2018/2019
Licenciatura em Engenharia Informática
Programação para a Internet

Relatório Técnico – Programação para a Internet

Docente: Pedro Albuquerque Santos

Alunos:

André Reis n.º 170221035

Bruno Alves n.º 170221041

Funcionamento geral

Esta aplicação, com o nome de “Tournament Invasion”, tem a finalidade de gerir os torneios dos seus utilizadores. Portanto, cada torneio tem o seu tipo (equipas ou individual), privacidade (publico ou privado), data de início, progresso e as suas respetivas equipas/jogadores e jogos.

Singlepage Aplication

Para criar uma aplicação com apenas uma página, adotamos um sistema de “section”. Com o auxilio da função “openSection(section)”, é possível através dos botões do site, trocar entre secções, parecendo que troca de página. Assim temos uma barra de navegação no topo da página e um footer fixos, onde todas as “sections” são criadas para aparecer no meio do ecrã.

Frameworks e Bibliotecas

Para o desenvolvimento desta aplicação utilizámos como frameworks o Vuejs e o bootstrap.

O Vuejs foi utilizado como auxílio para visualização das forms de criação e remoção de objetos, dado que estas aparecem sobrepostas ao resto da pagina.

O bootstrap foi utilizado sobretudo pensado na facilitação que este nos dispões em relação aos estilos e posicionamentos dos objetos.

Utilizadores registados e não registados

Qualquer pessoa pode aceder livremente ao gestor de torneio mesmo sem estar registados. Qualquer utilizador não registado poderá visualizar toda a informação do ranking de jogadores e de torneios e seus respetivos jogadores/equipas e jogos decorridos ou a decorrer.

Contudo, um utilizador poderá registar-se para ter mais permissões, este poderá não só visualizar, mas também criar e apagar torneios, jogos, jogadores/equipas dos torneios.

Um utilizador registado tem a possibilidade de adicionar um jogador a um torneio que não está registado, no entanto a única informação que terão do mesmo será apenas o nome de jogador. Este “jogadores” ficarão guardados na base de dados na tabela “jogadores” juntamente com alguns utilizadores que estejam inscritos em torneios, porém serão sempre vistos como utilizadores não registados e não poderão visualizar os torneios onde estão inscritos caso estes sejam privados.

Controlo de erros

O gestor de torneios possui um controlo de erros bastante rigoroso.

- Mensagem de erro quando as credenciais do login não estão corretas;
- É impedido a um utilizador não registado de fazer atividades para além de visualizar a informação que lhe é permitida:
    - Iniciar sessão para ver a aba “Meus torneios”;
    - Iniciar sessão para criar um torneio e/ou jogo;
- Mensagens de erro quando todos os parâmetros das “forms” não estão preenchidos;
- Erro quando se tenta adicionar um jogador/equipa que já existe no torneio;
- Não permite a criação de mais do que um jogo entre duas equipas no mesmo dia e à mesma hora;
- Não permite a criação de um jogo de uma equipa contra ela mesma.

Classes

- User – Guarda os dados de um utilizador
- Tournament – representa um torneio
- Modality – representa uma modalidade, esta está relacionada ao torneios
- Player – representa um jogador
- Team – representa uma equipa
- Game – representa um jogo, este está sempre relacionado a um torneio
- Information – responsável pela destão de todos os dados e funções da aplicação.

Leitura da base de dados

Todos os dados presentes na base de dados são lidos e escritos como foi lecionado nas aulas, onde no ficheiro “request-handler.js” temos todas as funções com esta finalidade.

Estes dados são posteriormente interpretados na aplicação em si a partir do texto json enviado pelas funções deste script.

O script envia as informações em formato json através dos seguintes atalhos:

- **GET “/modalities” –** retorna as modalidades presentes na base de dados;

- **GET “/users” –** retorna os utilizadores guardados na base de dados;
- **POST “/users” –** cria e guarda um utilizador na base de dados;
- **PUT “/users/:id” –** edita um utilizador com o id em parâmetro;

- **GET “/tournaments” –** retorna os torneios guardados na base de dados;
- **POST “/tournaments” –** cria e guarda um torneio na base de dados;
- **PUT “/tournaments/:id” –** edita um torneio com o id em parâmetro;
- **DELETE “/tournaments/:id” –** apaga um torneio com o id em parâmetro;

- **GET “/tournaments/:id/players” –** retorna os jogadores de um determinado torneio com id em parametro;

- **POST “/tournaments/:id/players” –** cria e guarda um jogador num torneio com id enviado em parâmetro;
- **POST “/players” –** edita um jogador de um torneio com o id em parâmetro;
- **DELETE “/tournaments/:idTor/players/:idPlayer” –** remove um jogador com o id em parâmetro de um determinado torneio com id enviado também em parâmetro;

- **GET “/tournaments/:id/teams”;**
- **POST “/tournaments/:id/teams”;**
- **POST “/teams”;**
- **DELETE “/tournaments/:idTor/teams/:idTeam”;**

- **GET “/tournaments/:idTor/games”**
- **POST “/tournaments/:idTor/games”**
- **DELETE “/tournaments/:idTor/games/:idGames”**