



Universidade do Minho
Escola de Engenharia

Mestrado em Engenharia Informática
Desenvolvimento Aplicações WEB

MYFACEBOOK

Projeto Final

Equipa de desenvolvimento:

Carla Costa - PG38409

Sandra Sousa – PG38938

Braga, janeiro de 2019

Índice

<i>Índice de tabela</i>	<i>2</i>
<i>Índice de Figura.....</i>	<i>2</i>
<i>1. Introdução.....</i>	<i>4</i>
<i>2. Requisitos.....</i>	<i>5</i>
<i>3. Domínio da aplicação</i>	<i>7</i>
<i>4. Funcionalidades da aplicação.....</i>	<i>8</i>
4.1. FrontEnd	8
4.2. BackEnd.....	12
<i>5. Conclusão.....</i>	<i>14</i>

Índice de tabela

<i>Tabela 1 - API de dados de publicações.....</i>	<i>7</i>
<i>Tabela 2 - API de dados de utilizadores</i>	<i>7</i>

Índice de Figura

<i>Figura 1 – View para adicionar publicações de texto livre.....</i>	<i>8</i>
<i>Figura 2 - View para adicionar Fotos/Videos/Ficheiros</i>	<i>9</i>
<i>Figura 3 - View para adicionar locais visitados</i>	<i>9</i>
<i>Figura 4 - View para adicionar Evento</i>	<i>10</i>
<i>Figura 5 - View do perfil.....</i>	<i>10</i>

<i>Figura 6 - View de Registo (Página Inicial)</i>	11
<i>Figura 7 - View de Login (Página Inicial)</i>	11
<i>Figura 8 - View página principal</i>	12
<i>Figura 9 - View para editar dados da publicação</i>	13
<i>Figura 10 - View para editar dados do utilizador</i>	13

1. Introdução

O seguinte relatório foi elaborado no contexto da Unidade Curricular de Desenvolvimento de Aplicações Web e consiste no desenvolvimento da aplicação designada de *myFacebook*. Esta tem como propósito retratar, como o próprio nome indica, uma versão do *facebook* atual, criada por nós.

Na elaboração desta aplicação utilizamos o *Node.js* com o auxílio do *IDE Visual Studio Code* e como base de dados o *MongoDB*.

Para uma melhor compreensão da aplicação, este documento está organizado por cinco capítulos, sendo o primeiro uma breve descrição dos objetivos, das ferramentas utilizadas e da organização do documento. O segundo capítulo é referente aos requisitos da aplicação, sendo que estes foram definidos aquando a atribuição do enunciado.

No terceiro capítulo abordamos o domínio da aplicação onde estão descritos os pressupostos assumidos antes da conceção. Para quarto capítulo são descritas as funcionalidades da aplicação, desde do login, à inserção de publicações.

Por fim, no quinto capítulo relatamos o que correu bem e o que correu mal na elaboração da aplicação, assim como o que teríamos da realização deste projeto.

2. Requisitos

Os requisitos permitem-nos definir o que vai ser aplicado na aplicação e por isso definimos os seguintes como sendo os principais:

1. O utilizador deverá registar-se na aplicação através de: nome, apelido, email (o email deverá ser único para cada utilizador) e password.
2. O utilizador deverá fazer login com email e password para aceder à aplicação e às funcionalidades que ela oferece.
3. Um utilizador logado deverá ter disponível um *feed* onde encontrará uma lista de publicações públicas.
4. Um utilizador logado poderá adicionar novas publicações, fotos, vídeos, ficheiros, locais visitados e eventos.
5. Um utilizador logado poderá visualizar uma publicação.
6. Um utilizador logado poderá filtrar as publicações por id, tipo, local e hashtag.
7. Um utilizador logado poderá escrever comentários nas publicações de outros utilizadores e nas suas.
8. Um utilizador logado poderá fazer *likes* nas publicações de outros utilizadores e nas suas.
9. Um utilizador logado deverá ter disponível um perfil onde encontrará informações sobre si e as suas próprias publicações, tanto públicas como privadas.
10. Um utilizador logado poderá editar informações sobre si (nome, apelido e email).

11. Um utilizador logado apenas poderá editar uma publicação (p. ex. mudar o estado da publicação) que lhe pertença.
12. Um utilizador logado apenas poderá eliminar uma publicação que lhe pertença.

3. Domínio da aplicação

A aplicação tem por objetivo principal a criação de publicações, estas poderão ser apenas publicações com texto, publicações com fotos ou vídeos, publicações de locais visitado e de eventos.

Associado a estas publicações estão os utilizadores, os comentários e os “likes” realizados. Para uma melhor interação de todos estes elementos foram criadas as seguintes API de dados:

PUBLICAÇÕES

<i>GET api/pubs</i>	Para consultar todas as publicações registadas
<i>GET api/pubs/:pid</i>	Para consultar uma publicação específica <i>ex.: Ver publicação com o id “123”</i>
<i>GET api/pubs/local/:l</i>	Para consultar todas as publicações com a mesma localização <i>ex.: Ver todas as publicações que tenham na sua localização Braga</i>
<i>GET api/pubs/tipo/:t</i>	Para consultar todas as publicações que sejam do mesmo tipo <i>ex.: Ver todas as publicações que sejam do tipo Evento</i>
<i>GET api/pubs/palavra/:p</i>	Para consultar todas as publicações que tenham o mesmo hashtag <i>ex.: Ver todas as publicações que contenham o hashtag culinária</i>
<i>POST api/pubs</i>	Para inserir novas publicações
<i>POST api/pubs/delete</i>	Para apagar uma publicação
<i>POST api/pubs/editar</i>	Para editar uma publicação
<i>POST api/pubs/comentar</i>	Para inserir um comentário numa publicação
<i>POST api/pubs/likes</i>	Para inserir um gosto numa publicação

Tabela 1 - API de dados de publicações

UTILIZADOR

<i>GET api/user/:uid</i>	Para consultar as publicações feitas pelo utilizador logado
<i>POST api/user/editar</i>	Para editar os dados de utilizador logado

Tabela 2 - API de dados de utilizadores

4. Funcionalidades da aplicação

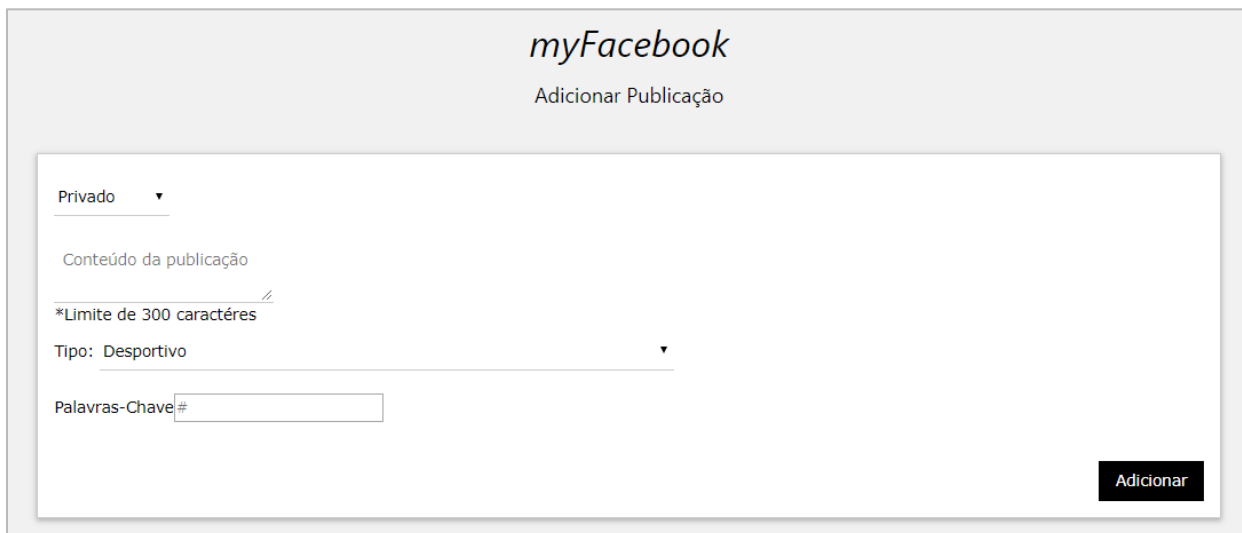
4.1. FrontEnd

A aplicação possui dois *FrontEnds*, um público designado de *feed* e um privado, designado de perfil.

Como já referido a aplicação segue o propósito de criar publicações, para isso possui um *schema* comum a todas, como é possível ver-se nas imagens abaixo, mas modifica-se conforme as *views* do tipo de publicações.

A cada publicação está associado o utilizador que a publicou através do *autorID* e do *autorNome*.

A aplicação permite fazer publicações de texto live, publicações de fotos, vídeos, ou ficheiros e ainda publicações de locais visitados e eventos. As imagens seguintes demonstram como foram contruídas as *views* de forma a permitir essas diferentes interações.



myFacebook

Adicionar Publicação

Privado ▼

Conteúdo da publicação

*Limite de 300 caracteres //

Tipo: Desportivo ▼

Palavras-Chave#

Adicionar

Figura 1 – View para adicionar publicações de texto livre

myFacebook

Adicionar nova foto/video/ficheiro

Privado ▼

Escolher ficheiro

Nenhum ficheiro selecionado

Legenda

Nome da publicação

Localização

Localização

Descrição

Descrição

Data

dd/mm/aaaa

Palavras-Chave

#

Adicionar

Figura 2 - View para adicionar Fotos/Videos/Ficheiros

myFacebook

Adicionar Localização

Privado ▼

Localização

Localização

Descrição

Descrição

Data

dd/mm/aaaa

Tipo Desportivo

▼

Adicionar Ficheiro(s)

Escolher ficheiro

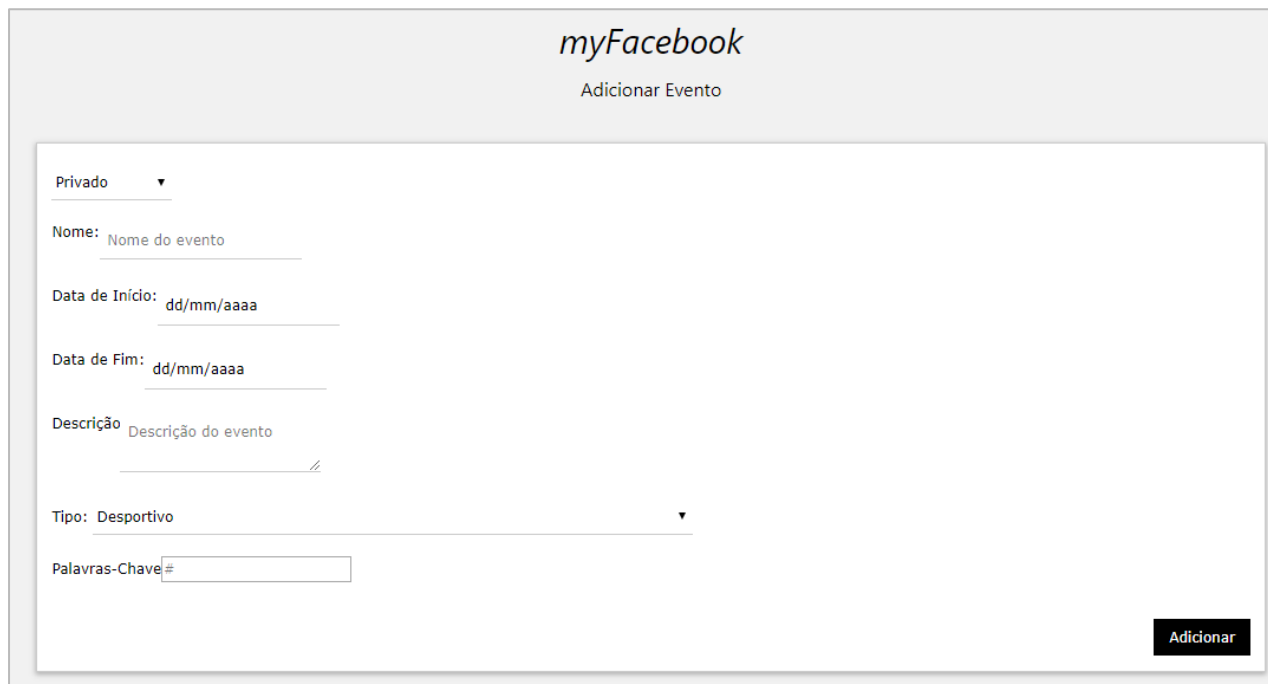
Nenhum ficheiro selecionado

Palavras-Chave

#

Adicionar

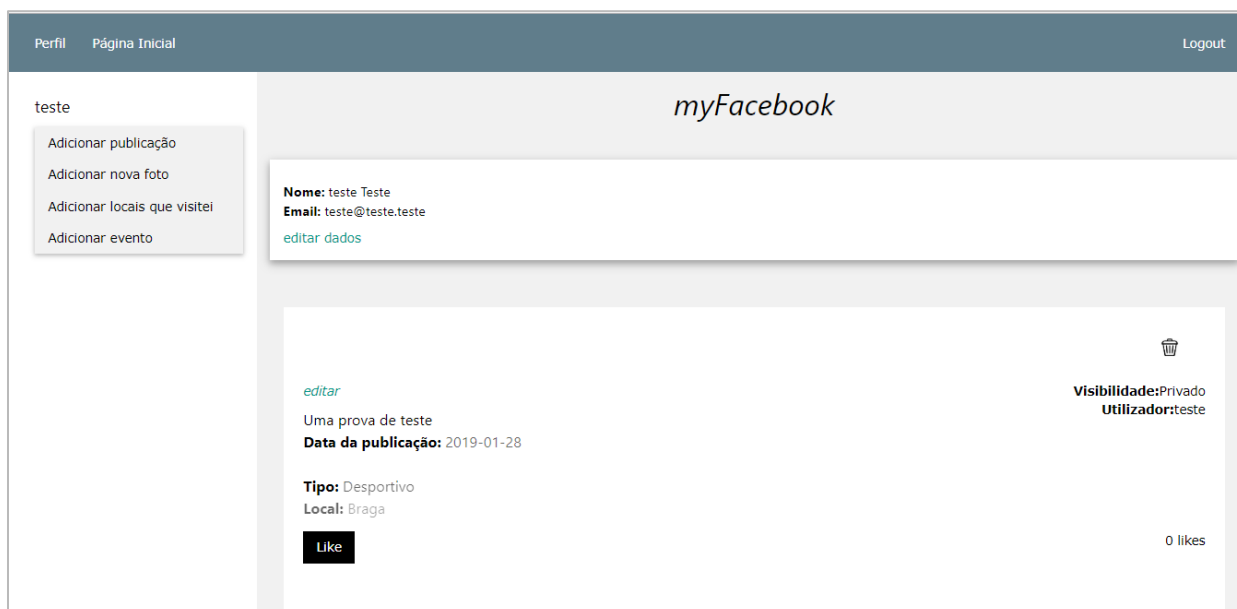
Figura 3 - View para adicionar locais visitados



The image shows a web form titled "myFacebook" with the subtitle "Adicionar Evento". The form is enclosed in a light gray border. At the top left, there is a dropdown menu set to "Privado". Below it, the "Nome:" field has a placeholder "Nome do evento". The "Data de Início:" field has a placeholder "dd/mm/aaaa". The "Data de Fim:" field also has a placeholder "dd/mm/aaaa". The "Descrição" field has a placeholder "Descrição do evento" and a small icon of two crossed lines. The "Tipo:" dropdown is set to "Desportivo". The "Palavras-Chave:" field has a placeholder "#". A black "Adicionar" button is located at the bottom right of the form.

Figura 4 - View para adicionar Evento

Também para o utilizador foi criado um *schema* e respetiva *view*, designada de perfil, juntamente com o login e o registo.



The image shows a web page titled "myFacebook" with the subtitle "View do perfil". The page has a dark blue header with "Perfil" and "Página Inicial" on the left and "Logout" on the right. On the left side, there is a sidebar with the name "teste" and four buttons: "Adicionar publicação", "Adicionar nova foto", "Adicionar locais que visitei", and "Adicionar evento". The main content area shows the user's profile information: "Nome: teste Teste", "Email: teste@teste.teste", and a link "editar dados". Below this, there is a post titled "Uma prova de teste" with the date "Data da publicação: 2019-01-28", type "Tipo: Desportivo", and location "Local: Braga". There is a "Like" button and "0 likes" text. On the right side of the post, there is a trash icon, the text "Visibilidade: Privado", and "Utilizador: teste".

Figura 5 - View do perfil

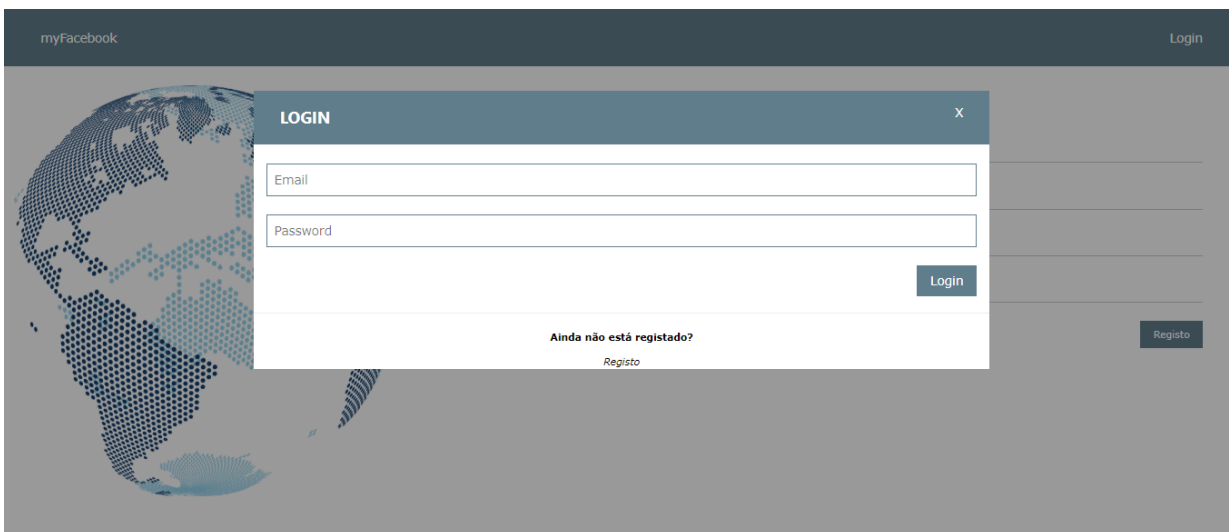
Na view acima apresentada é possível ver as publicações públicas e privadas feitas pelo utilizador teste. No perfil do utilizador é-lhe possível editar os dados das

publicações, assim como os seus próprios dados, mas estas funções pertencem ao *backend* que falaremos a seguir.



The image shows a web browser window with the URL 'myFacebook' in the address bar. The page has a dark blue header with 'myFacebook' on the left and 'Login' on the right. The main content area features a large, stylized globe on the left. On the right, there is a registration form titled 'Cria a sua conta'. The form includes input fields for 'Nome', 'Apelido', 'Email' (with the placeholder 'email@exemplo.com'), and 'Password' (with a masked password '*****'). A 'Registo' button is located at the bottom right of the form.

Figura 6 - View de Registo (Página Inicial)



The image shows a web browser window with the URL 'myFacebook' in the address bar. The page has a dark blue header with 'myFacebook' on the left and 'Login' on the right. The main content area features a large, stylized globe on the left. Overlaid on the page is a modal window titled 'LOGIN' with a close button 'X'. The modal contains input fields for 'Email' and 'Password', and a 'Login' button. Below the input fields, there is a link that says 'Ainda não está registado?' with the text 'Registo' underneath it. A 'Registo' button is also visible in the background on the right side of the page.

Figura 7 - View de Login (Página Inicial)

Optamos por colocar o registo e login na mesma página, a inicial, para assim o utilizador não ter de saltar de página em página. O registo é a primeira coisa que o utilizador vê e caso o deseje poderá criar uma conta.

Após registo poderá então iniciar sessão e é redirecionado para a página principal abaixo demonstrada.



Figura 8 - View página principal

Também é possível ao utilizador visualizar uma publicação específica, ou as publicações de um determinado tipo, ou publicações com determinada localização ou ainda com palavras-chaves comuns.

4.2. BackEnd

No *BackEnd* a aplicação trata utilizador, isto é, a esta dimensão apenas pertencem à edição de dados tanto de utilizador, que apenas ele pode modificar, como de publicações. Nas primeiras imagens podemos ver como é possível editar os dados de uma publicação.

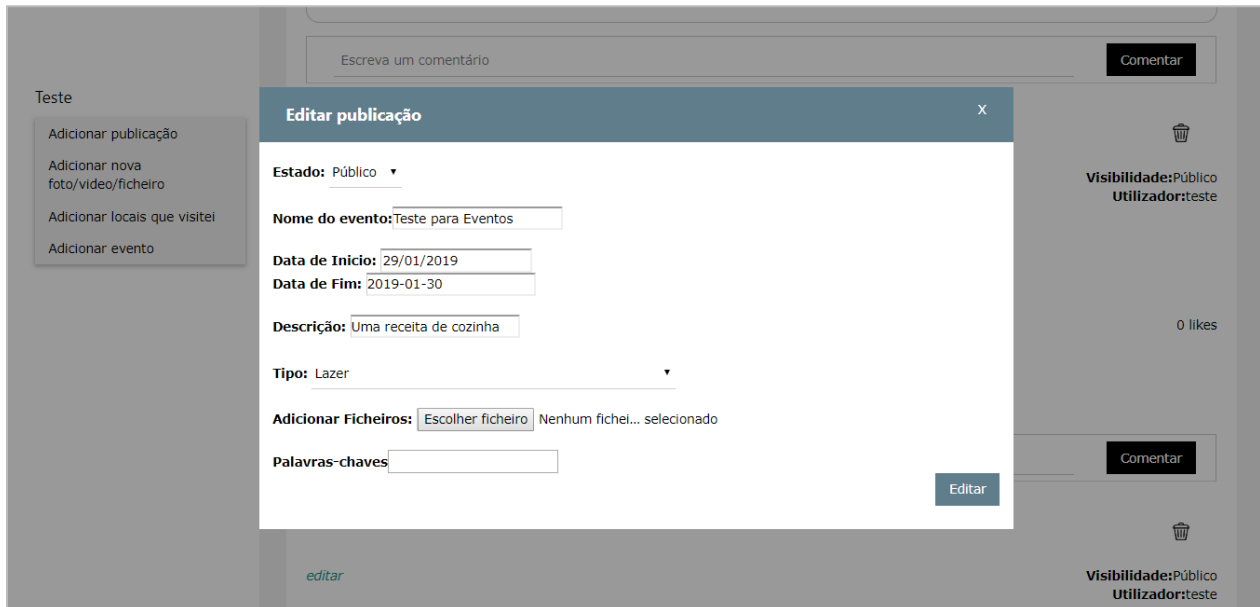


Figura 9 - View para editar dados da publicação

Nesta segunda imagem é possível ver como editamos os dados de utilizador.

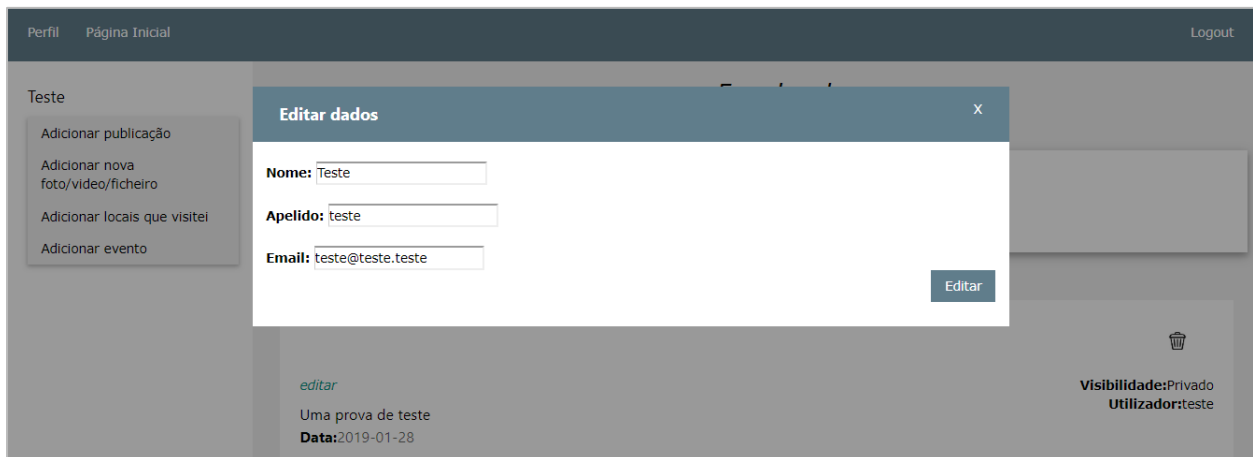


Figura 10 - View para editar dados do utilizador

5. Conclusão

O desenvolvimento deste trabalho permitiu-nos consolidar os conhecimentos adquiridos nas aulas acerca de Node.js e MongoDB.

Apesar de inicialmente ter havido algumas dificuldades nas instalações dos módulos, acreditamos que em geral os objetivos foram cumpridos, aplicamos a maioria dos requisitos solicitados e criamos os procedimentos necessários para a preparação da implementação da aplicação.

Poderiam ser feitas melhorias no que diz respeito a algumas funcionalidades, tais como a localização poderia ser automaticamente adquirida ao invés do utilizador ter de a escrever, nas hashtags ao escrever poderiam ser sugeridas algumas, entre outras pequenas coisas que acreditamos que tornariam a aplicação mais apelativa.

Posto isto, concluímos que este projeto foi um desafio para aplicar os conhecimentos das aulas, no qual consideramos que num modo geral obtivemos sucesso e ainda nos permitiu aprender mais sobre como uma aplicação pode ser estruturada de forma mais simples e organizada, já que está disposta por *models*, *controllers*, *routes* e *views*.