Ficha prática para a semana 10 de EngWeb2024

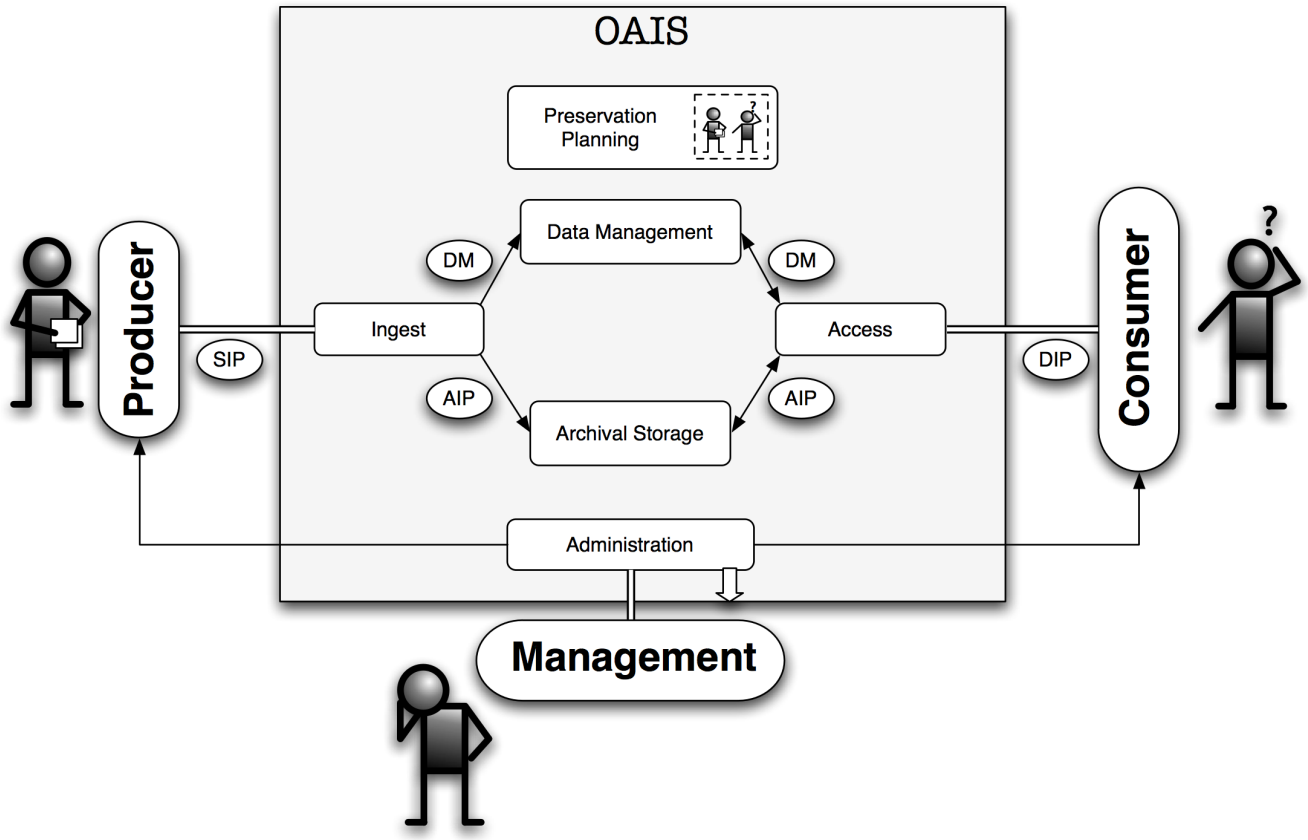
semana de 22 a 26 de Abril

@jcr

Entrega de projetos

Ao longo desta ficha, vão ser-te colocados problemas e desafios que no fim levarão à construção de uma aplicação web que permita a gestão das entregas de projetos de um conjunto de UC lecionadas por um determinado departamento.

Podemos pensar no sistema a desenvolver como seguindo o modelo OAIS já apresentado em várias aulas e que a seguir se apresenta numa imagem.



Neste modelo, podemos distinguir três tipos de atores: produtores de informação, gestores/administradores do sistema e consumidores finais. Três tipos de processos: ingestão, que trata da incorporação de novos materiais no sistema, gestão, que organiza e processa os materiais armazenados e a disseminação, que trata de disponibilizar a informação armazenada ao consumidor final. Três tipos de pacotes de informação: SIP ("Submission Information Package"), AIP ("Archival Information Package") e DIP ("Dissemination Information Package").

O SIP, no contexto desta ficha, será um ficheiro comprimido em formato ZIP, que deverá conter um manifesto (a definir numa linguagem de anotação à escolha: YAML, JSON, XML, ...) e um conjunto de

ficheiros. O AIP corresponde à forma armazenada do conteúdo do SIP e o DIP à informação que é consumida pelo consumidor, pode ser igual ao SIP ou pode ser simplesmente uma página web onde se pode consultar toda a informação.

Atores e ações

Na nossa aplicação, vamos ter dois tipos de produtores:

- **os docentes** que, sempre que o entenderem, vão criar um registo no sistema para a entrega de um projeto de uma determinada UC. Neste registo, deverão fazer parte os seguintes campos:
    - Data de criação do registo;
    - Data limite da entrega do projeto;
    - Ano letivo;
    - UC em que o projeto se realiza;
    - título/designação do projeto;
    - resumo do projeto (meia dúzia de parágrafos);
    - ficheiro PDF com o enunciado completo do projeto.
- **os alunos** que poderão fazer dois grupos de operações:
    1. Criação/alteração de um equipa de trabalho: deste registo farão parte os seguintes campos:
        - Data de criação do registo;
        - Ano letivo;
        - UC em que a equipa irá trabalhar;
        - Identificador da equipa;
        - Designação da equipa (nome de "guerra");
        - Nome e identificador de cada elemento da equipa;
        - Ficheiro GIF/JPEG/PNG com a foto de cada elemento da equipa;
        - Observações: alguma nota que queiram adicionar.
    2. Entrega de um projeto: deste registo farão parte os seguintes campos:
        - Data de criação do registo/entrega do projeto;
        - UC em que o projeto se realiza;
        - título/designação do projeto;
        - Identificador e designação da equipa;
        - Ficheiro ZIP com o projeto;
        - Observações: alguma nota que queiram adicionar.

API de dados

Numa primeira fase, vamos deixar de fora a autenticação e a gestão de utilizadores.

Dos requisitos acima, podemos identificar 4 coleções de dados: UC (com id e designação), Projeto (com os campos enumerados em cima), Equipa (com os campos enumerados em cima) e Entrega (com os campos enumerados em cima).

Esta semana, deverás implementar uma API de dados que suporte as operações CRUD sobre estas 4 coleções. A base de dados em MongoDB deverá ter a designação `entregas` e a API deverá responder na porta 13007.

A seguir descrevem-se as operações da API:

Coleção: uc

1. `GET /uc`: Devolve as UC registadas no sistema (já implementada no protótipo fornecido);
2. `GET /uc/:id`: Devolve a informação da UC, id, designação e entregas recebidas (parcialmente implementada no protótipo fornecido);
3. `POST /uc`: Permite criar uma nova UC e acrescentá-la à lista de UC disponíveis (já implementada no protótipo fornecido);
4. `PUT /uc/:id`: Permite alterar os dados de uma UC, apenas a designação (já implementada no protótipo fornecido);
5. `DELETE /uc/:id`: Permite remover uma UC, **se essa UC não tiver entregas registadas** (parcialmente implementada no protótipo fornecido).

Coleção: equipa

1. `GET /equipa`: Devolve as equipas registadas no sistema;
2. `GET /equipa/:id`: Devolve a informação da equipa, id, designação, membros, etc;
3. `POST /equipa`: Permite criar uma nova equipa;
4. `PUT /equipa/:id`: Permite alterar os dados de uma equipa, **se essa equipa não tiver entregas registadas**;
5. `DELETE /equipa/:id`: Permite remover uma equipa, **se essa equipa não tiver entregas registadas**.

Coleção: projeto

1. `GET /projeto`: Devolve os projetos registados no sistema (todos);
2. `GET /projeto?uc=id`: Devolve os projetos registados no sistema associados à `uc` com identificador `id`;
3. `GET /projeto/:id`: Devolve a informação do projeto, id, designação, membros, etc, e número de entregas que já estiverem registadas;
4. `POST /projeto`: Permite criar um novo projeto (quando for adicionada a autenticação, esta operação estará disponível apenas para docentes);
5. `PUT /projeto/:id`: Permite alterar os dados de um projeto, **se esse projeto não tiver entregas registadas**;
6. `DELETE /projeto/:id`: Permite remover um projeto, **se esse projeto não tiver entregas registadas**.

Coleção: entrega

1. `GET /entrega`: Devolve as entregas registadas no sistema (todas);
2. `GET /entrega?projeto=id`: Devolve as entregas registadas no sistema associadas ao `projeto` com identificador `id`;
3. `GET /entrega/:id`: Devolve a informação da entrega, id, equipa, material, etc;
4. `POST /entrega`: Permite criar uma nova entrega (numa primeira versão, com informação mínima, o identificador do projeto e o ficheiro ZIP correspondente ao SIP do projeto); Nesta operação, a informação da entrega deverá ser extraída do ZIP e colocada na base de dados; os ficheiros do projeto deverão ser armazenadas numa pasta específica para a entrega criada no âmbito de uma política a definir;

5. `PUT /entrega/:id`: Permite alterar os dados de uma entrega (o sistema deverá substituir a existente por esta);
6. `DELETE /entrega/:id`: Permite remover uma entrega (deverá ser colocada numa coleção de entregas removidas juntamente com um campo descritivo onde se deverá colocar uma justificação).

Interface Web

Mais tarde iremos criar uma interface web mas num serviço separado e independente que ira consumir a API de dados.