

Universidade do Minho

Mestrado Integrado em Engenharia Informática

Desenvolvimento de Aplicações Web



Adriana Meireles
A82582



Gonçalo Esteves
A85731

7 de fevereiro de 2021

Conteúdo

1	Introdução	2
2	Arquitetura da Aplicação	3
3	Base de Dados	4
3.1	Utilizadores	4
3.2	Recursos	4
3.3	Tipos	5
4	Funcionalidades do Sistema	6
4.1	Segurança	6
4.2	Armazenamento dos Recursos	6
4.3	Funcionalidades disponibilizadas ao Consumidor	6
4.4	Funcionalidades disponibilizadas ao Produtor	7
4.5	Funcionalidades disponibilizadas ao Administrador	7
5	Conclusão	9

1 Introdução

No âmbito da Unidade Curricular Desenvolvimento de Aplicações *Web* foi proposta a construção de uma Plataforma de Disponibilização de Recursos Educativos que permitisse a inserção, visualização, remoção de recursos bem como fazer comentários e pôr gostos nos mesmos, entre outras funcionalidades. Para tal, primeiramente foi idealizada uma possível arquitetura do nosso sistema. Posteriormente, foi pensado o formato dos documentos em Mongo para a apresentação dos dados. Seguidamente, procedeu-se à elaboração das funcionalidades do nosso sistema, com base nos documentos feitos anteriormente, dependendo do nível do utilizador. Por fim, foi feita uma apreciação crítica ao trabalho realizado.

2 Arquitetura da Aplicação

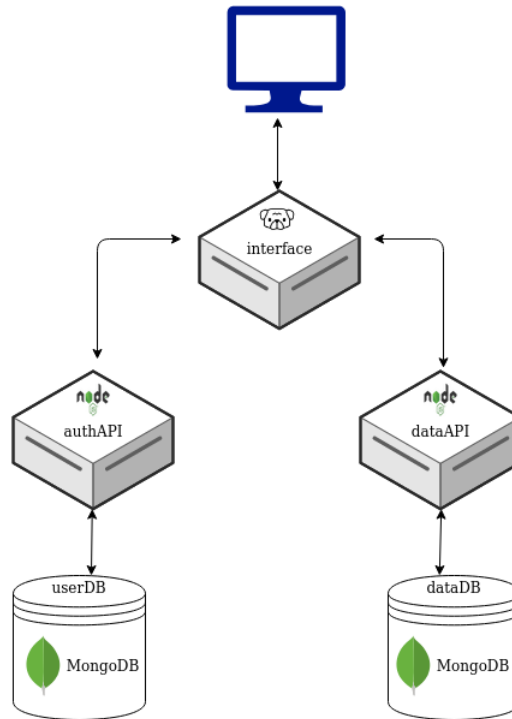


Figura 1: Arquitetura do sistema

Na figura acima está representada a arquitetura da aplicação.

Para o cliente visualizar as páginas *web* da interface, recorreu-se à ferramenta *Pug*.

No diagrama pode observar-se que existem dois serviços essenciais de backend da aplicação. Primeiramente, temos o *Auth Api*. Este microserviço é responsável por servir os dados relativos a todos os utilizadores do sistema. Também fornece os serviços necessários para a autenticação, mais especificamente através da posse de um *jwt*. Os dados foram guardados numa base de dados em *MongoDB*.

A *Data Api* trata dos dados dos recursos da aplicação, isto é, diferentes tipos de recursos, os recursos em si, comentários a recursos, gostos a recursos, entre outros dados. Os dados foram também guardados numa base de dados em *MongoDB*.

3 Base de Dados

No presente capítulo são apresentados os formatos dos documentos em *MongoDB* usados para assegurar a persistência dos dados do sistema. De forma sucinta, explica-se a que correspondem os campos de cada documento.

3.1 Utilizadores

Os utilizadores do sistema vão ser quem vai utilizar o sistema de modo a usufruir das suas funcionalidades. Para tal, os dados descritos abaixo são guardados pelo sistema:

- **nome:** Nome do utilizador (String);
- **email:** Email do utilizador (String);
- **filiacao:** Ligação ou cargo desempenhado, como estudante, docente, curso, departamento, entre outros (String);
- **nivel:** Nível de acesso do utilizador (Number);
- **dataRegisto:** Data de registo no sistema (String);
- **dataUltimoAcesso:** Data da última vez que acedeu ao sistema (String);
- **password:** Password do utilizador (String).

3.2 Recursos

Os utilizadores poderão fazer operações sobre recursos educativos. Deste modo, um recurso tem o seguinte formato na base de dados:

- **tipo:** Lista com os tipos do recurso (Lista de Strings);
- **titulo:** Título de um recurso (String);
- **subtitulo:** Subtítulo de um recurso (String);
- **dataCriacao:** Data que o recurso foi criado (String);
- **dataRegisto:** Data de registo do recurso no sistema (String);
- **visibilidade:** Visibilidade de um recurso. Pode ser público ou privado (Booleano);

- **nomeProdutor:** Nome do utilizador que submeteu o recurso (String);
- **emailProdutor:** Email do utilizador que submeteu o recurso (String);
- **nomeFicheiro:** Nome do ficheiro que o utilizador pretende submeter (String);
- **validado:** Valor que permite saber se um recurso está ou não validado pelo administrador (Booleano);
- **likes:** Lista com os emails dos utilizadores que colocaram gosto num recurso (Lista de Strings);
- **comentários:** Lista com o nome do utilizador e o respetivo comentário que fez a um recurso (Lista de Pares de Strings).

3.3 Tipos

Os administradores poderão adicionar novos tipos de recurso que possuem o seguinte formato:

- **tipo:** Nome do tipo que o administrador pretende adicionar (String).

4 Funcionalidades do Sistema

4.1 Segurança

Por forma a utilizar a aplicação, o utilizador deverá dirigir-se à página de registo, de modo a criar uma conta. Depois tem de fazer login, sendo aqui criados os cookies que guardam o email, nível de acesso e o token, necessários para validar algumas das operações a efetuar.

Relativamente à segurança nos servidores, para a interface conseguir comunicar com as outras duas APIs é necessário usar o JSON Web Token que foi gerado pela *authAPI* (nas routes) e que é verificado na *dataAPI* (no app.js). Em cada pedido ao servidor *dataAPI*, verifica-se não só o token mas também se o utilizador tem o nível de acesso necessário para o pedido ser realizado.

4.2 Armazenamento dos Recursos

De modo a armazenar os recursos no servidor *dataAPI*, estruturamo-los de forma semelhante a uma "bag" do género *BagIt*. Desta forma, podemos assegurar uma maior segurança no armazenamento dos dados, já que estes deverão respeitar uma estrutura pré-definida.

Após um utilizador submeter um recurso, este será guardado numa *bag*, que virá posteriormente a ser *zipada* e enviada ao servidor *dataAPI*, formando assim um SIP. Ao receber os SIP's, o *dataAPI* guarda-os numa pasta temporária, até serem aprovados por um administrador. Aquando da aprovação do recurso, o SIP é *unzipped* e verifica-se se toda a estrutura do mesmo respeita aquela que está pré-definida para pastas *BagIt*, bem como se os valores de *checksum* enviados nos ficheiros de manifesto correspondem aos obtidos com os recursos enviados. Caso tudo isto se confirme, a pasta com os recursos é guardada permanentemente, passando a ser um AIP, e o SIP é descartado. Por fim, sempre que há uma requisição de um recurso, toda a sua pasta é novamente *zipada* e enviada ao utilizador que a pediu, sendo estes os nossos DIP's.

4.3 Funcionalidades disponibilizadas ao Consumidor

As funcionalidades disponibilizadas ao consumidor são:

- Pode ver e fazer download de todos os recursos públicos;
- Pode fazer comentários em todos os recursos públicos;

- Pode pôr gosto nos recursos públicos;
- Pode fazer uma procura por tipo de modo a encontrar o que pretende;
- Pode ordenar os recursos de diferentes formas.

4.4 Funcionalidades disponibilizadas ao Produtor

As funcionalidades disponibilizadas ao produtor são:

- Pode adicionar um novo recurso;
- Pode ver e fazer download de todos os recursos públicos e recursos privados que tenha adicionado;
- Pode fazer comentários em todos os recursos públicos e recursos privados que tenha adicionado;
- Pode eliminar os comentários nos seus recursos;
- Pode eliminar os recursos se tiver sido ele próprio a adicionar;
- Pode pôr gosto nos recursos públicos e recursos privados que tenha adicionado;
- Pode fazer uma procura por tipo de modo a encontrar o que pretende;
- Pode ordenar os recursos de diferentes formas.

4.5 Funcionalidades disponibilizadas ao Administrador

As funcionalidades disponibilizadas ao administrador são:

- Pode aprovar um recurso;
- Pode adicionar um novo recurso;
- Pode adicionar um novo tipo de recurso;
- Pode ver todos os recursos tanto públicos como privados;
- Pode eliminar os comentários sobre qualquer recurso;
- Pode fazer download dos recursos que quiser;
- Pode eliminar os recursos que quiser;

- Pode pôr gosto nos recursos que quiser;
- Pode fazer uma procura por tipo de modo a encontrar o que pretende;
- Pode ordenar os recursos de diferentes formas.

5 Conclusão

A realização deste projeto permitiu não só pôr em prática, mas também aprofundar o nosso conhecimento no que toca às ferramentas leccionadas nesta unidade curricular, bem como outras ferramentas que decidimos explorar, o que foi bastante desafiador mas ao mesmo tempo interessante e enriquecedor.

O objetivo foi cumprido, tendo sido criado uma aplicação web cuja finalidade é ser uma Plataforma de Gestão e Disponibilização de Recursos Educativos, sendo que a maioria dos requisitos básicos foram implementados com sucesso e, para além destes, conseguimos ainda implementar outros tais como a barra de procura (*search bar*) e a possibilidade de ordenar recursos de diferentes formas.