



Universidade do Minho
Escola de Engenharia

Representação e Processamento de Conhecimento na Web

Projeto Final: Repositório de Recursos Didáticos

Guilherme Martins PG47225
Tiago Freitas PG47687



17 de junho de 2022

Conteúdo

1	Introdução	1
2	Requisitos	2
3	Estratégia e objetivos	4
4	Arquitetura	5
5	Web Server	6
5.1	Rotas e funcionalidades	6
6	API Server	8
6.1	Base de dados	8
6.1.1	Recursos (File_Info)	8
6.1.2	Avisos	8
6.1.3	Logs	8
6.2	Rotas e funcionalidades	9
7	Authentication Server	10
7.1	Base de dados	10
7.1.1	Utilizador	10
7.2	Rotas e funcionalidades	10
8	SIP - AIP - DIP	11
9	Permissões	12
10	Utilização e visão geral	13
11	Análise e Conclusão	15

Lista de Figuras

1	Modelo de referência OAIS	2
2	Arquitetura da aplicação	5
3	Página principal	13
4	Página de um recurso	13
5	Página de utilizadores	13
6	Perfil de um utilizador	14
7	Logs	14

1 Introdução

No âmbito da Unidade Curricular de Representação e Processamento de Conhecimento na Web, foi-nos proposto um trabalho prático onde desenvolvêssemos uma aplicação Web que suportasse um repositório de recursos didáticos (RRD), assistindo na atualização, manutenção e distribuição dos mesmos.

Na realização deste projeto foram utilizadas tecnologias lecionadas nesta cadeira este semestre, recorrendo à *framework* Express e utilizando as técnicas de autenticação e gestão de rotas e ficheiros aprendidas.

Ao longo do relatório, de forma a ajudar a compreender e explicar todas as decisões tomadas que nos levaram a conseguir concluir todas as tarefas pedidas, exibiremos que a aplicação Web desenvolvida compreende as funcionalidades necessárias para o seu bom funcionamento como autenticação dos utilizadores e o seu nível de controlo da aplicação, *upload* e *download* de recursos, consultas de recursos e avisos, fazer um comentário acerca dos recursos e registo, consulta e processamento dos logs. Através do seu nível de controlo da aplicação, um admin que pertence ao nível mais alto é ainda capaz de gerir utilizadores, recursos e avisos.

2 Requisitos

O objetivo principal deste projeto passa por criar um arquivo, seguro e com controlo de acesso, de recursos didáticos, que permite um acesso direto a um dado recurso em qualquer momento ou lugar com a devida conexão à internet.

Tem-se por objetivo que a versão final da aplicação desenvolvida tenha as seguintes principais funcionalidades:

- autenticação de utilizadores
- informação pública e privada
- gestão dos recursos adicionados ao repositório
- possibilidade de comentar os recursos disponíveis
- notificações

Além disso, o projeto deve seguir o seguinte modelo:

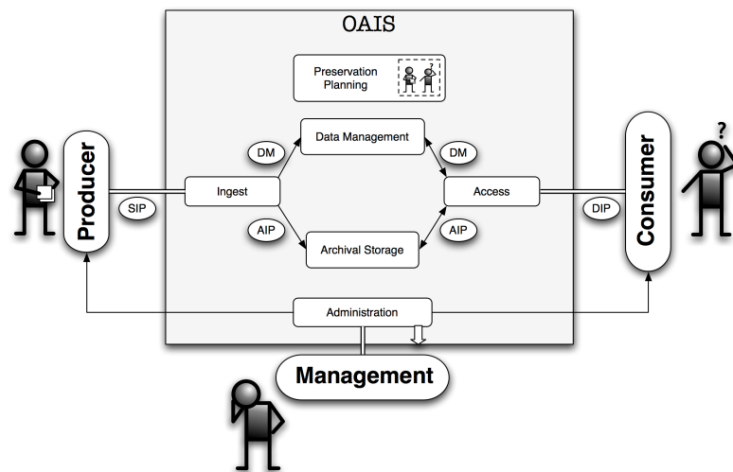


Figura 1: Modelo de referência OAIS

Portanto, devem existir 3 tipos de utilizadores/atores, de acordo com as permissões correspondentes:

- Consumidor: pode consultar, pesquisar informação e descarregar informação.
- Produtor: poder depositar recursos no repositório
- Administrador: tem acesso a todas as operações da aplicação e é responsável pela manutenção do sistema

De acordo com o OAIS, devem, também, existir 3 tipos de pacotes de informação:

- SIP: enviado pelo produtor ao sistema para ser processado e arquivado
- AIP: pacote arquivado, resultante do processamento e armazenamento do SIP
- DIP: pacote oferecido ao consumidor

3 Estratégia e objetivos

Antes de nos debruçarmos em implementações extras, o nosso foco principal no desenvolvimento da aplicação foi conseguirmos implementar todas as funcionalidades e requisitos pretendidos da forma mais eficiente possível sem despendar muitos esforços na parte estética e visual da aplicação. No entanto, e relativamente à interface com o utilizador, o objetivo consistiu em facilitar ao máximo a experiência do utilizador, com todas as funcionalidades a que este tem acesso evidenciadas e apresentadas da forma mais clara.

A nossa estratégia passou por proteger as rotas que cada tipo de utilizador tem permissão de acesso, quer visualmente, ocultando botões ou links para que não seja permitido ao utilizador sem autorização aceder a essas funcionalidades, por exemplo de controlo, quer ao nível das rotas/requests, verificando o nível do utilizador atual.

Em relação ao registo de um utilizador, nós optamos por permitir a qualquer pessoa definir o seu estatuto, ou seja, qualquer utilizador poderia se registar como administrador e ter acesso a todas as funcionalidades. Fizemos esta abordagem ao problema apenas para facilitar os nossos testes no decorrer do projeto.

4 Arquitetura

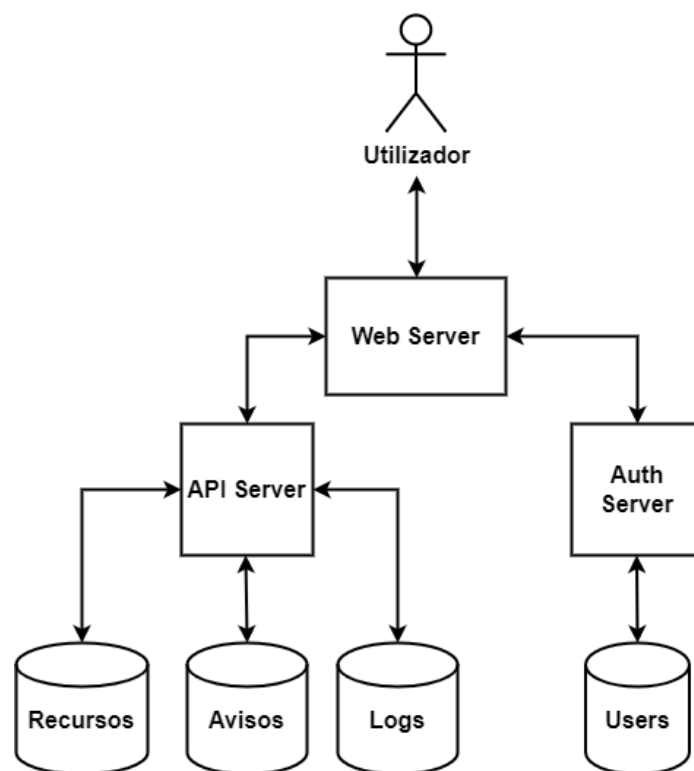


Figura 2: Arquitetura da aplicação

5 Web Server

Esta camada é responsável por fazer a ponte entre a aplicação e o utilizador, no sentido em que, apresenta uma interface gráfica e recebe todo o tipo de pedidos por parte dos utilizadores. Posteriormente, processará estes pedidos e encaminhá-los-á (de acordo com o tipo de pedido) para a *API Server* ou para a *Auth Sever*, que irão responder com um objeto em formato JSON contendo a informação necessária. Por fim, este servidor define a opção gráfica para o pedido realizado, que pode ser um atualizar da página atual ou o lançamento de outra página e vai depender do nível de utilizador atual, isto é, utilizadores com diferentes níveis irão ter páginas e respostas diferentes conforme o seu nível de permissão. Em todas as rotas, à exceção das de registo e *login*, verifica-se se a sessão está iniciada, caso não esteja o utilizador é redirecionado automaticamente para a página de *login*.

5.1 Rotas e funcionalidades

- **GET /rrd/registar** - pedido da página de registo
- **POST /rrd/registar** - tentativa de registo de novo utilizador
- **GET /rrd/login** - pedido da página de *login*
- **POST /rrd/login** - tentativa de *login*
- **GET /rrd/logout** - pedido para terminar sessão
- **GET /rrd** - pedido da página principal (conteúdo depende do nível do utilizador, pode ter secção de *upload*, recursos e avisos) ou da página relativa a um tipo de recursos (na barra de navegação aparecem listados apenas os tipos existentes no momento)
- **GET /rrd/users** - pedido da lista de utilizadores do sistema
- **GET /rrd/users/editar/:id** - pedido da página de edição de um determinado utilizador
- **POST /rrd/users/alterar/:id** - tentativa de guardar as alterações feitas a um determinado utilizador
- **GET /rrd/users/delete/:id** - tentativa de remoção de um determinado utilizador
- **GET /rrd/users/:id** - pedido da página relativa a um determinado utilizador
- **GET /rrd/perfil** - pedido da página do perfil do utilizador atual
- **GET /rrd/perfil/editar** - pedido da página de edição do perfil do utilizador atual
- **POST /rrd/perfil/alterar** - tentativa de guardar as alterações feitas ao perfil do utilizador atual
- **GET /rrd/perfil/delete** - pedido de remoção do perfil do utilizador atual
- **POST /rrd/upload** - tentativa de *upload* de um recurso
- **GET /rrd/recursos/download/:rid** - pedido de *download* de um determinado recurso
- **POST /rrd/aviso** - tentativa de inserção de um aviso

- **GET** /rrd/recursos - pedido da página com todos os recursos
- **GET** /rrd/avisos - pedido da página com todos os avisos
- **GET** /rrd/recursos/:rid - pedido da página relativa a um determinado recurso
- **GET** /rrd/avisos/:aid - pedido da página relativa a um determinado aviso
- **POST** /rrd/search - tentativa de pesquisa de recursos (barra de pesquisa)
- **GET** /rrd/recursos/editar/:rid - pedido da página de edição de um determinado recurso
- **POST** /rrd/recursos/alterar/:rid - tentativa de guardar as alterações feitas a um determinado recurso
- **GET** /rrd/recursos/delete/:rid - tentativa de remoção de um determinado recurso
- **POST** /rrd/recursos/comentar/:rid - tentativa de inserção de um comentário num determinado recurso
- **GET** /rrd/recursos/comentarios/:rid/delete/:cid - tentativa de remoção de um determinado comentário num determinado recurso
- **GET** /rrd/avisos/editar/:aid - pedido da página de edição de um determinado aviso
- **POST** /rrd/avisos/alterar/:aid - tentativa de guardar as alterações feitas a um determinado aviso
- **GET** /rrd/avisos/delete/:aid - tentativa de remoção de um determinado aviso
- **GET** /rrd/logs - pedido da página dos *logs*
- **GET** /rrd/logs/download - pedido de *download* dos *logs*
- **GET** /rrd/logs/reset - pedido para reiniciar o sistema de *logs*

De salientar que o mais correto seria a página relativa a um tipo de recursos se enquadrar na rota /rrd/recursos.

6 API Server

Esta camada é responsável por responder aos pedidos do *Web Server* relativos aos recursos, avisos e *logs* conectando-se às respectivas coleções da nossa base de dados do MongoDB (RRD). Ao nível da segurança e proteção, foi utilizado o *jsonwebtoken*, o que faz com que todos os pedidos que chegam só sejam processados e respondidos se neles estiver contido um *token* válido. Se tal acontecer, em alguns pedidos, é verificado o nível do utilizador de forma a poder autorizar ou não a operação em questão.

6.1 Base de dados

6.1.1 Recursos (File_Info)

Informação acerca dos recursos existentes no *file system*. Cada entrada é constituída por:

- **titulo:** nome do recurso
- **nome:** nome do ficheiro
- **tipo:** tipo de recurso
- **tamanho:** tamanho em bytes do recurso
- **extensao:** extensão dos ficheiros constituintes do recurso
- **ficheiros:** nomes dos ficheiros constituintes do recurso
- **data_criacao:** data de criação do recurso
- **data_submissao:** data de submissão do recurso
- **criador:** criador do recurso
- **utilizador:** utilizador que submeteu o recurso
- **descricao:** descrição do recurso
- **comentarios:** comentários do recurso (cada um possui id, autor, texto e data de submissão)

6.1.2 Avisos

Cada aviso contem:

- **autor:** utilizador que publicou o aviso
- **texto:** texto do aviso
- **data_submissao:** data de submissão do aviso

6.1.3 Logs

Cada *log* é constituído por:

- **texto:** texto do *log*
- **data:** data de realização da ação do *log*

6.2 Rotas e funcionalidades

- **GET** `/api/recursos` - pedido para listar recursos
- **POST** `/api/recursos` - tentativa de inserção de um recurso
- **GET** `/api/recursos/tipos` - pedido para listar os tipos dos recursos existentes
- **GET** `/api/recursos/path/:rid` - pedido para consultar a *path* de um determinado recurso
- **POST** `/api/recursos/comentar/:rid` - tentativa de inserção de comentário num determinado recurso
- **DELETE** `/api/recursos/comentarios/:rid/delete/:cid` - tentativa de remoção de um determinado comentário de um determinado recurso
- **GET** `/api/recursos/:rid` - pedido para consultar a informação de um determinado recurso
- **PUT** `/api/recursos/:rid` - tentativa de atualização da informação de um determinado recurso
- **DELETE** `/api/recursos/:rid` - tentativa de remoção de um determinado recurso
- **GET** `/api/avisos` - pedido para listar avisos
- **POST** `/api/avisos` - tentativa de inserção de um aviso
- **GET** `/api/avisos/:aid` - pedido para consultar um determinado aviso
- **PUT** `/api/avisos/:aid` - tentativa de atualização de um determinado aviso
- **DELETE** `/api/avisos/:aid` - tentativa de remoção de um determinado aviso
- **GET** `/api/logs` - pedido para consultar os *logs*
- **POST** `/api/logs` - tentativa de inserção de um *log*
- **DELETE** `/api/logs` - tentativa de *reset* dos *logs*

7 Authentication Server

Por fim, esta camada engloba todo o tipo de operações e funcionalidades associadas ao registo, autenticação e administração de utilizadores, guardados na nossa base de dados. Tal como a *API Server*, está protegida através da utilização de *jsonwebtoken* e é ocorre a verificação de permissões para operações que necessitem de tal.

7.1 Base de dados

7.1.1 Utilizador

Na base de dados cada utilizador contem:

- **username:** *username* do utilizador
- **email:** *email* do utilizador
- **password:** *password* do utilizador
- **level:** nível de permissão do utilizador (administrador, produtor ou consumidor)
- **data_registo:** data de registo do utilizador

7.2 Rotas e funcionalidades

- **GET /users** - pedido para listar utilizadores
- **POST /users/registar** - tentativa de registo de um utilizador
- **POST /users/login** - tentativa de *login* de um utilizador
- **GET /users/perfil/:username** - pedido da página do perfil do utilizador atual
- **PUT /users/perfil/:username** - tentativa de atualização da informação do perfil do utilizador atual
- **DELETE /users/perfil/:username** - pedido de remoção do perfil do utilizador atual
- **GET /users/:id** - pedido para consultar um determinado utilizador
- **PUT /users/:id** - tentativa de atualização da informação de um determinado utilizador
- **DELETE /users/:id** - tentativa de remoção de um determinado utilizador

8 SIP - AIP - DIP

Um pacote SIP para ser considerado válido e inserido na sistema tem que possuir a seguinte estrutura:

- ***RRD.json***: ficheiro que contem os metadados (iniciais) do recurso (título, tipo, data de criação, produtor e descrição)
- ***manifesto.txt***: ficheiro que contem o nome de todos os ficheiros pertencentes ao recurso (incluindo ele próprio e o ficheiro mencionado anteriormente)
- os documentos que se pretendem inserir no repositório (*.xml* ou *.pdf*)

Caso o *zip* seja válido, este é incluído no sistema, ou seja, é guardado no *file system* na diretoria corresponde ao seu tipo (é criada se ainda não existir) e os metadados são introduzidos na base de dados. Aos metadados iniciais são adicionados mais alguns (no ficheiro também), de forma a completar o objeto de acordo com o *schema* de um recurso já apresentado. A isto chama-se a passagem de um SIP para um AIP.

Sempre que existe a edição de um recurso o ficheiro *RRD.json* também é alterado, de forma a ficar sempre igual à respetiva entrada existente da base de dados e quando um recurso é removido, é apagado a entrada e o *zip* existente no *file system*.

Por fim, o processo de *download* (DIP) foi tornado muito mais fácil pois basta transferir o *zip* pretendido que nele estarão os ficheiros pretendidos e toda a informação acerca do recurso desde o momento da sua criação até ao momento do *download*.

9 Permissões

- **Inserção de recursos:** Administradores e produtores
- **Consulta de recursos:** Qualquer utilizador
- **Edição e remoção de um recurso:** Administrador e produtor desse recurso
- **Consulta de avisos:** Qualquer utilizador
- **Edição e remoção de um aviso:** Administrador e autor desse aviso
- **Comentar:** Qualquer utilizador
- **Eliminar comentário:** Administrador, produtor do recurso e autor do comentário
- **Administração de utilizadores:** Administrador
- **Visualização e administração do perfil:** Qualquer utilizador
- **Logs:** Administrador

De salientar que, para facilitar testes, dá-se a escolher no momento de registo, o nível de permissões ao utilizador, algo que não faz sentido se a aplicação for, de facto, usada.

10 Utilização e visão geral

Os servidores podem ser iniciados com os comandos *npm start* ou *npm run dev* (reinicia quando ocorre atualização no código) e correm nas seguintes portas:

- **Web Server:** porta 4100
- **API Server:** porta 4000
- **Auth Server:** porta 4200

De seguida vamos apresentar as páginas que consideramos mais importantes.

Título	Ficheiro	Tipo	Nº Ficheiros	Tamanho	Utilizador	Data de criação	Data de submissão	
Fichas2015	fichas_PG47687.zip	Fichas	3	6323 bytes	admin	2022-06-08	2022-06-30 14:14	Detalhes Download Editar Eliminar
Testes2021	testes_PG47225.zip	Teste	3	8973 bytes	producer	2022-06-08	2022-06-30 14:14	Detalhes Download Editar Eliminar
meus cvs	curriculos_TagaPreta.zip	Curriculos	2	318203 bytes	admin	2022-06-15	2022-06-30 14:14	Detalhes Download Editar Eliminar

Figura 3: Página principal

Comentários
<ul style="list-style-type: none">Publicado/alterado por consumer em 2022-06-30 14:15 bom trabalho

Figura 4: Página de um recurso

Username	Email	Password	Nível	Data de registo	
consumer	consumer@gmail.com	123	Consumer	2022-06-30 14:13	Detalhes Editar Eliminar
producer	producer@gmail.com	123	Producer	2022-06-30 14:12	Detalhes Editar Eliminar

Figura 5: Página de utilizadores

Página Inicial

Recursos

Tipos de Recursos

Pesquisar recurso

Q

Avisos

Utilizadores

Logs

Meu perfil

Logout

Meu perfil

Username: admin

Email: admin@gmail.com

Password: 123

Nível: Administrador

Data de registo: 2022-06-30 14:12

Recursos publicados:

Fichas2015

meus cvs

Editar

Eliminar

Figura 6: Perfil de um utilizador

Página Inicial

Recursos

Tipos de Recursos

Pesquisar recurso

Q

Avisos

Utilizadores

Logs

Meu perfil

Logout

Logs

2022-06-30 14:13: admin eliminou um recurso (ID: 62bda0eb3f67b0b501715676) com sucesso
2022-06-30 14:13: admin eliminou um recurso (ID: 62b09f0d167b0b501715666) com sucesso
2022-06-30 14:13: admin fez login com sucesso
2022-06-30 14:14: produtor fez o upload do recurso Testes2021 com sucesso
2022-06-30 14:14: produtor fez login com sucesso
2022-06-30 14:14: admin fez logout com sucesso
2022-06-30 14:14: admin fez o upload do recurso meus cvs com sucesso
2022-06-30 14:14: admin submeteu um aviso com sucesso
2022-06-30 14:14: admin fez o upload do recurso Fichas2015 com sucesso
2022-06-30 14:15: admin fez login com sucesso
2022-06-30 14:15: consumer fez logout com sucesso
2022-06-30 14:15: consumer fez o download de um recurso (ID: 62bda19d1f67b0b5017156ca) com sucesso
2022-06-30 14:15: consumer comentou um recurso (ID: 62bda19d1f67b0b5017156ca) com sucesso
2022-06-30 14:15: consumer fez login com sucesso
2022-06-30 14:15: produtor fez logout com sucesso
2022-06-30 14:15: produtor submeteu um aviso com sucesso

Download

Report

Figura 7: Logs

De realçar que as páginas acima surgiram na sessão de um a

11 Análise e Conclusão

Dado por concluído o trabalho prático da Unidade Curricular de Representação e Processamento de Conhecimento na Web, o grupo encontra-se realizado e feliz com o trabalho desenvolvido.

Este projeto permitiu ao grupo ganhar experiência numa área que nunca tinha sido abordada no curso. Ao longo do desenvolvimento do trabalho, deparamo-nos com situações às quais tivemos de improvisar e descobrir soluções dos problemas que enfrentamos.

No decorrer do desenvolvimento do projeto, decidimos focar-nos só e apenas no bom funcionamento da aplicação para que não houvesse erros e dando menos prioridade à parte estética e visual da aplicação, daí que também não tenhamos usado *jQuery*. No que toca às dificuldades encontradas, deparamo-nos com algumas ao nível da manipulação de *zips* e de módulos responsáveis por tal, que acabou por ser a razão pela qual implementamos uma abordagem mais simples do *BagIt*.

Em suma, consideramos que todo o projeto foi positivo, uma vez que desenvolvemos e implementamos tudo aquilo que nos foi pedido, sendo um projeto muito trabalhoso, mas que acabou por ser gratificante, já que aprendemos imenso.