Projeto Final: Repositório de Recursos Didáticos (RRD)

Projeto Final: Repositório de Recursos Didáticos (RRD)

Índice

Lista de Figuras

Repositório de Recursos Didáticos (RRD)

José Carlos Ramalho

Resumo

Neste projecto pretende-se que os alunos desenvolvam uma aplicação Web que implemente e dê suporte a um repositório de recursos didáticos.

A aplicação a desenvolver deverá ter todas as caraterísticas de uma aplicação web: autenticação de utilizadores, informação pública e privada, possibilidade de comentar os recursos disponíveis, notificações, e todas as outras operações e funcionalidades que a imaginação dos alunos conseguir criar.

Convem ir verificando o historial de alterações deste documento pois o mesmo irá sofrer alterações ao longo das próximas semanas.

Introdução

No dia a dia de um professor a necessidade de publicar conteúdos de formaorganizada para os seus alunos é uma necessidade constante.

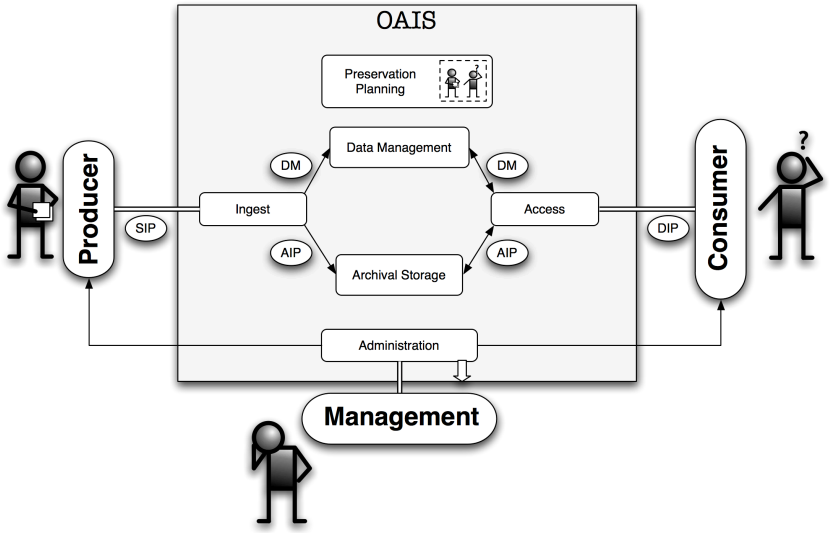
Neste projecto, pretende-se criar um arquivo seguro e com controlo de acesso, de recursos didáticos. Isto irá permitir um acesso direto a qualquer recurso em qualquer momento e lugar (desde que haja internet).

Para além deste serviço de acesso aos recursos, pretende-se que sejam implementados serviços adicionais: gestão de notícias, gestão dos utilizadores, gestão dos recursos que se vão adicionando ao repositório, possibilidade dos utilizadores comentarem e classificarem os recursos disponíveis.

O Repositório

O RRD deverá ser implementado seguindo as orientações do modelo OAIS (Figura 1, “Modelo de referência OAIS”).

Figura 1. Modelo de referência OAIS



Como se pode ver pela figura, o sistema terá de interagir com três tipos de actores:

Produtor: Corresponde a todos os que irão poder depositar recursos no repositório;

Administrador: Como o próprio nome indica é o administrador do sistema. Irá executar acções relacionadas com a manutenção do mesmo; Terá acesso a todas as operações da aplicação desenvolvida;

Consumidor: Corresponde ao futuro utilizador do repositório. Dirigir-se-á a este para consultar, pesquisar informação e descarregar informação selecionada.

Em termos funcionais, o sistema é constituído por 3 megaprocessos:

Ingestão: Processo responsável pela recepção/depósito de materiais a arquivar;

Administração: Processo responsável pela gestão interna do sistema: gestão dos objectos arquivados, gestão de utilizadores, produção de estatísticas, etc;

Disseminação: Processo responsável pela disseminação/distribuição/publicação dos objectos arquivados. Um disseminador tanto pode fornecer ao consumidor final um ZIP com a informação pretendida como pode gerar um website que permita áquele navegar no repositório e visualizar a informação pública.

De acordo com o OAIS, temos três tipos de *pacotes de informação* a circular no sistema:

SIP ("Submission Information Package"): Pacote que é enviado pelo produtor ao sistema para ser processado e arquivado. Deverá ser um arquivo comprimido ZIP. A sua estrutura é, normalmente, especificada no início do projecto, neste caso, um SIP deverá ter a seguinte constituição:

- Um manifesto especificado de acordo com a norma Bagit: Manual de referência [https://datatracker.ietf.org/doc/html/rfc8493];
- Ficheiro(s) PDF ou XML (associado a um Schema), correspondente(s) ao recurso didático que se pretende enviar ao RRD;
- Ficheiro com metainformação que será armazenada na Base de Dados e irá dar suporte às pesquisas. Campos de metainformação sugeridos:
  - data de criação
  - data de submissão
  - identificação do produtor
  - identificação de quem fez a submissão
  - Título do recurso
  - Tipo de recurso: sugere-se a criação de uma pequena taxonomia para o efeito, por exemplo, teste/exame, slides, manual, relatório, tese de mestrado, ...

;

AIP ("Archival Information Package"): O AIP corresponde ao pacote arquivado, ou seja, um SIP depois de processado e armazenado torna-se num AIP. Este terá uma determinada estrutura que irá depender da forma como será armazenado: base de dados relacional, ficheiro XML, conjunto de pares atributo valor, etc.

Neste projeto, sugere-se que a metainformação seja guardada numa base de dados em MongoDB e que os ficheiros sejam guardados no sistema de ficheiros de acordo com uma política definida pela equipe de desenvolvimento (não coloquem os ficheiros todos na mesma pasta).

DIP ("Dissemination Information Package"):

Pacote oferecido ao consumidor. Tanto poderá ser um website a partir do qual aquele consiga navegar nos conteúdos como pode ser um ficheiro ZIP com um conjunto de conteúdos previamente seleccionados.

Neste projeto, sugere-se que seja facilitada a consulta *online* para formatos suportados pelos browsers e que seja gerado um ZIP para download semelhante ao SIP.

Nas secções seguintes iremos detalhar um pouco os requisitos pretendidos para o projecto. Como estes estão muito relacionados com os pacotes de informação que circulam no sistema vamos começar por definir o que se pretende para cada um deles.

SIP e o processo de Ingestão

Começamos por definir o ponto de entrada no sistema, o SIP e o processo de ingestão que lhe está associado.

Nas primeiras aulas, discutiu-se a estrutura que deveria ter este pacote para o caso dos trabalhos de casa. Relembrando essa discussão apresentam-se os requisitos para o SIP do projeto:

- Um SIP é ficheiro comprimido no formato ZIP;
- O ZIP contem um conjunto de ficheiros e/pastas; um deles funciona como manifesto, descrevendo a estrutura e os restantes ficheiros e pastas que constituem o pacote;
- O manifesto deverá ser especificado em JSON ou XML, a escolha fica ao critério da equipa de implementação;
- O manifesto virá sempre num ficheiro de nome `RRD-SIP.(json|xml)`;
- Todos os outros ficheiros do pacote deverão estar ao nível do manifesto ou em subpastas e deverão ser todos referenciados por este.

O processo de ingestão deverá receber o ficheiro ZIP e realizar as seguintes tarefas:

- Verificar se o `RRD-SIP.(json|xml)` existe;
- Verificar se todos os ficheiros referenciados no `RRD-SIP.(json|xml)` existem no pacote enviado;
- Armazenar a metainformação do SIP na base de dados em Mongo definida para o efeito (“AIP e o armazenamento de projectos”);
- Armazenar os ficheiros do projecto na pasta correspondente na estrutura dentro do File System criada para o efeito.

Depois do processo de ingestão a informação do SIP foi armazenada e aquele foi convertido num AIP.

AIP e o armazenamento de projectos

AIP é a designação que se dá ao objecto intelectual depois deste ter ficado devidamente arquivado.

Neste projecto, o AIP irá corresponder a uma solução híbrida, com uma parte da informação a ser guardada numa base de dados em Mongo e outra no file system.

A metainformação relativa aos recursos, ou seja, a informação contida no `RRD-SIP.(json|xml)`, será guardada numa base de dados em Mongo.

Os ficheiros correspondentes aos recursos deverão ser guardados numa zona "especial" do file system. Os alunos deverão pensar numa maneira de fazer isto e numa forma de manter a ligação entre estes ficheiros e a respectiva informação guardada na base de dados.

A componente de registo de logs no sistema além de ser exibida na consola do administrador deverá também ser guardada num ficheiro cuja estrutura permita a sua consulta e processamento.

Os modelos e exemplos desenvolvidos nas aulas são apenas para orientação. Em muitas situações o aluno terá de estender os modelos especificados. Ou seja, não se limitem ao que foi discutido nas aulas, aproveitem essa informação e tentem ir mais além!

DIP e a disseminação/publicação de conteúdos

Um DIP corresponde à forma como se irão disponibilizar os conteúdos armazenados.

Para já, pretende-se que o consumidor/utilizador do sistema tenha à sua disposição um website a partir do qual possa explorar os conteúdos armazenados: listar os recursos, consultar a informação relativa a um recurso, consultar/descarregar um recurso, etc.

Uma outra forma de DIP será a possibilidade de exportação de um projecto. Neste caso, o DIP corresponderá a um ficheiro ZIP com uma estrutura semelhante ao SIP.

Processo de Administração

Além da interface de ingestão e da interface de consumo/disseminação o sistema deverá ter também uma interface de administração que deverá dar acesso ao seguinte conjunto de operações:

- Administração de utilizadores: registo, edição, remoção, listagem;
- Administração de recursos (AIPs): inserção, edição, remoção, listagem, exportação;
- Administração de notícias, a ser exibidas na página de entrada: criação, alteração, tornar visível/invisível;
- Estatísticas de utilização: processamento dos logs que dará indicadores sobre a consulta online (visualização) e descarregamento (download) dos projetos.

Arquitetura aplicacional

No seguimento do que se tem vindo a discutir nas aulas, sugere-se que os alunos sigam os seguintes passos:

- Desenvolvam uma base de dados em Mongo para assegurar a persistência dos dados;
- Desenvolvam uma primeira camada aplicacional, uma API de dados, que acedendo à base de dados responderá sempre com dados em JSON. Esta API será constituída por um modelo abstrato de acesso à base de dados, um controlador que implementa as operações de query e um roteador que recebe os pedidos e que lhes responde com os dados;
- Desenvolvam uma segunda camada aplicacional que será constituída pela implementação de uma interface para humanos. Esta camada será constituída por um roteador e várias templates das páginas

Web que se pretendem como resposta aos pedidos dos utilizadores. O fluxo será: o roteador recebe um pedido, para o satisfazer faz um pedido à API de dados, recebe os dados e envia como resposta a renderização de uma template parametrizada com os dados recebidos;

API de dados: pedidos

A API de dados deverá responder aos vários pedidos de dados. A título de exemplo apresentam-se algumas sugestões:

Sobre os Recursos:

GET /api/recursos	Devolve um array em JSON com a informação dos recursos ordenados por ordem (definida pela equipe);
GET /api/recursos/:rid	Devolve a informação em JSON relativa ao recurso com id `rid`;
GET /api/recursos?tipo=X	Devolve um array em JSON com a informação dos recursos do tipo `X`;
GET /api/recursos?q=pal	Devolve um array em JSON com a informação dos recursos que contêm `pal` no título;
POST /api/recursos	Regista um recurso no repositório;
PUT /api/recursos/:rid	Altera a informação do recurso com id igual a `rid`;

Extras

Como extra, para aqueles que gostam de desafios, pretende-se que no fim o deployment seja feito a partir de um `docker` ou um `docker-compose`.

Entrega do Projeto

De forma semelhante aos TPC, cada elemento da equipe, deverá criar uma pasta de nome "Projeto" no seu repositório do GitHub e colocar lá o projeto desenvolvido, até à meia noite de 19 de Junho de 2022.

Avaliação

A avaliação do projeto será constituída por várias componentes:

- Uma apresentação pública do projeto com o sistema a funcionar no servidor `epl.di.uminho.pt` (a responsabilidade da instalação ficará a cargo dos alunos);
- Um relatório a ser entregue ao docente antes da sessão de apresentação.