

SCOREDEI: RESULTADOS DESPORTIVOS EM DIRETO

ADRIANA GOMES DA SILVA LEOCÁDIO BERNARDO - 2019218086
PEDRO DUARTE SANTOS HENRIQUES - 2019217793



scoreDEI: Resultados desportivos em direto

Índice

Introdução	2
Arquitetura detalhada da plataforma e a sua descrição	3
Servidor de Base de Dados	3
Plataforma ScoresDEI (Web)	3
Comunicação com Clientes (HTTP)	3
Comunicação com Admins (HTTP)	3
Controladores	3
Detalhes sobre a implementação do backend da plataforma	5
Detalhes de implementações extra	5
Descrição dos testes feitos à plataforma	6
Diagrama Entidade Relacionamento	6
Conclusão	7

Introdução

O objetivo deste trabalho tem por base a criação de uma plataforma intitulada de “scoreDEI”. Esta plataforma serve para a consulta de resultados desportivos em tempo real e onde os utilizadores podem consultar e receber atualizações de resultados desportivos de jogos que se encontrem a decorrer.

O objetivo principal deste projeto tem por base a programação de um sistema Web onde deverá existir uma camada de dados e Frontend Spring Boot e Thymeleaf. E, ainda a programação de uma Base de dados através da Java Persistence Application Programming Interface (JPA).

Com esta necessidade em mente, concetualizámos e implementámos a plataforma scoreDEI. Esta plataforma que desenvolvemos permite realizar todas as operações para que os seus utilizadores possam interagir e recolher todas as informações e detalhes de um jogo desportivo em tempo real. Nesta plataforma são possíveis operações como: criar um jogo, visualizar detalhes de um jogo, criar um evento, etc. Assim como ter acesso a estatísticas das várias equipas e jogadores.

A parte final deste projeto, tem como objetivo a integração do projeto com uma API (api-sports.io) utilizando a tecnologia REST para a obtenção dos dados.

Arquitetura detalhada da plataforma e a sua descrição

A arquitetura que nós utilizámos está apresentada na Figura 1, onde mostra que a plataforma scoreDEI é composta por várias componentes: um servidor de base de dados, a própria plataforma web - ScoresDEI, comunicações com clientes e comunicações com admins.

Servidor de Base de Dados:

Este servidor de base de dados tem um papel crucial no que toca à salvaguarda de todos os dados, pois trata-se do repositório de dados da plataforma. É neste servidor que se encontram guardados todos os dados relativos a dados de utilizador, equipas, jogos, etc.

Plataforma ScoresDEI (Web):

A interface web da plataforma scoreDEI está dividida em dois grupos distintos, as vistas e funcionalidades a que todos os utilizadores têm acesso, e as vistas e funcionalidades a que somente os utilizadores com permissões de administração têm acesso. As vistas e funcionalidades devem estar acessíveis através de um menu que facilite a navegação. Opções de navegação às quais um determinado utilizador não tem acesso não devem estar visíveis no menu de navegação.

Comunicação com Clientes (HTTP):

Esta comunicação entre o cliente e a plataforma web é efectuada através do protocolo HTTP. A parte de backend consiste num servidor Web que responde aos diferentes tipos de pedidos da parte do frontend através de endpoints. É assim, através deste processo que os clientes se comunicam com o servidor de base de dados.

Comunicação com Admins (HTTP):

Esta comunicação entre o cliente e a plataforma web é efectuada através do protocolo HTTP. A comunicação entre a plataforma web e os admins é essencial para a gestão administrativa da informação que é apresentada, assim como a gestão dos próprios utilizadores da plataforma.

Controladores:

- **DefaultController:** Controlador que gere os Endpoints globais, isto é, a página inicial (“/”), a opção de criar dados (“/createData”) e a opção de estatísticas (“/stats”);

- **UsersController:** Este controlador efetua todas as operações relacionadas com a gestão de utilizadores, sejam estas de login (*/login*) ou registo (*/register*);
- **TeamsController:** Gere todos os Endpoints relacionados com dados de equipas, sejam operações de criar (*/createTeam*) ou editar equipas (*/editTeam*), sejam operações de listar todas as equipas (*/listTeams*) ou apresentar os detalhes de cada uma (*/teamDetails*);
- **GameController:** Controlador que efetua gestão de Endpoints relacionados com os jogos, seja de listar todos os jogos (*/listGames*), apresentar os detalhes de cada um (*/gameDetails*) ou de criar (*/createGame*) ou, ainda, editar jogos existentes (*/editGame*);
- **PlayerController:** Controlador que efetua gestão de Endpoints relacionados com os jogadores, seja de listar todos os jogadores (*/listPlayers*) ou de criar (*/createPlayers*) ou, ainda, editar jogadores existentes (*/editPlayers*);
- **GameEventController:** Controlador que efetua gestão de Endpoints relacionados com os eventos de um jogo, seja de criar um evento (*/createEvent*), editar detalhes de um evento existente (*/editEvent*) ou, ainda, adicionar detalhes a um evento existente (*/processEvent*).

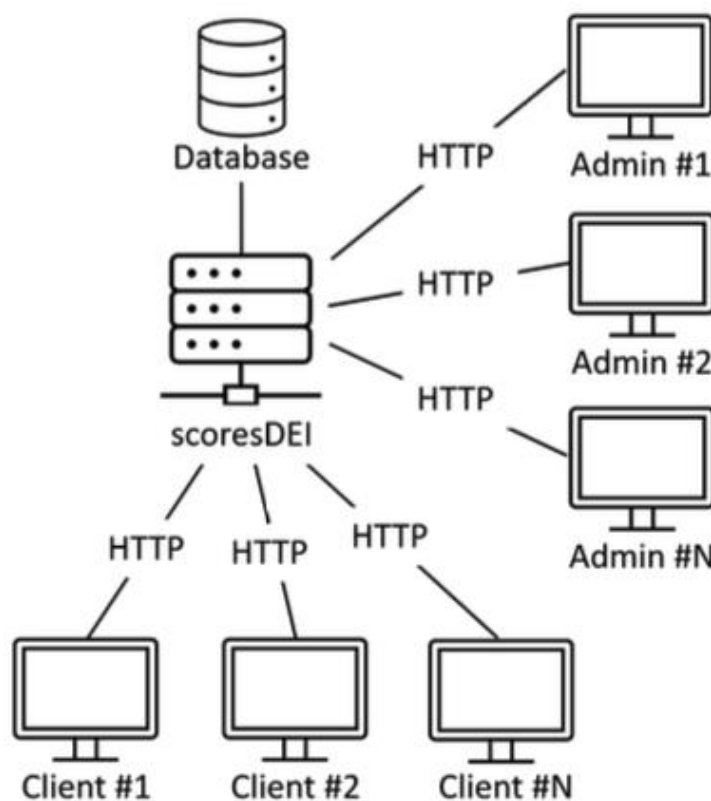


Figura 1: Arquitetura de alto nível da plataforma ScoreDEI

Detalhes sobre a implementação do backend da plataforma

A implementação backend desta plataforma - scoreDEI funciona da seguinte forma:

Da parte das funcionalidades de administradores, ou seja, utilizadores com permissões de administração, podem, Registrar utilizadores, Criar/Gerir Equipas, Criar/Gerir Jogadores e Criar/Gerir Jogos de Futebol, bem como editar Eventos de jogos existentes.

Da parte das funcionalidades de utilizadores registados, ou seja, utilizadores que tenham sido registados por um administrador, podem registar um novo evento num jogo.

Da parte das funcionalidades gerais, ou seja, funcionalidades que possam ser acedidas por qualquer utilizador, registado ou não, implementámos acesso a listagem de jogadores, equipas e jogos, bem como os seus detalhes.

Podem, ainda, consultar estatísticas:

- Estatística 1: Listagem ordenada das equipas com informação sobre jogos realizados: número de jogos, número de vitórias, número de derrotas, número de empates.
- Estatística 2: Nome do melhor marcador de golos.

Em termos de gestão de sessões na plataforma, tivemos algumas dificuldades em implementar um sistema, então, utilizamos uma dependência de segurança da framework Spring e a class *“WebSecurityConfigurerAdapter”*. Como nós implementámos um sistema de codificação de password para inserção em base de dados, definimos manualmente os Endpoints permitidos a cada tipo de utilizador (Administrador ou Utilizador).

Detalhes de implementações extra

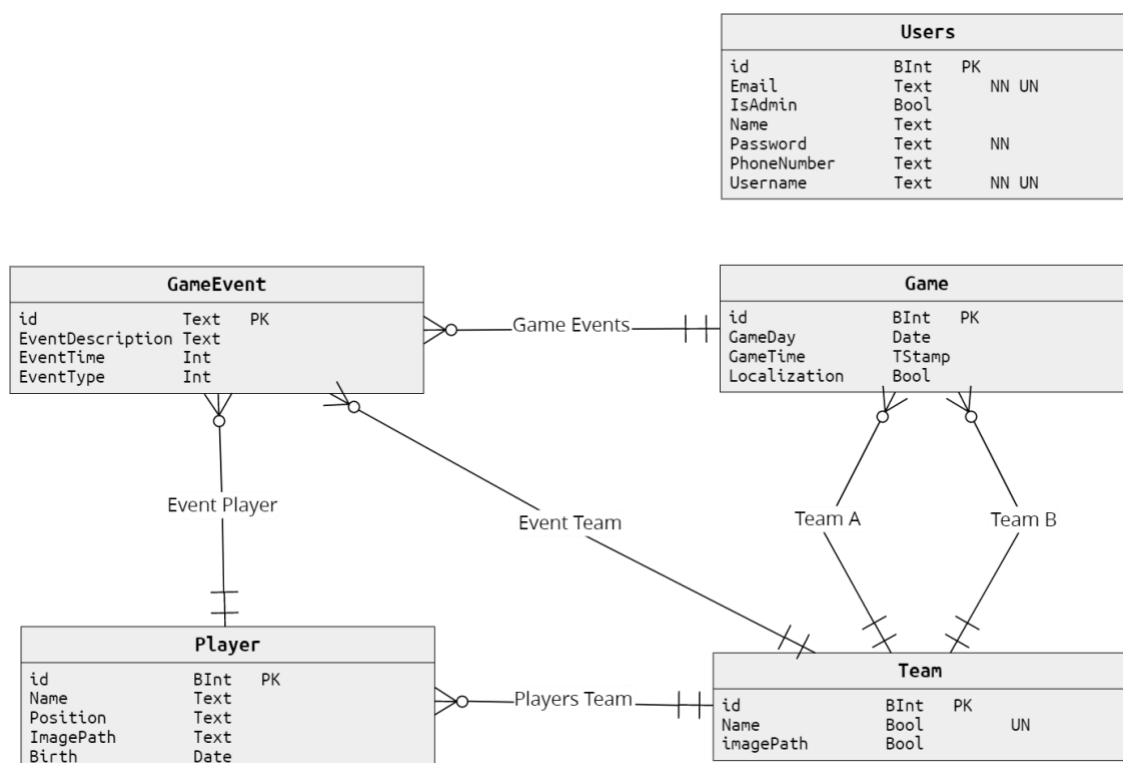
Para tornar a aplicação visualmente mais agradável, utilizámos a biblioteca bootstrap.

De forma a facilitar a inserção de dados na plataforma, utilizámos a Api Sports, utilizando dados da liga Portuguesa como base para a aplicação, de forma a expor todas as suas funcionalidades.

Descrição dos testes feitos à plataforma

Testes	Pass/Fail
Um admin quer registar um utilizador	Pass
Um admin quer criar uma equipa	Pass
Um admin quer alterar a posição de um jogador já existente	Pass
Um admin quer criar um jogo de futebol em Coimbra no dia 15/06/2022 às 14h00	Pass
Um utilizador registado quer criar um evento - o jogador X marcou um golo	Pass
Um utilizador não registado quer visualizar os detalhes de um jogo que está a decorrer	Pass
Um utilizador registado quer consultar o nome do melhor marcador de golos	Pass
Um utilizador não registado quer consultar o nome do melhor marcador de golos	Pass

Diagrama Entidade Relacionamento



Conclusão

Este trabalho foi realizado pelos dois elementos do grupo em conjunto na sua globalidade, para que pudéssemos completar os objetivos e ultrapassar os problemas mais eficazmente.

Concluindo, com este trabalho conseguimos aprender bastante não só sobre como desenvolver um sistema Web, com uma camada de dados e Frontend através do Spring Boot e Thymeleaf como também, a programação de uma base de dados através de JPA.

Gostámos muito de realizar este segundo trabalho desta cadeira, e consideramos que foi uma mais valia, pois conseguimos pôr em prática a matéria lecionada nas aulas teóricas e práticas. Apesar de termos tido algumas dificuldades ao longo do projeto, como na parte inicial, onde nos tivemos a ambientar melhor às linguagens de programação distintas, assim como na parte final, na integração com um sistema externo, no entanto, com esforço, conseguimos ultrapassar estas dificuldades, o que foi muito gratificante.