

Relatório do segundo trabalho de Tecnologias de Programação Web

1. Índice

Índice	2
Introdução	3
Aplicação feita em Angular	4
Estruturação do projeto	4
Aspetos importantes ao navegar na página	4
REST API	7
Alterações em relação ao trabalho anterior	7
Considerações finais	8

2. Introdução

Neste trabalho, foi-nos proposto implementar uma página web dinâmica, criada com as ferramentas introduzidas na disciplina de Tecnologias e Programação Web. Apesar do tema escolhido ter sido o mesmo do primeiro trabalho (página com informações sobre uma liga de football), foi desafiante alterar toda a estrutura do website já criado para desta feita usar uma REST API em django e todo o front end ser desenvolvido em Angular.

Segue-se desta forma, o relatório explicativo daquilo que, de uma forma geral, criamos neste trabalho.

3. Aplicação feita em Angular

3.1. Estruturação do projeto

Para construirmos as páginas deste website, usufruirmos dos componentes e serviços, disponibilizados pelo Angular. Desta forma, criamos componentes para todas as páginas de operações de edição da base de dados e para as de visualização destes.

Nas de obtenção de dados, pelo facto das páginas de edição e adição de novos dados de uma determinada entidade serem similares, para cada entidade foi usado o mesmo componente para editar, adicionar e remover entidades à base de dados.

Já no caso dos serviços criados, o que recebe principal destaque é o de comunicação com a REST API, que no fundo apresenta o conjunto de métodos para realizar as várias operações de CRUD que a REST API possibilita. Por fim, foram criados mais dois serviços, um para gestão de erros do servido atrás referido e outro para a leitura e conversão de ficheiros para base64, de forma a permitir o upload de imagens para a base de dados.


3.2. Aspetos importantes ao navegar na página

Apesar da estrutura do site ser similar ao do projeto passado, é importante relembrar que esta plataforma providência duas formas de navegação: uma como utilizador anónimo e outra como administrador da plataforma.


O utilizador anónimo tem a possibilidade de visualizar as equipas existentes (página das equipas), os jogadores, estádio e informação geral de cada uma (quando é seleccionada uma equipa na página das equipas) e de analisar todos os jogos que já houve (na página dos jogos), onde é possível verificar o resultado, o estádio onde foi jogado, as equipas que se defrontaram, as estatísticas do jogo e os eventos deste:

Data de fundação
1904-01-01


Estádio
Estádio do Sport Lisboa e Benfica (Luz)




Jogadores




ODYSSEAS
GUARDA-REDES




RUBEN DIAS
GUARDA-REDES




ANDRE ALMEIDA
DEFESA




FERRO
DEFESA



GRIMALDO
DEFESA



JARDEL
DEFESA




FC Porto
CASA


10 2019-11-05

4:2

MAIS INFO



Sporting
FORA




Benfica
CASA

10 2019-11-13

6:1

MAIS INFO




Braga
FORA

Já o administrador da plataforma tem o poder de adicionar, editar e remover dados da plataforma:

EQUIPAS
JOGOS
CONTA

Adicionar estádio
Adicionar equipa
Adicionar jogador
Adicionar jogo
Terminar sessão




FC Porto
CASA

10 2019-11-05

4:2

MAIS INFO



Sporting
FORA

ADICIONAR EVENTOS
EDITAR
ADICIONAR JOGADORES

REMOVER JOGADOR

Nome do jogador: *

Ruben Dias

Data de nascimento do jogador: *

01/01/1990

Foto do jogador:

Browse...

No file selected.

Alcunha do jogador:

Ruben Dias

Posição do jogador: *

Guarda-redes

Equipa do Jogador: *

Benfica

EDITAR JOGADOR

4. REST API

4.1. Alterações em relação ao trabalho anterior

Apesar do trabalho anterior ter sido desenvolvido em django, tal como a REST API, foram necessárias algumas alterações importantes para a implementar. Desta forma, foram criados novos urls para novos pedidos, necessários à página acima descrita e os urls já usados foram mantidos, mas desta vez, para lhes aceder, é necessário usar métodos GET (para obtenção de dados), POST (para adição de novos dados), PUT (para edição de dados já existentes) ou DELETE (para as operações de remoção). Já as queries que tinham sido criadas no ficheiro queries.py foram mantidas.

Também é importante evidenciar que foi usado django authentication, com tokens gerados aleatoriamente, de forma a permitir a autenticação dos administradores de forma segura.

Quanto aos modelos usados, estes foram mantidos em relação ao trabalho anterior sem qualquer alteração.

Tal como no trabalho anterior, neste usamos serializers, de forma a facilitar quer a validação dos parâmetros obtidos nos pedidos, quer a inserção de dados na base de dados, quer no retorno destes.

5. Considerações finais

Apesar de ter sido um projeto bastante trabalhoso, podemos nele fortalecer o nosso conhecimento em relação à criação de websites dinâmicos e de serviços RESTful. Apesar de termos usado as ferramentas django e angular para a criação do trabalho, foi adquirida prática suficiente para criar este tipo de plataformas e serviços em qualquer tipo de ferramenta que existe no mercado atual.

Credenciais de administrador:

- Username: *escaleira*
- Password: *olaadeus*

Url do website: <https://footballtpw.herokuapp.com>

Url da REST API: <https://escaleira.pythonanywhere.com/>

Repositório do projeto: https://github.com/oEscal/tpw_project_2