

UNIVERSIDADE DE COIMBRA

SISTEMAS DISTRIBUÍDOS

LEI - DEI - 2021/2022

---

## Relatório do trabalho 2

---



Joel Oliveira - 2019227468  
Tomás Mendes - 2019232272

1 de junho de 2022

# 1 Introdução

O projeto tinha como principal objetivo, o desenvolvimento de uma plataforma para de consulta de resultados desportivos em tempo real, scoreDEI. A plataforma está feita num sistema *web* em Java com uma camada de dados e *frontend* usando Spring Boot e Thymeleaf. A camada de dados é implementada com auxílio de JPA e uma base de dados PostgreSQL. É ainda utilizada uma api externa (api-sport.io) para que a plataforma use dados reais.

## 2 Arquitetura

### 2.1 C4

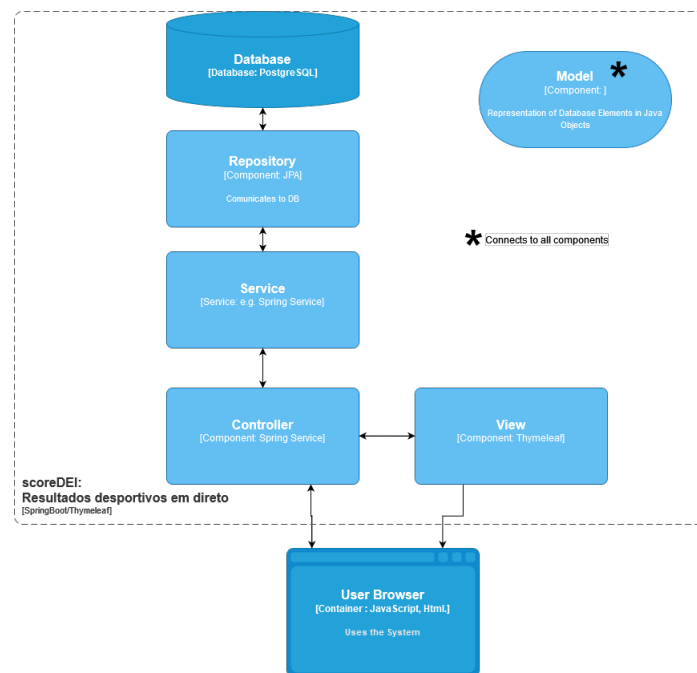


Fig. 1: Arquitetura da aplicação

A arquitetura implementada segue um esquema de arquitetura por camadas, como é visível na figura 1. O utilizador interage com a camada de visualização, definida pelo controlador e pelas vistas. O controlador comunica com a camada de serviços que, posteriormente, comunica com a camada de repositório que por sua vez comunica com a base de dados.

## 2.2 Entidade Relacionamento

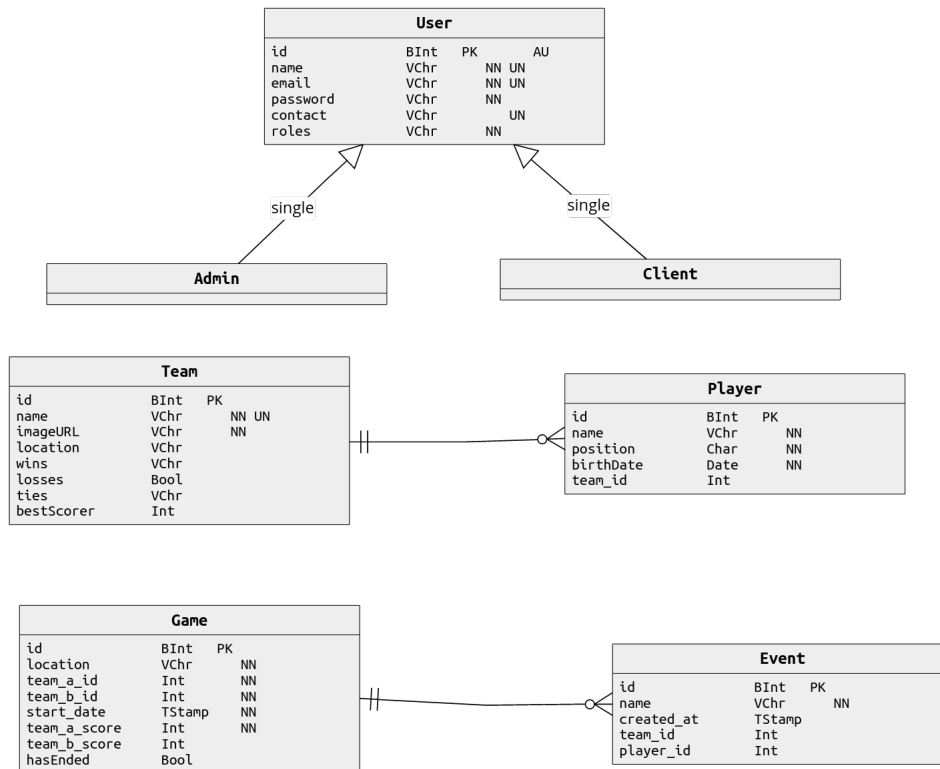


Fig. 2: Diagram de entidade relacionamento

## 3 Implementação

O *backend* segue a estrutura do Spring, em que os controladores usam métodos dos vários serviços, que por sua vez usam os repositórios, associados aos vários modelos, para interagir com a base de dados. Como tal, e de forma a organizar o código, foi feita uma divisão entre jogos, equipas, jogadores, eventos e autenticação. Cada uma destas divisões tem um controlador, com os *endpoints* relacionados.

A relação entre jogadores e equipas é de 1 para 1, i.e, 1 jogador apenas está associado a uma equipa. A relação equipa jogador é *OneToMany*, i.e, uma equipa tem vários jogadores, um *Set* neste caso, e é usado o argumento *mappedBy* para ser criada uma tabela adicional com o id da equipa e do jogador.

A autenticação dos utilizadores é feita através do Spring Security. A criação de utilizadores, por outro lado, foi implementada seguindo os métodos do Spring e Thymeleaf. O Spring Security converte a classe User para UserDetails, para fazer a autenticação. Este módulo permite também limitar os acessos a funcionalidades com base nas permissões de cada utilizador. Os utilizadores anónimos, que não têm conta, apenas podem visualizar o site. Não conseguem criar equipas, jogadores nem jogos e também não podem registar eventos nos jogos. Um utilizador comum, pode fazer tudo o que um anónimo consegue, e consegue registar eventos em jogos a decorrer. Um administrador, por contraste, tem permissões para fazer tudo. A primeira vez que a aplicação é corrida, a base de dados está vazia. De modo a poder criar equipas, jogos, etc, existe um endpoint (/fillAdmin) para adicionar um administrador.

Para conseguir testar as funcionalidades, existe um botão Fill na barra lateral que coloca algumas equipas, jogadores e jogos na base de dados. O *Custom* utiliza dados *hard coded* quando carregado, enquanto o *Sports API* utiliza dados reais fornecidos pela API: <https://api-sports.io/>

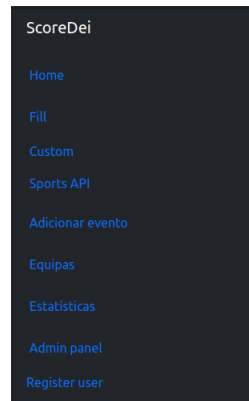


Fig. 3: Barra lateral

O *frontend* utiliza o Thymeleaf para fazer a integração com o *backend* e permitir páginas html dinâmicas. De forma a tornar o site visualmente apelativo, foi utilizado o *Bootstrap 5*, não sendo necessário CSS adicional.

Todas as vistas usam um *layout*[1] para que a barra lateral seja visível em todas e facilmente alterada.

A única vista que atualiza informação de forma automática é a *create-event.html*. Esta vista tem *forms* dinâmicos, com base nas escolhas dos anteriores. Após selecionar um jogo e o tipo de evento, é possível escolher a equipa e o jogador associado ao evento, se necessário. Os eventos gerais a um jogo, tais como: início, pausa, fim, etc, não precisam de ter um jogador nem uma equipa associados, mas o caso de golos e cartões necessita. Os *forms* atualizam de forma a restringir as seleções às condicionantes anteriores. Por exemplo, se um jogador marca um golo, efetuar a sua seleção é mais fácil se for feita uma subseleção prévia. Sabemos em que jogo foi marcado, e a que equipa ele pertence. Assim faz-se a seleção apenas a partir dos jogadores que servem estas restrições (4), e não a partir de todos (5).

Para isto, ao ser carregada a página, são guardadas nas opções atributos adicionais do tipo "data-\*", que permitem criar uma ligação entre jogo-equipas e equipa-jogador. São também incluídos *scripts* de *javascript*, ligados ao evento de atualizar o valor de uma *dropbox*. Quando isto acontece, os valores "dependentes" dessa *dropbox* são atualizados.

Para as estatísticas, foi criada uma vista com uma tabela dinâmica, que permite ordenar os valores, de forma crescente ou decrescente, de qualquer coluna. Assim não existe quase processamento nenhum do lado do *backend* para mostrar e ordenar as várias estatísticas.

## 4 Testes

Para garantir a integridade da plataforma, foram feitos alguns testes funcionais.

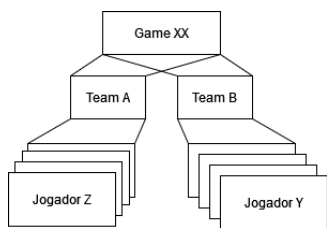


Fig. 4

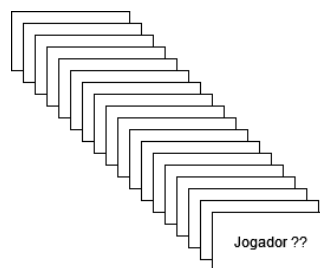


Fig. 5

Categoria	Validação	Resultado
Login	Fazer login na plataforma	✓
Registrar	Apenas administradores conseguem criar contas	✓
Criar jogo	Administradores conseguem criar jogos	✓
Criar jogo	Escolher as equipas, data e local do jogo	✓
Criar equipa	Administradores conseguem criar equipas	✓
Criar equipa	Escolher nome e imagem	✓
Criar jogador	Administradores conseguem criar jogadores	✓
Criar jogador	Escolher nome, posição, data de nascimento e equipa	✓
Criar eventos	Administradores e utilizadores autenticados conseguem criar eventos	✓
Lista de equipas	Qualquer utilizador consegue ver todas as equipas	✓
Lista de equipas	Qualquer utilizador consegue ver o perfil de uma equipa	✓
Estatísticas	Qualquer utilizador consegue ver as estatísticas	✓
Jogo	Jogo que já acabaram aparecem numa listagem separada	✓
Jogo	Não é possível criar eventos para jogos que já terminaram	✓
Jogo	Os jogos que estão a decorrer aparecem na lista	✓
Evento	Evento "Fim" termina um jogo	✓
Evento	Evento golo permite escolher uma equipa e jogador dessa equipa	✓
Evento	É possível ordenar os eventos pelo <i>timestamp</i>	✓
Evento	Evento golo atualiza as estatísticas do jogador	✓
Sports-API	Administradores conseguem ver o botão	✓
Sports-API	Administradores conseguem preencher a <i>db</i> com dados da API	✓

Tab. 1: Testes Funcionais

## Referências

- [1] Bootstrap 5 Sidebar Examples  
<https://dev.to/codeply/bootstrap-5-sidebar-examples-38pb>  
 acedido a 18/5/2022
- [2] Login form  
<https://mdbootstrap.com/docs/standard/extended/login/#section-1>  
 acedido a 18/5/2022