

Desenvolvimento de Aplicações Web

Trabalho Prático

-

Mestrado em Engenharia Informática
Universidade do Minho
Relatório

Grupo

PG41080	João Ribeiro Imperadeiro
PG41081	José Alberto Martins Boticas
PG41091	Nelson José Dias Teixeira

28 de Janeiro de 2020

Conteúdo

1	Introdução	2
2	A plataforma	3
2.1	Informatics Social Network (ISN)	3
2.2	Funcionalidades	3
2.2.1	Autenticação	3
2.2.2	Publicações	3
2.2.3	Grupos	3
2.2.4	Eventos	3
2.2.5	Informação de perfil	4
2.2.6	Amigos	4
2.2.7	Notificações	4
2.3	Arranque do sistema	4
3	Conclusão	5
4	Webgrafia	6
A	Token	7

Capítulo 1

Introdução

Neste trabalho prático é requerido o desenvolvimento de uma rede social de alunos para alunos onde possam partilhar materiais, discutir datas, combinar eventos, entre outros.

Os requisitos mínimos para este projeto são uma plataforma que:

- faça a gestão de utilizadores com autenticação;
- faça distinção do que é público e privado;
- permita aos utilizadores publicar conteúdos;
- permita partilhar ficheiros ou fotos;

É necessário ainda assegurar que só utilizadores autenticados poderão publicar ou comentar publicações de outros.

Neste projeto foram cumpridos estes requisitos e ainda se adicionaram mais alguns.

Capítulo 2

A plataforma

2.1 Informatics Social Network (ISN)

O sistema é dividido em duas partes principais: servidor de API e servidor de interface.

O servidor de API trata de toda a interação com a base de dados, servindo de intermediário entre a mesma e o servidor de interface.

O servidor de interface trata da apresentação das informações ao utilizador, sendo que, para isso, utiliza ficheiros `pug`, que geram `HTML`, aliados a `CSS`. As informações são obtidas recorrendo ao servidor de API.

2.2 Funcionalidades

De seguida, apresentam-se as principais funcionalidades do sistema.

2.2.1 Autenticação

De forma a poder usufruir da plataforma, um utilizador terá de se autenticar. Para isso, usará um email e uma *password*, caso já se encontre registado. Caso contrário, deverá registar-se, indicando os dados pedidos pela rede social.

Aquando de um novo registo, as informações respetivas são enviadas do servidor de interface para o servidor de API, com este último a ser responsável por guardar as mesmas na base de dados.

2.2.2 Publicações

Um utilizador pode publicar conteúdos. Assim, poderá escrever uma mensagem, à sua escolha, e ainda adicionar ficheiros, no número que entender. Para além disso, pode ainda escolher se essa mesma publicação poderá ser vista por qualquer pessoa, apenas pelos seus amigos ou pelos membros de um certo grupo.

Uma publicação é capaz de identificar e guardar as tags utilizadas na mesma, iniciadas por `"`.

2.2.3 Grupos

Um grupo pode ser criado por qualquer utilizador e pode ser público (qualquer pessoa pode aceder livremente ao mesmo e aos seus conteúdos) ou privado (apenas os seus membros podem aceder aos conteúdos partilhados no mesmo).

A página de um grupo apresenta três abas: as publicações do mesmo; os seus membros; os ficheiros partilhados em publicações no mesmo.

2.2.4 Eventos

Um evento pode ser criado por qualquer utilizador.

A página de um evento apresenta duas abas: as informações do mesmo (título e data, por exemplo); os seus participantes.

2.2.5 Informação de perfil

É possível visitar a página do perfil de um qualquer utilizador. Nesta são exibidas todas as informações desse utilizador, como o seu nome, email, data de nascimento, biografia, amigos e publicações. É disponibilizado um botão para adicionar utilizadores que ainda não sejam amigos e remover outros que já sejam. Caso esta página seja a do próprio utilizador, é ainda disponibilizado um botão "Editar". Este redireciona a mesma para uma outra página que permite editar as informações.

2.2.6 Amigos

Um utilizador pode adicionar e remover amigos, sendo que uma adição fica dependente da aceitação do outro utilizador envolvido.

É disponibilizada uma página com os pedidos de amizade que um utilizador recebe, na qual o mesmo pode aceitá-los ou recusá-los.

2.2.7 Notificações

Um utilizador receberá notificações sempre que:

- um amigo faz uma publicação
- é feita uma publicação num grupo ao qual o utilizador pertence
- um outro utilizador se junta a um grupo no qual o utilizador é membro
- um outro utilizador se junta a um evento no qual o utilizador participará
- um outro utilizador aceita um pedido de amizade do utilizador

2.3 Arranque do sistema

Vejamos como colocar a plataforma em funcionamento:

1. Executa-se o servidor de API, fazendo `npm i` seguido de `npm start`, na pasta `api/`.
2. Faz-se um pedido POST, colocando o conteúdo do ficheiro `rawDatabase.json` no *Body*, para a rota `http://localhost:5000/api/backup?token=eyJhb...` (ver token em anexo)
3. De seguida, executa-se o *script* `databaseFix.js`, na pasta principal, que estabelece relações entre as entidades adicionadas no ponto anterior. De notar que deverá ser necessário instalar o *package* `mongoose`.
4. Executa-se o servidor de interface, fazendo `npm i` seguido de `npm start`, na pasta `interface/`.
5. Por fim, num *browser*, acede-se a `http://localhost:7000/` e utiliza-se a plataforma.

Capítulo 3

Conclusão

Este trabalho permitiu a aplicação dos conteúdos lecionados ao longo da UC, bem como o aprofundamento e alargamento dos conhecimentos relativos ao desenvolvimento de aplicações web.

Por um lado, tivemos a oportunidade de implementar um servidor de API, o qual serve de intermediário para as operações efetuadas sobre a base de dados.

Por outro lado, desenvolvemos um servidor de interface, o qual responde aos pedidos do cliente, comunicando com o servidor de API sempre que necessário.

Em suma, construímos uma aplicação web, passando por todos os seus constituintes, desde o *back-end* ao *front-end*.

Capítulo 4

Webgrafia

- *W3*:
<https://w3schools.com>
- Documentação - *Bootstrap*:
<https://getbootstrap.com/docs/4.4/getting-started/introduction/>
- Documentação - *Express*:
<https://expressjs.com/en/5x/api.html>
- *Pug*:
<https://pugjs.org/api/getting-started.html>
- Documentação - *MongoDB*:
<https://docs.mongodb.com/>

Apêndice A

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdWIiOiJ0b2t1biBnZXJhZG8gcGFyYSBhIGxpdmUgZGVtbyIsImhhdCI6MTU4MDIwOTkwMn0.
.YXxS90uxAuBbRYXM65Mzjes_V8kEBW-MfUVzuH7TQ5Y