-time updates and concurrency:

- **Polling-based Request-Response Model**, where the client periodically requests updates from the server. Althoughugh simple to implement, this approach leads to unnecessary network requests, increased server load, and delayed updates.
- **Event-Driven Server-Sent Events (SSE)**, where the server pushes updates to clients as they occur. This enables real-time communication without the inefficiencies of constant polling.

We chose SSE for real-time updates due to its reliability and maintainability. To implement Server-Sent Events (SSE), we will use OkHttp to manage the connection, as Android lacks native SSE support. This ensures real-time updates without the inefficiencies of polling.

For order assignment, the platform will consider factors like **Proximity-based allocation**, **Fair distribution**, and **Traffic-aware routing**. An external geospatial API will handle location-based calculations (distance, routing, and traffic), simplifying the implementation by reducing the need for complex in-house geospatial logic.

To address safety concerns, the platform will integrate features such as neighborhood safety ratings, which provide security indicators based on user feedback. Couriers will receive alerts when they accept deliveries in high-risk areas. In addition, nighttime safety measures will offer visibility into risk-prone areas for deliveries made late at night.

Architecture and Tech Stack

The system architecture will consist of the following components:

- **Application for Android**: implemented using JetPack Compose.
- **Web API**: Provides backend logic and data management for real-time interactions.
- **PostgreSQL Database**: Stores connections between entities along with additional order and user-related data.

This architecture provides a robust and scalable foundation for the system, ensuring seamless interaction between the Android application, web API, and database.

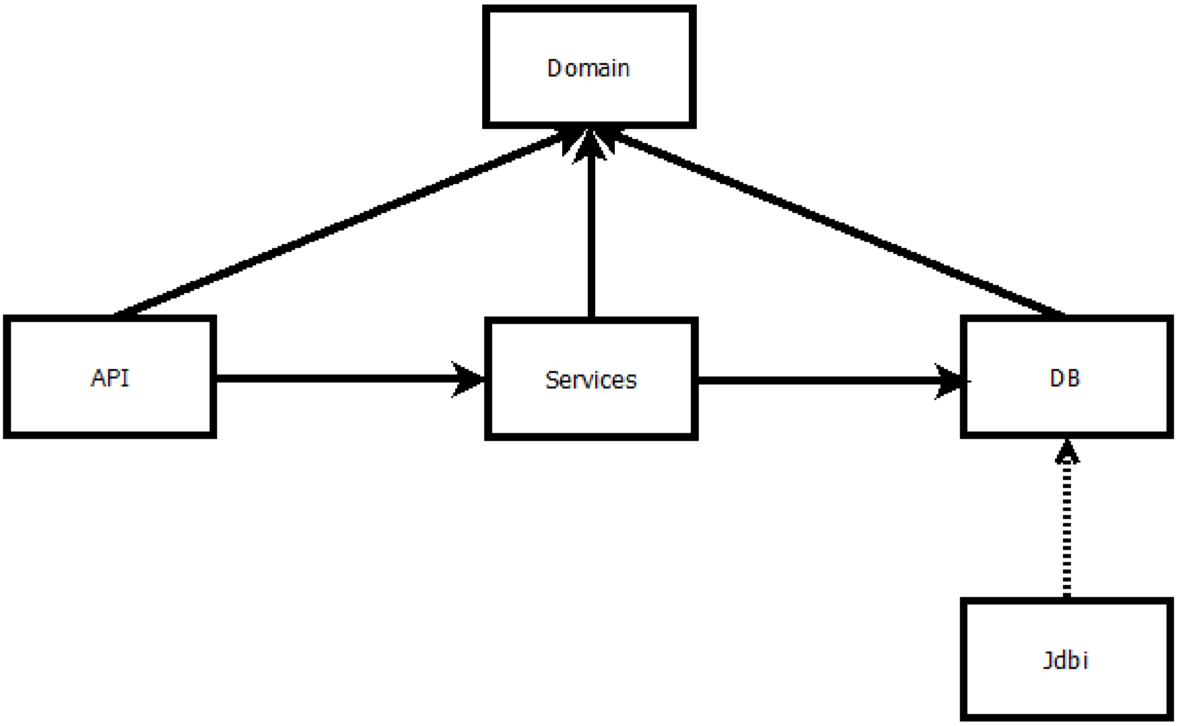


Figura 1: Module Diagram

Challenges Ahead

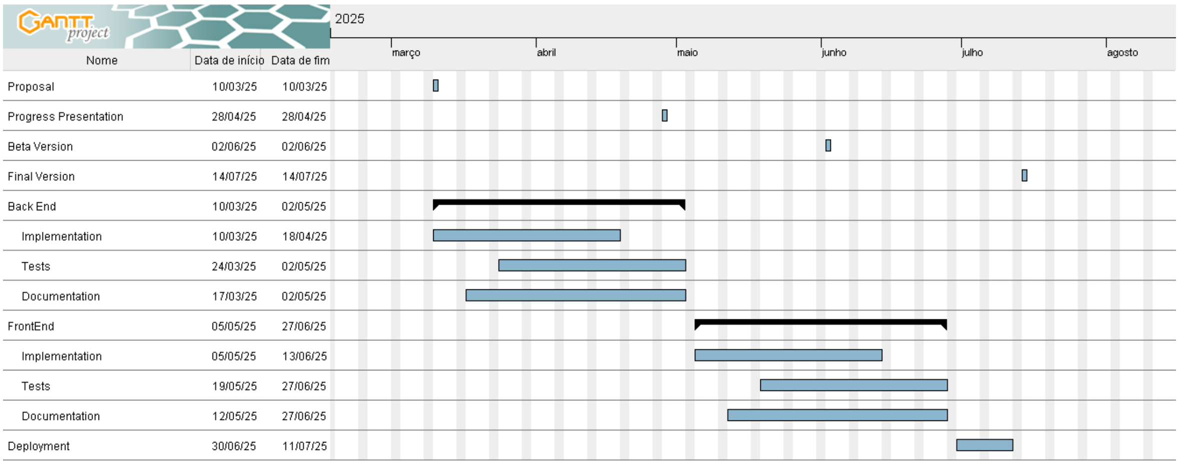
Although our project aims to improve the courier experience through innovative features, several challenges must be addressed during development and deployment.

- **Real-Time Data Synchronization**: Ensure consistent and reliable updates between couriers, the server, and the application without compromising performance or data integrity.
- **Scalability**: As the user base grows, maintain system performance and responsiveness while scaling up server resources and database capabilities.
- **User Interface Design**: Design an intuitive and efficient user interface using Jetpack Compose that meets the diverse needs of both couriers and administrators.

- **Integration Complexity**: Integrating third-party APIs for geospatial data and safety ratings while maintaining system stability and reliability.
- **Testing and Validation**: Rigorously testing the application using automatic tests.

Addressing these challenges proactively will be crucial to delivering a robust and effective mobile application that meets the demands of both couriers and users alike.

Project Plan



Nome	Data de início	Data de fim
Proposal	10/03/25	10/03/25
Progress Presentation	28/04/25	28/04/25
Beta Version	02/06/25	02/06/25
Final Version	14/07/25	14/07/25
Back End	10/03/25	02/05/25
Implementation	10/03/25	18/04/25
Tests	24/03/25	02/05/25
Documentation	17/03/25	02/05/25
FrontEnd	05/05/25	27/06/25
Implementation	05/05/25	13/06/25
Tests	19/05/25	27/06/25
Documentation	12/05/25	27/06/25
Deployment	30/06/25	11/07/25

Figura 2: Project Plan

Referências