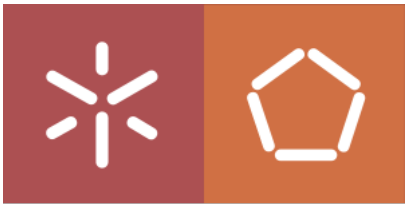

Universidade do Minho
Escola de Engenharia

Desenvolvimento de Aplicações Web

Dab Web

César Augusto, A79014
João Costeira, A78073
Mariana Fernandes,
A81728

Contents

1 Introdução

O trabalho prático da unidade curricular de desenvolvimento de aplicações Web, consiste na produção de uma plataforma onde utilizadores podem aderir a grupos e efectuar *posts* sobre a sua página pessoal ou nos respectivos grupos.
Desta forma é desenvolvida uma plataforma que permite a partilha de informações, semelhante a uma rede social.
Neste relatório encontra-se descrita a arquitectura da aplicação e as decisões tomadas de forma a implementar o sistema.

2 Descrição do Problema

Com o intuito de desenvolver uma rede social é necessário identificar as estruturas fundamentais da aplicação e respectivos utilizadores.
A seguinte tabela inclui de forma concisa os pontos fulcrais identificados no problema em questão, o desenvolvimento de uma rede social.

Descrição do Problema	
Utilizadores	Entidades que podem registar ou autenticar no sistema. Um utilizador pode efectuar/visualizar *posts*, seguir/criar grupos e efectuar pesquisas no sistema
Posts	Mensagens de texto que um utilizador pode publicar na sua página pessoal/-grupos. No seu conteúdo um utilizador pode colocar *hashtags*
Página Pessoal	Página onde um utilizador pode colocar *posts* não associados a nenhum grupo
Grupos	Página onde um conjunto de utilizadores podem seguir e efectuar *posts*. A sua visibilidade pode ser pública ou privada
Feed	Página principal onde cada utilizador pode visualizar *posts* efectuados em páginas que segue
Motor de Busca	Estrutura que permite um utilizador efectuar pesquisas sobre dados da aplicação.

Table 1: Descrição do problema

3 Descrição da Solução

Após a identificação das estruturas do problema, foi necessária a implementação de cada um dos elementos de forma a produzir a aplicação.

3.1 Utilizadores

Um dos pontos fundamentais para a consistência da aplicação é a identificação dos utilizadores.
Na solução proposta, cada um dos utilizadores possui dois identificadores únicos: o seu *email* e o seu *at* ou @.
O *email* é um identificador único universal que permite um utilizador autenticar-se na aplicação.
Por outro lado, o grupo decidiu a identificação extra de utilizadores por *ats* dentro da aplicação. A decisão foi tomada no sentido de simplificar a pesquisa de elementos,

tanto de utilizadores como grupos, semelhante a outras redes sociais existentes no mercado.
A seguinte tabela descreve o modelo de cada um dos utilizadores:

Descrição dos Utilizadores	
Name	Nome do utilizador
Email	*Email* do utilizador (identificador único), utilizado no processo de autenticação
At	Identificador único do utilizador dentro da aplicação
Password	Chave utilizada no processo de autenticação. No sistema é armazenado a sua *hash*
Following	Conjunto de elementos que o utilizador segue
Invites	Conjunto de grupos que este utilizador está convidado a aderir

Table 2: Modelação dos utilizadores

3.2 Grupos

De forma semelhante aos utilizadores, cada grupo é identificado por um *at* ou @. Desta forma a questão de não poder haver grupos com o mesmo nome foi resolvida, os grupos possuem como identificador único o seu *at* em vez do seu nome.
Complementarmente o *at* do grupo possui um segundo papel fundamental: caso o *at* do grupo for o mesmo do *at* do utilizador que criou esse grupo, estamos perante um grupo especial, a página pessoal do utilizador.
Desta forma a modelação de páginas pessoais e de grupos foi simplificada, porque ambas são simplesmente um conjunto de *post*, logo uma única estrutura pode representar tanto os grupos como páginas pessoais.

A seguinte tabela contem a modelação de cada grupo do sistema:

Descrição dos Grupos	
Name	Nome do grupo
Email	*Email* do utilizador (identificador único), utilizado no processo de autenticação
at_creator	Identificador do proprietário /administrador do grupo
At	Identificador único do grupo
Members	Conjunto de utilizadores membros do grupo
Invited	Conjunto de utilizadores convidados a aderir ao grupo
Posts	Conjunto de *posts* efectuados no grupo
Public	Variável que indica a visibilidade do grupo, público ou privado

Table 3: Modelação dos grupos

3.3 Posts

Os *posts* são a estrutura fundamental do sistema que permite a partilha de informação entre os diferentes utilizadores.
Todos os *posts* do sistema possuem uma origem, ou seja, o utilizador que emitiu esse *post* e um destino, que indica em que grupo esse *post* foi efectuado.
Caso o *post* foi partilhado no grupo do utilizador emissor (mesmo *at*), este é efectivamente publicado na sua página pessoal.
A visibilidade de cada um dos *posts* é coerente com a visibilidade da página/grupo sobre o qual foi efectuado. Deste modo só os utilizadores com permissões podem visualizar.
O corpo de cada *post* pode conter um conjunto de *hashtags* e desta forma a procura/filtro de temas de conversa ocorre.
De forma complementar, foi desenvolvida a possibilidade de partilha de ficheiros no momento de publicação dos *posts*.

A seguinte tabela descreve o modelo de cada *post*:

Descrição dos *Posts*	
Author	Autor da publicação
AuthorAt	Identificador do autor da publicação
GroupAt	Destino da publicação
Text	Corpo da publicação
HashTags	Conjunto de *hashTags* existentes no corpo da publicação
Files	Ficheiros partilhados com a publicação

Table 4: Modelação dos *posts*

3.4 Motor de Busca

Cada um dos utilizadores autenticados no sistema pode realizar pesquisas sobre os dados existentes.
De acordo com a solução proposta, tanto utilizadoras como os grupos são caracterizados por um identificador único, o seu *at*. Assim, a pesquisa por identificador permite a obtenção de resultados sem inconsistências devido à sua singularidade.

Sobre os *posts* do sistema, como no seu conteúdo é possível adicionar *hastags*, faz todo o sentido a possibilidade de efectuar procuras de acordo com esse identificador. Assim procura de *posts* associados a um certo tema de conversa pode ser realizada.

4 Arquitectura da aplicação

A aplicação desenvolvida encontra-se separada em duas camadas fundamentais, o *frontend* e o *backend*.
De seguida encontra-se descrito de forma sucinta da estrutra da aplicação.

4.1 Frontend

No *frontend* encontra-se o código da aplicação associado ao utilizador.
A comunicação entre o cliente e a *api* existente no servidor é efectuada com recurso ao módulo *axios*.
De forma a garantir a integridade da aplicação, é fundamental o bloqueio das funcionalidades a utilizadores não autenticados no sistema. Assim, com recurso ao módulo *passport-local* em conjugação com o módulo *axios*, o controlo de acesso a dados da aplicação são preservados.

De referir que foi utilizado o motor *pug* de forma a desenvolver as interfaces(*views*) da aplicação.

A seguinte tabela contem as principais *routes* da aplicação:

Descrição das Rotas	
Users	
/:at	Obter a página principal do utilizador, *posts* efectuados (*get*)
Posts	
/:groupat	Efectuar um *post* no sistema (*put*)
/:id/edit	Edição de um *post* (*post*)
/:id/delete	Apagar um *post* (*post*)
/downloadfile/	Efectuar o *download* de um ficheiro partilhado (*get*)
/:postid/uploadfile	*Upload* de um ficheiro (*post*)
Groups	
/new	Página de criar novo grupo (*get*)
/create	Criar novo grupo (*post*)
/:groupat/	Obtenção da página do grupo (*get*)
/:groupat/invite	Convite elementos para o grupo (*post*)
/:groupat/follow	Seguir grupo ou aceitar convite (*get*)
/:groupat/unfollow	Rota utilizada para deixar de seguir um grupo (*get*)
/:groupat/reject	Rejeitar convite de aderência a um grupo (*get*)
/	Página com todas as informações de grupos (grupos que criou, que segue e convites) (*get*)

Index	
/	Página inicial, Apresenta a página de *login* ou *feed* de acordo com a autenticação do utilizador (*get*)
/register	Registar um utilizador (*post*)
/login	Autenticar utilizador na aplicação (*post*)
/logout	Saída do estado autenticado (*get*)
/feed	Página com *posts* de grupos que o utilizador segue (*get*)
/search	Pesquisa na aplicação de acordo com *ats* ou *hashtags* (*get*)
Dashboard	
/	*Posts* de grupos seguidos pelo utilizador (*get*)
/posts/	Efectuar um *post* num grupo que o utilizador tem acesso (*post*)
/search	Efectuar pesquisas na aplicação (*get*)

Table 5: Descrição das principais rotas da aplicação

4.2 Backend

O *backend* é a estrutura da aplicação que representa o servidor do sistema.
De forma a responder aos pedidos dos utilizadores, foi desenvolvida uma *api*, que por sua vez comunica com uma base de dados orientada a documentos, o *mongodb*. A base de dados possui duas colecções, o *groups* e *users*, onde no qual os dados são armazenados de acordo com os modelos descritos anteriormente [3.2][3.3].

API

A seguinte tabela possui as principais funcionalidades desenvolvidas no *backend*:

Api da aplicação	
Api dos Grupos	
/follow/:groupat/:userat	Utilizador seguir um grupo (aceita o convite) (*post*)
/:groupat/invite/:invitedat/reject	Rejeitar convite para o grupo (*post*)
/unfollow/:groupat/:userat	Utilizador deixar de seguir um grupo (*delete*)
/	Criar um grupo (*post*)
/:groupat	Editar grupo (*put*)/ obter grupo(*get*)/ apagar grupo(*delete*)
/search/:groupat/:userat	Pesquisa dentro do grupo (*get*)
/usergroups/:userat	Obter informação do utilizador (*get*)
/:groupat/invite/:invitedat	Convidar utilizador para o grupo (*post*)
Api dos Utilizadores	
/	Inserir utilizador (*post*)
/:at	Pesquisar utilizador (get) / editar (put)
/email/:email	Pesquisa utilizador de acordo com o identificador *email* (*get*)
/:at/profile	Obter perfil do utilizador (*get*)
/:at/feed	Obter *feed* do utilizador (*get*)
Api dos *Posts*	
/	Inserir *post* na aplicação (*post*)
/:postid	Apagar *post* (*delete*) / alterar (*put*)
/hashtags/:hashtag	Pesquisa de *posts* por *hashtags* (*get*)
/:id	Pesquisa por identificador (*get*)
/groups/:groupat	Pesquisa de *posts* por grupo (*get*)
Api dos Ficheiros	
/addfile/:postid	Inserção de um ficheiro (*post*)
/getfile/:fileid/	Obtenção de um ficheiro (*get*)

Table 6: Descrição da *Api* da aplicação

5 Extras Desenvolvidos

A aplicação tem como objectivo principal a partilha de informações entre utilizadores. Desta forma a adição da possibilidade de partilha de ficheiros é fundamental para

melhorar a usabilidade da aplicação.
Assim, a estrutura dos *posts* foi expandida de forma a possibilitar para além da adição de *hashtags* no seu corpo, a incorporação de ficheiro na sua transmissão.

6 Dependências

Nos ficheiros *package.json* que estão em anexo tanto no *backend* como no *frontend*, encontram-se todos os módulos necessários a instalar de modo a executar a aplicação[9].
De forma a completar a instalação dos módulos como a execução da aplicação basta correr os seguintes comandos:

```
1 $npm install
2 $npm start
```

Por padrão, o *backend* executa na porta 5000 e o *frontend* na porta 7777.

7 Exemplo de Execução

De forma exemplificativa de uma execução da aplicação, de seguida encontra-se um conjunto de *secreenshots* que permite visualizar as principais funcionalidades da aplicação.

Figure 1: Autenticação/Registar na Aplicação

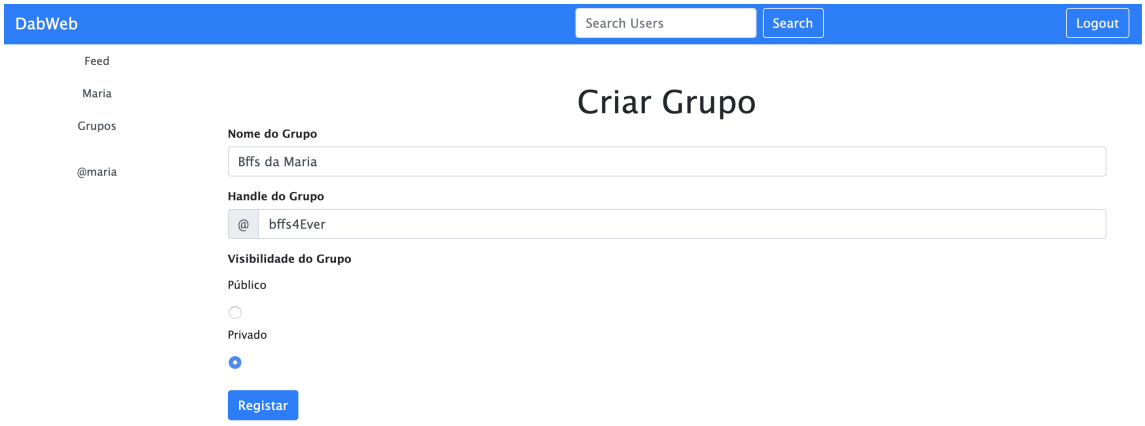

Figure 2: Criar um grupo na aplicação

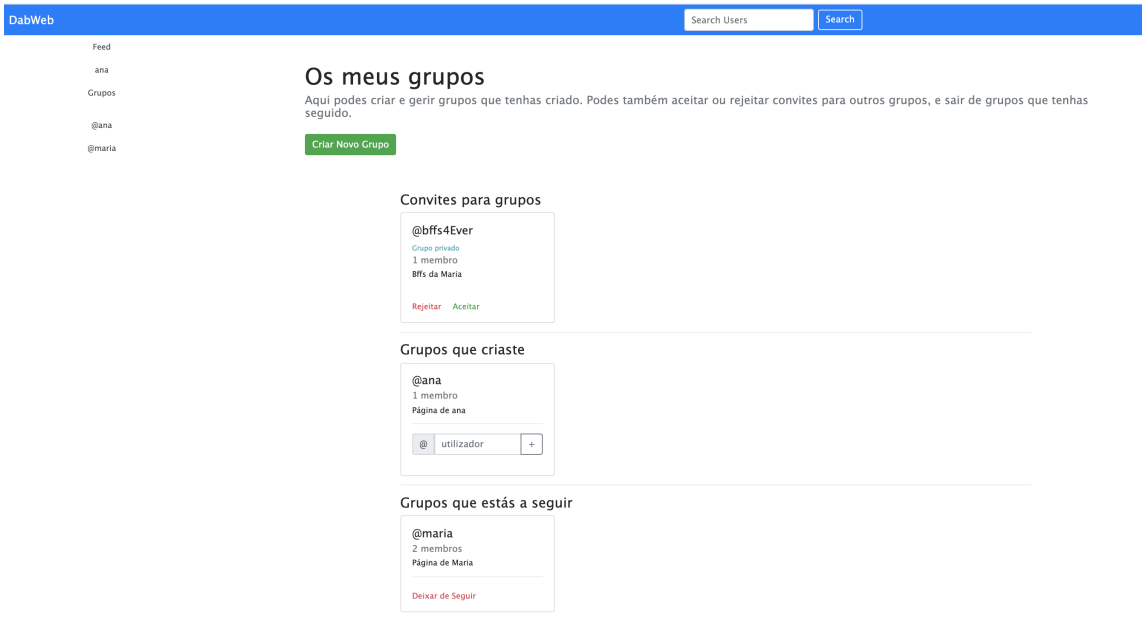

Figure 3: Página principal de grupos associados a um utilizador

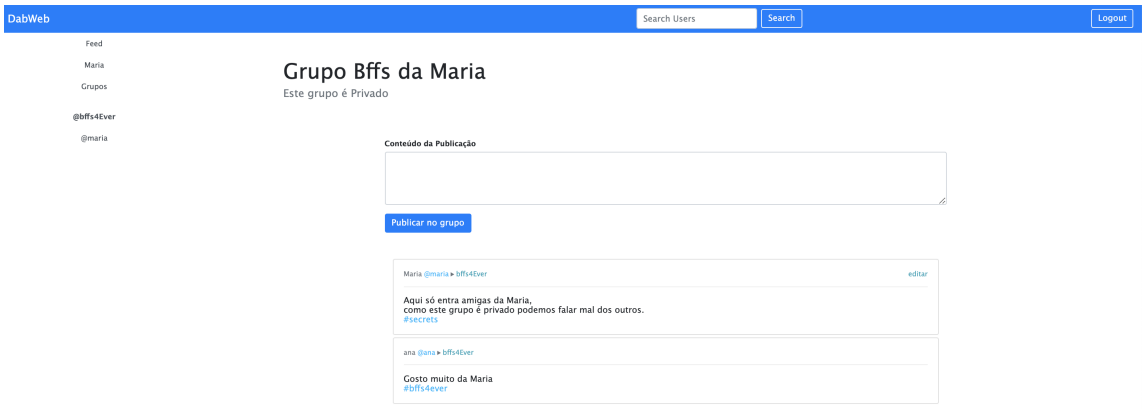

Figure 4: Dois utilizadores a interagir num grupo privado

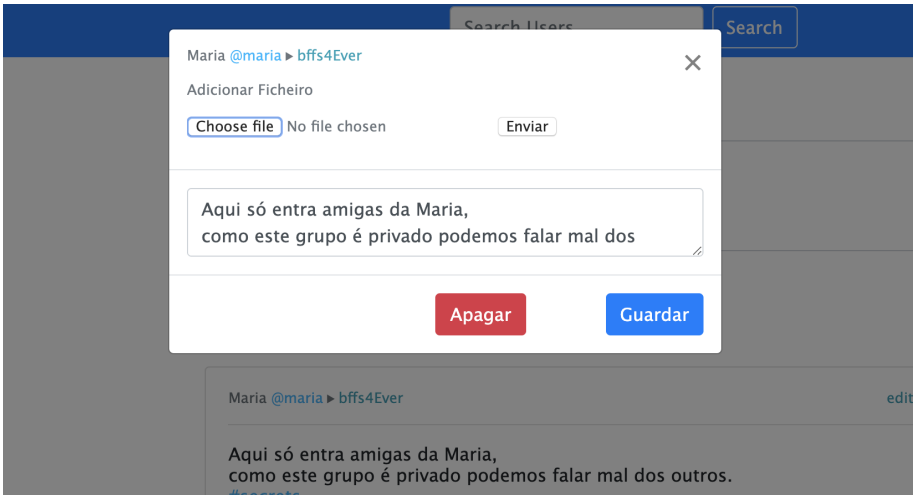

Figure 5: Edição de *posts* e partilha de ficheiros

8 Conclusão

Com este trabalho foi possível desenvolver uma pequena rede social de partilha de informação entre utilizadores.
Os requisitos pedidos foram implementados e complementarmente foi expandido o sistema com a possibilidade de partilha de ficheiros.

9 Referências

- https://www.npmjs.com/
- https://nodejs.org/
- https://expressjs.com/
- https://www.mongodb.com/
- http://www.passportjs.org/packages/passport-local/
- https://www.npmjs.com/package/axios
- https://pugjs.org/