

# Arquitetura do projeto

## Backend

A arquitetura implementada no nosso projeto segue uma estrutura em camadas. Esta arquitetura foi organizada da seguinte forma:

### Camada de Aplicação (Controller)

Esta camada é responsável por lidar com os pedidos HTTP. Contém os controladores da aplicação, que expõem os endpoints REST e fazem a ponte entre o cliente e as restantes camadas do sistema.

### Camada de Lógica de Negócio (Service)

Aqui encontram-se os handlers dos casos de uso da aplicação. Esta camada contém a lógica central do negócio. Ao manter esta lógica separada, garantimos que a aplicação é mais fácil de testar e manter.

### Camada de Persistência (Repository)

Esta camada é responsável pelo acesso aos dados. Utilizamos interfaces com a anotação `@Repository`, que permite realizar queries à base de dados. Graças ao Spring Data JPA, muitos destes repositórios foram definidos de forma declarativa.

### Camada de Transferência de Dados (DTO)

Os DTOs (Data Transfer Objects) são usados para transportar dados entre a camada de apresentação e as outras camadas da aplicação, evitando expor diretamente as entidades do domínio. Esta camada ajuda a encapsular e estruturar a informação de forma mais segura e controlada.

### Camada de Domínio (Model)

Aqui encontram-se todas as entidades do sistema, marcadas com a anotação `@Entity`. Estas classes representam os objetos principais da aplicação que serão persistidos na base de dados. Esta camada define a estrutura e relações dos dados.

## Frontend

### Interfaces JavaFX e Thymeleaf (Web)

Na segunda fase do projeto foram implementadas duas interfaces que permitem interagir com a aplicação: uma usando Thymeleaf e outra JavaFX. A maioria da lógica de negócio está no backend, portanto nestas interfaces apenas limitamo-nos a utilizar as suas ferramentas, com exceção de algumas verificações simples que podem ser feitas no frontend para facilitar o tempo de resposta.

A aplicação JavaFX permite realizar todos os casos de uso que foram requisitados:

- Realização de login com autenticação através do email (ao iniciar a aplicação é colocado um utilizador com as seguintes credenciais para efetuar o primeiro acesso à aplicação : nome: "Admin"; NIF: "123456789"; Email:

“admin@fc.ul.pt”). (A)

- Registo de utilizadores (jogadores e árbitros) apenas com os campos de Nome, Nif e Email, sendo os dois últimos únicos na base de dados. Os jogadores são criados sem posição preferida (atributo PrefPos) e os árbitros são sempre criados sem certificação (atributo certificado). Estes atributos devem ser alterados posteriormente na parte de “Verificar, remover e atualizar um utilizador”. Acreditamos que esta seja a melhor implementação porque facilita e simplifica a criação destas entidades e torna mais realista a aplicação. A verificação/atualização de utilizadores é feita com uma procura pelo NIF ou pelo Email do utilizador de forma a encontrar apenas uma entidade na base de dados. Na interface vão encontrar dois campos que permitem fazer a procura, mas decidimos dar prioridade ao Email, ou seja, se introduzir um email do user1 e o nif do user2, o utilizador devolvido é o user1. (B,C)
- A criação de equipas apenas necessita de um nome. Assim, os restantes atributos, que não sejam estatísticas, devem ser posteriormente adicionados, tal como os jogadores do plantel. Também é possível verificar, remover e atualizar uma equipa. (E, F)
- Em relação aos jogos também é possível marcar um jogo amigável ou de campeonato (dependendo do atributo campeonato). Alteramos a forma como é criado um jogo de forma a simplificar a sua utilização, sendo que agora não possui um atributo com horas e minutos, mas sim um atributo TURNO que pode ter três valores dependendo das horas definidas por nós: MANHA (8:00), TARDE (12:00) e NOITE (18:00). É possível registar um resultado de jogo e cancelar um jogo de campeonato que ainda não se tenha realizado. (H, I, M)
- É possível criar, verificar, remover e atualizar um campeonato. Apenas permitimos a criação de campeonato por começar, ou seja, não é possível criar campeonatos com equipas que possuem pontos e jogos realizados. (J,L)

A aplicação Web também também permite realizar todos os casos de uso pedidos:

- Tal como no JavaFX, é possível realizar a autenticação através do email indicado acima. (A)
- Também é possível realizar a busca por Jogadores, Equipas e Campeonatos (todas as entidades na base de dados das respectivas tabelas). (D,G,K)
- É possível, tal como no JavaFX, registar o resultado de um jogo que já se tenha realizado. As estatísticas são automaticamente adicionadas tanto aos jogadores, como aos árbitros e às equipas. (I)
- Todos os filtros dos casos de uso N, O, P, Q e R estão todos implementados e são intuitivamente fáceis de utilizar.
- Todos os filtros sugeridos no enunciado estão implantados e ainda mais alguns diferentes ou com nuances mais específicas, como procurar jogadores com mais ou menos de X golos marcados ou cartões recebidos.

## Testes

Foram implementados testes automáticos em classes que pretendem testar as funções do seu respetivo serviço e que são utilizadas pelos controllers. Decidimos fazer uma injeção dos serviços usando `@AutoWired`, tal como fazemos no controller. Contudo, anotamos a classe com `@SpringBootTest` e `@Transactional` o que permite dar rollback das alterações que fizemos na base de dados durante os testes.

## Campeonato

O Campeonato possui o campo Modalidade que mais tarde permitirá definir o tipo de campeonato. Para esta fase e por motivos de simplificação deixamos desativado, mas funcionaria no futuro.

Os jogos dos campeonatos são guardados logo na lista de jogos do campeonato. Após ser criado um resultado de um jogo, dependendo do vencedor, é atualizada a classificação no seu respetivo campeonato (3 pontos para o vencedor e 1 para cada uma das equipas no jogo). Definimos algumas regras como: só é possível a adicionar equipas ao campeonato se o campeonato estiver no estado `POR_COMEÇAR`; apenas é possível criar jogos para um campeonato no estado `EM_CURSO`; ao terminal o campeonato por um endpoint, são calculados os vencedores e atualizadas as listas das conquistas de cada equipa.