

scoreDEI: Resultados desportivos em direto

SD 2021/22 — Meta 2 — 30 de maio de 2022 (23:59)

Resumo

O scoreDEI é uma plataforma para consulta de resultados desportivos em tempo real. Este projeto consiste em implementar uma plataforma onde utilizadores podem consultar e receber actualizações de resultados desportivos de jogos que se encontrem a decorrer.

1 Objetivos do projeto

No final do projeto cada estudante deverá ter:

- programado um sistema de Web com: camada de dados e Frontend através do Spring Boot e Thymeleaf
- programado uma Base de dados através da Java Persistence Application Programming Interface (JPA)
- integração com serviço externo (api-sports.io) utilizando a tecnologia REST para obtenção de dados

2 Visão geral

O acesso a informação instantânea prolifera nos dias que correm. Desde a interacção com as redes sociais, à realização de operações bancárias, o utilizador digital actual tem acesso a uma vasta rede de serviços na palma da sua mão, com os quais a troca de informação é imediata. Também no acesso a resultados desportivos os utilizadores mais aficionados têm demonstrado, ao longo dos anos, a necessidade de acompanhar o resultado de encontros à medida que estes se desenrolam. Como tal foi identificada a oportunidade para a criação de um serviço que disponibilize, aos seus utilizadores, actualizações de resultados desportivos em tempo real.

3 Arquitectura

A plataforma scoreDEI deve permitir todas as operações necessárias para que seja possível aos utilizadores consultar resultados de eventos desportivos.

Para tal, a plataforma deve, por um lado disponibilizar funcionalidades de gestão administrativa da informação a ser apresentada assim como dos próprios utilizadores que irão ter acesso à plataforma, e por outro fornecer funcionalidades de login e consulta de informação aos utilizadores finais.

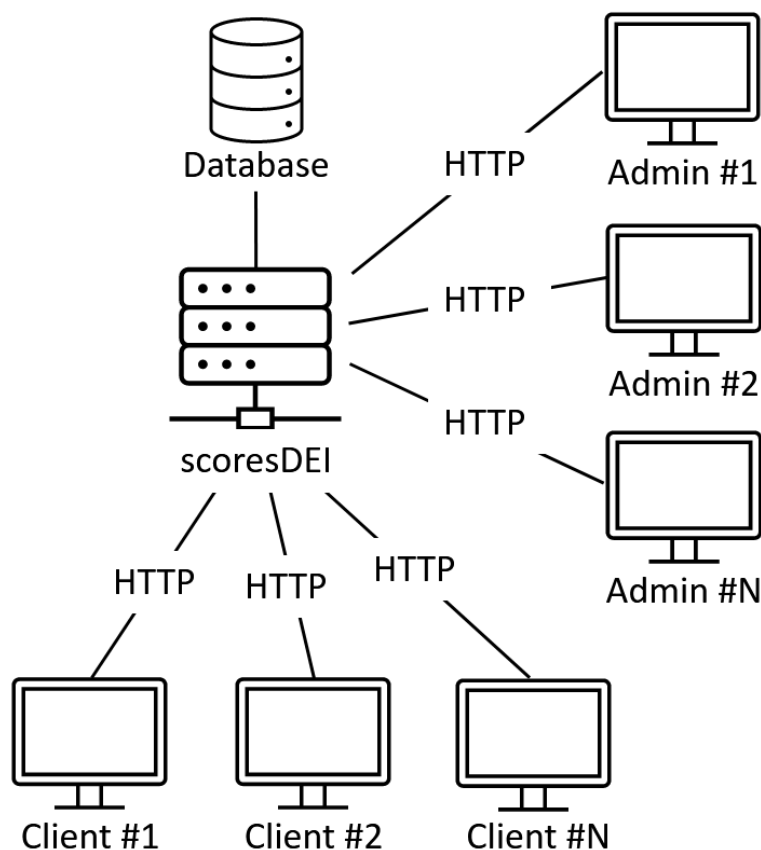


Fig. 1: Arquitectura de alto nível da plataforma scoresDEI

A plataforma que será implementada neste projeto deverá seguir a arquitectura apresentada na Figura 1. Como é habitual em websites, aplicações e plataformas web, a arquitectura da plataforma segue um *design pattern frontend* e *backend*, no qual o *frontend* representa o código fonte que é executado no *browser* do cliente (páginas HTML, CSS, Javascript), e o *backend* representa o conjunto de serviços e código fonte que existem do lado do servidor Web que serve a plataforma.

Como se pode observar na Figura 1, em termos de *frontend* existem diferentes vistas para utilizadores com privilégios de administração da plataforma, e para os outros utilizadores. A comunicação entre o cliente e a plataforma web é efectuada através do protocolo HTTP. Já o *backend* consiste num servidor Web, que disponibiliza um conjunto de endpoints para responder aos diferentes pedidos efectuados pelo *frontend*, e comunica com um servidor de base de dados que utiliza para consultar e salvaguardar dados (dados de utilizador, equipas, jogos, etc.).

Como todo e qualquer software está sujeito a falhas, a plataforma scoreDEI deve considerar a possibilidade de ocorrência de falhas e lidar com estas de forma adequada.

É sempre crucial o tratamento de excepções que possam existir, e no caso de ocorrer uma falha o utilizador deverá ser notificado do que sucedeu e o que poderá fazer nessa situação.

4 Interface 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 menú de navegação.

Neste ponto os alunos não se deverão preocupar muito com a qualidade da interface apresentada. Poderão utilizar Bootstrap ou outra framework frontend ou simples código HTML que simplifique o processo.

4.1 Funcionalidades Administradores

As funcionalidades que se seguem, e as vistas que permitem aceder às mesmas, deverão estar acessíveis apenas a utilizadores com permissões de administração.

1. **Registar utilizadores**

Deve ser possível efetuar o registo de utilizadores na plataforma. Existem dois tipos de utilizadores, os administradores, que são responsáveis pela gestão de jogos e de resultados em direto, e os utilizadores, que podem submeter resultados e consultar jogos que estão a decorrer. Para cada um deles deverá ser necessário preencher a seguinte informação: nome, email, password e contacto telefónico.

2. **Criar/Gerir Equipas**

Um administrador deve conseguir criar equipas, cada equipa terá um nome e uma imagem associada

3. **Criar/Gerir Jogadores**

Um administrador deve conseguir criar jogadores, cada jogador terá um nome próprio, posição em que joga (defesa, lateral, médio, etc), data de nascimento e deverá ser associada uma equipa já existente no sistema.

4. **Criar/Gerir Jogos de Futebol**

Deve ser possível aos administradores criar jogos de futebol. Cada jogo deverá contemplar a seguinte informação:

- (a) Equipa A e Equipa B (que deverão ser seleccionadas a partir da lista de equipas registadas no sistema)
- (b) Localização - Localização do evento

(c) Data - data em que ocorrerá o evento e hora de início

Cada jogo deverá também ter uma lista de eventos, que serão preenchidos pelas informações dos utilizadores em tempo real. O resultado de um jogo poderá ser calculado a qualquer momento através da análise destes eventos.

5. Gerir eventos num jogo - Grupos de 3

Para validar a veracidade dos eventos reportados pelos diferentes utilizadores, em tempo real, o administrador deverá ter a possibilidade de marcar um evento como verdadeiro ou rejeitá-lo. Assim, todos os eventos que estão a ser reportados, devem ser visíveis ao administrador num painel de administração de um jogo, podendo efectuar apenas estas duas acções para cada evento reportado.

4.2 Funcionalidades Gerais

As funcionalidades que se seguem, e as vistas que permitem aceder às mesmas, deverão estar acessíveis a diferentes grupos de utilizadores

1. Registar evento num jogo (utilizadores registados)

Cada utilizador ao entrar na plataforma, terá disponível uma lista de jogos a decorrer ou jogos futuros. Ao entrar num jogo, poderá submeter eventos para esse mesmo jogo. Os eventos que podem ser introduzidos na plataforma são os seguintes:

- (a) Início e Fim de Jogo - Sinalizar o início e o final da partida.
- (b) Golo - Seleccionar um jogador e reportar um golo desse jogador, e respectiva equipa, com uma data e hora associada
- (c) Cartão Amarelo - Seleccionar um jogador e reportar um cartão amarelo desse mesmo jogador, com uma data e hora associada
- (d) Cartão Vermelho - Seleccionar um jogador e reportar um cartão vermelho desse mesmo jogador, com uma data e hora associada
- (e) Jogo interrompido - Indicar que o jogo se encontra interrompido
- (f) Jogo resumido - Indicar que o jogo volta a estar a decorrer (após ter estado interrompido)

2. Acompanhar jogo (utilizadores não registados)

Um utilizador, que pode ser não autenticado, deverá conseguir visualizar os detalhes de um jogo a decorrer, incluindo resultado actual, tempo de jogo e listagem de eventos até ao momento. Assim, a página de um jogo deverá mostrar:

- (a) Nome do jogo (Equipa A vs Equipa B)
- (b) Resultado actual
- (c) Localização

(d) Data

(e) Listagem de eventos, ordenados de forma cronológica

Estes eventos só serão apresentados nesta vista após serem aprovados por um Administrador.

3. Consultar estatísticas (utilizadores não registados)

A plataforma scoresDEI pretende ser mais do que uma plataforma para acompanhar jogos ao vivo, para isso irá tirar proveitos dos dados registados sobre equipas e jogos realizados para apresentar aos utilizadores diversas estatísticas que estes achem interessantes. A vista de estatísticas deverá apresentar ao utilizador, pelo menos as seguintes estatísticas:

- (a) Listagens ordenáveis das equipas com informação sobre jogos realizados, nomeadamente: número de jogos, número de vitórias, número de derrotas, número de empates.
- (b) Nome do melhor marcador de golos.
- (c) **(Apenas grupos de 3 elementos)** Estatísticas adicionais para as listagens do ponto anterior: número médio de golos por jogo, número máximo de golos num jogo.
- (d) **(Apenas grupos de 3 elementos)** Um formulário, que permita escolher duas equipas e, após a submissão do mesmo, apresente uma página com o número de vezes que estas já se defrontaram e estatísticas adicionais sobre esses jogos: vitórias equipa A e equipa B, empates, total golos, total cartões amarelos e vermelhos.

5 Repositório de dados

Apesar de a plataforma scoresDEI estar ainda na sua infância, e como tal ainda não ter muita informação relativamente a equipas, jogadores e jogos realizados, pretende-se que ao longo do tempo isso mude, e que no futuro esta se torne uma plataforma de referência na disponibilização de dados e estatísticas sobre jogos de futebol.

Como tal pretende-se que a plataforma faça uso de uma base de dados SQL para persistência dos dados recolhidos para o futuro, ver Figura 2.

Para a implementação desta funcionalidade pretende-se que os alunos tirem partido do aprendido durante as aulas sobre Java Persistence Application Programming Interface (JPA) para implementar a comunicação entre o backend da plataforma e uma base de dados SQL. É aconselhável o uso de PostgreSQL, no entanto os alunos poderão escolher um qualquer outro motor de base de dados SQL com o qual se sintam mais confortáveis.

Para a implementação deste requisitos com sucesso, os alunos são aconselhados a primeiro identificar as entidades, e as suas propriedades, que melhor representam os dados que a plataforma irá receber e pretende guardar, assim como as suas relação.

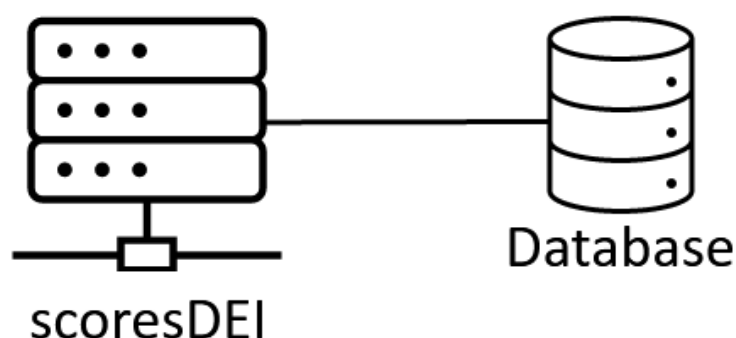


Fig. 2: Repositório de dados para a plataforma scoresDEI

Os alunos deverão construir um diagrama Entidade-Relacionamento da estrutura de dados e apresentar esse diagrama no relatório final do projecto.

6 Implementação

Nesta secção são apresentadas considerações adicionais de implementação que dizem respeito apenas aos alunos da LEI. Note-se que deverão ser utilizadas as seguintes tecnologias para este projeto: Java Spring Boot e Thymeleaf, outras abordagens não serão consideradas para avaliação do mesmo.

6.1 Integração com serviços externos

De forma a facilitar a inserção de dados na plataforma, e também no sentido de obter dados reais de equipas de futebol e os seus jogadores, pretende-se usar um serviço externo para obter esses dados. O serviço externo que iremos utilizar para obtenção de dados reais é a API de futebol disponibilizada pelo web site <https://api-sports.io/>.

Ainda que a API seja grátis dentro de determinados limites (30 pedidos por minuto e 100 pedidos por dia), os pedidos HTTP para os endpoints da API obrigam à inclusão de uma chave nos Headers do pedido para controlo de acessos.

Para obter uma chave, deverão efetuar o registo de uma conta através de <https://dashboard.api-football.com/register>, e uma vez tendo acesso ao Dashboard, terão acesso à API Key na secção Profile.

Os alunos deverão criar uma funcionalidade de inicialização de dados da plataforma scoresDEI, a qual faz uma sequência de pedidos HTTP à API REST referida, no sentido de obter dados de equipas de futebol e dos seus jogadores.

Algumas sugestões de leitura para a implementação desta funcionalidade:

1. **Documentação da API api-sports.io:**

<https://api-sports.io/documentation/football/v3>

2. **Tutorial sobre como consumir uma API REST em Java:**

<https://rapidapi.com/blog/how-to-use-an-api-with-java/>

6.2 Data Transfer Objects - Apenas grupos de 3 elementos

Para transferirem dados entre a camada de apresentação e a camada de negócio os grupos de 3 alunos deverão utilizar Data Transfer Objects (DTOs). Este requisito aplica-se aos dois sentidos da interação entre essas camadas.

7 Entrega do projeto

Os Alunos deverão ter atenção às considerações que se seguem relativamente à entrega do projeto.

7.1 Relatório

Devem reservar tempo para a escrita do relatório final do projeto, tendo em conta os passos anteriores. Devem escrever o relatório de modo a que um novo colega que se junte ao grupo possa perceber a solução criada, as decisões técnicas e possa adicionar novos componentes ou modificar os que existem. Deve incluir:

- Arquitetura detalhada da plataforma e a sua descrição. Deverá mostrar um nível mais detalhado da arquitectura do que a apresentada na Figura 1. Deverá por exemplo apresentar os diferentes controladores e/ ou Servlet, vistas e outros módulos que possam existir;
- Detalhes sobre a implementação do backend da plataforma. Como foram organizadas as diferentes funcionalidades, quais funcionalidades se revelaram mais complexas e como conseguiram ultrapassar esses desafios. Detalhes sobre o controlo de acesso de utilizadores;
- Detalhes sobre a implementação de qualquer vista que tenha actualização da sua informação de forma automática.
- Descrição dos testes feitos à plataforma (tabela com pass/fail).

7.2 Formato e conteúdo

O projeto deverá ser entregue num arquivo ZIP. Esse arquivo deverá conter um ficheiro README.TXT com toda a informação necessária para instalar e executar o projeto sem a presença dos alunos. Projetos sem informações suficientes, que não compilem ou não executem corretamente **não serão avaliados**.

Dentro do ficheiro ZIP deverá também estar um Javadoc/PDF/HTML com o relatório. O relatório deve seguir a estrutura fornecida, dado que a avaliação irá incidir sobre cada um dos pontos. Também no ficheiro ZIP deverá existir um ficheiro WAR com a aplicação Web pronta a executar, bem como os entregáveis da meta 1 prontos a correr.

Finalmente, o ficheiro ZIP deverá ter também **uma pasta com o código fonte completo do projeto**. A ausência deste elemento levará à anulação do projeto.

7.3 Deadline

O ficheiro ZIP com o projeto deverá ser entregue na plataforma Infoestudante até ao dia **30 de maio de 2022 (23:59)**, via <https://infoestudante.uc.pt>, entregas após o prazo terão uma penalização de **5%** na nota final.