

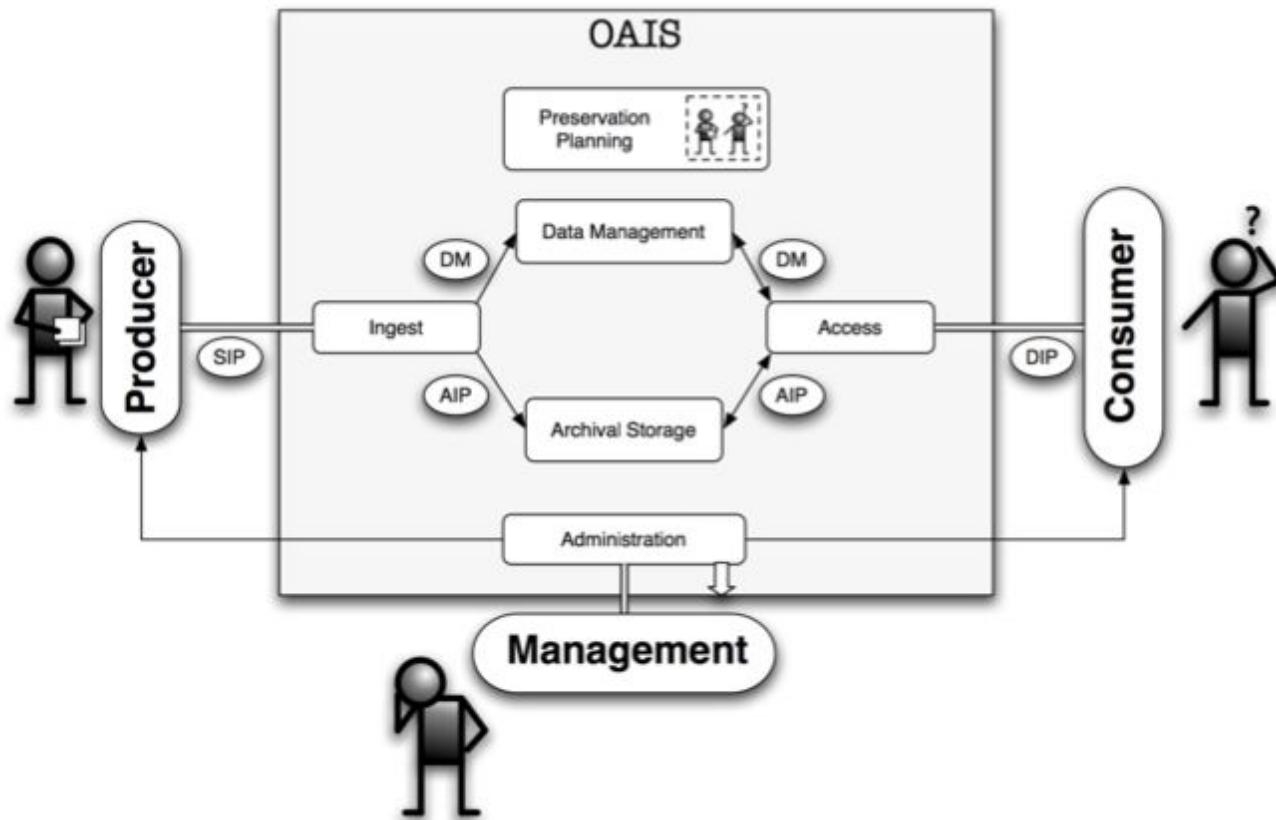
# Projetos 2024

José Carlos Ramalho

2024-04-14

Modelo comum a todas as propostas

# Open Archival Information System



# Processo de ingestão

- Upload de pacotes na plataforma;
- A estrutura do pacote deverá ser verificada contra o manifesto;
- Faça outras validações que entender por bem;
- Se tudo estiver bem o recurso é armazenado na plataforma ficando disponível para os utilizadores;
- Se houver erros de validação, um relatório de erros deverá ser enviado a quem fez a submissão.

# Processo de administração

- Conjunto de operações, CRUD e outras, sobre os recursos armazenados:
  - Listar, Consultar, Inserir um novo, Alterar um existente, Apagar.
- Responsável também por converter um SIP num pacote armazenado (AIP).

# Processo de disseminação

- Conversão de um AIP num DIP;
- Numa fase inicial, considerem  $DIP = SIP$ .

# Funcionalidades comuns a todos os projetos

- Autenticação de utilizadores (perfis?);
- Possibilidade de exportar toda a informação;
- Possibilidade de importar pelo processo inverso.

# **Proposta 1: Plataforma de Gestão e Disponibilização de Recursos Educativos**



# Objetivos

- Disponibilizar recursos educativos de vários tipos: livros, artigos, aplicações, trabalhos de alunos, monografias, relatórios, ...
- Permitir adicionar novos tipos de recursos e novos recursos;
- Ter os recursos classificados por ano, tipo, tema, ... (utilização de hashtags ou de uma taxonomia classificativa);
- Permitir que um utilizador faça um Post sobre um recurso;
- Permitir que os outros utilizadores comentem Posts;
- Criar um sistema de ranking para os recursos (atribuição de estrelas pelos utilizadores);
- E o que a imaginação ditar...

# Submission Information Package (SIP)

- Um ficheiro ZIP que deve seguir uma determinada estrutura (muito básica);
- Baseiem-se no BagIt: <https://tools.ietf.org/id/draft-kunze-bagit-16.html>
- Não precisam de seguir os requisitos todos...
  - Um manifesto e uma lista de ficheiros...

# Dissemination Information Package (DIP)

- Semelhante/igual ao SIP;
- O sistema deverá conseguir importar e exportar o mesmo pacote.

# Utilizadores

- O sistema deverá estar protegido com autenticação: username+password, chaveAPI, google, facebook, ...
- Deverão existir pelo menos 3 níveis de acesso:
  - a. Administrador - tem acesso a todas as operações;
  - b. Produtor (autor de recurso) - pode consultar tudo e executar todas as operações sobre os recursos de que é produtor/autor;
  - c. Consumidor - pode consultar e descarregar os recursos públicos.
- Dados sobre o utilizador a guardar (sugestão):
  - a. nome, email, filiação (estudante, docente, curso, departamento, ...), nível (administrador, produtor ou consumidor), dataRegisto (registo na plataforma), dataUltimoAcesso, password, outros campos que julgue necessários...

# Recursos educativos

- Podem ser de vários tipos: relatório, tese, artigo, aplicação, slides, teste/exame, problema resolvido, ... (defina um conjunto base e preveja o seu aumento);
- Metainformação comum a todos os recursos:
  - tipo, título, subtítulo (opcional), dataCriação, dataRegisto (entrada no sistema), visibilidade (público: todos podem ver e descarregar, privado: apenas disponível para administradores e seu produtor), produtor/autor, ...

# Notícias

- Na página principal, em local a definir e dependendo do design poderão aparecer notícias, criadas pelo admin ou geradas pelo sistema (exemplo: Nova submissão: o produtor X acabou de disponibilizar um artigo intitulado “...”).

# Dataset para a demonstração final

- Na demonstração final, a plataforma deverá ter algumas dezenas de entradas;
- Usem material real;
- Têm o material que circula pelos alunos de várias UCs;
- Têm o RepFichas do JCR:  
<http://www4.di.uminho.pt/~jcr/AULAS/didac/RepFichas/site/index-fichas.html>
- Programas exemplo do JCR:  
<http://www4.di.uminho.pt/~jcr/AULAS/didac/programasC/index.htm>
- Minitestes de Programação Imperativa:  
<http://www4.di.uminho.pt/~jcr/AULAS/didac/minitestesPI/>
- Compiladores: <http://www4.di.uminho.pt/~jcr/AULAS/didac/compiladores/>
- Têm os sites das UC deste ano com muito material:
  - <https://epl.di.uminho.pt/~jcr/AULAS/EngWeb2023/>
  - <https://epl.di.uminho.pt/~jcr/AULAS/RPCW2023/aulas2023.html>
  - <https://epl.di.uminho.pt/~jcr/AULAS/ATP2022/>

## **Proposta 2: Inquirições de G nere**



# O que são:

“Processos necessários para a ordenação dos párocos. Consistem na inquirição de testemunhas para comprovar a filiação, reputação, bom nome ou "limpeza de sangue" do requerente.

Incluem os seguintes documentos: requerimento inicial, carta de comissão, inquirições de genere a testemunhas e declarações dos párocos.

Estão incluídos nesta série processos de Justificação de Fraternidade que são requeridos quando há, na família, alguém ordenado e com inquirição de genere elaborada e concluída.”

<http://pesquisa.adb.uminho.pt/details?id=1193713>

Arquivo Distrital de Braga / Universidade do Minho

# Objetivos

- Analisar o dataset fornecido e tratá-lo de modo a carregá-lo no MongoDB:
  - Ter especial cuidado com o campo “Scope and Content”, há dados de relações genealógicas que se pretendem materializar em relações entre registos sempre que possível.
- Criar uma interface web de navegação nestes registos, especial atenção à navegação pelas relações entre registos;
- Permitir adicionar novos registos;
- Ter a possibilidade de navegar por nome (índice antroponímico), por lugar (índice toponímico) e data (índice cronológico);
- Permitir que um utilizador edite a informação de um registo:
  - Podendo acrescentar novas relações entre registos.
- Permitir que um utilizador faça um Post sobre um registo;
- Permitir que os outros utilizadores comentem Posts;
- E o que a imaginação ditar...

# Utilizadores

- O sistema deverá estar protegido com autenticação: username+password, chaveAPI, google, facebook, ...
- Deverão existir pelo menos 2 níveis de acesso:
  - a. Administrador - tem acesso a todas as operações;
  - b. Consumidor - pode consultar, fazer posts e sugerir alterações;
- Dados sobre o utilizador a guardar (sugestão):
  - a. nome, email, filiação (estudante, docente, curso, departamento, ...), nível (administrador, produtor ou consumidor), dataRegisto (registo na plataforma), dataUltimoAcesso, password, outros campos que julgue necessários...

# Dataset

- Disponível online:

<https://drive.google.com/file/d/1mEPD6bx9wjsuaDkwSZ5qXs1Tdo0CCN6M/view?usp=sharing>

## **Proposta 3: Mapa das Ruas de Braga**

# O que é?

“Passados duzentos e cinquenta anos da feitura do Mapa das Ruas de Braga pelo padre Ricardo Rocha, é justo que se evoque aquele que pode ser considerado como o documento mais sui generis de todo o manancial iconográfico de Braga. Embora inserido num processo mais vasto e pragmático que marcaria a história de uma das mais antigas e poderosas instituições da urbe – o Cabido da Sé Primacial – que o produziu, o Mappa, sempre deslumbrante, individualiza-se como uma janela aberta sobre a memória da cidade.”

<https://bragavirtual.pt/mappa-das-ruas-de-braga/>

# Ponto de partida

- Dataset composto por:
  - 60 Ficheiros XML com a informação de cada uma das ruas: casas, proprietários, rendas, ...
  - Várias imagens correspondentes a digitalizações dos desenhos originais do séc. XVIII;
  - Várias imagens correspondentes a um trabalho de campo realizado por alunos de engenharia biomédica que fotografaram as ruas na atualidade.
- Material disponível em:  
<https://drive.google.com/drive/folders/1hAScFhLmOsGKoZgaAjiAorDvK5zc4e nl?usp=sharing>

# Objetivos

- Analisar o dataset fornecido e tratá-lo de modo a carregá-lo no MongoDB ou noutro sistema de dados;
- Criar uma interface web de navegação em toda a informação disponibilizada;
- Permitir adicionar novos registos;
- Ter a possibilidade de navegar por nome (índice antroponímico), por lugar (índice toponímico) e data (índice cronológico);
- Permitir que um utilizador edite a informação de um registo:
  - Podendo substituir imagens, alterar campos, ...
- Permitir que um utilizador faça um Post sobre um registo;
- Permitir que os outros utilizadores comentem Posts;
- E o que a imaginação ditar...



# Utilizadores

- O sistema deverá estar protegido com autenticação: username+password, chaveAPI, google, facebook, ...
- Deverão existir pelo menos 2 níveis de acesso:
  - a. Administrador - tem acesso a todas as operações;
  - b. Consumidor - pode consultar, fazer posts e sugerir alterações;
- Dados sobre o utilizador a guardar (sugestão):
  - a. nome, email, filiação (estudante, docente, curso, departamento, ...), nível (administrador, produtor ou consumidor), dataRegisto (registo na plataforma), dataUltimoAcesso, password, outros campos que julgue necessários...

## Proposta 4: **Base de Dados de Acordãos**

# Contexto

O Ministério da Justiça português tem na sua organização vários tribunais.

Cada um com autonomia própria.

Cada tribunal disponibiliza periodicamente um conjunto de acordãos que se tornam públicos.

A estrutura da informação é mais ou menos semelhante mas as interfaces disponibilizadas para consulta não são...

# Ponto de partida

- Dataset composto por:
  - 14 bases de dados em JSON.
- Material disponível em:

<https://drive.google.com/drive/folders/1pJUTOVOGeo1uMxX9WywBrQUmKPKr-EIG?usp=sharing>

# Objetivos

- Analisar o dataset fornecido e tratá-lo de modo a carregá-lo no MongoDB ou noutro sistema de dados, ter o cuidado de perceber que campos comuns existem entre as várias bases de dados;
- Criar uma interface web de navegação em toda a informação disponibilizada;
- Permitir adicionar novos registos;
- Ter várias possibilidades de pesquisa (discutir com o docente);
- Permitir que um utilizador edite a informação de um registo:
  - Podendo substituir imagens, alterar campos, ...
- Permitir que um utilizador tenha uma “caixa” de favoritos, onde pode guardar o apontador para o registo e uma descrição introduzida por ele;
- Construir uma taxonomia de termos a partir dos descritores e usá-la na pesquisa (falar com o docente);
- E o que a imaginação ditar...

# Utilizadores

- O sistema deverá estar protegido com autenticação: username+password, chaveAPI, google, facebook, ...
- Deverão existir pelo menos 2 níveis de acesso:
  - a. Administrador - tem acesso a todas as operações;
  - b. Consumidor - pode consultar, guardar favoritos e sugerir alterações;
- Dados sobre o utilizador a guardar (sugestão):
  - a. nome, email, filiação (estudante, docente, curso, departamento, ...), nível (administrador, produtor ou consumidor), dataRegisto (registo na plataforma), dataUltimoAcesso, password, outros campos que julgue necessários...

## Proposta 5: **Gerador de websites para UC**

# Objetivos

- Analisar o dataset de uma UC fornecido e tratá-lo de modo a criar um modelo em MongoDB para o guardar;
- Criar uma interface web de navegação em toda a informação disponibilizada, semelhante ao das UC que se listam no slide seguinte (há espaço para melhorar/alterar o design e até sugerir novas funcionalidades);
- Criar uma funcionalidade para a criação de novas UC (devem implementar todas as operações de CRUD sobre uma UC);
- Ter várias possibilidades de pesquisa sobre as UC criadas e ter uma interface centralizada para aceder ao site de cada uma;
- Permitir que o utilizador que criou a UC edite a informação desta;
- E o que a imaginação ditar...



# Utilizadores

- O sistema deverá estar protegido com autenticação: username+password, chaveAPI, google, facebook, ...
- Deverão existir pelo menos 3 níveis de acesso:
  - a. Administrador - tem acesso a todas as operações;
  - b. Produtor (autor de recurso) - pode consultar tudo e executar todas as operações sobre os recursos de que é produtor/autor;
  - c. Consumidor - pode consultar e descarregar os recursos públicos.
- Dados sobre o utilizador a guardar (sugestão):
  - a. nome, email, filiação (estudante, docente, curso, departamento, ...), nível (administrador, produtor ou consumidor), dataRegisto (registo na plataforma), dataUltimoAcesso, password, outros campos que julgue necessários...

# Ponto de partida

- Datasets exemplo (em uso):

[https://drive.google.com/drive/folders/1jwYdwrQj4t-ZBTF674L4xWpi6\\_jq2Ax\\_  
?usp=sharing](https://drive.google.com/drive/folders/1jwYdwrQj4t-ZBTF674L4xWpi6_jq2Ax_?usp=sharing)

# Submissão do Projeto

# Como entregar o projeto

1. A entrega será feita criando uma pasta no repositório git de cada aluno de nome "Projeto2024" e colocando nela tudo o que concerne ao projeto, datasets, código desenvolvido, ..., e o relatório (4 a 6 páginas em formato Markdown - até podem usá-lo para documentar o Git);
2. Além disto deverão enviar um email a "jcr@di.uminho.pt" com o assunto "ENGWEB2024-Projeto-Axxxxx-Ayyyyy-Azzzzz" com um ficheiro ZIP onde deverá estar todo o conteúdo do repositório Git (Axxxxx é o número de aluno de cada membro do grupo);
3. As defesas serão marcadas, em folha de cálculo própria que irei disponibilizar em breve, uma vez definida a data final de entrega;
4. Se por algum motivo, algum grupo não puder estar presente para a defesa nas semanas definidas, o grupo deverá articular comigo o quanto antes uma data que sirva a todos;
5. Todas as defesas serão realizadas em sessões online que criarei para o efeito (link será disponibilizado na folha de marcações);
6. A ausência na defesa do projeto implica uma não avaliação no projeto, a presença é obrigatória.