

Aplicação web para Avaliar a Compatibilidade entre Usuários do Spotify

Davi F. S. Guimarães¹, João Paulo L. L. Oliveira¹, João Victor de Carvalho¹, Christian G. Herrera²

¹Técnico em Informática – Departamento de Computação
Centro Federal de Educação Tecnológica de Minas Gerais
(CEFET-MG)
Divinópolis – MG – Brazil

²Departamento de Computação
Centro Federal de Educação Tecnológica de Minas Gerais
(CEFET-MG)
Divinópolis – MG – Brazil

{guimaraes.davi.fs, joaopaulolacerdaleao, vicjoao2345}@gmail.com

christian.herrera@cefetmg.br

Abstract. *Music is recognized as an essential element of social cohesion, contributing significantly to the formation of interpersonal bonds. Research indicates that individuals with similar musical tastes tend to connect more easily, reinforcing the role of music preferences in the construction of social identity. However, there is still a gap in the availability of tools that can objectively and automatically measure musical compatibility between individuals based on their actual listening habits. In this context, this project aims to develop a web application capable of analyzing data from the Spotify API, such as most listened-to artists, tracks, and genres, applying data science techniques and clustering algorithms to calculate and visualize the degree of musical similarity between users. The system is expected to provide an interactive and intuitive interface, capable of presenting clear compatibility metrics and promoting social interactions based on real affinities.*

Palavras-chave: Spotify API, musical compatibility, data analysis, clustering, web application.

Resumo. *A música é reconhecida como um elemento essencial de coesão social, contribuindo significativamente para a formação de vínculos interpessoais. Pesquisas indicam que pessoas com gostos musicais semelhantes tendem a se conectar com mais facilidade, o que reforça o papel das preferências musicais na construção de identidades sociais. No entanto, ainda há uma lacuna na disponibilidade de ferramentas que permitam mensurar, de forma objetiva e automatizada, a compatibilidade musical entre indivíduos com base em seus hábitos reais de escuta. Diante disso, este projeto tem como propósito desenvolver uma aplicação web capaz de analisar dados da API do Spotify, como artistas, faixas e gêneros mais ouvidos, aplicando técnicas de ciência de dados e algoritmos de agrupamento (clustering) para calcular e visualizar o grau*

de similaridade musical entre usuários. Espera-se que o sistema ofereça uma interface interativa e intuitiva, capaz de apresentar métricas claras de compatibilidade e promover interações sociais baseadas em afinidades reais.

Palavras-chave: Spotify API, compatibilidade musical, análise de dados, *clustering*, aplicação web.

1. Introdução

As relações humanas são influenciadas por diversos fatores, sendo um deles a hipótese da atração por similaridade, proposta por Donn Byrne em seu estudo The Attraction Paradigm [Byrne 1997]. Nessa obra, o autor argumenta que tendemos a nos sentir mais atraídos por pessoas que compartilham nossos gostos, valores e comportamentos. As métricas para essa similaridade variam entre sociedades e gerações, sendo a música, atualmente, um dos indicadores mais significativos de afinidade social.

No entanto, com a quantidade massiva de músicas disponíveis nas plataformas digitais e a diversidade de consumo musical por parte do usuário médio, torna-se difícil realizar uma análise mais detalhada sobre a compatibilidade musical entre indivíduos.

Nesse sentido, discutir soluções tecnológicas que possibilitem uma análise mais profunda dessa similaridade musical significa também explorar caminhos para prever e fortalecer vínculos sociais. O artigo The role of music preferences in early adolescent's friendship formation and stability [Selfhout et al. 2009] reforça essa perspectiva ao demonstrar como a semelhança nos gostos musicais influencia na formação de amizades na adolescência. Tal argumento é também sustentado pela Teoria da Identidade Social [Tajfel 1974], segundo a qual as preferências musicais contribuem para a construção da identidade pessoal e para o sentimento de pertencimento a determinados grupos sociais. Ainda que ambos os estudos se concentrem na adolescência, seus efeitos são perceptíveis, ainda que em menor escala, em outras faixas etárias, evidenciando o papel contínuo da música como elemento mediador nas relações humanas.

Por conseguinte, o compartilhamento de preferências musicais pode impactar a formação e o fortalecimento de vínculos interpessoais [Lonsdale and North 2009]. Estudos indicam haver uma tendência significativa de pessoas se conectarem mais facilmente com aquelas que possuem gostos musicais semelhantes [Boer et al. 2011]. Embora plataformas de *streaming*, como o Spotify, tenham democratizado o acesso à música, ainda são escassas as ferramentas acessíveis que possibilitem a comparação estruturada entre perfis musicais distintos com base em dados objetivos.

Diante desse contexto, o presente projeto propõe o desenvolvimento e a validação de uma aplicação web voltada à mensuração e visualização da compatibilidade musical entre usuários. O sistema será baseado nos dados disponibilizados pela API do Spotify (como faixas, artistas e gêneros mais escutados) e utilizará técnicas de agrupamento (*clustering*) para identificar padrões de similaridade entre os perfis. A proposta alia fundamentos de ciência de dados a aspectos subjetivos da experiência musical, visando oferecer uma ferramenta interativa, tecnicamente fundamentada e potencialmente aplicável em contextos de recomendação e análise social.

2. Objetivos

2.1. Objetivo Geral

Desenvolver uma aplicação *web* capaz de mensurar e representar a compatibilidade musical entre usuários da plataforma Spotify, com base na análise automatizada de dados de escuta.

2.2. Objetivos Específicos

- Permitir a obtenção automatizada de informações sobre o perfil musical de usuários por meio da API do Spotify.
- Oferecer uma análise de similaridade entre perfis musicais com base em técnicas de agrupamento.
- Desenvolver uma interface *web* interativa para visualização da compatibilidade musical entre dois usuários.
- Validar a funcionalidade da aplicação com usuários reais ou por meio de testes simulados.

3. Referencial Teórico

3.1. Música como Elemento de Conexão nas Relações Humanas

A música, presente em todas as culturas conhecidas, sempre desempenhou um papel central nas relações sociais. A prática musical não se limita à expressão artística individual, mas funciona como uma atividade social capaz de criar e fortalecer vínculos entre as pessoas [Turino 2008]. Através da música, sentimentos e emoções podem ser compartilhados de forma que transcende as limitações da linguagem verbal, possibilitando a formação de conexões sociais profundas e duradouras.

O presente trabalho possibilita compreender a importância da música como elemento de conexão nas relações humanas, especialmente no contexto contemporâneo, em que as interações sociais são mediadas cada vez mais por plataformas digitais. Dessa forma, ele contribui para a análise de como a música continua a exercer seu papel social fundamental mesmo em ambientes virtuais, mantendo sua capacidade de unir pessoas e fortalecer vínculos.

3.2. Compartilhamento Musical em Plataformas Digitais

Na última década, o consumo musical passou por uma transformação significativa. Mais de 90% de toda a música consumida mundialmente ocorre em formato digital, sendo o *streaming* responsável por aproximadamente 65% desse total [IFPI 2023]. Plataformas como Spotify, Deezer e Apple Music não apenas facilitam o acesso à música, mas também funcionam como espaços de interação social, permitindo o compartilhamento de *playlists*, o acompanhamento de artistas e amigos, além da criação de comunidades virtuais baseadas em preferências musicais semelhantes.

Diante desse cenário, este trabalho permite entender como o compartilhamento de gostos musicais em plataformas digitais pode influenciar as conexões sociais. Considerando a centralidade dessas plataformas nas interações cotidianas, ele possibilita a análise das relações entre hábitos musicais compartilhados e a formação de vínculos entre indivíduos.

3.3. Afinidade Musical como Indicador de Compatibilidade entre Usuários

A música atua como um importante mecanismo de coesão social, sendo capaz de aproximar indivíduos por meio de gostos e preferências compartilhadas [Boer et al. 2011]. Em contextos digitais, essas preferências tornam-se ainda mais acessíveis e quantificáveis, uma vez que a maioria das pessoas consome música por meio de plataformas de *streaming* [IFPI 2023], como o Spotify.

Nesse contexto, surge a possibilidade de utilizar dados musicais como indicadores de compatibilidade entre usuários. Informações como artistas favoritos, gêneros predominantes e hábitos de escuta podem revelar padrões de afinidade que vão além da música em si, refletindo valores, estilos de vida e até aspectos emocionais. Assim, a análise da afinidade musical se apresenta como um caminho promissor para fomentar conexões entre pessoas em ambientes digitais, sobretudo quando combinada com técnicas computacionais capazes de identificar similaridades e agrupar perfis com base em critérios objetivos.

3.4. Trabalhos Relacionados

Durante a pesquisa de soluções existentes, foram encontradas ferramentas que propõem maneiras distintas de explorar a similaridade musical entre usuários. Entre elas, destaca-se o Spotify Blend, funcionalidade nativa da plataforma Spotify, que cria uma *playlist* mista com base nos hábitos de escuta de dois usuários. No entanto, essa funcionalidade não explica métricas de compatibilidade, nem oferece visualizações ou explicações sobre os dados considerados.

Outra ferramenta relevante é o site MusicTaste.space, que conecta duas contas Spotify e fornece um relatório simples com base em faixas e artistas em comum. Apesar de cumprir parcialmente o objetivo de comparar perfis, a aplicação não emprega técnicas de agrupamento nem visualizações avançadas para análise da afinidade musical.

Dessa forma, observa-se uma lacuna entre as soluções existentes e a proposta deste projeto, que visa oferecer uma análise mais técnica e objetiva da compatibilidade musical entre usuários, por meio do uso de ciência de dados e algoritmos de *clustering*.

4. Metodologia

O desenvolvimento da aplicação será realizado por meio de uma abordagem modular e incremental, dividida em cinco etapas principais: coleta de dados via Spotify API, armazenamento em banco de dados relacional, análise e processamento dos dados, criação do *backend* para exposição dos dados e desenvolvimento da interface interativa. Essa estrutura permite a separação clara entre as camadas da aplicação e favorece a escalabilidade do sistema. Para o controle de versão e o gerenciamento do código, será utilizado o sistema Git, em conjunto com a plataforma GitHub.

A análise de compatibilidade será baseada em métricas derivadas dos dados de escuta dos usuários, utilizando algoritmos de *clustering* para mapear padrões de afinidade musical. Para isso, serão utilizadas ferramentas de análise de dados, além de frameworks para o desenvolvimento web com suporte à integração entre *frontend* e *backend*.

Para assegurar a confiabilidade do fluxo de interação, será seguido o modelo definido no fluxograma (Figura 1) e na máquina de estados finita (FSM), vista na figura 2, garantindo que as transições ocorram da forma esperada e as interfaces exibam corretamente as telas correspondentes a cada estado do processo.

4.1. Representação Visual

4.1.1. Fluxograma

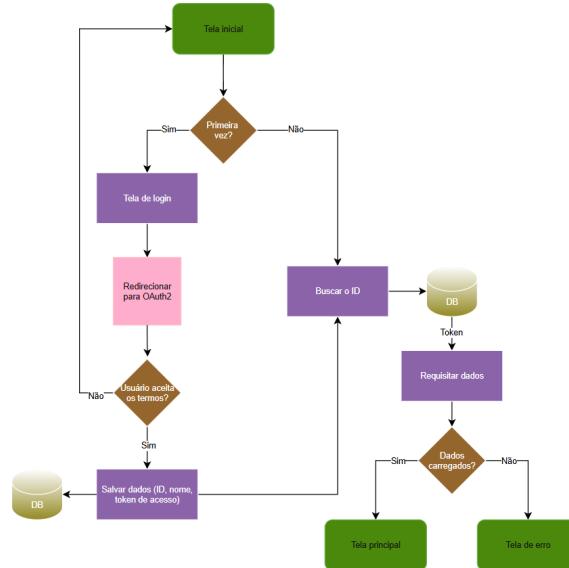


Figura 1. Fluxograma

O fluxograma apresentado (Figura 1) detalha o processo completo de login no sistema:

- Início: estado inicial em que o usuário acessa o site.
- Tela inicial: exibe opções ao usuário. Responsável por verificar se é ou não o primeiro acesso, e decidir se encaminhará o usuário para o login via OAuth2 ou para busca de usuários.
- Tela de Login: apresentada para novos usuários, solicitando que iniciem o processo de autorização.
- Redirecionamento para OAuth2: sistema envia o usuário para a autenticação via API do Spotify.
- Aceite de Termos: decisão que valida se o usuário aceitou os termos de uso. Em caso negativo, retorna para a tela inicial.
- Salvar Dados: salva ID, nome e token no banco de dados para buscas futuras.
- Buscar o ID: para usuários recorrentes, o sistema consulta o banco de dados em busca do identificador salvo.
- Request dos Dados: utilizando o token salvo, faz requisições à API do Spotify para carregar os dados do usuário. Se a requisição for bem sucedida, envia o usuário à tela principal. Do contrário, exibe uma tela de erro.
- Tela Principal / Tela de Erro: estados finais que encerram o fluxo, a depender do resultado anterior.

4.1.2. Máquina de Estados Finita (FSM)

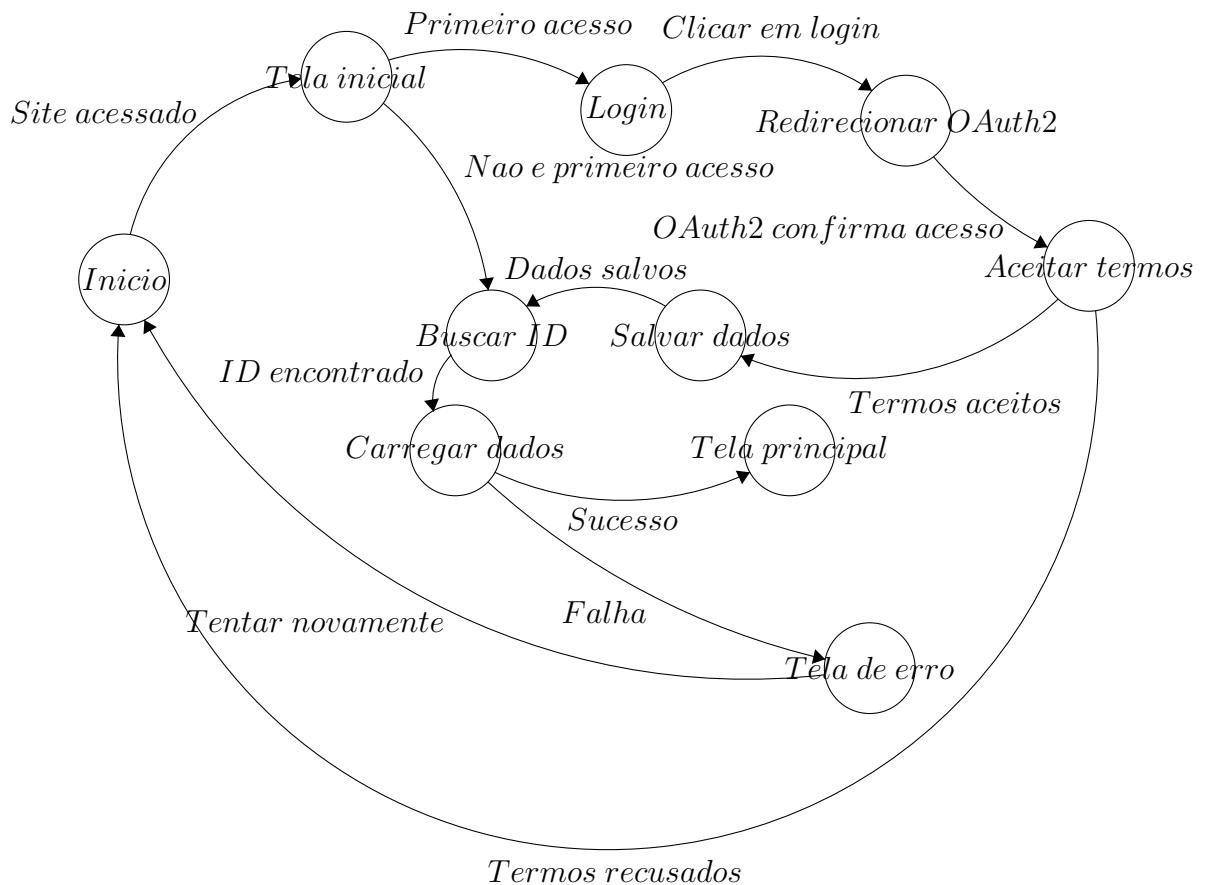


Figura 2. Diagrama de Máquina de Estados Finita (FSM)

Para melhor visualização dos estados do sistema, foi utilizada a Máquina de Estados Finita (Figura 2), com os seguintes estados principais:

- Início: ponto inicial do sistema.
- Tela inicial: exibição da tela inicial. Caso seja o primeiro acesso, redireciona para a tela de login. Do contrário, encaminha para a busca por usuários.
- Tela de login: para novos usuários. Redireciona para autenticação via API Spotify.
- Redirecionar ao OAuth2: transição para autorização externa. Encaminha ao aceite de termos de uso.
- Aceite de termos: validação do aceite do usuário. Caso positivo, executa o salvamento dos dados. Do contrário, retorna à tela inicial.
- Salvar dados: armazenamento do token de acesso. Encaminha à busca por usuários.
- Buscar ID: permite a busca por perfis de usuários. Caso encontre o usuário buscado, carrega os dados. Do contrário, continua na etapa atual para novas buscas.
- Carregar dados: carregamento de dados via API. Caso bem sucedido, leva à tela principal. Do contrário, encaminha à tela de erros.
- Tela principal: sucesso no login. Entra no sistema principal.
- Tela de erro: falha no processo. Retorna à tela inicial.

As transições entre esses estados são disparadas por eventos do usuário (clicar, aceitar termos) e por resultados do sistema (autorização bem-sucedida ou não). A FSM foi utilizada para garantir clareza no controle de fluxo e facilitar a implementação do código.

4.2. Detalhamento de Itens Importantes

4.2.1. draw.io e Finite State Machine Designer

A ferramenta diagrams.net (anteriormente conhecida como draw.io) foi utilizada para a elaboração de fluxogramas do sistema. Trata-se de uma aplicação web gratuita e multiplataforma que permite a criação de diagramas visuais com grande flexibilidade, exportação em diversos formatos e integração com serviços em nuvem [draw.io AG 2000]. Sua escolha se justifica pela interface intuitiva, pela compatibilidade com editores LaTeX e pela facilidade de colaboração entre os membros da equipe durante o design das etapas funcionais da aplicação.

Já o site FSM Designer foi utilizado para modelar uma Máquina de Estados Finita (FSM) que representa o fluxo lógico de navegação e autenticação dos usuários. A ferramenta oferece uma interface minimalista e direcionada à modelagem de autômatos, permitindo a definição clara de estados e transições [Wallace 2010]. Sua simplicidade de uso e a exportação vetorial facilitaram a integração com o documento técnico e a clareza da representação do controle de fluxo da aplicação.

4.2.2. Fast API

FastAPI é um *framework web* e de alto desempenho para construção de APIs em Python, baseado em *type hints* e com suporte nativo a programação assíncrona. Segundo a documentação oficial, ele permite a criação de APIs robustas com validação automática, documentação interativa e tempo de resposta reduzido, sendo ideal para aplicações que requerem escalabilidade e velocidade de desenvolvimento [Ramírez, Sebastián 2024].

No presente projeto, FastAPI será utilizado para construir o backend da aplicação, responsável por fornecer os dados analisados à interface e gerenciar a lógica de compatibilidade entre usuários. A escolha se deu devido à sua excelente integração com bibliotecas de análise de dados em Python, facilidade de documentação automática via Swagger, e suporte nativo a requisições assíncronas, fundamentais para comunicação eficiente com a API do Spotify.

4.2.3. Git

Git é um sistema de controle de versão distribuído e de código aberto que acompanha as alterações nos arquivos de forma eficiente [GitHub, Inc. 2024], permitindo a integração de contribuições de múltiplos desenvolvedores ao longo do tempo.

Neste projeto, o Git será utilizado para o controle das versões do código-fonte, bem como para viabilizar o desenvolvimento colaborativo entre os integrantes da equipe.

O Git foi escolhido em meio a outras alternativas por apresentar uma ampla documentação, grande popularidade na comunidade de desenvolvedores e compatibilidade com plataformas de hospedagem e colaboração, como o GitHub.

4.2.4. Github

GitHub é uma plataforma baseada em nuvem para hospedagem de repositórios Git, que possibilita armazenar, compartilhar e colaborar no desenvolvimento de projetos de software [GitHub, Inc. 2024].

No contexto desta aplicação, o GitHub será responsável pelo armazenamento do repositório principal e pela centralização do desenvolvimento colaborativo, atuando de forma integrada ao Git no controle de versões. A escolha do GitHub justifica-se por ser amplamente adotado pela comunidade de desenvolvedores e por oferecer integração nativa com o Git, o que facilita o gerenciamento do código-fonte e a colaboração entre os membros da equipe.

4.2.5. Pandas, NumPy e scikit-learn

Pandas e NumPy são bibliotecas fundamentais para manipulação e análise de dados em Python. Pandas oferece estruturas de dados como *DataFrame*, ideais para análise tabular [The Pandas Development Team 2024], enquanto NumPy fornece operações matemáticas otimizadas sobre *arrays* [Harris, Charles R. et al. 2024]. Já o scikit-learn é uma biblioteca de aprendizado de máquina, amplamente utilizada para tarefas como *clustering*, classificação e regressão [Scikit-learn Developers 2024].

Essas ferramentas serão empregadas para organizar, processar e extrair padrões dos dados obtidos via Spotify. A vetorização das preferências musicais, normalização de variáveis e aplicação de algoritmos de agrupamento serão realizadas com apoio dessas bibliotecas. A escolha se justifica por sua confiabilidade, grande base de documentação e ampla adoção na comunidade de ciência de dados.

4.2.6. PostgreSQL

PostgreSQL é um sistema de gerenciamento de banco de dados relacional, conhecido por sua robustez, conformidade com padrões SQL e extensibilidade. Ele oferece suporte a tipos de dados avançados, como JSON e arrays, além de facilitar consultas complexas e processamento de dados vetorizados [PostgreSQL Global Development Group 2024].

PostgreSQL será utilizado para armazenar os dados coletados dos usuários, tanto em sua forma bruta (JSON) quanto processada (vetores numéricos). Sua escolha se deve ao bom desempenho com dados semiestruturados e à flexibilidade para futuras expansões da aplicação, como rankings, filtros e cruzamento de dados entre múltiplos usuários.

4.2.7. React e Plotly.js

React é uma biblioteca JavaScript para construção de interfaces de usuário, baseada em componentes reutilizáveis e reatividade. Plotly.js é uma biblioteca para visualização de dados interativos e responsivos em JavaScript. Juntas, permitem criar painéis de análise de dados modernos e com alto grau de personalização visual [Plotly Technologies Inc. 2024].

O frontend da aplicação será construído com React, devido à sua ampla adoção, modularidade e compatibilidade com bibliotecas de visualização como o Plotly.js. A biblioteca Plotly será usada para gerar gráficos que representem a compatibilidade musical, como mapas de calor, gráficos de radar e dispersões comparativas. A escolha se justifica pela qualidade gráfica, interatividade e fácil integração com dados do backend.

4.2.8. Spotify API (Spotipy ou OAuth + Requests)

A Spotify Web API permite que aplicações obtenham dados de perfis de usuários, músicas, artistas e *playlists*. A autenticação é feita por meio do padrão OAuth 2.0, garantindo acesso seguro e consentido [Spotify for Developers 2024]. Spotipy é uma biblioteca Python que abstrai essas interações e facilita o uso da API por meio de funções já estruturadas [Paul Lamere and Contributors 2024].

A API será usada para coletar os dados necessários sobre hábitos de escuta, como top artistas, faixas favoritas e gêneros mais ouvidos. Dependendo do nível de controle desejado, será usada a biblioteca Spotipy (pela facilidade) ou requisições HTTP manuais com requests e tokens OAuth (pela flexibilidade). Essa coleta é fundamental para alimentar as etapas seguintes de análise e visualização.

4.3. Validação da Aplicação

Os testes serão realizados com usuários reais, considerando a necessidade de um histórico pré-existente de dados de escuta. O procedimento de validação incluirá, além da verificação das funcionalidades gerais e da análise de compatibilidade, a avaliação da integração completa entre a API do Spotify e o banco de dados relacional.

Serão verificadas a persistência e a recuperação correta dos dados, a possibilidade de exclusão segura das informações a pedido do usuário, o retorno eficiente de dados previamente armazenados em sessões subsequentes, bem como a robustez do sistema frente a falhas comuns, como tokens expirados ou negação de consentimento.

5. Cronograma

Tabela 1. Cronograma

Semanas	Atividades
1	Pesquisa inicial sobre o tema
2	Análise de viabilidade
3–4	Produção do artigo
5–7	Prototipagem da interface e definição da arquitetura do sistema (front-end e back-end), com foco em usabilidade e escalabilidade
8–10	Implementação da autenticação via Spotify e coleta dos dados de escuta (artistas, faixas e gêneros mais ouvidos)
10–12	Estruturação do banco de dados e pré-processamento das informações coletadas
13–17	Desenvolvimento e testes do algoritmo de agrupamento (<i>clustering</i>) para cálculo de similaridade entre perfis musicais
17–19	Integração entre backend, API, lógica de agrupamento e visualização da compatibilidade musical na interface
20–22	Testes com usuários reais ou simulados, coleta de feedback e avaliação dos resultados
23–24	Ajustes finais na aplicação, documentação técnica do sistema e finalização do texto acadêmico

Referências

- Boer, D., Fischer, R., Strack, M., Bond, M., Lo, E., and Lam, J. W. (2011). How shared preferences in music create bonds between people. *Personality and Social Psychology Bulletin*, 37:1159 – 1171.
- Byrne, D. (1997). An overview (and underview) of research and theory within the attraction paradigm. *Journal of Social and Personal Relationships*, 14(3):417–431.
- draw.io AG, J. L. . (2000). diagrams.net - online diagram software. <https://app.diagrams.net>. Acesso em: 6 jul. 2025.
- GitHub, Inc. (2024). About github - documentation. <https://docs.github.com/en/get-started/learning-about-github/about-github>. Acesso em: 3 jul. 2025.
- Harris, Charles R. et al. (2024). NumPy: the fundamental package for scientific computing with Python. <https://numpy.org/>. Acessado em: 5 jul. 2025.
- IFPI (2023). Global music report 2023: State of the industry. Relatório técnico –, International Federation of the Phonographic Industry. Disponível em PDF. Acesso em: 3 jul. 2025.
- Lonsdale, A. and North, A. C. (2009). Musical taste and ingroup favouritism. *Group Processes Intergroup Relations*, 12:319 – 327.
- Paul Lamere and Contributors (2024). Spotify: A lightweight Python library for the Spotify Web API. <https://spotipy.readthedocs.io/>. Acessado em: 5 jul. 2025.

- Plotly Technologies Inc. (2024). Plotly.js – Open-source JavaScript Graphing Library. <https://plotly.com/javascript/>. Acessado em: 5 jul. 2025.
- PostgreSQL Global Development Group (2024). PostgreSQL Documentation. <https://www.postgresql.org/docs/>. Acessado em: 5 jul. 2025.
- Ramírez, Sebastián (2024). FastAPI: FastAPI framework, high performance, easy to learn, fast to code, ready for production. <https://fastapi.tiangolo.com/>. Acessado em: 5 jul. 2025.
- Scikit-learn Developers (2024). scikit-learn: Machine Learning in Python. <https://scikit-learn.org/>. Acessado em: 5 jul. 2025.
- Selfhout, M. H. W., Branje, S. J. T., ter Bogt, T., and Meeus, W. (2009). The role of music preferences in early adolescent's friendship formation and stability. *Journal of adolescence*, 32 1:95–107.
- Spotify for Developers (2024). Spotify Web API Documentation. <https://developer.spotify.com/documentation/web-api/>. Acessado em: 5 jul. 2025.
- Tajfel, H. (1974). Social identity and intergroup behaviour. *Social Science Information*, 13(2):65–93.
- The Pandas Development Team (2024). Pandas: Python Data Analysis Library. <https://pandas.pydata.org/>. Acessado em: 5 jul. 2025.
- Turino, T. (2008). *Music as Social Life: The Politics of Participation*. University of Chicago Press.
- Wallace, E. (2010). madebyevan.com/fsm/ - finite state machine designer. <https://madebyevan.com/fsm/>. Acesso em: 6 jul. 2025.