



Universidade do Minho

UNIVERSIDADE DO MINHO

DESENVOLVIMENTO DE APLICAÇÕES WEB

Plataforma de Gestão e Disponibilização de Recursos Educativos

Autores:

Carlos Magalhães, PG42817

João Soares, PG42835

Tiago Gonçalves, PG42851

7 de fevereiro de 2021

Conteúdo

1	Introdução	3
2	Estrutura dos dados	3
3	Objetivos	3
4	Open Archival Information System	4
5	Autenticação	5
6	Autorização	5
7	Conclusão	6

1 Introdução

Neste trabalho abordamos os vários conteúdos lecionados ao longo do semestre na unidade curricular de Desenvolvimento de Aplicações Web. Desde a persistência dos dados à alteração da estrutura do DOM.

Para este projeto desenvolvemos uma plataforma Web de gestão e disponibilização de recursos educativos. Esta ferramenta permite a qualquer membro da comunidade académica publicar recursos educativos partilháveis com o resto da comunidade.

Esta plataforma Web foi desenvolvida utilizando a *framework Express.js* do *Node.js* e a persistência dos dados foi feita através de uma base de dados não relacional chamada *MongoDb*. A interface da nossa aplicação foi criada através da ferramenta *Pug* que compila o código para uma função de *javascript* e que retorna uma página HTML. De forma a tornar a nossa plataforma apresentável ao utilizador, utilizamos a *framework* de estilos do W3.CSS.

2 Estrutura dos dados

Como já foi referido anteriormente a nossa plataforma Web faz a persistência dos dados através de uma base de dados MongoDB e para que fosse possível preservar os dados criamos dois *model schemas* que definem a estrutura usada para guardar os dados. Para isso criamos um *model schema* para os utilizadores e outro para as publicações.

```
{
  "_id" : ID do Utilizador,
  "mail" : Email do Utilizador,
  "name" : Nome do Utilizador
  "course" : Curso do Utilizador,
  "activity" : Atividade do Utilizador,
  "department" : Departamento do Utilizador,
  "password_enc" : Password Encriptada,
  "data_register" : Data de Registo,
  "data_last_login" : Data do último login,
  "role" : Papel do Utilizador na aplicação,
  "pedido_produto" : Se tem pedido para ser produtor ativo
}
```

Figura 1: Model Schema do Utilizador

```
{
  "_id" : ID da Publicação
  "resources" : [],
  "comments" : [],
  "author" : Mail do autor da publicação,
  "theme" : Tema da publicação,
  "description" : Descrição da publicação,
  "visibility" : Visibilidade da publicação,
  "data_created" : Data de criação da publicação,
  "pub_rating" : Rating da publicação,
}
```

Figura 2: Model Schema da Publicação

3 Objetivos

Como foi explicado anteriormente, o objectivo do trabalho é criar uma plataforma WEB de gestão e disponibilização de recursos educativos. Tivemos por base o trabalho realizado ao longo das aulas para desenvolver os seguintes objetivos.

- *Disponibilização recursos educativos de vários tipos: livros, artigos, aplicações, trabalhos de alunos, monografias, relatórios;*
- *Permissão para adicionar novos tipos de recursos e novos recursos;*

- *Permissão para a criação de publicações por parte dos utilizadores;*
- *Organização dos recursos por ano, tipo, tema;*
- *Permissão para comentar as publicações;*
- *Criação de um sistema de ranking para os recursos;*
- *Criação de um Open Archival Information System;*
- *Criação de um sistema de autenticação;*
- *Criação de um sistema de autorização;*
- *Apresentação de notícias;*

4 Open Archival Information System

O Open Archival Information System (OAIS) é um modelo padrão da preservação de documentos digitais. Este modelo possui uma estrutura e uma linguagem própria que é aceite pelos especialistas de preservação de arquivos, tendo sido aceite pelo ISO (International Organization for Standardization) como um modelo standard. O conceito principal de um OAIS é o conceito de pacote, que não é mais do que um conjunto de informação estruturada.

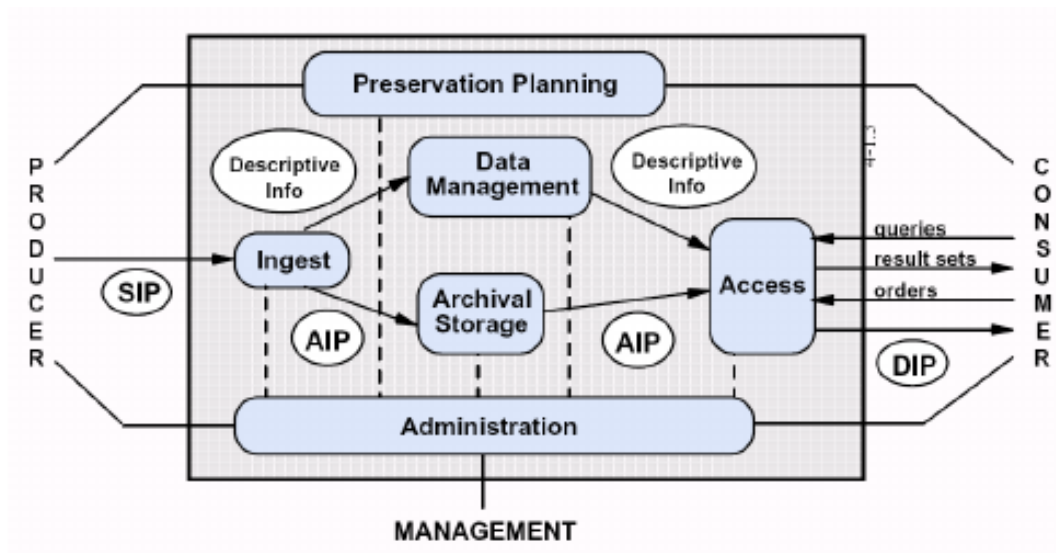


Figura 3: *Open Archival Information System*

Dentro de um OAIS existem vários tipos de pacotes. Sempre que um produtor pretende enviar ficheiros para o sistema de arquivos, é criado um pacote SIP que é o pacote responsável pela submissão de um ficheiro e dos seus metadados no sistema. O SIP após ser submetido no sistema origina um ou mais AIPs que são os pacotes responsáveis pelo armazenamento da informação. Sempre que um consumidor efetua um pedido ao sistema, seja ele uma *query* ou um pedido para a obtenção de um documento digital, o OAIS origina um pacote DIP com a informação estruturada que permite responder ao pedido do consumidor podendo o DIP conter a resposta a uma *query* ou um documento digital comprimido.

Para conseguirmos criar o nosso sistema de arquivos de informação, utilizámos o *Baglt*. A nossa implementação consiste então na utilização de diretorias zipadas com a estrutura de um *Baglt* simples. A estrutura que adotámos foi a seguinte: dentro da diretoria base que é zipada posteriormente, são criados dois ficheiros de texto informativos e uma subdiretoria que contém os ficheiros que se pretendem enviar. Os dois ficheiros de texto informativos são um ficheiro com informações sobre o formato e versão do *Baglt* implementado e o outro ficheiro contém uma linha para cada ficheiro a enviar, em que cada linha tem o código hash após aplicado o algoritmo *sha512* sobre o nome do ficheiro respetivo e a localização do ficheiro respetivo dentro do pacote.

5 Autenticação

A autenticação na nossa plataforma Web é um requisito para o trabalho, por isso decidimos recorrer a uma ferramenta do *Node.js* conhecida como *Passport.js* para implementarmos um sistema de autenticação local na nossa plataforma. Sempre que um utilizador pretende autenticar-se na nossa plataforma através da *interface de login*, é lhe pedido para preencher um formulário com o nome do utilizador e com a respetiva password da conta. Consequentemente inicia-se o processo de autenticação através do *passport*. Através desta ferramenta criamos o nosso sistema de autenticação que verifica se o utilizador se encontra registado na base de dados. Caso se confirme verifica se a palavra passe introduzida, após ser encriptada corresponde à palavra passe armazenada na base de dados correspondente a esse utilizador. Caso a autenticação se complete, os dados do utilizador são serializados e é criada uma *cookie* de sessão que será armazenada numa pasta juntamente com as restantes *cookies* de sessão dos outros utilizadores e enviada na resposta para ser armazenada também do lado do cliente. Esta *cookie* será válida e dará acesso aos utilizadores às várias páginas até que a mesma expire.

6 Autorização

Outro dos objetivos deste trabalho foi a criação de um sistema de autorização que consiste em atribuir um papel a cada um dos utilizadores da plataforma de forma a

definir as permissões de cada um dos papeis na plataforma

Para tal foram criados quatro papeis na nossa plataforma: Administrador; Produtor; Consumidor; Anónimo. Os utilizadores Anónimos são todos os utilizadores da plataforma que ainda não efetuaram o registo e por isso apenas tem permissões para aceder às páginas de registo e *login*. Os consumidores são todos os utilizadores registados na plataforma que não têm permissão para criar publicações e apenas podem consultar as publicações públicas. De forma a que estes possam fazer publicações foi criado um mecanismo que permite aos Consumidores efetuar o pedido aos Administradores da plataforma para alterarem o seu papel na plataforma para Produtor. Como referido um Produtor pode criar novas publicações, tal como o Administrador. No entanto um Administrador para além de todas as permissões de um Produtor também tem acesso a outras operações exclusivas tal como descrito na Tabela 1.

Tabela 1: Tabela de permissões por Papel

	Anónimo	Consumidor	Produtor	Administrador
Página Inicial	X	X	X	X
Registar	X			
Login		X	X	X
Mural		X	X	X
Notícias		X	X	X
Comentar		X	X	X
Avaliar		X	X	X
Publicar			X	X
Aceitar Produtores				X
Editar Publicações			X	X
Apagar Publicações			X	X

7 Conclusão

Com este trabalho conseguimos consolidar as nossas capacidades de desenvolvimento Web e ao cumprir todos os objetivos percebemos que possuímos as ferramentas necessárias para continuarmos a evoluir nesta área de forma autónoma, aprendendo por exemplo, a criar novas APIs de dados utilizando outras ferramentas.

Embora a nossa plataforma cumpra com todos os objetivos, consideramos que ainda poderíamos melhorar a mesma. Por exemplo poderíamos dividir a plataforma em dois servidores distintos: *front-end* e *back-end*. Isto levaria a um melhor encapsulamento de código, permitindo uma maior escalabilidade e portabilidade.