



Universidade do Minho
Escola de Engenharia

Processamento e Representação de Informação **WorkIT - Rede Social de Trabalho**

Diego Porto PG3714
José Duarte Freitas A63129
Ricardo Neves A78764
Rui Costa A79947

Janeiro 2020

Resumo

Neste relatório pretendemos mostrar quais os passos que seguimos para o bom desenvolvimento de uma aplicação Web. Esta constitui uma rede social direcionada aos alunos de informática da Universidade do Minho, proporcionando-lhes um bom ambiente para partilhar dúvidas, materiais de estudo e trabalhos. Assim, podemos afirmar que a nossa rede social tem como objetivo aumentar o rendimento escolar dos alunos utilizadores da plataforma.

Conteúdo

1	Introdução	4
2	Estrutura da Aplicação	4
2.1	Entidades do Sistema	4
2.2	Definição das Estruturas de Dados	4
3	Principais Funcionalidades	6
3.1	Funcionalidades de Utilizador	6
3.2	Funcionalidades do Administrador	9
4	<i>Back-end</i>	10
4.1	Servidor de dados	10
4.2	Servidor da interface	10
5	<i>Front-end</i>	10
6	Integração com <i>ANTLR</i>	11
7	Conclusões e Trabalho Futuro	12

1 Introdução

A aplicação desenvolvida tem como principal objetivo proporcionar aos alunos de informática da Universidade do Minho uma plataforma para publicarem e consultarem, com facilidade, apontamentos dos seus colegas e docentes, de forma a facilitar o estudo de todos. Isto, como será de esperar, irá aumentar o rendimento de todos os alunos envolvidos. Toda a camada de apresentação ao utilizador está escrita em inglês, para possibilitar o uso do sistema em outros países.

As tecnologias usadas para o desenvolvimento desta aplicação foram *Node.js*, usado para todo o desenvolvimento da parte lógica do *Front-end* e *Back-end*. Para armazenamento e persistência de todos os dados, usamos o *MongoDB*, que funciona como base de dados (não-relacional), e que proporciona uma boa escalabilidade e flexibilidade.

Este relatório está formulado de forma a, numa primeira fase, apresentar a modelagem e a estrutura para a arquitetura da aplicação e, na segunda parte, apresentar os métodos utilizados para o funcionamento da aplicação e manipulação de informação na mesma.

Por último, apresentamos também o resultado final e o comportamento da aplicação.

2 Estrutura da Aplicação

2.1 Entidades do Sistema

Após estudar o problema em questão, o grupo chegou à conclusão que teríamos dois tipos de entidades a interagir com o sistema:

- Utilizador normal (alunos ou docentes)
- Administrador do sistema

As diferentes funcionalidades destas entidades serão referidas posteriormente neste documento.

2.2 Definição das Estruturas de Dados

Nesta aplicação, dados os requisitos propostos, chegamos à conclusão que seria necessário definir várias estruturas de dados, que serão úteis para a divisão da informação:

- **Users:** Define toda a informação guardada acerca de cada utilizador, sendo constituída por:
 - **name:** Nome do utilizador;
 - **number:** Número mecanográfico do utilizador, é usado como identificador para todas as operações;
 - **email:** *Email* do utilizador;
 - **password:** Palavra-chave do utilizador, usada para verificar o acesso ao sistema. É encriptada, para evitar furtos de informação;
 - **groups:** Lista dos grupos aos quais o utilizador se juntou;
 - **events:** Lista dos eventos aos quais o utilizador se juntou;
 - **photo:** Fotografia do utilizador.

- **Publications:** Define toda a informação guardada acerca de cada publicação, sendo constituída por:
 - **title:** Título da publicação;
 - **text:** Conteúdo textual da publicação;
 - **date:** Data da publicação;
 - **tags:** Palavras chave da publicação, usadas para facilitar a pesquisa de publicações;
 - **author:** Identificador do autor, usado para associar corretamente a publicação ao respetivo autor;
 - **authorName:** Nome do autor, usado para poder mostrar o nome do autor de cada publicação.
 - **target:** Alvo da publicação, serve para definir se a publicação em questão é publica ou direcionada a um grupo/evento em específico;
 - **comments:** Lista dos comentários da publicação;
 - **files:** Lista dos ficheiros anexados à publicação.
- **Messages:** Define toda a informação guardada acerca de cada mensagem, sendo constituída por:
 - **date:** Data da mensagem;
 - **author1:** Emissor da mensagem em questão;
 - **author2:** Remetente da mensagem em questão;
 - **text:** Texto que vai na mensagem;
 - **file:** Lista de ficheiros enviados anexados à mensagem.
- **Groups:** Define toda a informação guardada acerca de cada grupo, sendo constituída por:
 - **name:** Nome do grupo;
 - **description:** Descrição do grupo.
- **Events:** Define toda a informação guardada acerca de cada evento. É em tudo semelhante aos Grupos, pelo que é constituída por:
 - **name:** Nome do evento;
 - **description:** Descrição do grupo em questão;
 - **startDate:** Data de início do evento;
 - **endDate:** Data do final do evento.

De forma a guardar toda esta informação que é fornecida ao sistema, foi usada, como já foi referido na **Introdução**, uma base de dados *MongoDB* que terá armazenada todos os dados referidos acima.

3 Principais Funcionalidades

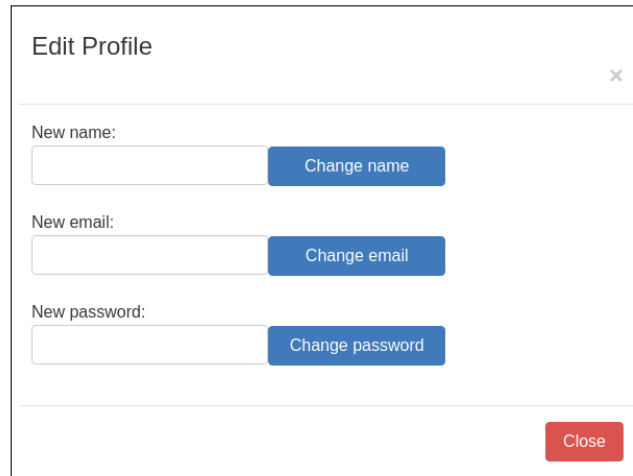
Neste projeto, as funcionalidades desenvolvidas podem ser separadas em dois grandes grupos:

- Funcionalidades de Utilizador;
- Funcionalidades de Administrador.

3.1 Funcionalidades de Utilizador

Para que o utilizador tenha acesso às funcionalidades do sistema, tem que introduzir as suas credenciais próprias, de modo a fazer *login* na aplicação. Após ser aceite no sistema, é apresentada a página do *feed*, sendo que é a principal página e todas as funcionalidades estão acessíveis a partir da mesma. Estas funções baseiam-se em:

- **Ver publicações** - Na página do *feed* é apresentada ao utilizador uma lista com todas as publicações que este pode ver.

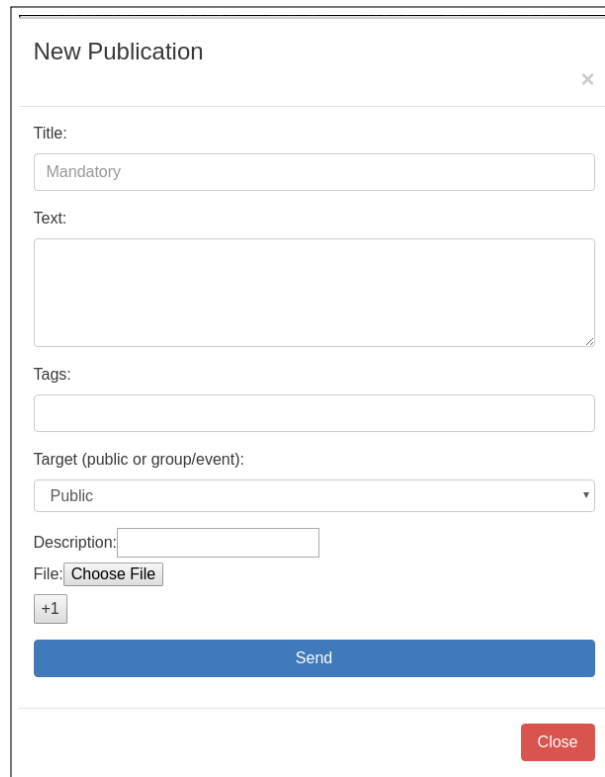


The image shows a web form titled "Edit Profile" with a close button (X) in the top right corner. The form contains three sections, each with a text input field and a corresponding button:

- New name:** A text input field followed by a blue button labeled "Change name".
- New email:** A text input field followed by a blue button labeled "Change email".
- New password:** A text input field followed by a blue button labeled "Change password".

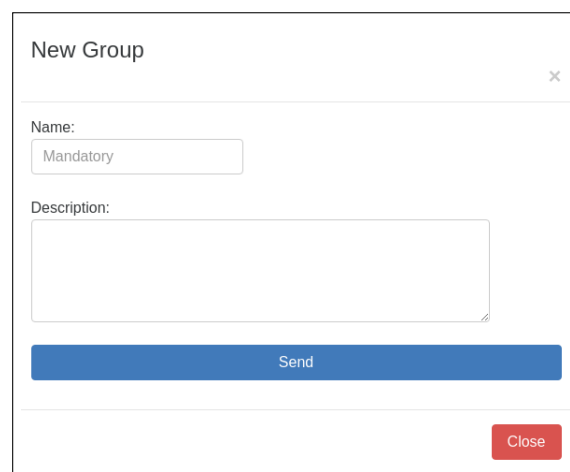
At the bottom right of the form, there is a red button labeled "Close".

- **Criar Publicação** - A partir da página inicial ao clicar no botão de criar publicação é apresentado um *pop-up* com o formulário de criação de publicação.



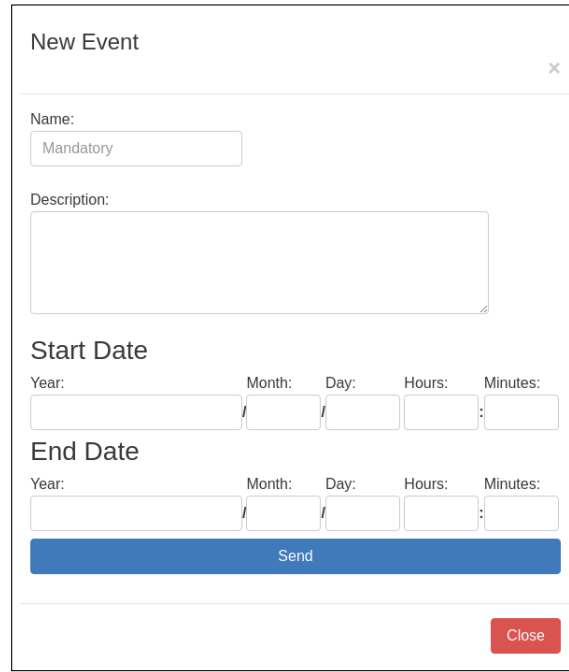
The 'New Publication' pop-up form is a vertical rectangle with a light gray border. At the top, it has a title bar with the text 'New Publication' on the left and a close button (an 'x' icon) on the right. Below the title bar, the form contains several input fields: a 'Title:' label followed by a text box containing the word 'Mandatory'; a 'Text:' label followed by a large, empty text area; a 'Tags:' label followed by a text box; a 'Target (public or group/event):' label followed by a dropdown menu currently showing 'Public'; a 'Description:' label followed by a text box; and a 'File:' label followed by a 'Choose File' button. Below these fields is a '+1' button. At the bottom of the form, there is a wide blue button labeled 'Send' and a red button labeled 'Close' in the bottom right corner.

- **Criar Grupo** - A partir da página inicial ao clicar no botão de criar grupo é apresentado um *pop-up* com o formulário de criação de grupo.



The 'New Group' pop-up form is a vertical rectangle with a light gray border. At the top, it has a title bar with the text 'New Group' on the left and a close button (an 'x' icon) on the right. Below the title bar, the form contains two input fields: a 'Name:' label followed by a text box containing the word 'Mandatory'; and a 'Description:' label followed by a large, empty text area. At the bottom of the form, there is a wide blue button labeled 'Send' and a red button labeled 'Close' in the bottom right corner.

- **Criar Evento** - A partir da página inicial ao clicar no botão de criar evento é apresentado um *pop-up* com o formulário de criação de evento.



The image shows a 'New Event' pop-up form. It has a title bar with 'New Event' and a close button (X). The form contains the following fields:

- Name:** A text input field with the placeholder text 'Mandatory'.
- Description:** A large text area for entering the event details.
- Start Date:** A date and time picker with fields for Year, Month, Day, Hours, and Minutes.
- End Date:** A date and time picker with fields for Year, Month, Day, Hours, and Minutes.
- Send:** A blue button to submit the event creation.
- Close:** A red button to close the pop-up.

- **Aderir/Sair do Grupo** - Ao clicar no botão de grupo na *navbar* é apresentada uma lista com todos os grupos. Nos grupos em que o utilizador está inserido, aparece a opção de sair; e nos grupos que o utilizador não está inserido, aparece a opção de aderir.
- **Aderir/Sair do Evento** - Ao clicar no botão de evento na *navbar* é apresentada uma lista com todos os eventos. Nos eventos em que o utilizador está inserido, aparece a opção de sair; e nos eventos que o utilizador não está inserido, aparece a opção de aderir.
- **Ver Publicações do Grupo** - Na página do *feed*, ao clicar no botão do grupo em questão, no *feed* passa só a ter as publicações referentes a esse grupo.
- **Ver Publicações do Evento** - Na página do *feed*, ao clicar no botão do evento em questão, no *feed* passa só a ter as publicações referentes a esse evento.
- **Ver o Próprio Perfil** - Na página do *feed*, ao clicar no botão de ver o próprio perfil, é apresentada uma página com todas as informações do próprio utilizador, assim como as suas publicações.

- **Editar o Próprio Perfil** - Na página do próprio perfil, ao clicar no botão de editar perfil, é apresentado um menu em *pop-up* com os diferentes campos que o utilizador pode editar.

- **Ver Perfil de Outro Utilizador** - Ao clicar no botão de utilizadores na *navbar* é apresentada uma lista com todos os utilizadores. Para cada elemento da lista, é possível clicar num botão que apresenta a página de perfil do utilizador em questão, contendo todas as respetivas informações.
- **Comentar Publicação** - Na página do *feed*, em cada publicação é possível clicar no botão referente a adicionar comentário, que abre um menu *pop-up*, apresentando o menu de escrita do comentário.
- **Enviar Mensagem Privada** - Ao clicar no botão de utilizadores na *navbar* é apresentada uma lista com todos os utilizadores. Para cada elemento da lista é possível clicar num botão que apresenta um menu em *pop-up* de escrita e envio de mensagem privada. As mensagens podem ser consultadas acedendo ao menu de mensagens pelo botão presente na *navbar*.

3.2 Funcionalidades do Administrador

De modo a ter controlo sobre o funcionamento do sistema a partir da interface, foi criada uma partição de administrador que tem o objetivo de garantir que há um moderador da aplicação. Este administrador tem acesso a todas as informações constituintes do sistema, desde as publicações, grupos, eventos e utilizadores registados. De salientar que esta rota de Administrador está protegida, sendo que é necessária uma autenticação especial. Estas funcionalidades de moderação baseiam-se em:

- **Eliminar Publicações** - Ao clicar no botão do *feed* na *navbar*, é apresentada uma lista com todas as publicações. Para cada elemento da lista é possível carregar no botão de eliminação que o apaga do sistema.
- **Eliminar Grupos** - Ao clicar no botão de Grupos na *navbar*, é apresentada uma lista com todos os grupos. Para cada elemento da lista é possível carregar no botão de eliminação que o apaga do sistema.
- **Eliminar Eventos** - Ao clicar no botão de Eventos na *navbar*, é apresentada uma lista com todos os eventos. Para cada elemento da lista é possível carregar no botão de eliminação que o apaga do sistema.
- **Eliminar Utilizadores** - Ao clicar no botão de utilizadores na *navbar*, é apresentada uma lista com todos os utilizadores. Para cada elemento da lista é possível carregar no botão de eliminação que o apaga do sistema.

4 *Back-end*

4.1 Servidor de dados

A primeira parte da implementação do servidor foi dedicada à criação da *API* de dados do sistema. Aqui, definimos os modelos e controladores que irão interagir com a base de dados e que já foram identificados na secção da **Definição de Estruturas de Dados**.

À medida que a implementação da aplicação ia avançando, este servidor de dados foi sendo atualizado, de modo a satisfazer os pedidos provenientes do **Servidor da interface**, que irá ser detalhado de seguida.

4.2 Servidor da interface

O servidor da interface é responsável pela comunicação entre o **Servidor da interface** e o utilizador, e assegura que não há falhas no que toca a redirecionamento do utilizador entre o sistema.

A comunicação entre o servidor da interface e a API de dados é realizada através de pedidos *axios*. Estes pedidos são emitidos para a API (contendo informações no *header* e/ou *body*), sendo que a mesma retorna, em formato *JSON*, o resultado do pedido. Por isto, todos os dados que são dispostos no monitor do utilizador passaram por um pedido *axios* e tratados posteriormente.

Para esta aplicação, foram definidas oito rotas distintas, sendo uma delas dedicada à autenticação. Esta autenticação baseia-se na ferramenta *JWT*, apresentada nas aulas de PRI.

5 *Front-end*

Em relação à construção do *Front-end*, o grupo tentou criar uma interface acessível e intuitiva para todos os tipos de utilizadores sabendo, à partida, que a grande fatia é dedicada aos jovens estudantes da Universidade.

Assim, após uma pequena pesquisa, decidimos construir a interface com a ajuda da ferramenta *Bootstrap*. O *Bootstrap* é uma *framework web* aberta para desenvolvimento de componentes da interface de *websites* e aplicações *WEB*. Esta possibilita um design moderno, amigável e responsivo, melhorando a experiência do utilizador.

6 Integração com *ANTLR*

Foi proposto pelo docente da Unidade Curricular de GCS que o grupo integrasse a ferramenta *ANTLR* no projeto, servindo como um possível caminho para registo de novos utilizadores.

Desta forma, aplicando todo o conhecimento adquirido nas aulas de GCS, concebemos a seguinte gramática:

```
grammar register;

// Parser
register returns [String name, String number, String email, String password]:
NAME ASSIGN username{$name = $username.out;} COMMA
NUMBER ASSIGN usernumber{$number = $usernumber.out;} COMMA
EMAIL ASSIGN useremail{$email = $useremail.out;} COMMA
PASSWORD ASSIGN userpassword{$password = $userpassword.out;};

username returns [String out]: TEXT {$out = $TEXT.text;};

usernumber returns [String out]: TEXT {$out = $TEXT.text;};

useremail returns [String out]: TEXT {$out = $TEXT.text;};

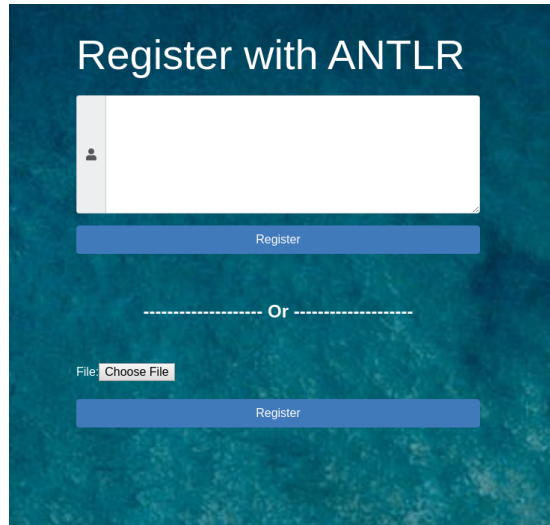
userpassword returns [String out]: TEXT {$out = $TEXT.text;};

// Lexer
DOT: '.';
COMMA: ',';
ASSIGN: ':';
NAME: 'name';
NUMBER: 'number';
EMAIL: 'email';
PASSWORD: 'password';
TEXT: [0-9a-zA-ZáéíóúÁÊÏÓÜÀàÇ. @ ]+;
WS: [\r\n\t] -> skip;
```

Podemos reparar acima que a gramática retorna quatro *Strings*: nome, numero, *email* e *password*. Estes dados serão tratados para registar um novo utilizador que se pretenda registar através desta funcionalidade.

Com a gramática criada num ficheiro *register.g4*, foi aplicado o seguinte comando no terminal: `antlr4 -Dlanguage=JavaScript register.g4`.

Foram assim gerados três novos ficheiros *JavaScript*, contendo o *Lexer*, *Parser* e *Listener* da gramática. Estes *scripts* são importados no servidor e são responsáveis por transformar o conteúdo disponibilizado pelo utilizador num novo registo na base de dados.



Pela imagem acima, podemos verificar que o utilizador tem duas opções de uso do *ANTLR*: pode escrever diretamente os seus dados na interface, ou pode dar *upload* a um ficheiro. É importante referir que, em ambos os casos, o *input* do utilizador deverá estar semântica e sintaticamente correto.

7 Conclusões e Trabalho Futuro

Para concluir este documento, podemos dizer que o grupo se encontra bastante satisfeito com o resultado obtido, uma vez que cumpriu grande parte dos requisitos delineados numa fase inicial. De facto, superou algumas expectativas do grupo, devido a tratar-se de um trabalho rigoroso e exigente a vários níveis.

No entanto, como na grande maioria dos projetos, há aspetos que poderiam (e deveriam) ser melhorados. Assim, para um possível trabalho futuro, o nosso objetivo decai sobre a melhoria de alguns aspetos na interface, inserção de fotos de perfil, melhoramento na fiabilidade da autenticação, serviço de *chat* mais rápido e intuitivo, entre outros detalhes e melhoramentos.

