



Representação e Processamento de Conhecimento na Web

Projeto - Repositório de Recursos Didáticos

André Carvalho da Cunha Martins PG47009

António Jorge Nande Rodrigues PG47030

José Pedro Castro Ferreira PG47375



Figura 1: PG47009



Figura 2: PG47030



Figura 3: PG47375

19 de junho de 2022

Conteúdo

1	Introdução	2
2	Arquitetura da Plataforma	3
2.1	Camadas aplicacionais	3
2.2	Web-Server	3
2.3	Api-Server	4
2.4	Auth-Server	5
3	Persistência de Dados	6
3.1	Base de Dados MongoDB e Collections	6
4	Funcionalidades da plataforma	8
4.1	Inserção de novos recursos	8
4.2	Consulta de ficheiros	8
4.3	Edição e remoção de ficheiros	8
4.4	Edição de um utilizador	8
4.5	Notícias	8
4.6	Favoritos, gostos e comentários	8
4.7	Área de administração	9
5	Ilustração da aplicação	9
5.1	Página de login ou registo	9
5.2	Página inicial <i>producer</i>	9
5.3	Página inicial <i>consumer</i>	9
5.4	Página inicial <i>admin</i>	10
5.5	Notícias para todos os utilizadores	10
5.6	Comentários	11
5.7	Favoritos	11
5.8	Consulta de ficheiros	11
5.9	Search-bar	11
5.10	Opções de administração	11
5.11	Edição de utilizadores	12
5.12	Erros na inserção de ficheiros	12
6	Conclusão	13

1 Introdução

Como projeto prático da Unidade Curricular de Representação e Processamento de Conhecimento na Web foi-nos proposto o desenvolvimento de uma plataforma Web para armazenamento, consulta e acesso a recursos didáticos e formativos. Nesta plataforma teríamos que identificar e desenvolver funcionalidades específicas para tipos de utilizadores diferentes, realizar a validação dos ficheiros que eram adicionados, bem como tratar do seu processamento, armazenamento e disseminação.

Além destas funcionalidades que seriam o cerne da nossa aplicação, foram desenvolvidas funcionalidades extra tais como permitir dar gosto nos recursos adicionados, comentar recursos adicionados, gestão de notícias por parte de um administrador, bem como realizar o download de um ficheiro de logs que retrata todos os acontecimentos relevantes ocorridos durante a execução da aplicação.

De seguida iremos realizar uma breve explicação acerca da arquitetura de toda a aplicação, baseada no modelo de referência OAIS, e sobre a organização da base de dados não-relacional, suportada em MongoDB. Por fim, apresentaremos imagens relativas ao funcionamento da aplicação para retratar as diferentes funcionalidades implementadas.

2 Arquitetura da Plataforma

Para uma melhor compreensão e visualização de toda a arquitetura da aplicação e de todo o processamento necessário para responder aos pedidos de um cliente arbitrário elaboramos um pequeno diagrama apresentado abaixo sobre o qual iremos fazer uma pequena descrição de todos os seus componentes.

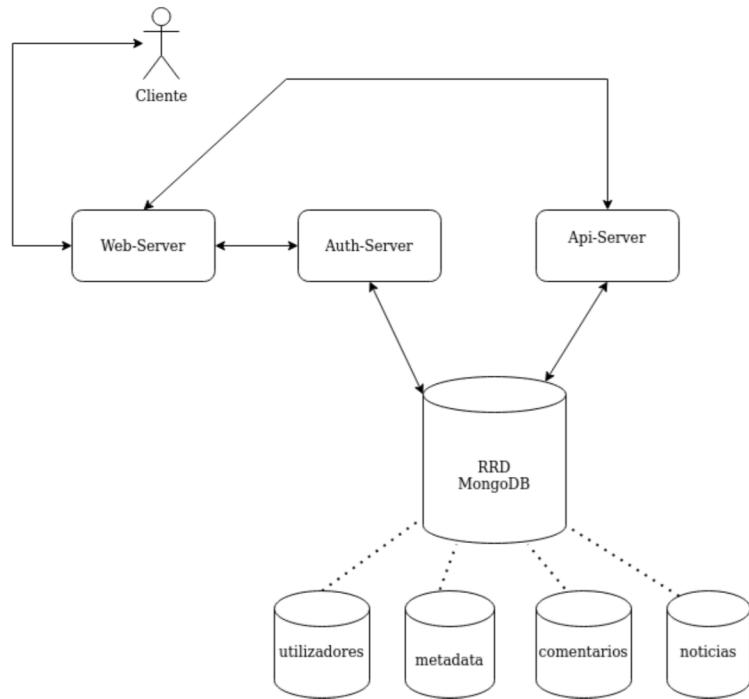


Figura 4: Arquitetura aplicacional

2.1 Camadas aplicacionais

Como é possível observar na imagem, a nossa plataforma pode ser dividida em 3 camadas aplicacionais heterogéneas, cada uma com um papel e funcionalidades específicas.

2.2 Web-Server

Esta camada está responsável pela construção e implementação de uma interface gráfica para apresentar aos clientes da aplicação. É a esta camada que todos os clientes irão realizar os seus pedidos. Através das rotas definidas, serão feitos pedidos quer à camada Auth-Server quer à API-Server.

Focando na comunicação à API-Server, esta irá receber o pedido, realizar o processamento necessário para o pedido feito (uma vez que está conectada à base de dados da nossa aplicação) e irá responder com os dados em formato JSON. A Web-Server ao receber a resposta ao seu pedido, irá processar os dados em JSON e irá renderizar uma das várias páginas Web que tem definidas para apresentar a informação pretendida ao cliente.

2.3 Api-Server

Como descrito em cima, a camada API-Server retrata a construção da API de dados da nossa aplicação. Conectando-se à base de dados, esta camada está responsável por responder ao pedidos da Web-Server em formato JSON. Iremos apresentar todas as rotas definidas e as respostas que se obtém quando é realizado um pedido para essas mesmas rotas. De salientar que toda a API está protegida através da utilização de **jsonwebtoken** fazendo com que os pedidos só possam ser realizados sob a existência de um token válido. Além dessa verificação, também são realizadas verificações específicas acerca do nível de acesso do utilizador correspondente ao token, uma vez que alguns pedidos são específicos e protegidos apenas para administradores da aplicação.

Assim, apresentamos todas as rotas suportadas pela nossa API de dados:

- **GET /recursos:** Pedido para obter os recursos que estão guardados na base de dados. Existem 3 tipos de query que podem ser definidos, originando um retorno de dados diferentes.
Tipo: Retorna apenas recursos de um de 4 tipos possíveis (Tese, Slides, Fichas e Exames)
Q: Retorna apenas os recursos que contenham a palavra na query no seu título. Rota utilizada para listar recursos através da search-bar.
User: Retorna os recursos que o utilizador da query tem um gosto. Rota utilizada para a funcionalidade Favoritos, em que aparecem os recursos nos quais o utilizador com sessão iniciada tem gosto.
Se não houver definição de query, será feita a listagem de todos os recursos.
- **GET /noticias:** Rota protegida apenas para administradores, devolve todos os acontecimentos que ocorreram no sistema, quer sejam eles visíveis para os utilizadores quer sejam invisíveis.
- **GET /noticiasUser:** Rota que devolve as notícias que se encontram visíveis para os utilizadores e que serão apresentadas na página inicial da aplicação. Essas notícias retratam inserção de recursos, eliminação de recursos, edição de um recurso, colocar gosto num recurso e comentar um recurso.
- **GET /recursos/:filename** Rota que devolve toda a informação do recurso que possui o nome *filename*. De salientar que todos os recursos guardados na base de dados possuem um nome único e irrepetível, fazendo com que este pedido apenas retorne um e um só recurso.
- **DELETE /recursos/remove/:filename** Rota que irá remover da base de dados o recurso que possui o nome *filename*. Nesta rota é feita a verificação se o nível do utilizador é diferente de consumidor, uma vez que os consumidores não podem remover recursos da plataforma.
- **POST /recursos** Rota que irá adicionar um novo recurso à base de dados. Novamente, é realizada a verificação se o utilizador não é um consumidor.
- **PUT /recursos** Rota que irá editar um recurso existente na base de dados. Novamente, é realizada a verificação se o utilizador não é um consumidor.
- **POST /recursos/addLike/:filename** Rota que irá adicionar um utilizador à lista de utilizadores com gosto no recurso *filename*.
- **POST /recursos/removeLike/:filename** Rota que irá remover um utilizador à lista de utilizadores com gosto no recurso *filename*.

- **POST /noticias** Rota que irá adicionar uma nova notícia à base de dados.
- **DELETE /noticias/remove/:id** Rota que irá remover uma notícia de id *id*. Apenas um administrador pode realizar este pedido.
- **GET /noticias/deleteAll** Rota que irá eliminar todas as notícias da base de dados. Apenas um administrador pode realizar este pedido.
- **GET /noticias/editV/:id** Rota que irá tornar a notícia de id *id* visível para todos os utilizadores. Apenas um administrador pode realizar este pedido.
- **GET /noticias/editI/:id** Rota que irá tornar a notícia de id *id* invisível para todos os utilizadores, exceto administradores. Apenas um administrador pode realizar este pedido.
- **GET /comentarios/:filename** Rota que irá devolver todos os comentários relativos ao recurso *filename*
- **POST /comentarios** Rota que irá acrescentar um novo comentário.
- **DELETE /comentarios/:id** Rota que irá eliminar o comentário de id *id*
- **PUT /comentarios/:id** Rota que irá editar o comentário de id *id*

2.4 Auth-Server

Esta camada aplicacional está, principalmente, responsável por todas as questões e funcionalidades relacionadas com a autenticação e registo de utilizadores da plataforma. Como referido acima, recorremos à utilização de **jsonwebtoken** para realizar a autenticação de todos os utilizadores, recorrendo, depois, a esse token para obter dados da API.

A cada utilizador é também associado um nível de acesso de 3 níveis possíveis (*consumer*, *producer* e *admin*), podendo os utilizadores do tipo *producer* adicionar recursos à plataforma, editar e eliminar os seus próprios recursos. Já os utilizadores do tipo *consumer* apenas podem consultar os recursos existentes, não podendo alterar os mesmos. Já todos os utilizadores do tipo *admin* têm acesso a todas as funcionalidades da plataforma, bem como a dados e funcionalidades específicas para administração.

De salientar que todas as funcionalidades como dar gosto num recurso, comentar um recurso, editar o perfil de utilizador, utilização da search bar são abertas a todos os tipos de utilizadores.

Por questões de segurança decidimos que a inserção de novos admins na base de dados apenas pode ser feita com um POST manual e não através da aplicação.

3 Persistência de Dados

Tal como era proposto no enunciado, toda a persistência dos dados da plataforma, tais como informações acerca dos recursos, dos seus utilizadores, entre outras informações relevantes é feita recorrendo à utilização de uma base de dados definida em MongoDB e ao módulo do JavaScript FileSystem.

Fazendo uma breve explicação acerca da utilização do FileSystem, este módulo foi utilizado para guardar os recursos que são inseridos na plataforma, uma vez que os metadados de cada recurso são armazenados na devida *collection* na base de dados. No Web-Server existe uma diretoria chamada *uploads* onde serão guardados todos os ficheiros que forem colocados na nossa plataforma. Dentro dessa diretoria, existem 4 sub-diretorias correspondentes aos 4 tipos de ficheiros que a nossa plataforma suporta (Tese, Fichas, Exames e Slides). Assim, cada ficheiro será guardado na sub-diretoria correspondente e com um nome único e irrepetível. Assim, e através dos metadados armazenados na nossa base de dados é possível "fazer a ponte" entre as informações e realizar todas as operações relativas aos recursos e ao tratamento dos mesmos.

3.1 Base de Dados MongoDB e Collections

A base de dados definida para este projeto nomeou-se RRD e possui 4 collections distintas face ao tipo de pedidos que iriam ser feitos ao sistema. Assim, faremos uma breve descrição de cada uma delas e de como são constituídas.

- **Metadata:** Collection onde serão armazenadas as informações acerca dos recursos guardados no FileSystem, sendo cada entrada desta collection constituída por:
 - **date_creation:** Data de criação do recurso
 - **date_submission:** Data de submissão do recurso
 - **producer:** Autor do recurso
 - **owner_submission:** Utilizador que realizou a submissão do recurso
 - **title:** Título do recurso
 - **type:** Tipo do recurso
 - **filename:** Nome único do recurso
 - **filesZip:** Ficheiros PDF ou XML que constituem o recurso
 - **liked_by:** Conjunto de usernames que gostaram do recurso
- **Notícias:** Collection onde serão armazenadas todas as informações relativas a eventos que ocorram na plataforma durante a sua execução. As notícias podem dividir-se em 2 tipos: As que só são visíveis aos administradores, funcionando como logs do sistema e retratando todo o tipo de acontecimentos, e as que são visíveis a todos os utilizadores, funcionando como avisos para eventos importantes que tenham ocorrido. Cada entrada desta collection é constituída por:
 - **username:** Username do utilizador que realizou o evento
 - **action:** Descrição do evento ocorrido
 - **date:** Data e hora do evento
 - **visible:** Flag para saber se evento é visível a todos os utilizadores ou não.

- **Comentarios:** Collection onde serão armazenados os comentários relativos aos recursos presentes na plataforma. Ao associar um comentário a um *filename*, é possível fazer a ligação entre os recursos e os comentários que lhe foram efetuados. Cada entrada desta collection é constituída por:
 - **username:** Username de quem realizou o comentário
 - **descricao:** Texto que constitui o comentário
 - **date:** Data e hora do comentário
 - **filename:** Ficheiro comentado
- **Utilizadores:** Collection onde serão armazenados todos os utilizadores da plataforma. Cada entrada desta collection é constituída por:
 - **name:** Nome do utilizador
 - **username:** Username do utilizador. Único e irrepetível.
 - **password:** Password do utilizador
 - **level:** Nível de acesso entre 3 possíveis, *producer*, *consumer* ou *admin*.

4 Funcionalidades da plataforma

Faremos, agora, um pequeno resumo das principais funcionalidades da nossa plataforma, apresentando de seguida capturas de ecrã da aplicações onde é possível observar o diferentes conjunto de operações que é possível realizar.

4.1 Inserção de novos recursos

Sendo esta plataforma um repositório de recursos, estes têm que seguir algumas regras para poderem ser aceites na nossa plataforma, começando por terem que estar no formato ZIP. Necessitam, também, de possuir 2 ficheiros obrigatórios e que se encontrem válidos. O ficheiro *manifest.txt* que indica os ficheiros que fazem parte do recurso e o ficheiro *RRD-SIP.json* que indica os metadados do recurso que será adicionado. Estes 2 ficheiros têm que seguir as regras definidas e estarem totalmente válidos. De seguida, o recurso apenas poderá possuir ficheiros do tipo XML ou PDF, não sendo aceites quaisquer tipos de outros ficheiros. Após estas validações é verificado se o nome do ficheiro é único e ainda não existe na base de dados. Ultrapassada esta última validação, o ficheiro é adicionado com sucesso à nossa plataforma.

Este processo de inserção de novos recursos apenas está disponível a utilizadores *producer* ou *admin*.

4.2 Consulta de ficheiros

Esta funcionalidade está disponível para todos os utilizadores, podendo utilizar a search-bar existente e procurar ficheiros através do seu título ou indo à área específica para cada tipo de ficheiro, onde estão apresentados todos os ficheiros existentes daquele tipo. Todos os utilizadores podem, também, realizar o download de qualquer recurso.

4.3 Edição e remoção de ficheiros

Esta funcionalidade apenas está disponível para administradores ou para utilizadores *producer*, limitados a poderem apenas editar ou eliminar os ficheiros que submeteram. A edição de um ficheiro trata-se de alterar o seu tipo para um dos 3 tipos restantes disponíveis.

4.4 Edição de um utilizador

Todos os utilizadores podem alterar o seu username para um novo, desde que ainda não esteja utilizado e podem, também, editar o seu nome que lhes está associado.

4.5 Notícias

Todos os utilizadores têm acesso a um *pop-up* que mostra as notícias mais relevantes que aconteceram.

4.6 Favoritos, gostos e comentários

Todos os utilizadores podem dar, ou remover, um gosto num dado recurso. Após darem gosto num recurso, este será adicionado aos seus favoritos, tornando a consulta de todos os ficheiros que tiveram um gosto mais rápida e fácil. É também possível a todos os utilizadores realizarem comentários nos recursos, bem como a possibilidade de os poderem alterar ou mesmo apagar, desde de que sejam os autores do comentário. Exceção feita aos administradores que podem editar e apagar quaisquer comentários.

4.7 Área de administração

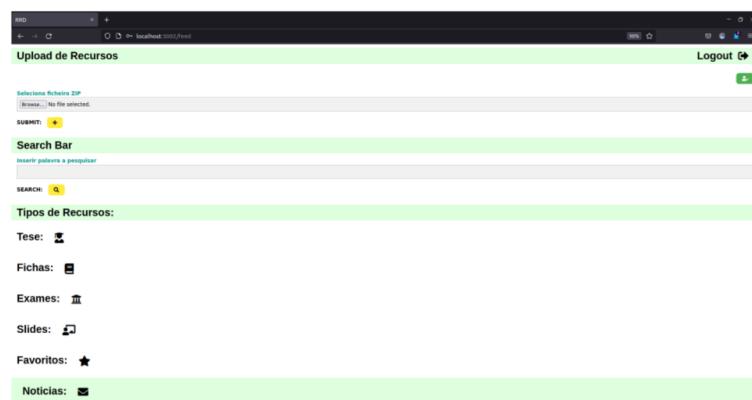
Esta área é restrita apenas a administradores, onde poderão ter acesso a uma listagem de todos os recursos, utilizadores e estatísticas/notícias do sistema. Aliada a esta consulta estão todas as funcionalidades de inserção, edição e remoção.

5 Ilustração da aplicação

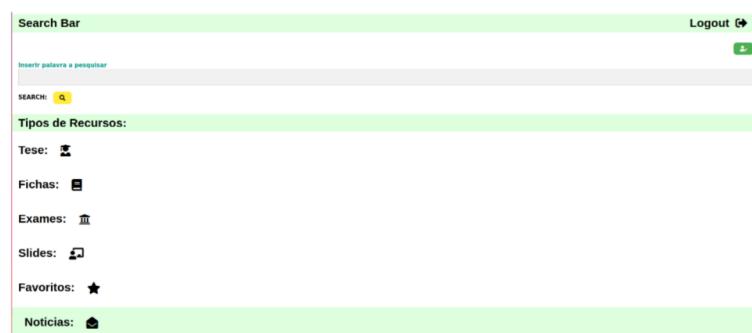
5.1 Página de login ou registo



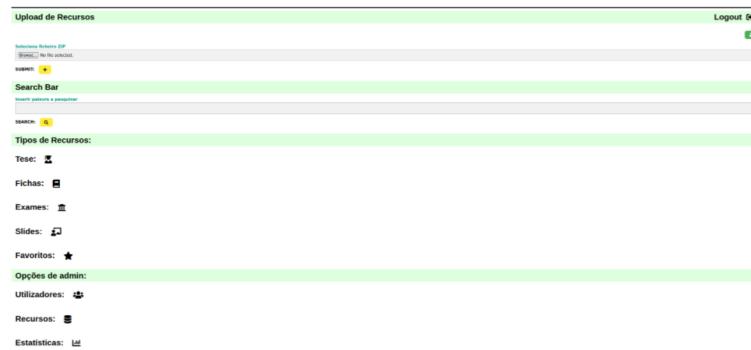
5.2 Página inicial producer



5.3 Página inicial consumer



5.4 Página inicial *admin*

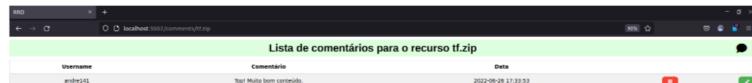


5.5 Notícias para todos os utilizadores

A modal window is displayed, showing a table of recent user activities. The table has columns for 'Username', 'Acontecimento' (Event), and 'Data' (Date). The data is as follows:

Username	Acontecimento	Data
andre141	O utilizador andre141 comentou o recurso tf.zip	2022-06-26 17:33:53
andre141	O utilizador andre141 gostou do recurso tf.zip publicado por andre141	2022-06-26 17:33:20
andre141	O utilizador andre141 inseriu o ficheiro tf.zip do tipo Slide	2022-06-26 17:29:17

5.6 Comentários



Screenshot of a web browser showing a comment for a resource named tf.zip. The comment text is "Top! Muito bom conteúdo." and it was posted on 2022-09-26 17:33:53 by user andrei141.

5.7 Favoritos



Screenshot of a web browser showing a list of favorite resources. One resource is listed: "Slides sobre tolerância a faltas" submitted by Pedro Gonçalves on 2022-06-26 17:29:17.

5.8 Consulta de ficheiros



Screenshot of a web browser showing a list of slides. One slide is listed: "Slides sobre tolerância a faltas" submitted by Pedro Gonçalves on 2022-06-26 17:29:17.



Screenshot of a web browser showing a detailed view of a slide titled "Slides sobre tolerância a faltas" submitted by Pedro Gonçalves on 2022-06-26 17:29:17.

5.9 Search-bar



Screenshot of a web browser showing a search results page for resources containing the word "faltas" in the title. One resource is listed: "Slides sobre tolerância a faltas" submitted by Pedro Gonçalves on 2022-06-26 17:29:17.

5.10 Opções de administração



Screenshot of a web browser showing a list of users and a "Registrar novo utilizador" (Register new user) button. The users listed are André Martins, João Pátria, Pedro Souza, Pedro Barbosa, and Rui Miguel Campos, each with their respective roles (producer, admin, consumer) and status indicators.

Lista de recursos						
Data de Criação	Data de Submissão	Produtor	Autor da Submissão	Título	Tipo	
13-06-2022	2022-06-26 17:29:17	Pedro Gonçalves	andre141	Síntese sobre tolerância a falhas	Slide	

Lista de acontecimentos				Fazer download:		Apagar todas:	
Username	Acontecimento	Data	Visibilidade				
admin1	O utilizador admin1 efetuou login	2022-06-26 17:35:49	Invisível				
ruicampose17	O utilizador ruicampose17 procurou ficheiros com a palavra falhas	2022-06-26 17:35:38	Invisível				
ruicampose17	O utilizador ruicampose17 consultou os recursos do tipo Slide	2022-06-26 17:35:20	Invisível				
ruicampose17	O utilizador ruicampose17 efetuou login	2022-06-26 17:35:18	Invisível				
andre141	O utilizador andre141 comentou o recurso tifp	2022-06-26 17:33:53	Visível				
andre141	O utilizador andre141 gostou do recurso tifp publicado por andre141	2022-06-26 17:33:20	Visível				
andre141	O utilizador andre141 consultou os recursos do tipo Slide	2022-06-26 17:33:20	Invisível				
andre141	O utilizador andre141 consultou os recursos do tipo Slide	2022-06-26 17:33:18	Invisível				
andre141	O utilizador andre141 efetuou login	2022-06-26 17:32:39	Invisível				
andre141	O utilizador andre141 consultou os recursos do tipo Slide	2022-06-26 17:29:26	Invisível				
andre141	O utilizador andre141 consultou os recursos do tipo Ficheiro	2022-06-26 17:29:23	Invisível				
andre141	O utilizador andre141 consultou os recursos do tipo Exame	2022-06-26 17:29:21	Invisível				
andre141	O utilizador andre141 consultou os recursos do tipo Teste	2022-06-26 17:29:19	Invisível				
andre141	O utilizador andre141 inseriu o ficheiro tifp do tipo Slide	2022-06-26 17:29:17	Visível				
andre141	O utilizador andre141 efetuou login	2022-06-26 17:28:58	Invisível				
admin1	O utilizador admin1 efetuou login	2022-06-26 17:28:46	Invisível				
admin1	O utilizador admin1 consultou os recursos do tipo Teste	2022-06-26 14:23:45	Invisível				

5.11 Edição de utilizadores

Informação do utilizador selecionado				
Name	Username	Password	Level	
André Martins	andre141	123	producer	
Submeta o novo username:				
<input type="text"/>				
Submeta o novo nome:				
<input type="text"/>				

5.12 Erros na inserção de ficheiros

Lista de erros ao efetuar o upload	
Os ficheiros só podem ter o tipo PDF ou XML	
O ficheiro manifesto.txt não se encontra válido	
Não existe o ficheiro RRD-SIP.json	

6 Conclusão

Dado por concluído o projeto prático da Unidade Curricular, cremos que atingimos com sucesso os objetivos pretendidos para a realização deste projeto. O desenvolvimento da plataforma revelou-se bastante desafiante, uma vez que foi a aplicação de maior dimensão que tínhamos feito até ao momento, quer no desenvolvimento de frontend, quer no desenvolvimento de backend, contudo os desafios propostos ajudaram sempre a querer ir mais além e ir adicionando cada vez mais funcionalidades à nossa aplicação.

Como aspecto menos positivos apontamos a pouca utilização de JQuery, o que teria tornado a nossa aplicação visualmente mais estética e com um maior dinamismo, contudo cremos ter aplicado com bastante sucesso todo o conhecimento e material utilizado durante todo o semestre nas aulas práticas.

Assim, ficamos satisfeitos com o resultado final alcançado, tendo este projeto despertado em nós ainda mais o "bichinho" do desenvolvimento web e da construção de aplicação cada vez maiores e mais robustas.