

1 2



9 0

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

Relatório Trabalho Prático 2
scoreDei



Sistemas Distribuídos
2021/2022

Marco da Cruz Pais, 2019218981
Simão Carvalho Monteiro, 2019215412

Índice

Introdução	3
Arquitetura	3
Arquitetura Geral	3
Base de dados	4
Detalhes de Implementação Backend	5
Entities	6
Controllers	6
Services	6
Repositories	6
Detalhes extra de implementação	6
Distribuição de tarefas	7
Descrição dos testes feitos à plataforma	8
Conclusão	9
Referências	9

Introdução

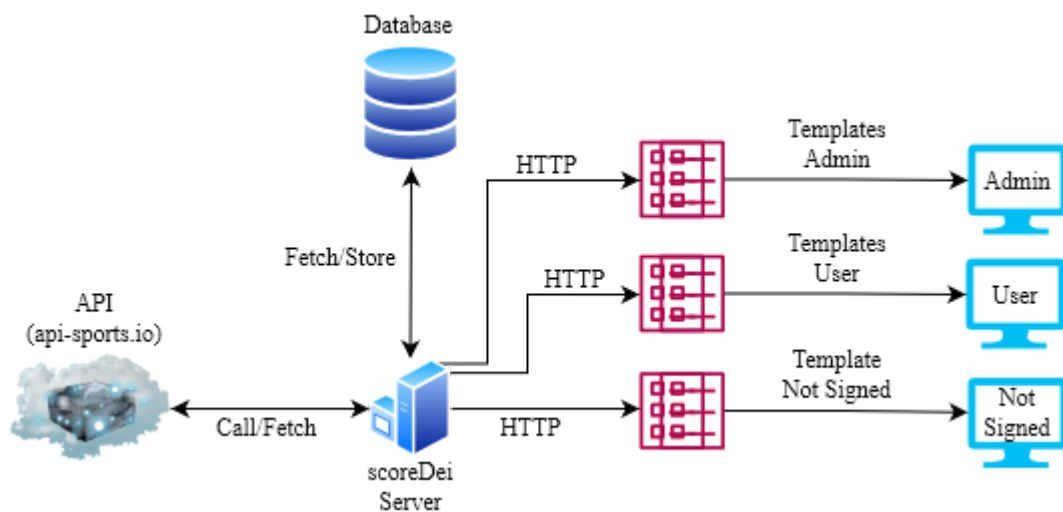
No âmbito da cadeira de Sistemas Distribuídos foi proposta a realização de uma plataforma para consulta de resultados desportivos em tempo real.

Neste projeto foram utilizadas tecnologias como Spring Boot para construir a camada de dados, Thymeleaf e Bootstrap para a construção do frontend. Foi, também, utilizado Java Persistence Application Programming Interface (JPA) para a construção do sistema de banco de dados de objetos relacionais e tecnologia REST para obtenção de dados e população da base de dados local, através do serviço externo (api-sports.io).

Arquitetura

Geral da Aplicação

Na figura abaixo está representada a estrutura e arquitetura da nossa Web App scoreDei.



A nossa aplicação tem 4 estruturas muito importantes, das quais:

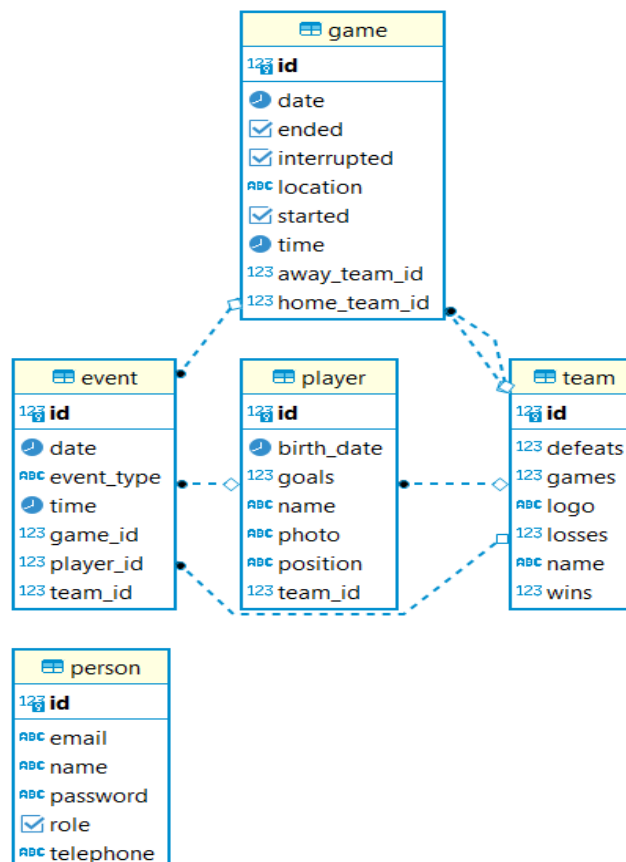
- scoreDei Server - Este servidor é onde está armazenada toda a aplicação desde os templates frontend até ao código backend e é, também, onde todos os pedidos feitos pelos diferentes utilizadores irão ser respondidos.
- API (api-sports.io) - A API é um serviço externo que, utilizando tecnologia REST, permite-nos obter informação sobre jogadores e

equipas e assim preencher a nossa base de dados mais rapidamente com informação já verificada como fidedigna.

- Database - Nesta estrutura são guardadas todas as informações como os diferentes utilizadores, jogadores, equipas, jogos e eventos. É fulcral neste projeto pois permite-nos obter informação de maneira fácil e em caso de falha ter um backup de todas as informações guardadas até esse momento.
- Templates Frontend - Permite-nos renderizar para os diferentes utilizadores toda a informação necessária para estes utilizarem, de forma mais fácil e intuitiva, a aplicação. Os templates, também, nos ajudaram a dividir os diferentes utilizadores na renderização das diferentes funções a que cada um tem permissões.

Base de Dados

Na figura abaixo está representado o diagrama de Entidade-Relacionamento da estrutura de dados da nossa Web App scoreDei.



A base de dados é constituída por 4 tabelas:

- **Person** - Esta tabela armazena toda a informação dos utilizadores como o seu nome, email, telefone, password e o grupo de permissões em que se insere.
- **Game** - Guarda toda a informação de um jogo como data e hora, equipa visitada, equipa visitante, localização e outros fatores que verificam se o jogo já começou, terminou ou se está interrompido. Esta tabela tem relações com as tabelas da Team e Event. A partir da tabela Team são associadas tanto a equipa visitada, como a visitante. A partir da tabela Event associa-se uma lista de eventos.
- **Team** - Esta tabela guarda toda a informação de uma equipa como o seu logo, nome, número de vitórias, derrotas e empates. Tem relação com as tabelas de Player, Event e Game. Esta tem uma lista de jogadores associados através da tabela Player, e, também, é associada a eventos e a jogos.
- **Player** - Guarda toda a informação de um jogador como nome, data de nascimento, posição, golos, foto e a sua equipa. Esta tabela tem relações com as tabelas Event e Team pois os eventos estão associados a um jogador e um jogador tem sempre uma equipa, respetivamente.
- **Event** - A tabela de eventos guarda todas as informações de eventos em todos os jogos como o tipo de evento, a data e o tempo, o jogador, a equipa e em que jogo aconteceu esse evento. Esta tabela tem relações com as três últimas tabelas mencionadas pois cada evento tem uma equipa, um jogo e um jogador associado.

Detalhes de Implementação do Backend

Entities

Foram criadas 5 entidades, que estão representadas na figura da base de dados apresentada acima e estas são Person, Player, Team, Game e Event. A anotação @Entity permite associar um objeto/classe java a uma tabela de base de dados de forma automática. Mais tarde, a criação desses objetos e a gravação dos mesmos na base de dados torna-se muito mais fácil e eficiente.

Controllers

No total criamos 6 controladores entre estes, o PersonController, PlayerController, TeamController, GameController, EventController e ViewsController. Estes controllers em combinação com métodos de manipulação baseados na anotação @RequestMapping são usados para marcar uma classe como um path da web. Todos os controllers, à exceção do último, servem para criar os paths relacionados com as diferentes entidades e as suas funções. Já o último controller serve para mapear os paths daquelas funções mais gerais da aplicação web.

Services

Tal como nas Entities foram criados 5 Services entre os quais, PersonService, PlayerService, TeamService, GameService e EventService. Estes services servem para aceder a funções que contactam com o repositório Crud para obter informação da base de dados. Cada Entity tem um service próprio para lidar com as funcionalidades e dados de cada um.

Repositories

Tal como nas Entities e nos Services foram criados 5 Repositories entre os quais, PersonRepository, PlayerRepository, TeamRepository, GameRepository e EventRepository. Estes Repositories implementam um CrudContainer para aceder à base de dados e neles são criadas queries JPQL associadas a funções para se tornar possível aceder à base de dados através dessas queries. Essas funções irão ser chamadas através dos Services respetivos.

Detalhes extra de Implementação

Para a construção dos diferentes templates do frontend, foi utilizado bootstrap para tornar o design mais elegante, para além de alguns atributos de CSS.

Foram, também, criadas listagens ordenáveis mais intuitivas de forma ao utilizador se sentir mais confortável na sua utilização.

Por fim, criou-se um logo básico para representar esta aplicação.

Distribuição de tarefas pelos elementos do grupo

Primeiramente, o trabalho foi dividido em 6 grandes tarefas:

- Backend, que foi dividido em:
 - Entidades;
 - Controladores;
 - Serviços;
 - Repositórios;
- Frontend, que foi dividido em:
 - Templates;
 - Bootstrap;
 - CSS;
- Dados da API;
- Tratamento de Exceções;
- Qualidade de Código;
- Presente relatório.

O primeiro ponto foi dividido de forma igual pelos dois elementos, sendo que cada um implementou entidades, controladores, serviços e repositórios.

O segundo ponto foi, também, dividido de forma igual devido à quantidade de templates a construir e devido à utilização de bootstrap e css.

O terceiro ponto foi feito em conjunto, pois foi aquele que se tornou mais trabalhoso devido à leitura da documentação e, também, devido à implementação.

Um de nós fez o ponto quatro e o outro fez o ponto cinco.

Por fim, o presente relatório foi feito por ambos.

Descrição dos testes feitos à plataforma

Nº Teste	Descrição	Resultado
1	Login e Logout	Passou
2	Login sem conta	Passou
3	Carregamento de dados através da API (Api-Sports)	Passou
4	Criação e eliminação de Utilizadores	Passou
5	Criação e eliminação de Equipas	Passou
6	Criação e eliminação de Jogadores	Passou
7	Criação e eliminação de Jogos	Passou
8	Edição de Utilizadores	Passou
9	Edição de Equipas	Passou
10	Edição de Jogadores	Passou
11	Edição de Jogos	Passou
12	Listagem de Utilizadores	Passou
13	Listagem de Equipas	Passou
14	Listagem de Jogadores	Passou
15	Listagem de Jogos	Passou
16	Listagens por várias ordens (Ascendente e Descendente)	Passou
17	Registo de Eventos	Passou
18	Renderização das opções necessárias consoante a role do utilizador	Passou

Conclusão

Este trabalho prático ajudou-nos a consolidar o conhecimento obtido nas aulas teóricas e práticas sobre SpringBoot, Thymeleaf, Java Persistence Application Programming Interface (JPA) e REST services.

A implementação do projeto foi um sucesso pois completamos todos os requisitos e objetivos pedidos. A construção dos diferentes requisitos exigiu de nós uma procura sobre o material relacionado com o que estava a ser desenvolvido e, ainda, expôs de uma forma diferente os vários temas lecionados nas aulas o que, por sua vez, ajudou estendeu a nossa compreensão sobre os mesmos em relação ao dado nas diferentes aulas.

Referências

- [API-Sports - Documentation Football](#)
- [Learn Spring Boot | Baeldung](#)
- [Stack Overflow](#)
- [Slides Teóricos de Sistemas Distribuídos](#)