



Gonçalo Vasconcelos Correia, nº 2019216331, pdcorreia@student.dei.uc.pt

João Garcia nº 2019216375, uc2019216375@student.uc.pt

Sistemas Distribuídos

scoreDei: Resultados desportivos em direto

Projeto 2- Meta 2- Turma PL 4

Coimbra, 2 de abril 2022

Índice

1. Arquitetura detalhada da plataforma e a sua descrição	3
2. Entidades e relações.....	5
3. Detalhes sobre a implementação do backend da plataforma	6
3.1 Endpoints e funcionalidades.....	6
4. Testes feitos à plataforma	7
5. Conclusão	11

1. Arquitetura detalhada da plataforma e a sua descrição

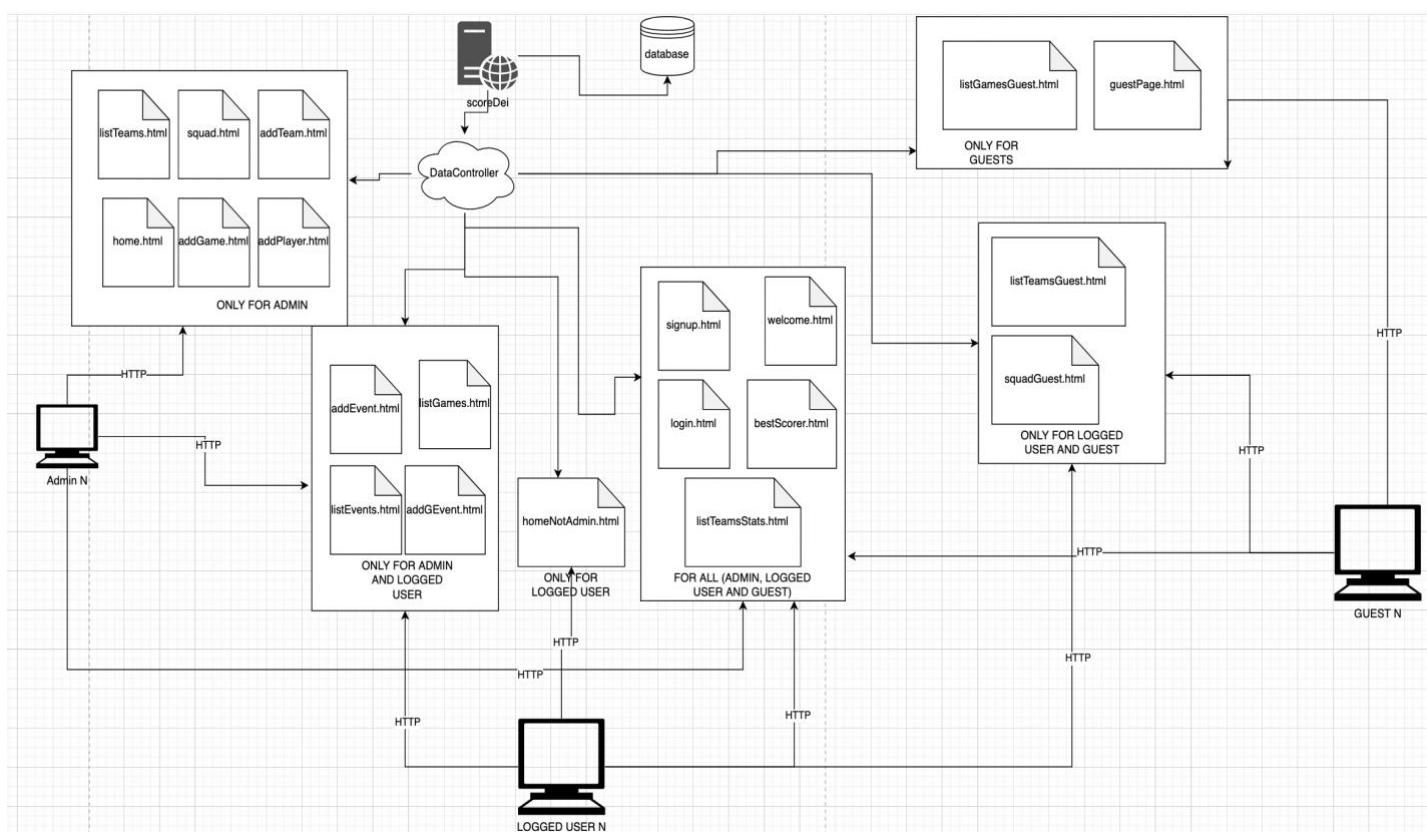


Fig 1 – Arquitetura da plataforma

A partir da imagem é possível observar que existe um servidor de backend, uma database que contém as entidades que posteriormente serão detalhadas, bem como as suas relações, e um controlador denominado de **DataController**. Este controlador contém vários endpoints que, posteriormente, serão enumerados e explicados. Cada endpoint, por sua vez, retorna páginas html ou redireciona para outros endpoints que só são possíveis de aceder tendo em conta o tipo de utilizadores. De facto, como é possível analisar pela imagem, existem três tipos de

utilizadores: Admins, Utilizadores registados (que fazem login) e Utilizadores não registados (guests). Deste modo, temos os seguintes grupos de páginas:

- *Only for admin*
 - listTeams.html
 - squad.html
 - addTeam.html
 - home.html
 - addGame.html
 - addPlayer.html
- *Only for admin and Logged User*
 - addEvent.html
 - listaGames.html
 - listEvents.html
 - addGEvent.html
- *Only for Logged User*
 - homeNotAdmin.html
- *For All (Admin, Logged User and Guest)*
 - signup.html
 - login.html
 - welcome.html
 - bestScorer.html
 - listTeamsStats.html
- *Only for Logged User and Guest*
 - listTeamsGuest.html
 - squadGuest.html
- *Only for Guest*
 - guestPage.html
 - listGamesGuest.html

2. Entidades e relações

Em seguida são apresentadas as várias entidades bem como as relações entre elas

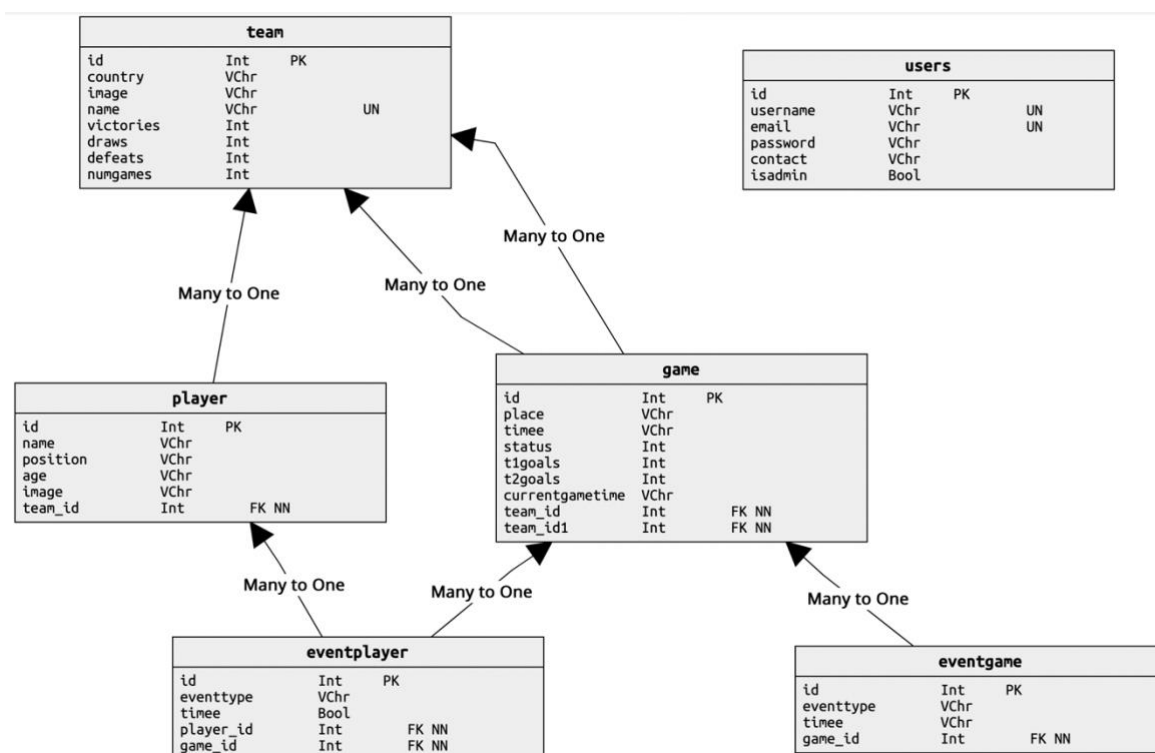


Fig 2 – Tabelas e suas relações

Como é possível observar pela figura existem seis tabelas. A entidade Player tem uma relação de Many to One com a entidade Team, ou seja, uma equipa pode ter vários jogadores, sendo que na entidade jogadores irá aparecer uma chave forasteira para a equipa a que pertence. Deste modo a busca pelos jogadores de uma equipa é facilitada, e caso se queira aceder a todos os atributos da equipa a partir de um jogador dela também é trivial. Do mesmo modo, a tabela Game tem uma dupla relação de Many to One para com a entidade Team. Deste modo, a tabela dos jogos vai ter duas chaves forasteiras que permitem aceder facilmente às duas equipas (e a todos os seus atributos) que se vão defrontar. Para os eventos, optámos por fazer a divisão em duas tabelas, de modo a separar os eventos que são para jogadores e os eventos relacionados com o jogo. De facto, era importante que o a tabela dos eventos tivesse uma chave forasteira para o jogador relacionado com o evento, no entanto se tivessmos apenas uma tabela isto não seria possível, visto que há eventos que não dizem respeito a nenhum jogador. Por consequente, o seu valor ficaria a nulo, o que não pode acontecer caso seja uma chave forasteira. Com a divisão das tabelas foi possível ter eventos que para além de uma chave forasteira para o jogo (visto que as duas tabelas têm uma relação Many to One com o jogo (um jogo pode ter vários eventos) também, uma delas, pode ter uma chave forasteira para o jogador. Por outro lado, a tabela dos utilizadores não tem qualquer relação com as outras tabelas, visto que achámos que não haveria necessidade tendo em conta as funcionalidades descritas no projeto.

3. Detalhes sobre a implementação do backend da plataforma

A implementação do backend está dividida em duas partes: Data e Demo. Na diretoria Demo estão presentes as entidades acima descritas: EventGame, EventPlayer, Game, Player, Team e Users. Na diretoria Demo estão presentes os services e repositories para cada uma destas entidades, bem como uma classe Evento e a classe AES256. A classe Evento acabou de referir foi feita com o intuito de auxiliar a colocar todos eventos respetivos a um jogo no mesmo local. De facto, como os eventos foram divididos em duas tabelas (EventGame e EventPlayer) foi necessária uma classe que os reunisse para depois serem mostrados por ordem cronológica. Por outro lado, a classe AES256 tem como função fazer a encriptação da password.

3.1 Endpoints e funcionalidades

As funcionalidades abaixo descritas tiveram como referência os diferentes tipos de utilizadores que existem, bem como os services e repositories de cada entidade. De facto, as funcionalidades que dizem respeito às equipas, por exemplo, fazem o uso do serviço de equipas e do seu repositório. De igual modo as outras entidades seguem o mesmo raciocínio.





- <http://localhost:8080/welcome> : Página inicial.
- <http://localhost:8080/login> : Funcionalidade de login.
- <http://localhost:8080/createUser> : Funcionalidade que permite ao utilizador registar-se na plataforma.
- <http://localhost:8080/guest> : Página com as funcionalidades disponíveis para um utilizador não registado.
- <http://localhost:8080/listGamesGuest> : Página que permite ao utilizador visualizar os jogos que estão a decorrer sem poder criar eventos para eles.
- <http://localhost:8080/listTeamsGuest> : Página que permite ao utilizador visualizar as equipas existentes sem as poder editar.
- <http://localhost:8080/squadByTeamGuest?id=3> : Página que permite ao utilizador visualizar os jogadores de cada equipa pelo seu ID sem as poder editar o jogador.
- <http://localhost:8080/bestScorer> permite visualizar o melhor (ou melhores no caso de empate) melhor marcador.
- <http://localhost:8080/listTeamStats?order=0> : Permite visualizar as estatísticas de cada equipa por uma ordem a escolher (order = 0 -> Ordem pelo número de jogos, order =1 -> Ordem pelo número de vitórias, order =2 -> Ordem pelo número de empates, order = 3-> Ordem pelo número de derrotas) .
- <http://localhost:8080/addTeam> : Funcionalidade que permite para adicionar equipas.
- <http://localhost:8080/addPlayer> : Funcionalidade que permite adicionar jogadores.
- <http://localhost:8080/addGame> : Funcionalidade que permite adicionar jogos.
- <http://localhost:8080/listTeams> : Permite listar as equipas com opção de as editar.
- <http://localhost:8080/editTeam?id=3> : Funcionalidade que permite editar a equipa com o id pretendido.
- <http://localhost:8080/squadByTeam?id=3> : Funcionalidade que permite listar os jogadores da equipa com o id pretendido, com a opção de editar.


- <http://localhost:8080/editPlayer?id=67> : Funcionalidade que permite editar o jogador com o id pretendido.
- <http://localhost:8080/listGames>: Permite listar os jogos com a opção de adicionar eventos.
- <http://localhost:8080/listEventsForGameId?id=1>: Permite listar os eventos do jogo com o id pretendido.
- <http://localhost:8080/eventGForGameId?id=2> : Permite adicionar eventos relacionados com o jogo (início, interromper, continuar e fim) ao jogo com o id pretendido.
- <http://localhost:8080/eventPForGameId?id=2> : Permite adicionar os eventos relacionados com um jogador escolhido ao jogo com o id pretendido.
- <http://localhost:8080/createData> : Permite ir buscar as equipas e jogadores à API externa.

Em termos de navegabilidade todos os endpoints são acessíveis a partir da plataforma. De facto, os endpoints específicos para cada tipo de utilizadores são apenas acessíveis para eles tendo em conta a navegabilidade. Por outras palavras, por exemplo, ou utilizador que não efetue login não consegue aceder ao endpoint de adicionar equipas através da navegação. No entanto, no desenvolvimento do projeto tivemos dificuldade em criar sessões para os utilizadores, por isso, alterados os links manualmente, qualquer utilizador da plataforma consegue usar todas as funcionalidades. Para além como foi feita a divisão dos eventos em duas tabelas, por motivos abordados anteriormente, a tarefa da listagem por ordem cronológica foi dificultada, pois tínhamos informações de eventos para um jogo em dois locais distintos. Com o intuito de contrariar esta adversidade, obtemos os eventos respetivos ao jogo pretendido das duas tabelas e convertemo-los para um objeto do tipo “Event”, armazenando-os todos na mesma lista do tipo “Event”. Assim, através do método `compareTo()`, já foi possível executar o ordenamento cronológico dos eventos.




4. Testes feitos à plataforma

1. Criação/Gestão de utilizadores








Testes	Desfecho Esperado	Pass/Fail
Criar utilizador	Criação com sucesso	
Criar utilizador com email repetido a um utilizador já registado	Criação recusada	
Criar utilizador com username repetido a um utilizador já inserido	Criação recusada	
Utilizadores apenas conseguirem as suas funcionalidades através da navegabilidade	Navegabilidade bem-sucedida	

Utilizadores não poderem aceder a endpoints de funcionalidades que não lhes pertencem	Acesso interdido	
---	------------------	---





2. Login

Testes	Desfecho Esperado	Pass/Fail
Login com user já registado	Login com sucesso	
Login com user ainda não registado	Login recusado	
Login com username de user já registado, mas palavra passe errada	Login recusado	




3. Criação/ Gestão de equipas

Testes	Desfecho Esperado	Pass/Fail
Criação de uma equipa	Criar Equipa com sucesso	
Criação de uma equipa sem um dos seus atributos	Criação recusada	
Criar equipa com nome de uma já na base de dados	Criação recusada	
Editar nome da equipa	Edição com sucesso	
Editar country da equipa	Edição com sucesso	
Editar imagem da equipa	Edição com sucesso	
Editar um dos campos da equipa para algo vazio	Edição recusada	




4. Criação/ Gestão de jogadores

Testes	Desfecho Esperado	Pass/Fail
Criação de um jogador	Jogador criado com sucesso e associado à equipa correta	
Criação de um jogador com os dos parâmetros vazio	Criação recusada	
Editar um dos parâmetros do jogador	Edição com sucesso	
Editar equipa de um jogador	Edição com sucesso e jogador aparecer na listagem da equipa para o qual foi mudado	






5. Criação e Gestão de jogos

Testes	Desfecho Esperado	Pass/Fail
Criar um jogo	Criação com sucesso	
Criar um jogo com a mesma equipa selecionada duas vezes	Criação recusada	
Criar um jogo com um dos parâmetros vazios	Criação recusada	





6. Criações de evento para um jogo

Testes	Desfecho Esperado	Pass/Fail
Criar evento para um jogo pretendido	Criação com sucesso	
Criar evento com um dos parâmetros vazios	Criação recusada	
Criar um evento e o tempo de jogo ser alterado com sucesso	Na listagem de jogos tempo ser alterada	



7. Visualizar detalhes de um jogo

Testes	Desfecho Esperado	Pass/Fail
Ser possível visualizar local do jogo	Visualização com sucesso	
Ser possível visualizar tempo de jogo	Visualização com sucesso	
Ser possível visualizar as equipas que se defrontam	Visualização com sucesso	
Ser possível visualizar resultado de jogo	Visualização com sucesso	
Ser possível observar os eventos do jogo por ordem cronológica de quando ocorreram	Visualização com sucesso	



8. Estatística 3. A)

Testes	Desfecho Esperado	Pass/Fail
Visualizar equipas ordenadas pelo número de jogos	Visualização com ordenamento correto com sucesso	
Visualizar equipas ordenadas pelo número de vitórias	Visualização com ordenamento correto com sucesso	
Visualizar equipas ordenadas pelo número de derrotas	Visualização com ordenamento correto com sucesso	
Visualizar equipas ordenadas pelo número de empates	Visualização com ordenamento correto com sucesso	

9. Estatística 3. B)

Testes	Desfecho Esperado	Pass/Fail
Visualizar melhor marcador	Visualização com sucesso	
Existir mais que um melhor marcador	Visualização dos melhores marcadores com sucesso	

10. Fazer importação a uma API externa

Testes	Desfecho Esperado	Pass/Fail
Equipas inseridas com sucesso	Inserção nas tabelas com sucesso	
Jogadores inseridos com sucesso	Inserção nas tabelas com sucesso	

5. Conclusão

Em suma, a partir deste projeto conseguimos perceber como é que são feitas as ligações entre uma base de dados, um servidor de backend e o frontend. Para além disso, como foi um tema cativante e fácil de entender, o desenvolvimento deste webService tornou-se em algo mais interessante. Apenas tivemos dificuldades em criar uma sessão para cliente, de modo a evitar que um user não registado conseguisse ter acesso a funcionalidade de admin, por exemplo. Para além disso, de modo a tentar tornar o projeto desenvolvido em algo mais “real” e que se parecesse mais com um website verdadeiro, aprendemos e implementámos algum css em adição ao html.