



WaveCoach

Training Management Platform for Surf

Tiago Canilhas, n.º50472, e-mail: a50472@alunos.isel.pt, tel.: 924115540
João Barrisco, n.º50476, e-mail: a50476@alunos.isel.pt, tel.: 967487235

Supervisor: Filipe Freitas, e-mail: ffreitas@cc.isel.ipl.pt

March 9, 2025

1 Introduction

The goal of this project is to develop a web application that allows the management of surfing athletes. This application is mainly aimed at coaches, but athletes can also access it to view their performance. Coaches will be able to register athletes, log different training sessions (e.g. gym, surf), record competition results, and monitor performance through interactive charts. This will help them analyze progress and adjust training plans. This project is being developed to meet coaches' need for a more efficient and effective platform, replacing manual registration.

2 System Requirements

2.1 Functional Requirements

- **User registration:** Coaches should be able to register by providing a username and password;

- **Athlete management:** Coaches should be able to add and remove their athletes. Adding an athlete will only require their name and date of birth. A code will be generated to share with the athlete to access the web application;
- **Athlete profile:** Coaches should be able to add and view information related to their athletes, such as weight, height, body mass index, and more. Athletes can also view this same information;
- **Athletes' training records:** Coaches, through interactive tables and charts, should be able to record and track each athlete's training sessions. Each training session contains different information:
 - **Surf training sessions:** Sea conditions, maneuvers performed, and a training-related questionnaire completed by the athlete;
 - **Gym training sessions:** Type of exercise, repetitions, and weight.
- **Athletes' summaries:** The application should allow, weekly or monthly, summaries so that coaches can analyze their athletes' progress;
- **Athletes' competition records:** Coaches can store data from their athletes' competitions, such as the date, location, and score achieved.

2.2 Non-Functional Requirements

- **User-friendly:** The application should be easy to understand, and therefore, it should have a simple and intuitive user interface;
- **Cloud Deployment:** The application will be hosted on a cloud platform to ensure easy access. This will allow users to access the platform from anywhere without needing local installations.
- **Tested:** The application must include tests to ensure that all functionalities work correctly.

2.3 Optional Features

- **PDF Generation:** The application should be able to generate a PDF file with the athlete's summaries, so the coach can interact with the summaries more easily;
- **Planning of future training sessions:** In addition to accessing past training sessions, the coach will be able to schedule new training sessions by providing the date and type of training. Athletes will be able to access this information;
- **Email notifications:** The application will notify athletes by email if any training session is scheduled or changed. For this, athletes will need to associate their email address.

3 Technologies

In the development of this project different technologies will be employed, such as the Kotlin programming language [1] and the Spring framework [2], which will be utilized for the creation of the API. PostgreSQL [3] will also be used for data storage. Finally, TypeScript [4] will be applied, together with the React library [5], for the creation of the website

4 Risks

- Having no knowledge of surfing practice may lead to difficulties in creating a practical design for coaches, as we don't know the best way to present information to them. Consequently, frequent communication with the practitioners will be essential.
- In frontend development, we may encounter more challenges because, although these technologies aren't new to us, we have less experience with TypeScript and React.

5 Plan

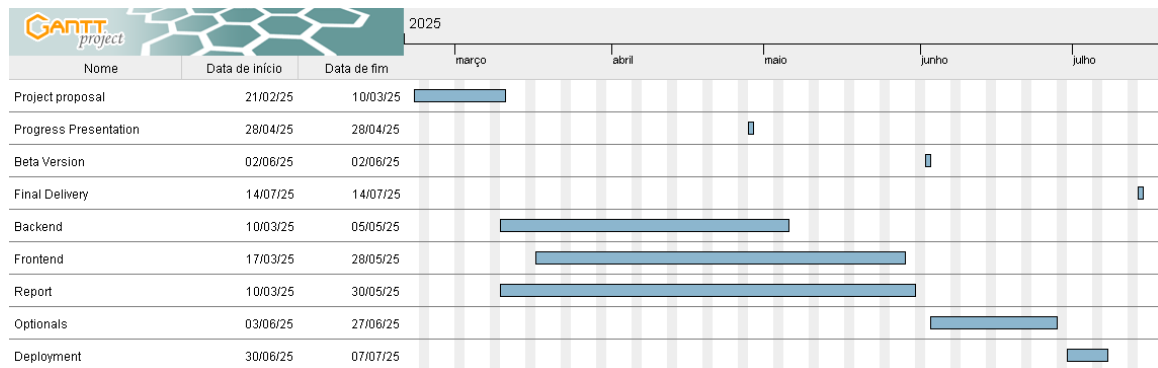


Figure 1: Gantt Chart plan.

References

- [1] Kotlin. <https://kotlinlang.org/>. [Accessed 23-February-2025].
- [2] Spring framework. <https://spring.io/>. [Accessed 23-February-2025].
- [3] PostgreSQL. <https://www.postgresql.org/>. [Accessed 23-February-2025].
- [4] Typescript. <https://www.typescriptlang.org/>. [Accessed 23-February-2025].
- [5] React. <https://react.dev/>. [Accessed 23-February-2025].