



Universidade do Minho
Escola de Engenharia

myFacebook

Desenvolvimento de Aplicações Web

Mestrado Integrado em Engenharia Informática

2º Semestre

2018-2019

A74745 - Carlos José Gomes Campos

A74219 - Hugo Alves Carvalho

29 de Janeiro de 2019
Braga

Resumo

Este documento relata o projeto prático desenvolvido no âmbito da unidade curricular de **Desenvolvimento de Aplicações Web**, tendo como objetivo o desenvolvimento e implementação de um repositório digital, onde os utilizadores possam armazenar registos de ideias, fotos e outros objetos. A aplicação em si é criada através da framework **Express**. O servidor é implementado em **NodeJS**, sendo que o *frontend* é renderizado através da template engine **PUG**, e a base de dados é gerida em **MongoDB**.

Conteúdo

1	Introdução	5
2	Requisitos da Aplicação	6
3	Domínio da Aplicação	7
3.1	Backend	7
3.2	Frontend	7
4	Rotas	8
4.1	Rotas - Users	9
4.2	Rotas - Posts	9
4.3	Considerações Adicionais	9
5	Arquitetura do Sistema	10
5.1	Model	10
5.2	Controller	10
5.2.1	Posts	10
5.2.2	Posts	10
5.3	View	11
6	Resultados	12
6.1	Login	12
6.2	Registrar	12
6.3	Feed de Notícias	13
6.4	Perfil Público/Privado	13
6.5	Adicionar Publicação	13
6.6	Sobre Mim	14
6.7	Editar Conta	14
6.8	Importador/Exportador	15
7	Conclusões e Trabalho Futuro	16

Lista de Figuras

1	Crescimento do número de utilizadores da rede social Facebook [1]	5
2	Página de Log In	12
3	Página de Registo	12
4	Página de Feed de Notícias	13
5	Página de Perfil Público	13
6	Página para Adicionar Publicação	14
7	Página Sobre Mim	14
8	Página Editar Conta	15

1 Introdução

Com o acentuado crescimento do número de utilizadores dos mais variados serviços de internet, torna-se cada vez mais importante que o cidadão esteja presentes nestes ambientes. Neste sentido, é possível afirmar assertivamente que as redes sociais se tornaram num meio de comunicação muito importante para grande parte dos relacionamentos pessoais e até de relacionamentos profissionais. De um leque alargado de redes sociais, é essencial identificar o Facebook como o meio de comunicação mais utilizado e com mais forte impacto nos utilizadores.

Esta solução de divulgação comercial, para além de ser do agrado das empresas, uma vez que permite alcançar um maior número de utilizadores, é também bastante apreciada pelos utilizadores pois apresenta facilidade de acesso à informação, bem como de possível interação através de comentários, gostos ou partilhas.

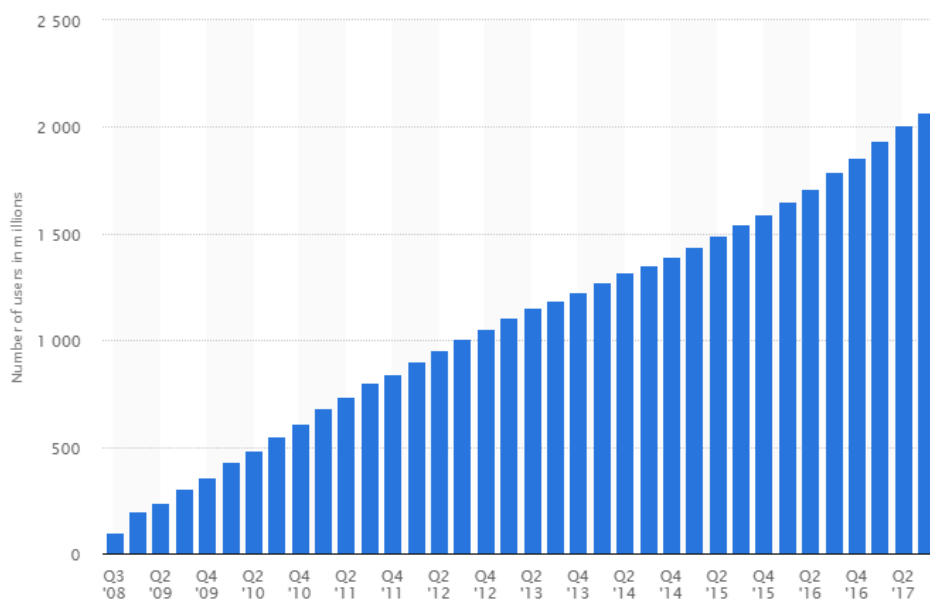


Figura 1: Crescimento do número de utilizadores da rede social Facebook [1]

Tendo em conta o crescimento da utilização de redes sociais e importância da utilização de repositórios digitais, foi pedido aos alunos, no âmbito da unidade curricular de Desenvolvimento de Aplicações Web, para desenhar e implementar um repositório digital onde os utilizadores possam armazenar registos de ideias, fotos e outros objetos. Este documento relata todo o trabalho desenvolvido, as decisões tomadas, os resultados e a sua respectiva análise.

2 Requisitos da Aplicação

1. A aplicação é destinada a apenas um utilizador, ou seja, depois de instalada, o *backend* (BE) tratará apenas um “eu” individual. No entanto, o *frontend* (FE) de acesso à parte pública deverá estar disponível na Web para qualquer utilizador.
2. Para se poder distinguir entre um acesso público e um acesso privado no FE deverá haver um módulo de autenticação transversal à aplicação, usando módulo *passport* e a implementar as autenticações: tradicional(`username/passwd`) e *facebook*.
3. A linguagem usada para o desenvolvimento da aplicação é o *JavaScript*.
4. A aplicação terá como base o eixo temporal e a disposição cronológica de eventos e conteúdos será a sua base.
5. Cada elemento granular de conteúdo deverá ficar sempre classificado com um classificador (“hash-tag”), por exemplo: desporto, família, educação, etc.
6. A classificação de conteúdos deverá materializar-se em páginas de navegação com ordenação e agrupamento semântico que serão alternativa à navegação cronológica base.
7. Cada utilizador vai ter uma lista de itens publicados. Um item pode ser constituído por bloco de texto e fotografia associada.
8. Numa fase inicial, a equipa deverá fazer um levantamento de todos os tipos de item que irá suportar e definir quais os campos de meta-informação que ficarão associados a cada item. Estes campos permitirão filtrar os itens para os mais variados fins: extrair o meu percurso desportivo, extrair o meu percurso académico, calcular a lista de locais turísticos que já visitei.
9. Um item pode ser público ou privado, por omissão deverá ser privado.
10. Se for privado apenas aparece no FE do seu criador, se for público deverá aparecer no FE público da aplicação.
11. Um item público pode ser tornado privado e vice-versa.
12. Deve ser possível incluir comentários em cada item: serão notas que ficarão associadas ao item e que permitirão qualificar melhor o item quanto ao seu âmbito e conteúdo e mesmo à sua aplicação.
13. Pretende-se que haja uma ligação às redes sociais. Por exemplo, deverá haver uma operação para partilhar um item no *facebook*.

3 Domínio da Aplicação

3.1 Backend

De modo a satisfazer os requisitos desta aplicação, usamos a tecnologia **Express** para gerar a topologia usada para implementar o *BE*. Para além dos pacotes que são instalados de origem, também foram instalados os seguintes:

- *Axios* - Este pacote é usado para transitar informação contida em ficheiros **JSON**, de uma determinada rota, para que esta seja renderizada com ficheiros **PUG**.
- *Mongoose* - Este pacote é o responsável pela ligação da aplicação a uma base de dados **MongoDB**.
- *Passport* - Este pacote é usado no âmbito da autenticação.
- *Express-session* - Este pacote é usado no que toca a sessões.
- *Session-file-store* - Este pacote é usado no que toca a sessões.
- *Uuid/V4* - Este pacote é usado no âmbito da autenticação.
- *Passport-local* - Este pacote é usado no âmbito da autenticação.
- *Passport-facebook* - Este pacote é usado no âmbito da autenticação.

3.2 Frontend

A interface pode ser acedida através da porta local 5000, e de modo a cumprir todos os requisitos definidos no capítulo anterior, foi feita uma divisão da mesma em várias páginas, e essas são:

- Sobre mim - Esta página apresenta a informação sobre o utilizador que está logado.
- Adicionar Publicação - Esta página apresenta um formulário para que se possa adicionar uma nova publicação.
- Editar Conta - Esta página permite editar a informação sobre o utilizador que está logado.
- Página de erro - Esta página apresenta o erro que é gerado caso algum de inesperado aconteça.
- Feed de notícias - Esta página apresenta todas as publicações, de todos os utilizadores, que são públicas.
- Log In - Esta página apresenta um formulário que permite a realização de Log In de utilizador na aplicação.
- Registar - Esta página apresenta um formulário que permite a realização deo registo de utilizador na aplicação.
- Feed Público - Esta página apresenta todas as publicações, de um utilizador, que são públicas.
- Feed Privado - Esta página apresenta todas as publicações, do utilizador logado, que são privadas.

4 Rotas

É de identificar as rotas que foram criadas para fazer a navegação na aplicação, tais como:

- <http://localhost:5000/> - Se o utilizador já tiver logado: redireciona para a página <http://localhost:5000/feed>. Se não tiver logado: <http://localhost:5000/login>.
- <http://localhost:5000/login> (GET) - Se o login for válido (POST): <http://localhost:5000/feed>. Se não for, volta para: <http://localhost:5000/login>. Se quiser registar: <http://localhost:5000/registar>.
- <http://localhost:5000/registar> - Após preencher o formulário, deve enviar um POST para: <http://localhost:5000/registar>. Se o registo for válido: <http://localhost:5000/login>. Se não for, volta para: <http://localhost:5000/registar>.
- <http://localhost:5000/feed> - Mostra todas as publicações públicas. Suporta também rotas como:
 - <http://localhost:5000/feed?categoria=XXXXXX&hashtag=YYYYYY>
 - <http://localhost:5000/feed?categoria=XXXXXX>
 - <http://localhost:5000/feed?hashtag=YYYYYY>
- <http://localhost:5000/perfilpublico> - Mostra todas as publicações públicas de um determinado utilizador. Suporta também rotas como:
 - <http://localhost:5000/perfilpublico?categoria=XXXXXX&hashtag=YYYYYY>
 - <http://localhost:5000/perfilpublico?categoria=XXXXXX>
 - <http://localhost:5000/perfilpublico?hashtag=YYYYYY>
- <http://localhost:5000/perfilprivado> - Mostra todas as publicações privadas do utilizador que está logado. Suporta também rotas como:
 - <http://localhost:5000/perfilprivado?categoria=XXXXXX&hashtag=YYYYYY>
 - <http://localhost:5000/perfilprivado?categoria=XXXXXX>
 - <http://localhost:5000/perfilprivado?hashtag=YYYYYY>
- <http://localhost:5000/adicionarpost> (GET) - Rota para adicionar uma publicação - faz render do ficheiro *addPost.pug*
- <http://localhost:5000/adicionarpost> (POST) - Envia no POST todo o conteúdo do post a publicar. Se correr tudo bem, direciona para a página <http://localhost:5000/feed>. Caso contrário, permanece na página <http://localhost:5000/adicionarpost>.
- <http://localhost:5000/post/:idPost> (PATCH) - Rota para alterar a privacidade de um post.
- <http://localhost:5000/auth/facebook> - Rota para autenticação com **facebook**.
- <http://localhost:5000/sobremim> (GET) - Rota para aceder à página Sobre Mim - faz render do ficheiro *aboutMe.pug* com os dados do utilizador que está logado (*req.user*).
- <http://localhost:5000/editarconta> (GET) - Rota para aceder à página Editar Conta - faz render do ficheiro *editCount.pug* com os dados do utilizador que está logado (*req.user*)
- <http://localhost:5000/importar> (GET) - Rota para fazer render da página *import.pug*, que posteriormente permite importar um ficheiro JSON para carregar publicações (POST para <http://localhost:5000/importar>)
- <http://localhost:5000/exportar> (GET) - Rota para fazer download do ficheiro em JSON

4.1 Rotas - Users

- `http://localhost:5000/users/:idUser` (GET) - Rota para obter os dados de um utilizador.
- `http://localhost:5000/users/atualiza` (POST)- Rota para atualizar a descrição do utilizador atual (`req.user`)

4.2 Rotas - Posts

- `http://localhost:5000/posts/post/:idPost` (GET) - Rota para obter os dados de uma publicação.
- `http://localhost:5000/posts/privacidade/:idPost` (GET)- Rota para atualizar a privacidade de uma publicação.
- `http://localhost:5000/posts/like/:idPost` (GET) - Rota para fazer um *like* numa publicação.
- `http://localhost:5000/posts/comentario/:idPost` (POST) - Rota para adicionar um comentário a uma publicação.

4.3 Considerações Adicionais

Para uma correta utilização das rotas e dos pedidos definidos, foi necessário implementar um mecanismo que permitisse aceder a certas rotas apenas se o utilizador tivesse logado. Em seguida apresenta-se um exemplo de como esta verificação foi feita:

```
function verificaAutenticacao(req, res, next){
  if(req.isAuthenticated()) next()
  else res.redirect("/login")
}

router.get('/perfilpublico', verificaAutenticacao, (req,res) => {
  axios.get('http://localhost:5000/api/posts/perfilpublico/'+req.user.nickname)
    .then(resposta=> res.render('userPubs', { posts: resposta.data }))
    .catch(erro => {
      console.log('Erro ao carregar dados da BD.')
      res.render('error', {error: erro, message: "Erro ao carregar dados da BD."})
    })
})
```

5 Arquitetura do Sistema

Esta aplicação foi desenvolvida segundo a arquitetura *MVC*. Esta arquitetura é desenvolvida por uma metodologia que propõe a divisão da aplicação em três camadas distintas: Modelo(*Model*), Visualização(*View*) e Controlo (*Controller*), de modo a promover uma melhor compreensão do código que incorpora a aplicação, reutilização do mesmo e o seu desenvolvimento paralelo de modo eficiente.

5.1 Model

Nesta camada, foram desenvolvidos dois modelos, um para os utilizadores, e outro para as publicações. Sendo que o utilizador tem os campos: nome, email, password, dataNascimento, genero, localidade, descricao, imagem e facebook. Sendo que este ultimo é um objeto constituído por: id, token, email e name. Todos estes campos são campos gerados na autenticação usada pelo *facebook*.

O modelo criado para as publicações, é constituído por: autor, texto, data, comentarios, categoria, likes, hashtag, ficheiro (sendo que este é o caminho para o ficheiro) e privacidade. Por sua vez, o campo comentário é um array de objetos, em que cada um é constituído por autor e o próprio comentário.

5.2 Controller

No controlador da aplicação encontram-se as consultas à base de dados e sua respetiva execução. Listam-se, em seguida, todas as consultas permitidas.

5.2.1 Posts

- `getPost` - Obtém os dados referentes a uma publicação, fornecendo o seu respetivo *_id*.
- `listaTodosUser` - Obtém todas as publicações de um determinado autor.
- `inserirPost` - Insere uma publicação na Base de Dados, recebendo como parâmetro todos os seus campos.
- `comentar` - Insere um comentário na Base de Dados, recebendo como parâmetro todos os seus campos.
- `listaPostsPublicosAutor` - Obtém todas as publicações publicas de um determinado autor.
- `listaPostsPrivadosAutor` - Obtém todas as publicações privadas de um determinado autor.
- `atualizaPrivacidade` - Faz uma atualização do valor de privacidade de uma determinada publicação (*true/false*)
- `fazerLike` - Incrementa um valor ao campo *like* de uma publicação
- `addComentario` - Acrescenta um objeto comentário ao campo *comentario* de uma publicação.

5.2.2 Posts

- `users` - Devolve uma lista de todos os utilizadores
- `post` - Devolve a informação de um utilizador pelo seu mail
- `createUser` - Cria um objeto utilizador e guarda-o na base de dados.
- `getUserByEmail` - Obtém os dados de um utilizador pelo seu email.
- `getUserByNick` - Obtém os dados de um utilizador pelo seu nickname.
- `comparePassword` - Compara duas passwords cifradas (usadas no formulário de registo)
- `alteraImagem` - Altera o *path* do campo *imagem* do utilizador
- `alteraDescricao` - Altera o campo *descricao* do utilizador

5.3 View

Para cada janela mencionada no capítulo anterior, foi criado um ficheiro **PUG** que trata de fazer a renderização da informação que é enviada pelo **BE** para o **FE**. Foram criados os ficheiros:

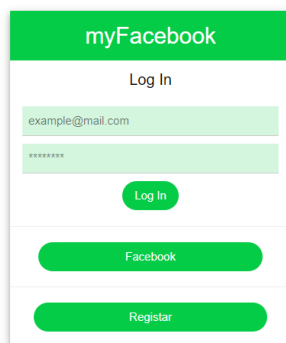
- *aboutMe.pug* - que representa a página do sobre mim.
- *addPost.pug* - que representa a página usada para adicionar uma publicação.
- *editCount.pug* - que representa a página usada para editar os dados de uma conta.
- *error.pug* - que representa a página de erro.
- *feed.pug* - que representa a página usada para mostrar todas as publicações que são públicas, de todos os utilizadores.
- *header.pug* - que é um fragmento de **HTML**, usado para navegação, que aparece em algumas páginas.
- *layout.pug* - que é um fragmento de **HTML**, dá início a todas as páginas aqui mencionadas.
- *login.pug* - que representa a página usada para fazer o Log In na aplicação.
- *register.pug* - que representa a página usada para fazer o registo na aplicação.
- *userPosts.pug* - que representa a pagina usada para mostrar todas as publicações de um utilizador, que são publicas ou privadas, consoante o pedido feito.

6 Resultados

Neste secção apresentam-se os resultados obtidos depois do desenvolvimento da aplicação, bem como as suas funcionalidades disponíveis.

6.1 Login

A página de log in permite ao utilizador fazer a sua autenticação com email e password, bem como com ligação direta ao seu Facebook. Por outro lado, caso não esteja registado e pretenda criar conta, pode aceder à secção "Registar".

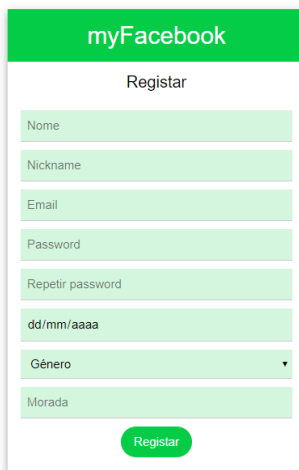


The image shows a mobile app interface for 'myFacebook'. At the top is a green header with the text 'myFacebook'. Below the header, the title 'Log In' is centered. There are two input fields: the first contains the email 'example@mail.com' and the second contains a masked password '*****'. Below these fields is a green button labeled 'Log In'. Underneath the button are two more green buttons: 'Facebook' and 'Registar'.

Figura 2: Página de Log In

6.2 Registar

A página de registo consiste num formulário simples, onde os utilizadores preenchem com os seus dados pessoais. A verificação de passwords é feita pela função descrita na secção 5.2.2. - *comparePasswords*. Ao submeter o registo válido, um novo utilizador é criado na base de dados.



The image shows a mobile app interface for 'myFacebook'. At the top is a green header with the text 'myFacebook'. Below the header, the title 'Registar' is centered. There are seven input fields: 'Nome', 'Nickname', 'Email', 'Password', 'Repetir password', 'dd/mm/aaaa', and 'Género' (which has a dropdown arrow). Below these fields is a green button labeled 'Registar'.

Figura 3: Página de Registo

6.3 Feed de Notícias

A página de Feed de Notícias mostra todas as publicações públicas de todos os utilizadores. Nesta página, é possível fazer consultas por categoria, hashtag. Por outro lado, é ainda possível fazer *like* na publicação, partilhar no facebook e adicionar comentário.

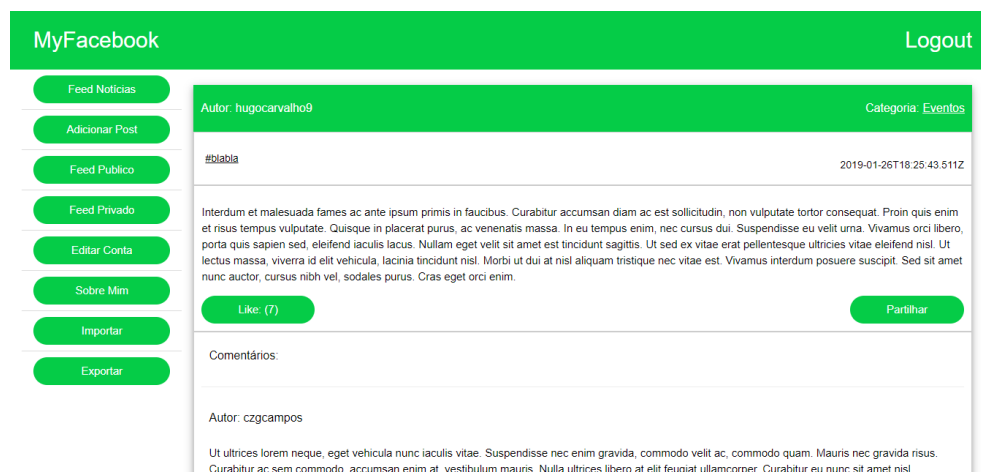


Figura 4: Página de Feed de Notícias

6.4 Perfil Público/Privado

As páginas de perfil público e privado seguem o mesmo do feed noticias, com o facto acrescido de ser possível alterar a privacidade de uma publicação, uma vez que apenas apresenta as publicações referentes ao utilizador autenticado.

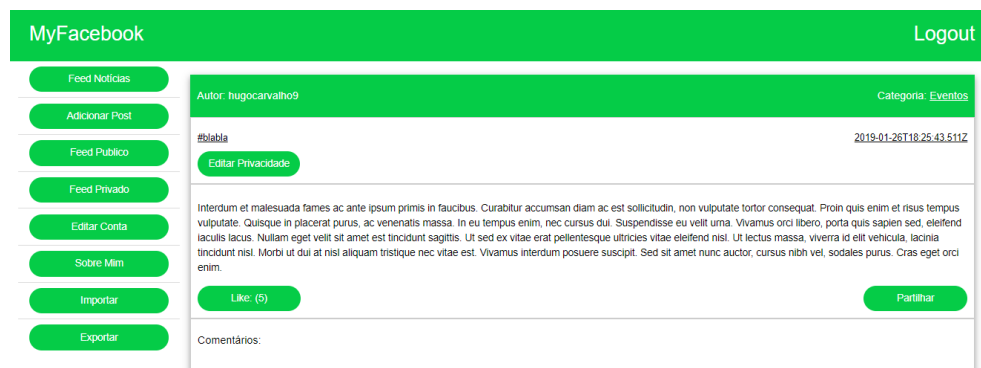


Figura 5: Página de Perfil Público

6.5 Adicionar Publicação

Na página de adicionar publicação, o utilizador pode definir a hashtag pretendida, a categoria, a sua privacidade e respectiva descrição.

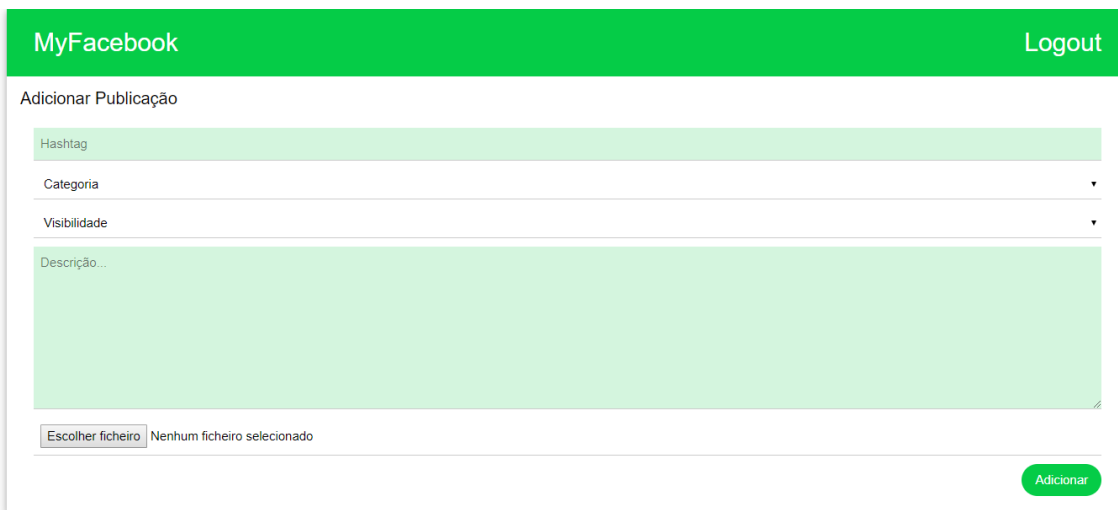


Figura 6: Página para Adicionar Publicação

6.6 Sobre Mim

Quando um utilizador pretende verificar todas as suas informações, deve aceder à página Sobre Mim. Se pretender alterar a foto ou a descrição, pode clicar em "Editar", que tem com rota o render da página de editar a conta.

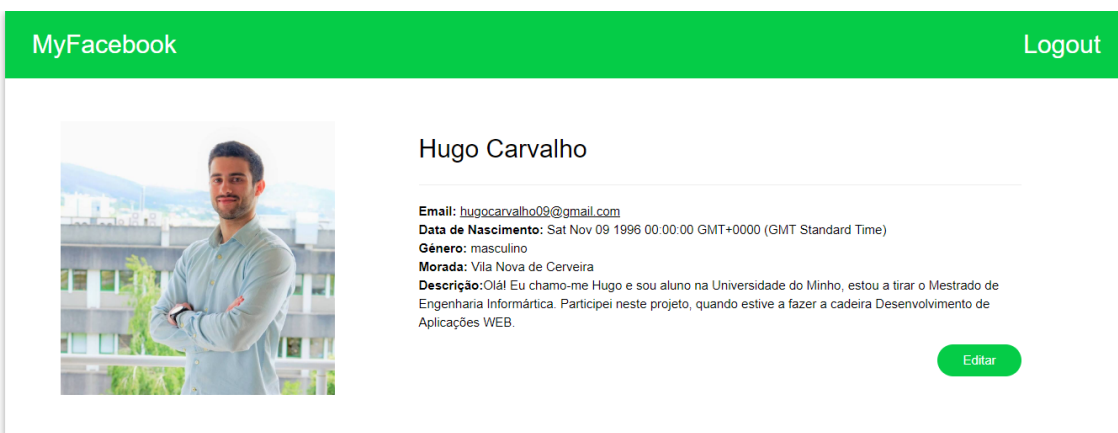


Figura 7: Página Sobre Mim

6.7 Editar Conta

Tal como descrito anteriormente, nesta secção é possível alterar a imagem de perfil do utilizador autenticado, bem como alterar a sua descrição.



Escolher ficheiro Nenhum fich... elecionado

Enviar

Hugo Carvalho

Email: hugocarvalho09@gmail.com

Data de Nascimento: Sat Nov 09 1996 00:00:00 GMT+0000 (GMT Standard Time)

Género: masculino

Localidade: Vila Nova de Cerveira

Descrição: Olá! Eu chamo-me Hugo e sou aluno na Universidade do Minho, estou a tirar o Mestrado de Engenharia Informática. Participei neste projeto, quando estive a fazer a cadeira Desenvolvimento de Aplicações WEB.

Descrição...

Figura 8: Página Editar Conta

6.8 Importador/Exportador

Para além das funcionalidades descritas, foi ainda desenvolvido um importador e um exportador JSON para carregar publicações de um utilizador.

7 Conclusões e Trabalho Futuro

Terminada a realização do projeto, é então possível concluir praticamente todas as funcionalidades pretendidas nos requisitos do trabalho prático.

Em primeiro lugar, foi definido o domínio da aplicação, bem como das tecnologias a utilizar para o correto desenvolvimento do projeto. Tal como leccionado nas aulas práticas, o grupo optou por utilizar *Node.JS* com *Pug* para a interface visual. Foram também utilizados vários *packages* necessários para a realização das funcionalidades: *axios*, *mongoose*, *passport*, *fs*, *express*, *passport*, *passport-facebook*.

Seguidamente, definiu-se quais as vistas e as rotas necessárias. Assim, decidiu-se que deviam existir as páginas: sobre mim, adicionar publicação, editar conta, feed de notícias, log in, registar, perfil público e perfil privado. Todas as rotas desenvolvidas estão detalhadas no capítulo 4 deste relatório. Por outro lado, toda a aplicação foi desenvolvida segundo a arquitetura MVC.

Como ponto menos positivo deste trabalho, o grupo considera que deveria ter explorado tecnologias como *jQuery* em situação de *likes* ou de comentário em publicações. Por outro lado, deveria também ter sido explorada a publicação de eventos ou videos na funcionalidade "Adicionar Publicação-

Em suma, o grupo ficou bastante satisfeito com o trabalho desenvolvido, uma vez que elaborou e compreendeu as principais etapas de desenvolvimento de uma aplicação Web.

Referências

- [1] <https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/>
- [2] NODE.JS - Construção de Aplicações Web - Luís Abreu, 2016, FCA.