



**Universidade do Minho**  
Escola de Engenharia

Processamento de Linguagens e Conhecimento  
Processamento e Representação de Informação  
Trabalho de Grupo

**COLLEGIFY**

Equipa: Feliz Bertos

|        |   |
|--------|---|
| A74618 | Bernardo Miguel Teixeira Dias Viseu de Carvalho |
| A75480 | Marco Matias Pereira Gonçalves                  |
| A75310 | Renato Jorge Cruzinha da Silva                  |

Braga,  
30 de novembro de 2020

# Conteúdo

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Introdução &amp; Contextualização</b> | <b>2</b> |
| <b>2</b> | <b>Desenvolvimento da Aplicação</b>      | <b>3</b> |
| 2.1      | Base de Dados . . . . .                  | 3        |
| 2.2      | Utilizadores . . . . .                   | 3        |
| 2.3      | Post de Recursos . . . . .               | 3        |
| 2.4      | Pesquisa . . . . .                       | 3        |
| 2.4.1    | Pesquisa de <i>Posts</i> . . . . .       | 4        |
| 2.4.2    | Pesquisa de <i>Tags</i> . . . . .        | 4        |
| 2.5      | <i>Feed</i> de Notícias . . . . .        | 4        |
| 2.6      | Autenticação . . . . .                   | 4        |
| 2.7      | <i>Common.js</i> . . . . .               | 5        |
| <b>3</b> | <b>Conclusão</b>                         | <b>6</b> |

# 1 Introdução & Contextualização

No âmbito da Unidade Curricular de Processamento e Representação de Informação (PRI), foi-nos proposto este trabalho prático que consiste no desenvolvimento de uma página web com capacidade de partilha de conteúdo universitário, com um sistema de usuários que podem fazer *upload* e *download* do vários conteúdos disponíveis.

Um utilizador pode identificar o tipo de conteúdo que quer publicar, (por exemplo: TPC, Relatório, Teste, Tese, etc...) de maneira a termos uma boa organização e representação dos recursos. É também indicado o curso, ano e semestre relativo ao conteúdo.

Outros objectivos deste projecto foram a capacidade de comentar, classificar, e adicionar aos favoritos os recursos disponíveis.

Neste relatório vamos explicar o desenvolvimento deste projecto, assim como as partes que se mostraram mais difíceis de implementar.

## 2 Desenvolvimento da Aplicação

Para o desenvolvimento deste projecto, utilizamos métodos ensinados em PRI, de maneira a termos uma aplicação utilizável e satisfatória tendo em conta os objectivos do enunciado.

### 2.1 Base de Dados

Para implementarmos consistência de dados na nossa aplicação, usamos a base de dados em *MongoDB* onde guardamos as nossas colecções de dados. Para tal, na nossa aplicação tivemos de definir os vários modelos que iríamos precisar para manter esta consistência de dados e de modo a que pudéssemos armazenar e manipular essa informação. Sendo assim, analisámos que necessitaríamos de ter três modelos distintos, de modo a fazer a distinção entre *users*, *posts*, *comments* e *news*.

Desenvolvemos os seguintes modelos de dados:

- User
- Post
- Comment
- News

### 2.2 Utilizadores

De modo a criar vários níveis de acesso, tivemos de distinguir os vários níveis de utilizadores. Para isso, usamos a variável *level* de um *user* no base de dados anteriormente explícita, e admitimos três níveis distintos de acessos. Estes três níveis são diferenciados por três *strings* diferentes, sendo elas registadas aquando do registo de utilizador. Fazemos portanto a distinção entre **Admin**, **Produtor** e **Consumidor**.

### 2.3 Post de Recursos

Na parte de criar um recurso na base de dados, é pedido ao utilizador para criar um post, onde é apresentada uma página em *pug* em que são exibidos os vários campos necessários para a criação com sucesso de um post. É aqui que o utilizador faz o upload do ficheiro que quer publicar no sistema, sendo que pode optar por fazê-lo de modo *Público* ou *Privado*, sendo que em caso deste último, apenas ficaria disponível para ele e para os *admins*, e também não é lançada uma notícia para o feed dos consumidores.

O método de *upload* consiste em utilizar a ferramenta *multer* de maneira semelhante ao que foi feito nas aulas, dando *upload* de um ficheiro e guardando-o numa pasta *postStore*.

### 2.4 Pesquisa

De modo a facilitar a vida do utilizador, a nossa aplicação tem o modo de pesquisa implementado. Esta pesquisa pode ser feita através de *tags* ou de parâmetros variados dos *posts*.

### 2.4.1 Pesquisa de Posts

Neste sentido, damos ao utilizador a opção de efectuar uma pesquisa de um post através da selecção do curso pretendido. Para isso, definimos uma rota de pesquisa de cursos, que iremos ver à frente, na secção das rotas, todas as que estão definidas, mas as de pesquisa, são efectuadas acrescentados o nome parâmetro e acrescentando em variável a no endereço. Deste modo, uma pesquisa sobre um recurso do curso de MIEI da cadeira de PRI, do 4º ano do 1º semestre, ficaria um pedido igual ao seguinte: `GET /cursos/MIEI/ano/4/semestre/1/uc/PRI`. Para cada um dos parâmetros é apresentada a lista de possibilidades, ou seja, o utilizador nunca através da interface web vai parar a uma rota que não tenha nenhum recurso disponível. Sendo assim, para isto ser possível, todos os parâmetros intermédios têm uma view em *pug* para apresentar os resultados e redireccionar o utilizador para a página de resultados pretendida.

### 2.4.2 Pesquisa de Tags

Quando o utilizador está a visualizar a lista de posts, quer a lista total, quer como resultado de uma pesquisa, tem a possibilidade de, caso encontro uma *tag* que seja do seu interesse, procurar por uma lista de todos os posts que contêm essa mesma *tag* no seu registo. O resultado é apresentado numa view como resultado do pedido `GET /posts/tag/:tag`

## 2.5 Feed de Notícias

Na *Homepage* da nossa aplicação, o utilizador tem acesso a uma lista das últimas 10 notícias publicadas no nosso servidor. Sendo assim, são apresentadas quer novas publicações, quer novos comentários. De notas, que caso um produtor faça *upload* de um recurso privado, não será lançada uma notícia, e consequentemente não aparecerá no feed de notícias dos consumidores. Cada vez que um utilizador faça um comentário num post público, será lançada também uma notícia para o servidor, aparecendo também no respectivo *feed*. Quando o utilizador está na *view* do feed de notícias, apresentado na página principal, tem acesso directo a cada *post*, clicando directamente na publicação da notícia, ou do comentário, facilitando o acesso aos posts e dando um valor adicional à *feature* dos notícias.

## 2.6 Autenticação

Para tratar a autenticação utilizamos as *libraries* *passport*, *passport-local*, *passport-google-oauth20* e *passport-facebook*. Por norma o utilizador pode autenticar-se de uma forma simples utilizando o seu username e a password (*LocalStrategy*). De forma a agilizar o processo de autenticação foram ainda implementadas autenticações através do Google (*GoogleStrategy*) e do Facebook (*FacebookStrategy*). De forma a modularizar o código foi criada uma pasta *config* que contem os ficheiros *key.js* que guarda as credencias para serem utilizadas as *FacebookStrategy* e *GoogleStrategy*, já o ficheiro *passport-config.js*. Caso o utilizador já esteja autenticado na aplicação este se entrar pelas rotas `/auth/google` ou `/auth/facebook` ele simplesmente irá acrescentar um *googleID* ou *facebookID* ao seu utilizador fazendo com que este também consiga entrar na sua conta através do Google ou Facebook.

## **2.7 *Common.js***

É de destacar a utilização de uma função contida num ficheiro `common.js`, a *verificaAutenticacao*, que é utilizada para verificar que um utilizador não registado, não consegue aceder a rotas protegidas destinadas a utilizadores já registados no sistema, redireccionando-os assim para a página de Login.

### 3 Conclusão

Em suma, com a realização deste trabalho, pudemos por em prática o conhecimento adquirido ao longo das aulas, consolidando assim a matéria relacionada com páginas web, acesso e utilização de bases de dados autenticação, entre outros.

As nossas maiores dificuldades passaram pela implementação de algumas *features*, como a divisão de utilizadores por níveis de maneira a não permitir certos utilizadores o acesso a funcionalidades do sistema.

Este trabalho foi muito esclarecedor e ajudou muito no entendimento de muitos conceitos de programação que até antes nos eram desconhecidos, o que fez com que este trabalho fosse muito didáctico.

O resultado deste projecto é, no nosso ponto de vista, satisfatório, sendo que conseguimos realizar um bom sistema de partilha de recursos académicos. Achamos que existe ainda funcionalidades com mais potencial, como a melhor organização das páginas, assim como uma melhor implementação do sistema de *rankings* e *likes*.