I am ... (in bits and bytes)

O meu eu digital...

Projeto Final

Engenharia Web 2025

José Carlos Ramalho

7 de Abril de 2025

Resumo

Este documento descreve o projeto final que os alunos a frequentar a UC de Engenharia Web terão de realizar.

O projeto consistirá no desenvolvimento de uma aplicação Web, constituída por um frontend (interface pública), backend (interface de administração e privada), persistência de dados em base de dados, ficheiros e outros e lógica de controlo em JavaScript.

A aplicação irá suportar o *"eu digital"* do utilizador, ou seja, será uma espécie de diário digital. Diário digital quer dizer que as operações sobre os dados têm uma cronologia associada e que a linha temporal será o eixo principal da aplicação. Em termos de conteúdos, pretende-se o máximo de possibilidades como por exemplo: fotografias, registos desportivos, crónicas, pensamentos soltos, resultados académicos, participação em eventos, organização de eventos, opiniões, comentários sobre outros recursos Web, etc.

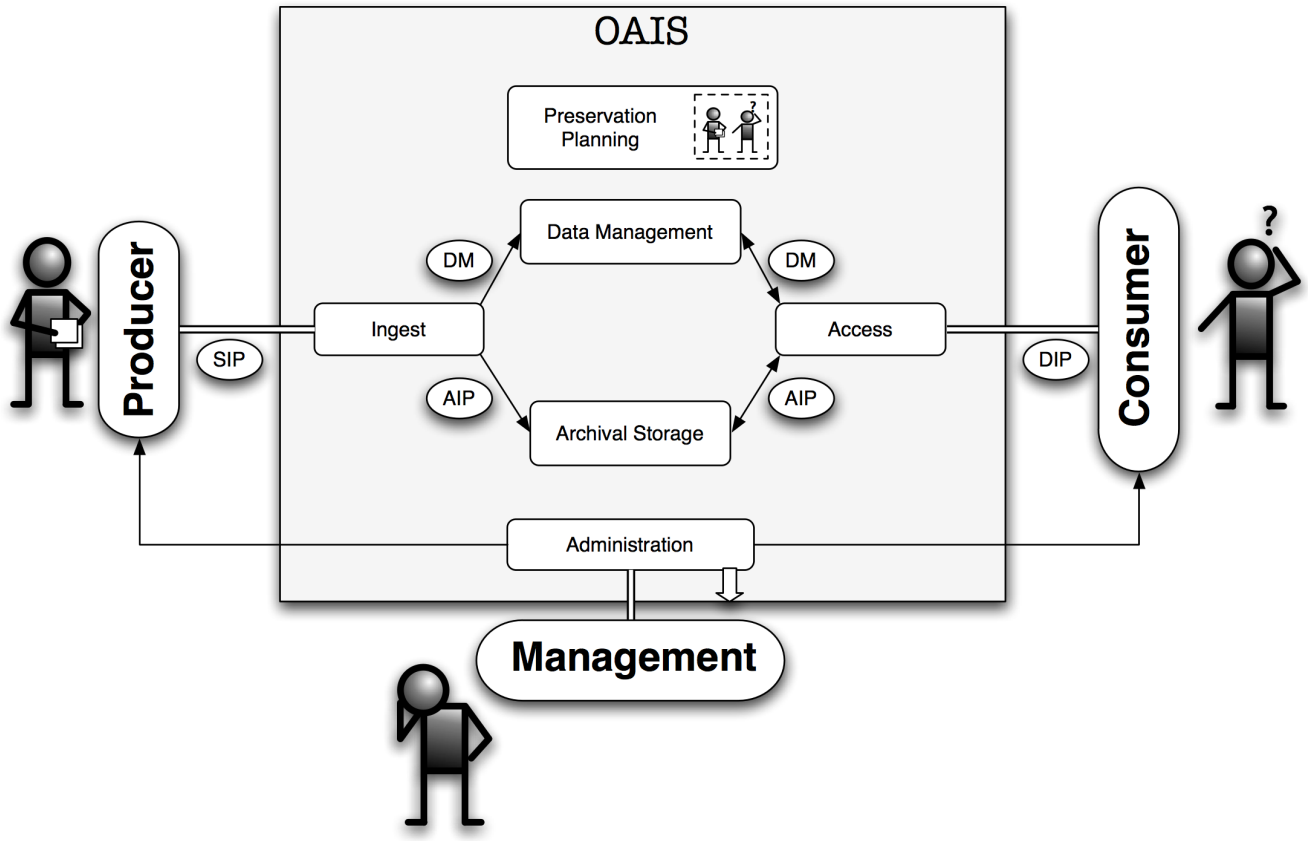
A imaginação do aluno definirá os limites do domínio em termos de conteúdo.

Requisitos

A seguir explicam-se os requisitos dos mais gerais para os mais específicos.

Requisitos sobre o repositório dos materiais digitais

O Repositório onde serão guardados todos os arrtefactos digitais deverá ser implementado seguindo as orientações do modelo OAIS (Figura 1, "Open Archival Information System").



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

- **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 proces- sado e arquivado. Deverá ser um arquivo comprimido ZIP. A sua estrutura é, normalmente, especificada no início do projec- to, neste caso, um SIP deverá ter a seguinte constituição:

- Um manifesto especificado de acordo com a norma Bagit: Manual de referência;
- Ficheiro(s) de vários formatos;
- Ficheiros com metainformação de cada recurso 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 ta- xonomia para o efeito, por exemplo, foto, jantar de aniversário, passeio de bicicleta, viagem, treino de natação, ...;

- **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 ar- mazenado: file system, base de dados, ...

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 pre- viamente 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.

Requisitos Gerais

- A aplicação é destinada a apenas um utilizador, ou seja, depois de instalada, o backoffice tratará apenas um *"eu" individual*". No entanto, o frontend de acesso à parte pública deverá estar disponível na Web para qualquer utilizador;
- Se a equipa de trabalho decidir preparar a aplicação para suportar simultâneamente múltiplos "eus" poderá fazê-lo o que será considerado como um extra;
- Para se poder distinguir entre um acesso público e um acesso privado deverá haver um módulo de autenticação transversal à aplicação. É aconselhado o uso do módulo passport e a implementação das autenticações: tradicional (username/passwd), facebook e google;
- Independentemente, dos limites que as equipes de trabalho definirem, a aplicação terá de ter um formato/linguagem para importação e exportação de dados;
- O domínio da aplicação deverá ser especificado e devidamente caraterizado antes do desenvolvimento ter início;
- A aplicação terá como base o eixo temporal e a disposição cronológica de eventos e conteúdos será a sua base;
- Cada elemento granular de conteúdo deverá ficar sempre classificado com um ou mais classificadores;

- Os classificadores do conteúdo deverão vir de um vocabulário controlado/taxonomia que terá de ser especificado pela equipa de trabalho (poderá e deverá ser discutido com o docente);
- A classificação de conteúdos deverá materializar-se em páginas de navegação com ordenação e agrupamento semântico que serão alternativa à navegação cronológica base;
- Um diário digital vai ser uma lista de itens publicados. Um item pode ser constituído por vários componentes: bloco de texto, fotografia, ficheiro associado, etc;
- Numa fase inicial, a equipa deverá fazer um levantamento de todos os tipos de item que irá suportar e definir quais os campos de metainformação que ficarão associados a cada item. Estes campos permitirão filtrar os itens para os mais variados fins: extrair o meu percurso desportivo, extrair o meu percurso académico, calcular a lista de locais turísticos que já visitei, etc;
- Um item pode ser público ou privado, por omissão deverá ser privado;
- Se for privado apenas aparece no backoffice do seu criador, se for público deverá aparecer no frontoffice público da aplicação;
- Um item público pode ser tornado privado e vice-versa;
- Deve ser possível associar comentários a um item: serão notas que ficarão associadas ao item e que permitirão qualificar melhor o item quanto ao seu âmbito e conteúdo e mesmo à sua aplicação;
- Pretende-se que haja uma ligação às redes sociais. Por exemplo, deverá haver uma operação para partilhar um item no facebook, ou no twitter, ou no strava, ou noutra (eventualmente, com as devidas adaptações);

Requisitos sobre o 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 manifesto-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 manifesto-SIP.(json|xml) existe;
- Verificar se todos os ficheiros referenciados no manifesto-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 artefactos");
- 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.

Requisitos sobre o AIP e o armazenamento de artefactos

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 manifesto-SIP.(json|xml), deverá ser 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.

Requisitos do DIP e da 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 dado item. Neste caso, o DIP corresponderá a um ficheiro ZIP com uma estrutura semelhante ao SIP.

Requisitos sobre o 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;
- Depois de devidamente de tudo devidadmente testado, deverá ser criado um Docker para cada serviço e um docker-compose a orquestrar tudo.