



FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE  
COIMBRA

---

## *Sistemas Distribuídos*

*Trabalho Prático Nº2:      Score Dei*

---

## Índice

<b>1. Introdução</b>	<b>3</b>
<b>2. Membros do Grupo</b>	<b>3</b>
<b>3. Descrição da arquitetura da plataforma</b>	<b>4</b>
<b>4. Detalhes sobre a implementação do backend da plataforma</b>	<b>5</b>
<b>5. Testes realizados à plataforma</b>	<b>6</b>
<b>6. Dificuldades</b>	<b>8</b>
<b>7. Conclusão</b>	<b>8</b>

## 1. Introdução

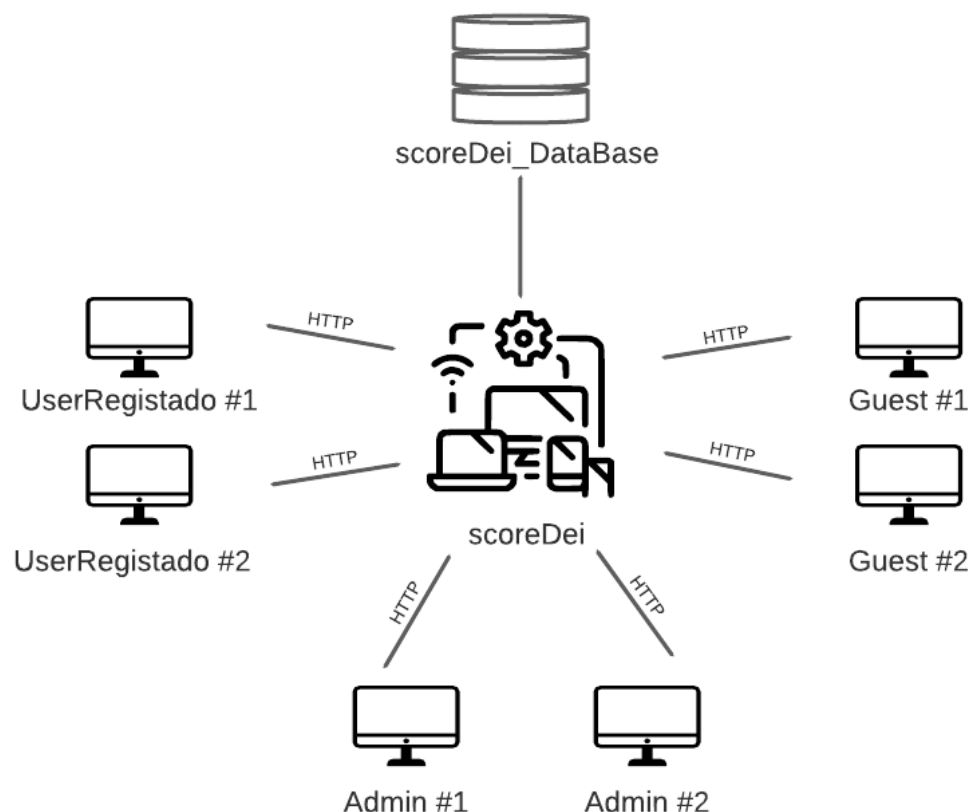
No âmbito da unidade curricular de Sistemas Distribuídos, foi pedido a realização de uma plataforma com a finalidade de consultar e receber atualizações de resultados desportivos de jogos que se encontrem a decorrer. Esta API, tem como funcionalidade consolidar o conhecimento e domínio através da Framework Spring Boot e Thymeleaf acompanhado por um pouco de linguagem SQL como repositório de dados e um API exterior ao projeto que contém algumas informações relevantes ao desporto para que a App seja mais realista com uma boa quantidade de informação.

## 2. Membros do Grupo

Nome	Nº de Estudante	Email	PL
Gonçalo Correia	2018306132	uc2018306132@student.uc.pt	PL4
João Marques	2018278463	uc2018278463@student.uc.pt	PL4
Maria Sarmento	2018279801	uc2018279801@student.uc.pt	PL4

### 3. Descrição da arquitetura da plataforma

**Figura 1** - Arquitetura da scoreDEI



A nossa API é constituída por um servidor principal, o *AppController.java*, que irá interagir com o usuário, administrador ou convidado. Serve como elo de ligação entre estas 3 classes, guardando as informações e pedidos de cada uma na base de dados.

Este servidor, funciona como o cérebro da API, governando todas as orientações dadas pelos utilizadores ou pelos administradores. Os convidados, apenas tem acesso à aplicação para fins de demonstração sem poder interagir com o servidor.

O administrador, tal como o nome indica, possui regalias em comparação aos outros usuários, nomeadamente criar jogadores, criar equipas, criar jogos, validar eventos criados por utilizadores, como também possui as outras funcionalidades presentes nos outros usuários. Um usuário normal, apenas pode para além da observação das estatísticas, criar e dominar um jogo, como por exemplo, começar uma partida, acabá-la, interrompê-la, arbitrar a partida (cartão vermelho, cartão amarelo, golo).

## 4. Detalhes sobre a implementação do backend da plataforma

Para dar início ao funcionamento da scoreDEI, é necessária a criação de, pelo menos, 2 equipas bem como o respetivo jogo entre elas. Esta ação é realizada por um user ou por um admin.

Recorrendo à rota */getAPI*, é possível retirar estas informações da api-sports, em formato Json, onde foi utilizado postman para a facilitação da visualização do ficheiro. Foi utilizada a biblioteca Jackson para a extração da informação e, de seguida, procedeu-se à criação das variáveis das equipas e dos jogadores.

Para além desta rota, foram também criadas as seguintes:

- */register* ou */process\_register* - trata do registo do utilizador na base de dados, encriptando a password com recurso ao MD5.
- */login* ou */process\_login* - encontra o utilizador e a respetiva pass encriptada de modo a efetuar o login.
- */notLogged* - página para os utilizadores não registados.
- */register\_team* - cria a equipa.
- */register\_player* - cria o player.
- */create\_game* - cria o jogo.
- */gameEvent* - cria um evento de jogo.
- */playerEvent* - cria um evento para o jogador.
- */listTeams* - mostra todas as equipas.
- */players\_team* - mostra a lista de todos os jogadores de uma equipa.
- */details\_game* - amostragem dos detalhes do jogo que foi escolhido.
- */game\_details* - onde é feita a escolha do jogo que se quer analisar.
- */stats* - mostra as estatísticas do jogo.
- */tablescore* - mostra a tabela dos resultados.
- */bestScore* - mostra o melhor resultado.
- */teamVsTeam* - mostra todos os jogos entre 2 equipas escolhidas pelo user.

Por último, como dito anteriormente, foram criadas as classes, respectivos repositórios e serviços, nos repositórios foi usada a linguagem sql para obter as informações da base de dados.

## 5. Testes Realizados à plataforma

Foram realizados testes de acordo com as funcionalidades referidas no enunciado, tendo obtido sucesso em todos, como referido na introdução. Alguns extras também foram testados, como por exemplo a password ser hashed para nunca ser exposta.

Teste	Sucesso	Notas
Registar utilizador	Sim	O utilizador é registado com sucesso. O admin é reconhecido e tratado como tal, sendo possível aceder a opções específicas apenas para um administrador.
Login	Sim	É possível a qualquer utilizador registado fazer o login.
Guest	Sim	Caso um utilizador não esteja registado e não queira proceder ao registo, tem a opção de entrar como convidado, sendo que tem funcionalidades mais limitadas
Download da API	Sim	Os conteúdos da API são colocados corretamente na base de dados.
Lista Das Equipas	Sim	As equipas são listadas de forma correta, com a respetiva imagem.
Criar Jogo	Sim	É possível criar um jogo com as equipas existentes na base de dados.
Ações de Jogo	Sim	É possível começar um jogo, interromper um jogo, recomeçar um jogo e acabar

		um jogo corretamente.
Ação de Jogador	Sim	É possível adicionar ações a jogadores de uma partida.
Rank De Equipas	Sim	Após o final de um jogo, as pontuações das equipas são corretamente atualizadas.
Registrar Evento Num Jogo	Sim	Como utilizador registado, é possível enviar eventos para uma partida, como por exemplo golos e cartões.
Acompanhar Jogo	Sim	O utilizador deve conseguir visualizar os detalhes de um jogo a decorrer.
Consultar Estatísticas	Sim	As estatísticas são mostradas quando pedido, de forma correta.
Estatísticas a)	Sim	Mostradas corretamente.
Estatísticas b)	Sim	Mostradas corretamente.
Estatísticas c)	Sim	Mostradas corretamente.
Estatísticas d)	Sim	Mostradas corretamente.
Jogo com as mesmas equipas	Sim	Não é possível realizar jogos com a própria equipa.
Jogo com as mesmas equipas na mesma data	Sim	Não é possível realizar jogos com equipas que já têm jogo nessa data.
Utilizadores já registados	Sim	Não é possível registar um user que já tenha sido registado.
Campos não preenchidos	Sim	Não é possível submeter campos não preenchidos.
Password Encriptada	Sim	Password é encriptada antes de ser enviada para a base de dados.
Impossibilidade de arbitrar jogo finalizado	Sim	Feito corretamente.

Impossibilidade de arbitrar jogo não iniciado	Sim	Feito corretamente.
Impossibilidade de arbitrar jogadores de jogos não escolhidos	Sim	Não é possível criar uma ação relativamente a um jogador que não consta no jogo selecionado.
Tabela Team Vs Team	Sim	É possível ver a tabela com os resultados entre duas equipas selecionadas pelo user
Lista de Jogadores por Equipa	Sim	Tabela com todos os jogadores pertencentes a uma certa equipa.
DTO	Não	

## 6. Dificuldades

Existiram inúmeras dificuldades na realização deste projeto no que diz respeito à complexidade na importação de bibliotecas e das dependências. Neste projeto foi preciso interagir com outras bibliotecas e frameworks como, por exemplo, MD5, bibliotecas de segurança e tantos outros, e garantir que todas as bibliotecas estão presentes na aplicação acaba por ser massacrante.

Apresentação dos erros, também é alvo de queixa, pelo facto de existirem demasiadas dependências de código e os erros dificilmente serem detetados. Alguns bugs também foram encontrados, em que se passou tempo significativo e crucial à procura de um erro de parse quando tudo o que era preciso era iniciar a App.

## 7. Conclusão

Concluindo, este projeto contribui em grande parte para uma consolidação de uma framework desconhecida super interessante e construtiva, apesar de ter sido consolidada em pouco tempo, e inclusive o tempo de resolução do projeto.

Apesar de ter sido uma boa experiência, neste projeto perdeu-se algum tempo e focamo-nos em algumas coisas repetidas, gastando pouco tempo com algo que foi aprendido novo, nomeadamente a autenticação e proteção de rotas, criação de tokens, etc.



