

# Relatório Técnico

## Projeto *SoccerNow*

Grupo 25  
Martim Neto - 56279  
Francisco Carvalho - 52838

5 de junho de 2025

## 1 Descrição da Arquitectura do Projeto

O projeto *SoccerNow* é composto por:

- Um **backend** desenvolvido em *Java com Spring Boot*, responsável por fornecer uma API REST para a gestão de dados relacionados com jogadores, equipas, jogos e competições.
- Uma **interface gráfica** construída com *JavaFX*, que permite aos utilizadores interagir com os dados de forma intuitiva e local.
- Uma **interface Web**, desenvolvida com *Thymeleaf*, que permite visualizar e gerir as mesmas entidades através de um navegador.

A arquitetura segue uma abordagem em camadas com separação clara entre apresentação, lógica de negócio e persistência de dados.

## 2 Decisões Técnicas na Arquitectura da Aplicação

O backend recorre a tecnologias como Spring Boot, Spring Data JPA e PostgreSQL. As principais decisões técnicas incluem:

- Estrutura em **camadas** com os pacotes `controller`, `services`, `repository`, `model` e `dto`.
- Uso de **Spring Data JPA** para abstrair o acesso à base de dados.
- Implementação de tratamento de erros com `@ControllerAdvice` e exceções personalizadas.
- Utilização de DTOs para proteger a estrutura interna dos dados e simplificar a comunicação com a interface.

## Padrões de design aplicados:

- MVC (Model-View-Controller)
- DAO (Data Access Object)
- DTO (Data Transfer Object)
- Injeção de Dependências via `@Autowired`
- Singleton nos serviços
- Tratamento global de exceções com `@ControllerAdvice`

## 3 Decisões Técnicas no Desenho da Interface Web

A interface Web foi desenvolvida com **Thymeleaf**, tirando partido da sua integração com Spring Boot.

- Utilização de um template base (`layout.html`) com `th:fragment` para garantir consistência visual entre páginas.
- Páginas individuais para entidades específicas: `player.html`, `team.html`, `referee.html`, entre outras.
- Estilo personalizado via CSS na pasta `static/css`.
- Separação clara entre lógica de apresentação (HTML) e lógica de controlo (Controladores Spring).

## 4 Decisões Técnicas no Desenho da Interface JavaFX

A interface JavaFX foi desenhada para ser uma aplicação desktop eficiente e intuitiva.

- Uso de **FXML** para separar a interface gráfica da lógica de controlo.
- Controladores dedicados para cada vista, promovendo organização e manutenção.
- Navegação entre cenas feita de forma modular e dinâmica.
- Filtros foram feitos automaticamente pelo `tableview` no frontend.