



LiftDrop

Mobile App for Deliveries

Gonalo Morais, n.^o 49502, e-mail: a49502@alunos.isel.pt, tel.: 927468061
Joo 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

Maro 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 delivery-focused mobile application 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. While 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'll 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 accepting deliveries in high-risk areas. Additionally, nighttime safety measures will offer visibility into risk-prone areas for deliveries made late at night.

Architecture and Tools (TO DO)

The system architecture will consist of the following components:

- **Application for Android:** Implements the user interface and core functionality for couriers.
- **Web API:** Provides backend logic and data management for real-time interactions.

- **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.

Challenges Ahead (TO DO)

Project Plan

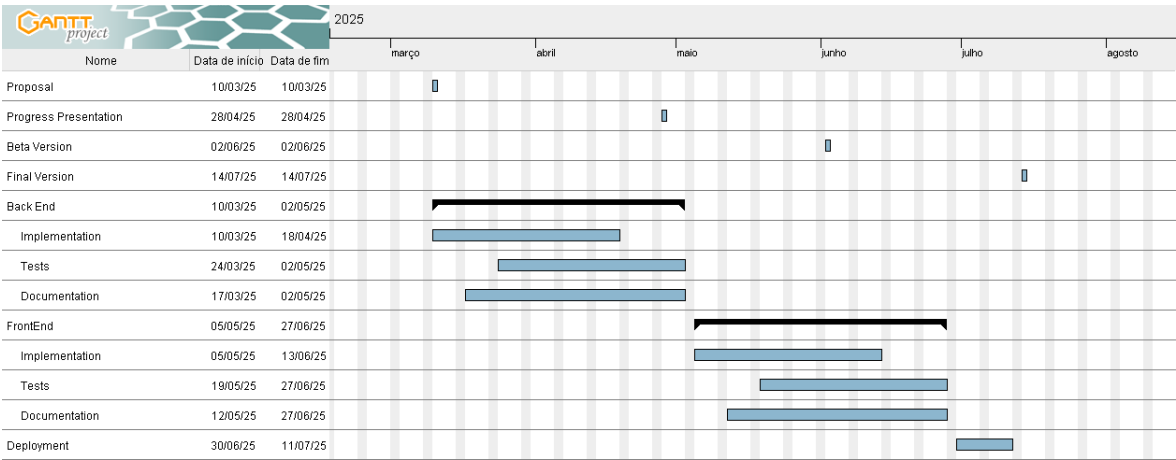


Figura 1: Project Plan

Referências