1 2 9 0 FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE D COIMBRA

Licenciatura em Engenharia Informática
Sistemas Distribuídos
2021/2022

scoreDEI 

Resultados desportivos em direto

Relatório

Professor orientador:
Professor Nuno Seixas

Realizado por:
Sofia Santos Neves - Nº2019220082
Tatiana Silva Almeida - Nº2019219581

1 de Junho de 2022

Índice

Introdução

Este projeto surgiu no âmbito da cadeira Sistemas Distribuídos, de Engenharia Informática, com o intuito dos utilizadores poderem consultar os resultados de jogos desportivos em tempo real.

O objetivo é ter um sistema Web com uma camada de dados Frontend utilizando o Spring Boot e Thymeleaf. Para a base de dados, utilizamos Java Persistence Application Programing Interface (JPA) e, por fim, fizemos a integração com uma API externa - api-sports.io - utilizando a tecnologia REST para a obtenção de dados para a aplicação. Todas estas tecnologias permitiram com que o scoreDEI se tenha tornado uma realidade.

Arquitetura

De uma maneira geral, foi nos pedido que existisse uma plataforma que tivesse associada uma Base de dados onde se pudesse guardar todos os dados dos utilizadores, jogos, equipas, jogadores e eventos como demonstrado no ER - Figura 1. Foi considerado que um utilizador (User_) não tem associações entre as demais tabelas. Para além disso, um jogador (Player) tem apenas uma equipa e pode ter zero ou mais eventos associados a ele. Já uma equipa (Team), tem vários jogadores, e pode ter vários jogos associados e eventos. Cada jogo (Football_Game) tem duas equipas associadas e zero ou mais eventos. Por fim, um evento (Event) que seja a marcação de um golo, cartão amarelo ou cartão vermelho tem um jogador, equipa e jogo associados, enquanto que um evento de começar jogo, terminar jogo, resumir jogo ou terminar jogo apenas tem um jogo associado.

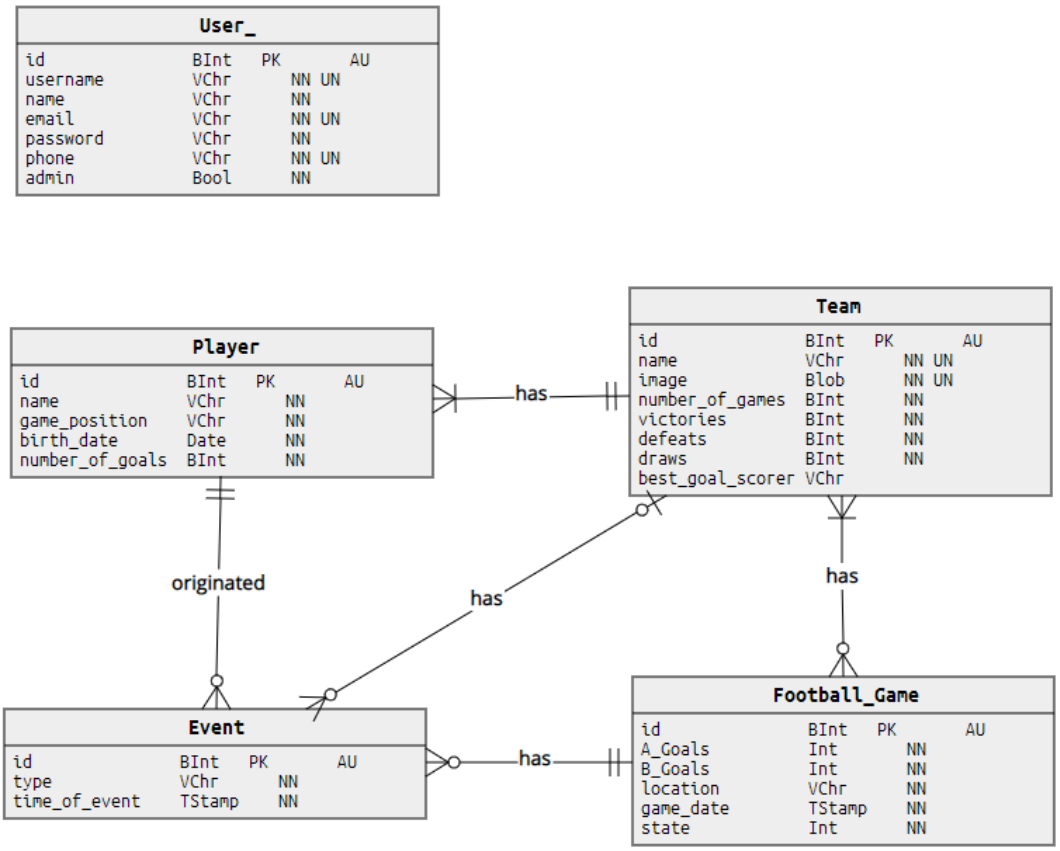

Figura 1 *- Modelo ER da base de dados*

Arquitetura do software

Encontra-se abaixo a arquitetura pela qual nos guiamos.

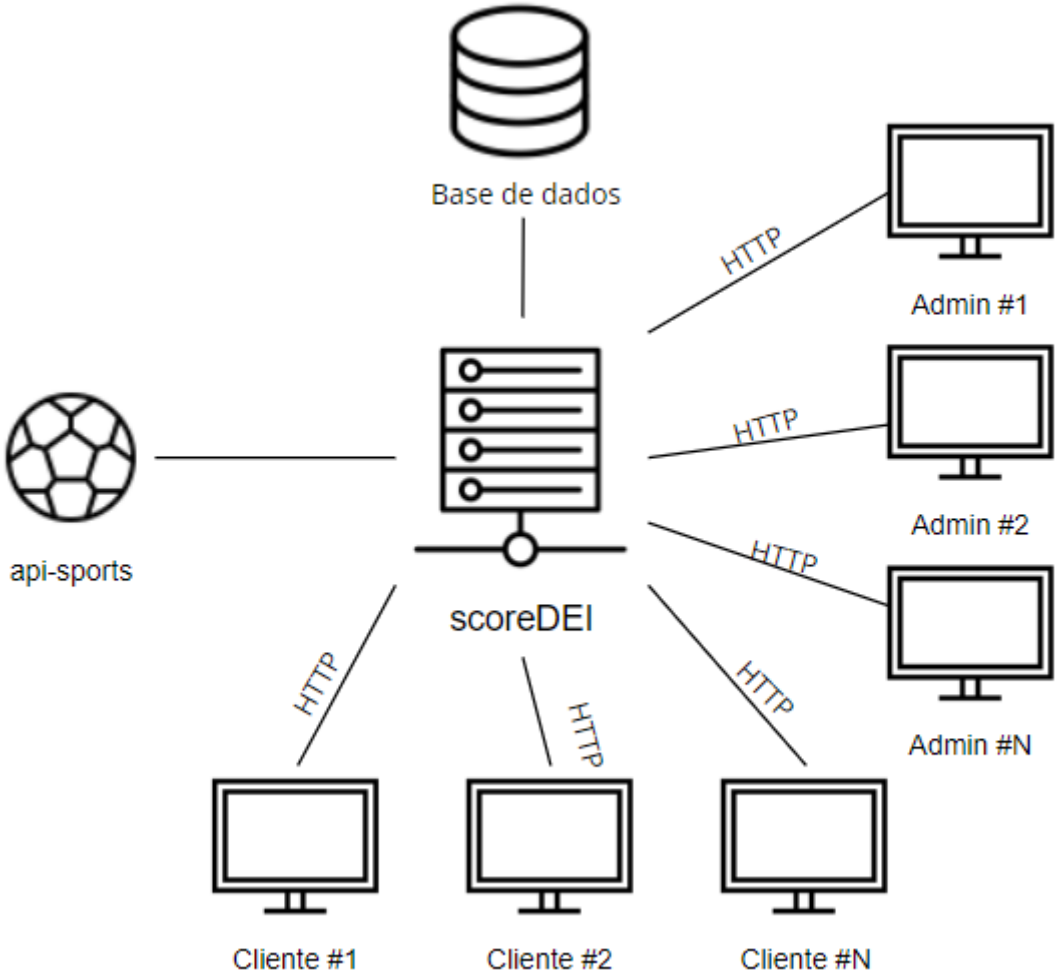

Figura 2 - *Arquitetura de alto nível*

Tal como apresentado no enunciado, seguimos a arquitetura recomendada e achamos também importante incluir a API como uma ligação relevante, uma vez que é lá que vamos buscar os dados para popular a base de dados.

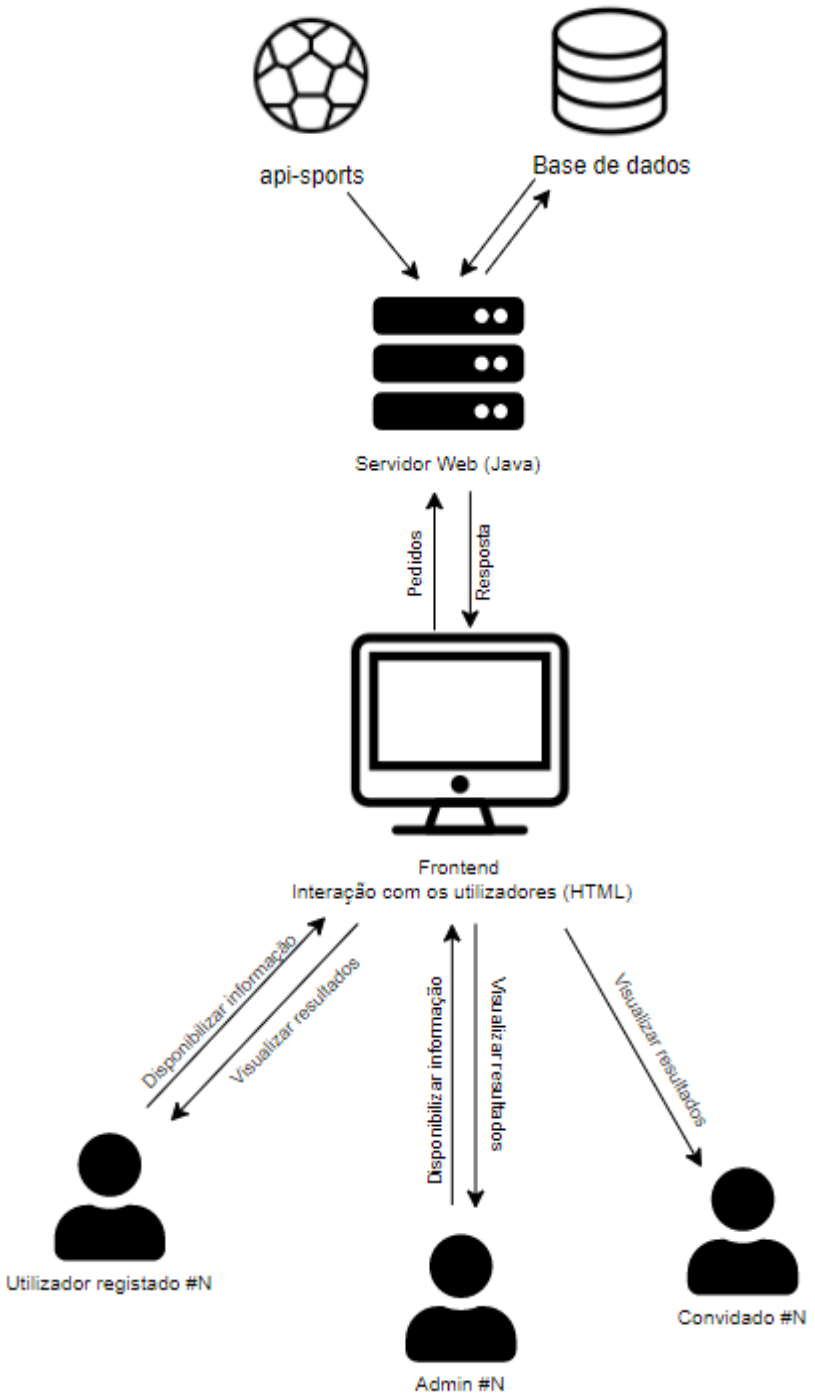

Figura 3 *- Arquitetura*

Numa tentativa de explorar um pouco mais a arquitetura da figura 2, surgiu a arquitetura da figura 3. Nesta última conseguimos perceber a presença de três tipos de utilizadores:

1. **Utilizador registado**: Utilizador que efetua login mas não é administrador
2. **Admin:** Utilizador que efetua login e é administrador
3. **Convidado:** Utilizador que não efetua login

Todos os utilizadores têm a possibilidade de visualizar os resultados mas apenas os que efetuam login é que disponibilizam informação, sendo superior a informação disponibilizada pelo

administrador. Na **tabela 1,** apresentada mais abaixo, encontram-se as permissões de cada um dos tipos de utilizadores.

Os utilizadores interagem com o frontend feito por nós em HTML que, consecutivamente, interage com o backend enviando pedidos e recebendo respostas.

O nosso backend é constituído por três componentes:

1. **api-sports:** API online onde são feitos pedidos pelo servidor web;
2. **Servidor web:** Faz pedidos de informação à API referente a jogadores e equipas. Coloca essa informação na base de dados e comunica com o frontend dando-lhe respostas aos seus pedidos;
3. **Base de dados:** Guarda a informação da API enviada pelo servidor web, e disponibiliza essa mesma informação ou guarda nova informação dependendo das necessidades dos utilizadores.

1. Controladores/Servlet

A nossa organização em termos de controladores passou por uma divisão tendo em conta os diferentes tipos de permissões.

1. **AdminController:** controlador onde se encontram as funcionalidades que dizem respeito unicamente ao admin;
2. **DataController:** controlador onde se encontram as funcionalidades comuns a todo o tipo de utilizadores e também os métodos que dizem respeito à recolha de informação da API e respectivo armazenamento na base de dados;
3. **UserController:** controlador onde se encontram as funcionalidades que podem ser feitas por um user registado que não seja administrador.

2. Vistas

Para desenvolvermos o projeto houve uma série de passos que seguimos antes de começarmos a sua implementação. O primeiro passo foi criar o modelo ER, apresentado anteriormente na Figura 1**,** que foi sofrendo ligeiras alterações quando achávamos que seria pertinente guardar outro tipo de informação.

Posteriormente ao ER, inicializamos a construção das vistas. Para isso, usámos como ferramenta a plataforma *figma* e construímos apenas um protótipo do que seria o nosso frontend. O nosso objectivo não foi organizar esteticamente a informação, mas sim perceber que campos é que teríamos de apresentar em cada página.

Em anexo [1] encontra-se o link para a plataforma figma onde podem ser visualizadas as diferentes vistas.

Implementação

Funcionalidades

Para apresentar quais as funcionalidades desenvolvidas, restringimos o acesso de acordo com o tipo de utilizador que estivesse a utilizar a sessão naquele momento. Por esse mesmo motivo, apresentamos primeiro quais são as permissões de acesso de cada um dos diferentes tipos de utilizadores - convidado, utilizador registado e admin.

	Convidado	Utilizador registado	Admin
Visualizar quais os jogos se encontram em progresso	✔	✔	✔
Visualizar quais os jogos futuros	✔	✔	✔
Visualizar informações de cada jogo	✔	✔	✔
Visualizar as estatísticas	✔	✔	✔
Criar um evento num determinado jogo	✘	✔	✔
Criar e gerir jogos (editar e eliminar)	✘	✘	✔
Criar e gerir jogadores (editar e eliminar)	✘	✘	✔
Criar e gerir equipas (editar e eliminar)	✘	✘	✔
Criar e gerir utilizadores (editar e eliminar)	✘	✘	✔

Tabela 1: *permissões dos diferentes tipos de utilizadores*

Para a implementação destas mesmas funcionalidades organizámos o nosso código em algumas secções principais:

1. **data:** package onde se encontram as classes respectivas às entidades - Event, Football_Game, Player, Team, User;
2. **demo:** package que contém ''DemoApplication'', todos os ficheiros de repositório e serviços e ainda uma package *controllers* - AdminController, DataController, UserController;
3. **formData**: package que contém as classes para formulários - FromGame, FormPlayer, FormTeam, FormTimeOfEvent, FormUser, Login;
4. **templates:** pasta que contém os ficheiros HTML dividida em outras pastas devidamente identificadas.

A maior parte da implementação passou pelas secções acima descritas.

Apesar do nosso foco não ser a parte visual do projeto (frontend), explorámos também o bootstrap, framework que disponibiliza código aberto para que possamos desenvolver componentes de interface e Frontend para sites e aplicações web, usando HTML, CSS e Javascript. Apesar do interesse na framework, apenas uma das nossas vistas se encontra implementada com o uso da

mesma. Gostaríamos, no entanto, de dar continuidade ao projeto fazendo todas as vistas recorrendo a essa mesma framework.

Ao desenvolver esta plataforma consideramos que o processo mais difícil tenha passado por resolver alguns tipos de erros que iam surgindo tendo em conta os diferentes tipos de dados que tínhamos (String, Timestamp, Date) e a forma como gostaríamos de apresentar esses mesmos dados em algumas das páginas.

Das funcionalidades todas que implementamos, consideramos mais desafiadora a funcionalidade de mostrar as informações de um jogo e de criar e gerir eventos uma vez que tentamos seguir rigorosamente a sequência pela qual podem surgir os eventos no futebol (nenhum jogador com vermelho pode marcar golo, um jogador recebe cartão vermelho quando recebe cartão amarelo, nenhum jogo pode começar 2 vezes, etc).

Vistas com info atualizada automaticamente

De modo a que pudesse haver uma atualização automática de páginas como: homepage, gameInfo e statistics optámos por colocar nos HTML's o seguinte código que permite a atualização destas mesmas páginas de 20 em 20 segundos:

```
<meta http-equiv="refresh" content="20">
```

Testes

Tipo de utilizador	Teste	Procedimento	Sucesso	Insucesso
Admin/ Utilizador registado/ Convidado	Landing Page	Carregar no botão de "Login" deve-me levar para o HTML de preenchimento do login	Aparece o HTML loginResult	-
Admin/ Utilizador registado/ Convidado	Landing Page	Carregar no botão de "Login as guest" deve-me levar para a homepage	Aparece o HTML homepage	-
Admin/ Utilizador registado	Login	Selecionar o botão "Login" depois de ter escrito o username correto e password deve-me levar para o HTML homepage	Aparece o HTML homepage	-
Admin/ Utilizador registado	Login	Selecionar o botão "Login" depois de ter escrito o username incorreto e/ou password deve-me apresentar uma mensagem de erro	Aparece o HTML com a mensagem de erro personalizada	-
Admin/	Home	Selecionar o botão	Aparece o HTML das	-

Utilizador registado/ Convidado	Page	“Statistics” deve-me levar para o HTML statisticsMenu	estatísticas - statisticsMenu	
Admin	Home Page	Selecionar o botão “Settings” deve-me levar para o HTML settings	Aparece o HTML das settings	-
Admin/ Utilizador registado/ Convidado	Home Page	Selecionar o botão com um jogo por ex: “SC Sporting x Benfica” deve-me levar para o HTML gameInfo	Aparece o HTML do gameInfo	-
Admin/ Utilizador registado/ Convidado	Statistics Menu	Deve-me apresentar as estatísticas de todas as equipas - Nº de jogos, Nº de derrotas, Nº de vitórias e Nº de empates. Para além disso, também deve apresentar o melhor marcador de golos.	Aparece o HTML das estatísticas com as estatísticas atualizadas. (A atualização das estatísticas apenas é feita após ter terminado um jogo)	-
Admin/ Utilizador registado/ Convidado	Statistics Menu	Ao carregar no botão “Games”mostrar as equipas ordem decrescente de nº de jogos. Ao carregar no botão “Victories” deve colocar por ordem decrescente de nº de vitórias. Ao carregar no botão “Defeats” deve colocar por ordem decrescente de nº de derrotas. Ao carregar no botão “Draws” deve colocar por ordem decrescente de nº de empates.	Aparece o HTML das estatísticas com as estatísticas ordenadas por nº de jogos ou nº de vitórias ou nº de derrotas ou nº de empates.	-
Admin	Settings	Selecionar o botão “Teams” deve-me levar para o HTML manageTeams	Aparece o HTML manageTeams com a informação correta sobre cada equipa, botão para editar, apagar e criar uma nova equipa.	-
Admin	Settings	Selecionar o botão “Users” deve-me levar para o HTML manageUsers	Aparece o HTML manageUsers com a informação correta sobre cada utilizador, botão para editar, apagar e criar um novo	-

			utilizador.	
Admin	Settings	Selecionar o botão “Games” deve-me levar para o HTML manageGames	Aparece o HTML manageGames com a informação correta sobre cada jogo, botão para editar, apagar e criar um novo jogo.	-
Admin	Settings	Selecionar o botão “Players” deve-me levar para o HTML managePlayers	Aparece o HTML manageUsers com a informação correta sobre cada jogador, botão para editar, apagar e criar um novo jogador.	-
Admin	Manage Teams	Selecionar o botão “Add team” deve-me levar para o HTML addNewTeam	Aparece o HTML addNewTeam	-
Admin	Manage Users	Selecionar o botão “Add team” deve-me levar para o HTML addNewUsers	Aparece o HTML addNewUsers	-
Admin	Manage Games	Selecionar o botão “Add team” deve-me levar para o HTML addNewGames	Aparece o HTML addNewGames	-
Admin	Manage Players	Selecionar o botão “Add team” deve-me levar para o HTML addNewPlayers	Aparece o HTML addNewPlayers	-
Admin	Add New Team	Selecionar o botão “Add team” depois de ter escrito o nome e URL da imagem correto deve-me levar para o HTML homepage e guardar a informação da nova equipa	Aparece o HTML da Homepage e a equipa é adicionada	-
Admin	Add New Team	Selecionar o botão “Add team” depois de não ter escrito o nome ou URL da imagem, ou introduzir nome ou URL de equipas que já se encontrem na base de dados, deve-me levar para o HTML AddNewTeam e apresentar uma mensagem de erro	Aparece o HTML AddNewTeam e uma mensagem de erro	-
Admin	Add New User	Selecionar o botão “Add User” depois de ter escrito o nome, username, password,	Aparece o HTML da Homepage e o utilizador é adicionado	-

		email, telefone e/ou se é admin correto deve-me levar para o HTML homepage e guardar a informação do novo utilizador		
Admin	Add New User	Selecionar o botão “Add team” depois de ter escrito o nome, username, password, email, telefone e/ou se é admin incorreto ou incompleto deve-me levar para o HTML AddNewUser e apresentar uma mensagem de erro	Aparece o HTML AddNewUser e uma mensagem de erro	-
Admin	Add New Game	Selecionar o botão “Add Game” depois de ter escrito o nome da equipa A, nome da equipa B, localização e data correto deve-me levar para o HTML homepage e guardar a informação do novo jogo	Aparece o HTML da Homepage e o jogo é adicionado	-
Admin	Add New Game	Selecionar o botão “Add team” depois de ter escrito o nome da equipa A, nome da equipa B, localização e data incorreto ou incompleto deve-me levar para o HTML AddNewGame e apresentar uma mensagem de erro	Aparece o HTML AddNewGame e uma mensagem de erro	-
Admin	Add New Player	Selecionar o botão “Add Player” depois de ter escrito o nome , posição de jogo, aniversário e equipa corretamente deve-me levar para o HTML homepage e guardar a informação do novo jogador	Aparece o HTML da Homepage e o jogador é adicionado	-
Admin	Add New Player	Selecionar o botão “Add Player” depois de ter escrito o nome , posição de jogo, aniversário e equipa incorretamente deve-me levar para o HTML AddNewPlayer e apresentar uma mensagem de erro	Aparece o HTML AddNewPlayer e uma mensagem de erro	-

Admin/ Utilizador registado/ Convidado	Game info	Deve aparecer toda a informação relativa ao jogo: Nome, resultado, localização, data e eventos.	Aparece toda a informação de um jogo corretamente.	-
Admin/ Utilizador registado	Game info	Selecionar o botão “Add Event” deve-me levar para o HTML eventsMenu	Se carregar no botão “Add Event” aparece o HTML eventsMenu.	-
Admin/ Utilizador registado	Events Menu	Deve aparecer os botões “Game started”, “Game finished”, “Goal”, “Yellow Card”, “Red Card”, “Game interrupted” e “Resume Game”.	Aparecem todos os botões corretamente.	-
Admin/ Utilizador registado	Events Menu	Carregar no botão “Game started”, caso ainda não tenha começado o jogo, deve criar um evento em que um dado jogo começa. Caso o jogo já tenha começado, deve gerar uma mensagem de erro.	Caso o jogo ainda não tenha começado cria o evento “Game started” à hora em que se carregou no botão. Caso contrário, aparece uma mensagem de erro.	-
Admin/ Utilizador registado	Events Menu	Carregar no botão “Game finished”, caso já tenha começado o jogo e não esteja interrompido, deve criar um evento em que um dado jogo termina. Caso o jogo ainda não tenha começado ou esteja interrompido, deve gerar uma mensagem de erro.	Caso o jogo já tenha começado e não esteja interrompido cria o evento “Game finished” à hora em que se carregou no botão. Caso contrário ou caso esteja interrompido, aparece uma mensagem de erro.	-
Admin/ Utilizador registado	Events Menu	Carregar no botão “Game interrupted”, caso já tenha começado o jogo e não esteja interrompido, deve criar um evento em que um dado jogo fica interrompido. Caso o jogo ainda não tenha começado ou esteja interrompido, deve gerar uma mensagem de erro.	Caso o jogo já tenha começado e não esteja interrompido cria o evento “Game interrupted” à hora em que se carregou no botão. Caso contrário ou caso esteja interrompido, aparece uma mensagem de erro.	-

Admin/ Utilizador registado	Events Menu	Carregar no botão “Resume game”, caso já tenha começado o jogo e esteja interrompido, deve criar um evento em que um dado jogo fica resumido. Caso o jogo ainda não tenha começado ou não esteja interrompido, deve gerar uma mensagem de erro.	Caso o jogo já tenha começado e esteja interrompido, cria o evento “Resume game” à hora em que se carregou no botão. Caso contrário ou caso não esteja interrompido, aparece uma mensagem de erro.	-
Admin/ Utilizador registado	Events Menu	Carregar no botão “Goal” caso o jogo já tenha começado deve apresentar o HTML newPlayerTimeEvent. Caso o jogo ainda não tenha começado ou esteja interrompido, deve aparecer uma mensagem de erro.	Aparece o HTML newPlayerTimeEvent caso esteja tudo correto. Caso contrário, aparece uma mensagem de erro.	-
Admin/ Utilizador registado	Events Menu	Carregar no botão “Yellow Card” caso o jogo já tenha começado deve apresentar o HTML newPlayerTimeEvent. Caso contrário, deve aparecer uma mensagem de erro.	Aparece o HTML newPlayerTimeEventcaso esteja tudo correto. Caso contrário, aparece uma mensagem de erro.	-
Admin/ Utilizador registado	Events Menu	Carregar no botão “Red Card” caso o jogo já tenha começado deve apresentar o HTML newPlayerTimeEvent. Caso contrário, deve aparecer uma mensagem de erro.	Aparece o HTML newPlayerTimeEventcaso esteja tudo correto. Caso contrário, aparece uma mensagem de erro.	-
Admin/ Utilizador registado	New Player Time Event - Goal	Não colocar todos os parâmetros do input necessários deve gerar uma mensagem de erro.	Aparece uma mensagem de erro gerada por nós.	
Admin/ Utilizador registado	New Player Time Event - Goal	Colocando todos os parâmetros de input e clicando no botão “Add event” deve adicionar o evento ao jogo e redirecionar para a homepage.	O evento é adicionado ao jogo e a pessoa é redirecionada para a homepage.	-

Admin/ Utilizador registado	New Player Time Event - Goal	Caso se tente adicionar o golo de alguém que já está fora de jogo (jogador com dois eventos de cartão amarelo ou um vermelho) deve aparecer uma mensagem de erro.	O evento não é adicionado ao jogo e aparece uma mensagem de erro gerada por nós.	-
Admin/ Utilizador registado	New Player Time Event - Yellow Card	Não colocar todos os parâmetros do input necessários deve gerar uma mensagem de erro.	Aparece uma mensagem de erro gerada por nós.	-
Admin/ Utilizador registado	New Player Time Event - Yellow Card	Se já existem mais que dois cartões amarelos ou um vermelho para o mesmo jogador deve aparecer uma mensagem de erro.	Aparece uma mensagem de erro gerada por nós.	-
Admin/ Utilizador registado	New Player Time Event - Yellow Card	Colocando todos os parâmetros de input e clicando no botão “Add event” deve adicionar o evento ao jogo e redirecionar para a homepage. Caso já exista um cartão amarelo para o mesmo jogador, o segundo é adicionado e é também gerado automaticamente um cartão vermelho.	O evento é adicionado ao jogo e a pessoa é redirecionada para a homepage. Caso já exista um cartão amarelo, então é adicionado o evento “Yellow Card” e “Red Card”.	-
Admin/ Utilizador registado	New Player Time Event - Red Card	Não colocar todos os parâmetros do input necessários deve gerar uma mensagem de erro.	Aparece uma mensagem de erro gerada por nós.	-
Admin/ Utilizador registado	New Player Time Event - Red Card	Se já existir algum cartão vermelho para o mesmo jogador deve aparecer uma mensagem de erro.	Aparece uma mensagem de erro gerada por nós.	-
Admin/ Utilizador registado	New Player Time Event - Red Card	Colocando todos os parâmetros de input e clicando no botão “Add event” deve adicionar o evento ao jogo e redirecionar para a homepage.	O evento é adicionado ao jogo e a pessoa é redirecionada para a homepage.	-
Admin/	Create	Clicando no botão “Create	Ao clicar no botão a	-

Utilizador registado/ Convidado	Data - popular base de dados	Dummy Data” deve popular a base de dados com dados da api-sports e dados pré-criados no servidor scoreDEI caso ainda não tenha sido excedido o limite de pedidos feitos à API. Caso contrário, deve apenas criar os utilizadores e ser possível criar teams e players de forma manual.	base de dados é populada com dados da api-sports e dados pré-criados no servidor scoreDEI caso não tenha sido excedido o limite de acessos. Caso contrário, são apenas criados os utilizadores e é possível criar teams e players manualmente.	

Tabela 2: *Descrição dos testes feitos à plataforma*

Referências

[1] - https://getbootstrap.com/
[2] - https://www.w3schools.com/TAgs/default.asp

Anexo

[1]-https://www.figma.com/file/1ltglJ8e7K2k1Ebyyb5LkP/%5BSD%5D-Projeto-2?node-id=0%3A1