



UNIVERSIDADE DO MINHO

DEPARTAMENTO DE INFORMÁTICA

MusicPedia

João Gomes
A82238

Sofia Teixeira
A80624

(27 de junho de 2021)

Conteúdo

1	Introdução	3
2	Persistência de Dados	4
2.1	Obtenção dos Dados	4
2.2	Ontologia	5
2.3	GRAPHDB	5
3	Arquitetura da solução	6
3.1	Servidor API	6
3.2	Servidor APP	6
4	Funcionalidades Implementadas	7
5	Conclusão	8

1 Introdução

Neste projeto pretende-se aplicar as tecnologias lecionadas e desenvolvidas no decorrer da Unidade Curricular Processamento e Representação de Conhecimento, desde tratamento de data-sets, criação de ontologias, utilização de GraphDB como base de dados baseada em ontologias e desenvolvimento de aplicações WEB com **Vue**.

Pretende-se fazer uso da *API* do **Spotify** para criar dados referentes a artistas, seus álbuns e músicas, sendo estes dados depois convertidos para um formato **Turtle** para poderem ser importados no GraphDB. Depois criar-se-á uma aplicação WEB que consumirá estes dados e os apresentará no formato de um Web Site para qualquer utilizador poder consultar.

Pretende-se que a aplicação tenha algumas destas funcionalidades:

- Consultar o top 60 dos artistas.
- Consultar o top 100 dos álbuns.
- Consultar o top 100 das músicas.
- Consultar a página de um artista, que conterà informação relativa ao artista e aos seus álbuns.
- Consultar a página de um álbum, que conterà informação relativa ao álbum e às suas músicas.
- Consultar a página de uma música, que conterà informação relativa à mesma.
- Função de procura, onde o utilizar poderá procurar por músicas dado um nome.
- Filtrar as músicas por alguns parametros.

2 Persistência de Dados

2.1 Obtenção dos Dados

Como foi referido na secção anterior, a aplicação explorará dados de artistas e dos seus respectivos álbuns e músicas. O processo de obtenção destes dados realizou-se em duas fases.

A primeira fase focou-se em arranjar um dataset que continha o nome dos melhores artistas dos últimos anos, segundo o canal **MTV**. De seguida, procurou-se vários datasets que permitissem obter informação sobre estes artistas, mas sem sucesso. Decidiu-se então construir um dataset com os dados pretendidos, fazendo uso da *API* pública do **Spotify**.

A *API* do **Spotify** permite, de entre muitas funcionalidades, obter toda a informação de cada álbum que um dado artista tem, bem como toda a informação de cada música que um dado álbum tem. Após o uso desta *API*, obtiveram-se os seguintes dados referentes a cada entidade:

- **Artista:**
 - Nome.
 - Lista de Álbuns.
- **Álbum:**
 - Nome.
 - Número de Músicas.
 - Imagem.
 - Data de publicação.
 - Lista de Músicas.
- **Música:**
 - Nome.
 - Duração.
 - Popularidade.
 - Se tem conteúdo para Adultos.
 - Danceability.
 - Posição no Álbum.

2.2 Ontologia

Uma Ontologia é uma metodologia de modelar dados de um domínio específico que permite representar conceitos, relações entre estes e atributos, tendo sido o formato escolhido para realizar a persistência de dados.

Para conseguir representar os dados numa ontologia, teve de se definir as Classes, *Data Properties* e *Object Properties*. As classes criadas que correspondem às entidades foram: Artista, Álbum e Música. As *Data Properties* criadas que correspondem aos atributos que as classes podem ter foram: Nome, Imagem, Data, Duração, Popularidade, Danceability, Posição no álbum e conteúdo para adultos. As *Object Properties* criadas com o intuito de representar as relações entre as classes foram: hasAlbum, hasMusic, ofAlbum.

Nas tabelas seguintes, pode ver-se a relação entre as classes, bem como os *Data Properties* que cada classe tem:

Sujeito	Property	Objeto
:Artista	:hasAlbum	:Album
:Artista	:nome	"Nome"
:Album	:hasMusic	:Musica
:Album	:nome	"Nome"
:Album	:numberTracks	"Number"
:Album	:image	"Imagem"
:Album	:data	"Data"
:Music	:ofAlbum	:Album
:Music	:duracao	"Duração"
:Music	:nome	"Nome"
:Music	:musicNumber	"Number"
:Music	:danceability	"Danceability"
:Music	:popularity	"Popularidade"
:Music	:ratedR	"RatedR"

Após a definição do esqueleto da ontologia, procedeu-se a povoá-la com os dados obtidos a partir da *API* do **Spotify**. Este processo foi feito recorrendo a um script de Python que converte a informação vinda do **Spotify** para sintaxe de Ontologia. Neste projeto, a sintaxe utilizada foi **Turtle**.

2.3 GRAPHDB

O **GraphDB** é um motor de base de dados que usa grafos como estrutura usada para representar e armazenar informação. Uma das suas funcionalidades, é a possibilidade de criar um repositório com os dados importados a partir de um ficheiro **Turtle**.

Usando a sua linguagem de *Querying* **SPARQL**, o **GraphDB** permite consultar, atualizar, inserir e remover os dados do repositório e é desta forma que o Servidor API interage com a base de dados.

3 Arquitetura da solução

De forma a realizar o projeto planeado, este foi arquitetado seguindo uma solução baseado em micro-serviços: Servidor API e o Servidor APP.

3.1 Servidor API

É o servidor *back-end* da aplicação. Foi implementado em **Express** e é o serviço responsável pela comunicação com o repositório do **GraphDB** onde é feita a persistência dos dados. Fazendo uso da *API* que o **GraphDB**, este servidor envia queries que serão executadas no repositório de dados, recebendo depois os dados que correspondem à query enviada.

3.2 Servidor APP

É o servidor *front-end* da aplicação. Foi implementado recorrendo à framework **Vue + Vuetify** e é o responsável por apresentar ao utilizador as diferentes páginas da aplicação. Após receber pedidos do cliente, comunica com o Servidor API para receber os dados pretendidos. Após a recepção destes mesmos, apresenta uma nova página com o conteúdo pedido originalmente pelo cliente.



Figura 1: Arquitetura do sistema.

4 Funcionalidades Implementadas

- **Página Top**

- Consultar Top artistas
- Consultar Top álbuns
- Consultar Top músicas

- **Página Dance**

- Consultar Top álbuns para dançar
- Consultar Top músicas para dançar

- **Página Artistas**

- Consultar Lista com todos os artistas.
- Procurar um dado artista pelo seu nome.
- Filtrar todos os artistas que comecem por uma dada letra.
- Clicar num dado artista e entrar na sua página individual

- **Página Artista**

- Consultar todos os álbuns do artista.
- Filtrar os álbuns pelo seu nome.
- Ordenar os álbuns por nome.
- Ordenar os álbuns por data.
- Clicar num dado álbum e entrar na sua página individual.

- **Página Álbuns**

- Consultar lista com todos os álbuns.
- Procurar um dado álbum pelo seu nome.
- Filtrar todos os álbuns que comecem por uma dada letra.
- Ordenar os álbuns pelo seu nome.
- Ordenar os álbuns pelo seu artista.
- Ordenar os álbuns pela sua data.
- Clicar num dado álbum e entrar na sua página individual.

- **Página Álbum**

- Consultar todas as músicas do álbum.
- Procurar uma dada música deste álbum pelo nome.
- Ordenar as músicas por nome.
- Ordenar as músicas por duração.
- Ordenar as músicas pela sua posição no álbum.
- Clicar numa dada música e entrar na sua página individual.
- Clicar no artista e entrar na sua página individual.

- **Página Músicas**

- Consultar tabela com todas as músicas.
- Procurar uma dada música pelo seu nome.
- Filtrar todos as músicas que comecem por uma dada letra.
- Ordenar as músicas pelo seu nome.
- Ordenar as músicas pelo seu álbum.
- Ordenar as músicas pelo seu artista.
- Ordenar as músicas pela sua data.
- Clicar numa dada música e entrar na sua página individual.

- **Página Música**

- Consultar informação da música.
- Clicar no seu artista para ver a sua informação.
- Clicar no seu álbum para ver a sua informação.

- **Filtro geral - R-Rated**

- Retira as músicas *R-Rated*
- Retira os álbuns que apenas têm músicas *R-Rated*
- Retira os artistas que apenas têm músicas *R-Rated*

5 Conclusão

O uso de bases de dados baseadas em ontologias tem visto um crescimento notável nos últimos tempos. Este projeto, focou-se em grande parte na modelação e construção de uma ontologia que representasse, da melhor maneira possível os dados sobre artistas, álbuns e músicas. Foram usadas a *Framework Express* para a implementação de um servidor *back-end*, bem como a *Framework Vue* para a implementação do servidor *front-end*. Para auxiliar na construção do website, foi também usado o *plugin Vuetify*, que facilita o uso de diversos componentes nas interfaces.

O trabalho futuro neste projeto focar-se-ia no aumento do conjunto de dados disponíveis, bem como na implementação de mais funcionalidades como por exemplo: a criação de um sistema de autenticação que permitisse a cada utilizador criar playlists com músicas.