



UNIVERSIDADE DO MINHO
MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA
DEPARTAMENTO DE INFORMÁTICA

PROCESSAMENTO E REPRESENTAÇÃO DA
INFORMAÇÃO

iBanda - Arquivo Digital Musical

Grupo 2

Eduardo Gil Ribeiro da Rocha - **A77048**
Lázaro Vinícius Oliveira Lima - **ID8120**
Manuel Gouveia Carneiro de Sousa - **A78869**

1 de Fevereiro de 2019

Resumo

Este relatório pretende exibir quais os passos tomados para o desenvolvimento de uma aplicação Web que gerencia um repositório digital de obras musicais e respectivas partituras. O repositório foi construído respeitando o modelo de referência internacional *OAIS*-(*Open Archive Information System*).

Conteúdo

1	Introdução	3
2	Arquitetura da Aplicação	3
2.1	Modelo de Referência OAIS	3
2.2	Definição das Estruturas dos Ficheiros	4
2.3	Estrutura de Armazenamento	5
3	SIP e o processo de Ingestão	6
4	AIP e o armazenamento de projetos	6
5	Principais Funcionalidades	7
5.1	Consumidor	7
5.2	Produtor	7
5.3	Administrador	8
6	Conclusões	8

1 Introdução

A aplicação proposta tem como principal objetivo proporcionar a uma banda filarmónica (ou utilizador independente) a visualização e navegação de obras e respetivas partituras.

As tecnologias utilizadas para o desenvolvimento da aplicação são *Node.js*, desenvolvido para aplicações escaláveis e para armazenamento *MongoDB*, que funciona como base de dados com escalabilidade e flexibilidade.

O presente relatório está organizado de forma a que a primeira secção apresente a modelagem e estrutura para a arquitetura da aplicação, e a segunda parte a metodologia adotada para o funcionamento da aplicação e manipulação da informação que é adicionada à aplicação.

Por último, o relatório apresenta os resultados obtidos e o comportamento da aplicação.

2 Arquitetura da Aplicação

2.1 Modelo de Referência OAIS

A solução desenvolvida teve como modelo de referência o OAIS (Open Archival Information System)[?], modelo esse que desde 2003 é norma internacional ISO 14721:2003, usado como referência na padronização de atividades de preservação digital de informações.

Na figura 1 podemos observar com alguns detalhes a arquitetura de funcionamento do modelo OAIS.

Como referência de modelo, podem ser observados no modelo OAIS na figura 1 três principais atores da aplicação, estes são:

- Produtor musical;
- Administrador da aplicação;
- Consumidor da aplicação.

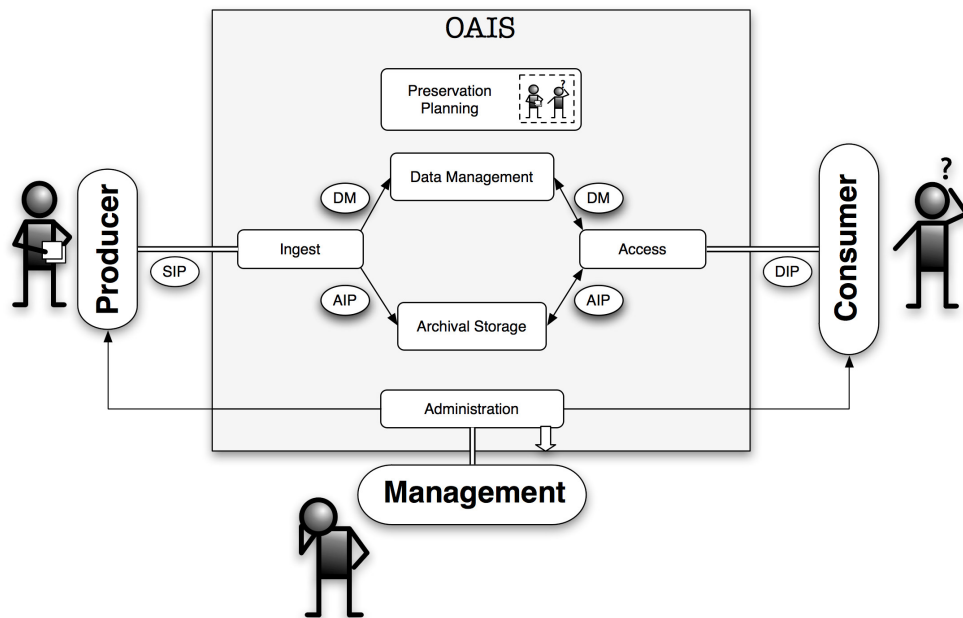


Figura 1: Modelos de Referência OAIS.

2.2 Definição das Estruturas dos Ficheiros

O ficheiro que acrescenta obras à aplicação é um ficheiro *zip* que contém uma pasta denominada de “instrumentos”, e ao mesmo nível, um ficheiro *XML* que corresponde ao manifesto da obra. Este manifesto, obrigatoriamente denominado de “iBanda-SIP.xml”, tem uma estrutura específica, a qual foi criada através de um ficheiro *XSD*. De notar a escolha de um ficheiro *XML* para manifesto, visto ter uma estrutura que se torna intuitiva em termos de escrita. Já dentro da pasta “instrumentos”, existe uma pasta respetiva a cada instrumento da obra, e dentro destas, diversas partituras.

A título de exemplo, esta foi a estrutura adotada para o manifesto:

```
1 <?xml version="1.0" encoding="UTF-8"?>
  <manifesto>
3    <meta>
      <titulo>Homenagem ao Dr Manuel Vaz</titulo>
5      <tipo>Marcha de Desfile</tipo>
      <compositor>Americo Cilindro</compositor>
7    </meta>
    <instrumentos>
9      <inst>
        <nome>Sax Baritono</nome>
11       <partitura>HomenagemMVaz-saxB.jpg</partitura>
      </inst>
13     <inst>
        <nome>Trombone</nome>
15       <partitura>HomenagemMVaz-trombone2.pdf</partitura>
        <voz>II</voz>
17       <clave>Fa</clave>
        <afinacao>Do</afinacao>
19     </inst>
    </instrumentos>
21 </manifesto>
</xml>
```

Esta estrutura permite a comparação entre os ficheiros que estão armazenados nas pastas com as informações contidas no XML, verificando assim se todos os ficheiros indicados existem.

2.3 Estrutura de Armazenamento

De forma a guardar toda a informação que é fornecida à aplicação, é utilizada uma base de dados em *MongoDB*, a qual terá as informações de todas as obras, bem como a informação dos utilizadores da aplicação. Relativamente aos ficheiros enviados, estes são guardados no *filesystem* do servidor, dentro de uma pasta cujo nome corresponde ao identificador (único) da obra na base de dados.

Para além disso, foi ainda criado um índice de todas as obras (em *JSON*), organizado pelo tipo de obra, permitindo assim uma simples e organizada listagem das mesmas.

Através deste tipo de armazenamento, era possível obtermos o tipo de informação necessária de forma flexível.

3 SIP e o processo de Ingestão

A ingestão da aplicação foi desenvolvida tendo como base o *SIP* (*Submission Information Package*)/ Pacote de Submissão de Informação. Esse pacote compactado é enviado pelo produtor à aplicação para processamento e armazenamento.

Depois de enviado o pacote, este será validado pelo servidor. Numa primeira fase, é analisado se o pacote enviado é do tipo *.zip*, descompactando-o então se for o caso. Posto isto, é imediatamente confirmada a existência da pasta “instrumentos” e do manifesto “iBanda-SIP.xml”, e se estes existem, é feita a verificação da estrutura do ficheiro *XML* através de um ficheiro *XSD*.

Caso tudo isto seja válido, é inicializada a comparação do que foi referido no manifesto com a respetiva pasta de instrumentos. Importante referir que este ficheiro *XML* foi convertido em *JSON*, para dessa forma trabalharmos com objetos nativos do *Javascript*. Neste momento, é construído o objeto que irá ser posteriormente guardado na base de dados, bem como a entrada no índice de obras.

Depois de processado, o SIP é transformado em *AIP* (*Archival Information Package*)- Pacote de Armazenamento de Informação.

4 AIP e o armazenamento de projetos

Armazenado o pacote fornecido pelo cliente, as informações contidas neste são guardadas na base de dados, e associadas a este pacote, uma pasta relativa às partituras e manifesto de uma obra.

De forma a mantermos a informação da base de dados “conectada” com a pasta previamente criada, a pasta foi definida com o identificador gerado pelo *MongoDB* aquando a inserção das informações da obra. Desta forma, para obtermos alguma partitura, saberíamos “a priori” o nome da pasta onde essa está localizada, visto ser o identificador do objeto da obra.

Posto isto, e tendo a informação estruturada e organizada, passaremos a exemplificar as funcionalidades da aplicação.

5 Principais Funcionalidades

Toda a aplicação concentra-se na interface principal, que contém opções de acesso e uma lista de notícias criadas pelo administrador do sistema. Estas opções, darão acesso às diversas funcionalidades da aplicação, como a listagem das obras ou até o *upload* de um pacote.

Para ter acesso a funcionalidades como o *upload* é necessário estar registado no sistema, sendo considerado assim um *Produtor* do mesmo. Este acesso é controlado através de autenticação baseada em *passport.js*, que é um middleware para *Node.js*. Através desta biblioteca, e aplicando a estratégia *JWT - JSON Web Token*, é gerado um *Token* o qual é atribuído ao utilizador em cada sessão, permitindo-o aceder a certas áreas restritas - este *token* é gerado com 30 minutos de duração, expirado este tempo, o utilizador é reencaminhado para a página de *login*.

De seguida, apresenta-se as funcionalidades às quais o utilizador comum (Produtor e Consumidor) pode aceder, e as funcionalidades que estão restritas ao uso do Administrador.

5.1 Consumidor

O consumidor é o utilizador na aplicação que pode usar certas funcionalidades sem que esteja previamente registado na mesma. Neste caso, poderá verificar quais são as obras no sistema, consultando ou descarregando a obra pretendida. Outra ação permitida pelo consumidor é de efetuar registo e respetivo *login* na aplicação, se assim o desejar.

5.2 Produtor

O Produtor é o utilizador o qual efetuou o registo na plataforma, tendo assim o acesso a todas as funcionalidades abertas ao cliente. Este, para além de ter a possibilidade de listar as obras na aplicação, consultar ou descarregar uma certa obra, poderá enviar novas obras através do menu de *upload* de pacotes. Nessa etapa é necessário ter em atenção o formato que foi definido para este pacote, como foi referido anteriormente.

5.3 Administrador

Por fim e não menos importante, temos na aplicação a interface de administração que dá acesso às ações que serão executadas para a manutenção da aplicação. Esta interface é então acedida por um Administrador da aplicação, previamente registado, o qual tem acesso não só à sua própria interface, mas também à interface que é disponibilizada a um Produtor.

Relativamente à interface do Administrador, estas são funcionalidades disponíveis:

- Administração dos utilizadores;
- Administração das obras (AIPs);
- Administração de notícias;
- Administração da agenda de eventos.

Através destas funcionalidades, é possível uma gestão de diversas entidades que compõem a aplicação. Aqui, é possível ao Administrador gerir os utilizadores, tendo a possibilidade de os remover e alterar os seus dados; Listar e remover obras do sistema; Criar e remover notícias, tornando-as visíveis ou invisíveis para todos os utilizadores; E por fim, criar ou remover eventos, criando assim uma agenda com os mesmos.

6 Conclusões

É importante referir os pontos que ainda não foram implementados, bem como aqueles que poderão levantar algumas dúvidas. De facto, na parte do Administrador, não é possível a edição de notícias/eventos, sendo a única alternativa possível a remoção. Para além disso, vale a pena realçar a maneira de como é feita a junção da entrada de uma obra na base de dados com os seus respetivos ficheiros. Ainda se formam dependências aquando a remoção de uma obra. De facto, para remover por completo uma obra do sistema, é necessária a remoção da entrada na base de dados, dos ficheiros das partituras na respetiva pasta, bem como a entrada no ficheiro de índice. Ao remover uma obra, e supondo que existirá uma falha em alguma dessas remoções, continuará a existir conteúdo de uma obra que foi erradamente eliminada.

Para além disso, não foi possível a implementação das gramáticas em *ANTLR* realizadas para a verificação de eventos, bem como a estruturação e organização de uma banda e seus membros, faltando então essas funcionalidades.

Por outro lado, a criação de um índice logo após o processo de ingestão fez com que a listagem e organização das obras, o principal foco da aplicação, se tornasse o mais eficiente possível, visto que o necessário para continuamente listar as obras estaria num ficheiro *JSON*, não sendo necessária a aplicação efetuar uma *query* à base de dados sempre que o utilizador quer visualizar uma lista de todas as obras.

Em suma, este relatório expôs as principais motivações e aspetos para criação de uma aplicação Web para gerência de obras digitais. Foi observada a necessidade de disponibilizar para os utilizadores uma ferramenta que auxiliasse na gestão e acesso às obras e partituras, bem como a sua visualização.

Como principal ação futura, é necessária a implementação de uma biblioteca ou tutor de suporte e construção iterativa de uma pequena enciclopédia do material armazenado para geração da visualização da informação carregada.

Referências

- [1] - OAIS Reference. The fundamental standard for digital preservation.
Disponível em: <http://www.oais.info/> - Acesso em: 27 de Janeiro de 2019