

Documento do Projeto – Recomendação Musical com Grafos e Busca

1. Introdução

Este projeto consiste em gerar sequências de músicas com transições naturais. Cada música é representada como um nó em um grafo, e as arestas representam a diferença entre suas características musicais. O sistema deve ser capaz de, dada uma música inicial e uma música alvo, encontrar o caminho mais suave entre elas usando algoritmos de busca. O projeto será desenvolvido seguindo boas práticas exigidas pela disciplina, incluindo TDD, testes estáticos, métricas e relatório Think Aloud.

2. Qual o projeto a ser desenvolvido?

O projeto consiste no desenvolvimento de um sistema capaz de gerar **sequências de músicas com transições suaves**, utilizando **grafos** e **algoritmos de busca**. Cada música é representada como um **nó**, e a similaridade entre duas faixas é representada por **arestas ponderadas**, cujos pesos são calculados a partir de características musicais extraídas automaticamente, como *tempo (BPM)*, *energy*, *danceability*, *valence* e *acousticness*.

A partir de uma música inicial e uma música alvo, o sistema deverá encontrar o **caminho mais natural** entre elas, passando por músicas com características semelhantes. Para isso, algoritmos como **Dijkstra** ou **A*** serão aplicados para determinar o caminho de menor custo no grafo de similaridade musical.

O objetivo final é permitir a criação de playlists coerentes, onde cada transição reproduz uma evolução musical fluida, simulando um processo semelhante ao de recomendações inteligentes observadas em plataformas musicais modernas.

3. Equipe

- Allan de Albuquerque Monteiro
- Denis Almeida Ferreira
- Emerson Henrique Sulpino de Araújo
- Everton Daniel de Lima Romualdo
- Pedro Arthur de Oliveira Barreto
- Vinicius Candido Firmino

4. Tarefas do Projeto

A seguir, estão listadas **todas as tarefas necessárias** para desenvolver o sistema, suas descrições detalhadas e os responsáveis.

4.1. Coleta de Informações e Definição das Características Musicais

Responsáveis:

- Everton Daniel
- Allan de Albuquerque

Descrição da tarefa:

- Estudar como funciona a Spotify Web API.
- Escolher quais atributos musicais melhor representam similaridade (tempo, energy, danceability, valence, acousticness, etc.).
- Justificar por que esses atributos foram escolhidos.
- Definir como será estruturado o dataset final (CSV).
- Criar o script inicial para coleta de características usando *spotipy*.
- Montar o arquivo `songs.csv` com um conjunto de músicas inicial para testes.

Produtos gerados:

- Script `features.py`
- Dataset inicial `dataset/songs.csv`
- Documento curto explicando a escolha dos atributos

4.2. Modelagem do Grafo

Responsáveis:

- Emerson
- Everton Daniel

Descrição da tarefa:

- Definir como o grafo será representado (NetworkX).
- Criar uma função para transformar o dataset em um grafo.
- Definir a métrica de distância (euclidiana, cosseno, etc.).

- Construir todas as arestas calculando similaridade entre músicas.
- Armazenar corretamente pesos nas arestas.
- Criar testes unitários para esta etapa usando Pytest.

Produtos gerados:

- Arquivo `graph.py`
- Grafo inicial funcional
- Testes: `test_graph.py`

4.3. Implementação dos Algoritmos de Busca (Dijkstra e/ou A)*

Responsáveis:

- Emerson
- Allan de Albuquerque

Descrição da tarefa:

- Implementar o algoritmo de busca mais adequado:
 - Dijkstra (caso não seja definida heurística)
 - A* (se grupo decidir criar uma heurística musical baseada no vetor do destino)
- Criar uma função que recebe música inicial e final e retorna a sequência.
- Criar testes automatizados para cada algoritmo.
- Garantir que o algoritmo esteja retornando o caminho mais suave de acordo com a métrica definida.

Produtos gerados:

- Arquivo `search.py`
- Testes: `test_search.py`

4.4. Visualização

Responsável:

- Allan de Albuquerque

Descrição:

- Criar função simples usando NetworkX + Matplotlib para visualizar o grafo.
- Visualização pode ser apenas para debugging interno.

Produto gerado:

- Opcional: `visualization.py`

4.5. Testes e Métricas

Responsáveis:

- Denis Almeida, Vinicius Firmino

Descrição da tarefa:

Inclui todas as métricas exigidas pelo PDF da disciplina:

● Testes Unitários (TDD)

- Cobertura usando `pytest-cov`.
- Testar individualmente:
 - Extração de características (quando possível)
 - Construção do grafo
 - Cálculo de distâncias
 - Algoritmo de busca

● Testes Estáticos – Pylint

- Executar Pylint

- Avaliar problemas e escrever relatório curto sobre os resultados

● Métricas Estáticas do Algoritmo

- Medir tempo de execução da busca com grafos pequenos/médios
- Medir impacto da quantidade de nós
- Medir complexidade geral
- Criar relatório curto (~1 página)

Produtos gerados:

- Testes completos em /tests
- Relatório Pylint
- Relatório de métricas

4.6. Think Aloud

Responsáveis:

- Pedro Arthur de Oliveira Barreto

Descrição da tarefa:

- Escolher uma parte do código considerada difícil (ex: implementação de A^* , cálculo de similaridade).
- Realizar entrevista Think Aloud durante o desenvolvimento.
- Gravar e transcrever (pode usar ferramenta automática).
- Responder o conjunto de perguntas do roteiro fornecido.
- Gerar documento final contendo:
 - Respostas
 - Transcrição
 - Lista dos 10 pontos mais relevantes discutidos durante a sessão

Produto gerado:

- relatorio_think_aloud.pdf

5. Resumo das Tarefas e Responsáveis

Tarefa	Responsáveis
Coleta de características musicais	Everton Daniel, Allan de Albuquerque
Modelagem do grafo	Emerson, Everton Daniel
Algoritmos de busca	Allan de Albuquerque, Emerson
Visualização	Allan de Albuquerque
Testes e métricas	Denis Almeida, Vinicius Firmino
Think Aloud	Pedro Arthur