

RONALD CARVALHO BOADANA

**MATCHFAI: SISTEMA DE RECOMENDAÇÃO HÍBRIDO DE MÚSICAS USANDO
INTELIGÊNCIA ARTIFICIAL GENERATIVA E AGENTES INTELIGENTES**

Trabalho de Conclusão de Curso apresentado à banca avaliadora do Curso de Engenharia de Computação, da Escola Superior de Tecnologia, da Universidade do Estado do Amazonas, como pré-requisito para obtenção do título de Engenheiro de Computação.

Orientador(a): Prof. Dr. Fábio Santos da Silva

Manaus – Dezembro – 2024

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).
Sistema Integrado de Bibliotecas da Universidade do Estado do Amazonas.

B662m

Boadana, Ronald Carvalho

MATCHFAI: SISTEMA DE RECOMENDAÇÃO HÍBRIDO DE
MÚSICAS USANDO INTELIGÊNCIA ARTIFICIAL

GENERATIVA E AGENTES INTELIGENTES / Ronald Carvalho

Boadana . Manaus : [s.n], 2024.

111 f.: color.; 21,0 cm.

TCC - Graduação em Engenharia de Computação- Universidade
do Estado do Amazonas, Manaus, 2024.

Inclui Apêndice.

Orientador: Fábio Santos da Silva.

1. Inteligência Artificial. 2. Sistemas de Recomendação. 3.
Agentes Inteligentes. 4. Large Language Models. I. Fábio Santos da
Silva (Orient.) II. Universidade do Estado do Amazonas.

CDU(1997)62:004



FOLHA DE APROVAÇÃO

MATCHFAI: SISTEMA DE RECOMENDAÇÃO HÍBRIDO DE MUSICAS USANDO INTELIGENCIA ARTIFICIAL GENERATIVA E AGENTES INTELIGENTES

Ronald Carvalho Boadana

Trabalho de Conclusão de Curso defendido e aprovado pela banca avaliadora constituída pelos professores:

Fabio Santos da Silva
Presidente

Carlos Maurício Serodio Figueiredo
Avaliador(a)

Tiago Eugenio de Melo
Avaliador(a)

Manaus, 20 de Dezembro de 2024

Resumo

Este trabalho explora o uso de grandes modelos de linguagem (*large language models*, em inglês - LLMs) um sistema de recomendação de músicas híbrido, juntamente com agentes de inteligência artificial, apresentando uma nova abordagem para recomendação musical. A proposta inclui uma arquitetura que utiliza diversos agentes para realizar uma filtragem híbrida, combinando filtragem baseada em conteúdo e colaborativa, além de outros agentes de suporte. Diversos LLMs, como Gemini, LLaMA e Mixtral, são empregados como motores de recomendação. Os resultados finais demonstram que o modelo Gemini 1.5 Pro apresentou o melhor desempenho entre os modelos avaliados, conforme os dados obtidos por meio de testes realizados com os usuários.

Palavras-Chave: Inteligência Artificial; Sistemas de Recomendação; *Large Language Models*; Agentes Inteligentes.

Abstract

This work explores the use of Large Language Models (LLMs) in a hybrid music recommendation system, combined with artificial intelligence agents, presenting a new approach to music recommendation. The proposed architecture includes various agents to perform hybrid filtering, combining content-based and collaborative filtering, along with other support agents. Several LLMs, such as Gemini, LLaMA, and Mixtral, are employed as recommendation engines. The final results show that the Gemini 1.5 Pro model achieved the best performance among the evaluated models, based on data obtained from user testing.

Keywords: Artificial Intelligence; Recommendation Systems; Large Language Models; Intelligent Agents.

Agradecimentos

Primeiramente, gostaria de agradecer a Deus por me guiar e me dar o suporte ao longo de toda a minha vida até aqui. Agradeço também, de coração, aos meus incríveis pais e a minha família, que sempre estiveram ao meu lado, me apoiando com muito incentivo e amor. E aos meus grandes amigos em especial (Arthur, Debora, Fernando, Ademir, Gustavo, Geovani, Adriano, Adevan, José e Adelmo) e tantos outros que conquistei ao longo desses 5 anos, com os quais vivemos muitas experiências e dificuldades ao longo do caminho. Por fim, agradeço ao meu orientador, Fábio Santos, por me acolher e compartilhar seus conhecimentos, contribuindo de forma fundamental para o desenvolvimento deste trabalho.

Sumário

Lista de Tabelas	x
Lista de Figuras	xv
1 Introdução	1
1.1 Justificativa	3
1.2 Objetivos	3
1.2.1 Objetivos Específicos	4
1.3 Metodologia	4
2 Fundamentação Teórica	5
2.1 Sistemas de Recomendação	5
2.1.1 Filtragem Baseada em Conteúdo	6
2.1.2 Filtragem Colaborativa	6
2.1.3 Filtragem Híbrida	7
2.2 Inteligência Artificial	9
2.2.1 Processamento de Linguagem Natural	10
2.2.2 <i>Large Language Models</i>	11
2.2.3 Agentes Inteligentes	14
2.3 <i>Application Programming Interface</i>	14
2.3.1 HTTP	15
2.3.2 REST	19

2.4	Conclusão	19
3	Trabalhos Relacionados	20
3.1	<i>LLM Based Generation of Item-Description for Recommendation System</i>	20
3.2	<i>TPTU: Large Language Model-based AI Agents for Task Planning and Tool Usage</i>	21
3.3	<i>Music Personalized Recommendation System Based on Hybrid Filtration</i>	22
3.4	<i>Uncovering ChatGPT's Capabilities in Recommender Systems</i>	23
3.5	<i>Recommender Systems in the Era of Large Language Models</i>	25
3.6	Análise Comparativa	26
4	Proposta de Arquitetura e Implementação	27
4.1	Arquitetura	27
4.1.1	Nível 1: Sistema	27
4.1.2	Nível 2: Contêiner	28
4.1.3	Nível 3: Componentes	29
4.2	<i>Chartsfy</i>	32
4.2.1	Cadastro de usuários	33
4.2.2	Fluxo do Funcionamento	34
4.2.3	Base de Dados	35
4.3	MatchfAI - API	36
4.3.1	Agentes	37
4.3.2	<i>Tools</i>	38
4.3.3	Tarefas	39
4.4	Estruturação dos Prompts e Ferramentas	39
4.4.1	Ferramentas	39
4.4.2	Agentes - Filtro Baseado em Conteúdo	41
4.4.3	Agentes - Filtro Colaborativo	43
4.4.4	Agentes - Filtro Híbrido	46
4.4.5	Agentes - Nova recomendação	48

4.4.6	Tarefas - Filtro Baseado em Conteúdo	48
4.4.7	Tarefas - Filtro Colaborativo	50
4.4.8	Tarefas - Filtro Híbrido	53
4.4.9	Tarefas - Nova recomendação	54
4.5	Banco de Dados	55
4.5.1	<i>genres</i>	55
4.5.2	<i>artistinfos</i>	56
4.5.3	<i>songscatologs</i>	56
4.5.4	<i>userinfos</i>	57
4.5.5	<i>userrrchistories</i>	57
4.5.6	<i>musichistories</i>	57
4.5.7	<i>userdislikedsongs</i>	57
4.5.8	<i>userlikedsongs</i>	57
4.5.9	<i>userevaluations</i>	58
4.5.10	<i>usersrelevants</i>	58
4.6	Rotina de Atualização de Dados	58
4.6.1	Pré-Processamento dos Dados	58
4.7	Fluxos de Geração da Recomendação	60
4.7.1	Fluxo Geral	60
4.7.2	Fluxo - Sem Músicas Avaliadas	62
4.7.3	Fluxo - Sem Histórico de Músicas Ouvidas	63
4.8	<i>MatchfAI - Backend</i>	64
4.9	Modelos Utilizados	66
4.10	<i>MatchfAI - FrontEnd</i>	66
4.11	Métricas de Avaliação	68
4.12	Avaliação das Recomendações	68
4.12.1	Processo de Testes	69
4.13	Conclusão	74

5	Resultados e Discussões	75
5.1	Comparação dos Modelos	75
5.2	Custos	76
6	Considerações Finais	77
6.1	Trabalhos Futuros	80
A	Perfis de Usuários	85
B	Avaliações dos Usuários	88
C	Avaliações dos Usuários - Itens Relevantes	100

Lista de Tabelas

2.1	Comparação dos modelos de LLM com o benchmark MMLU	12
2.2	Relação do CRUD com os métodos HTTP	19
4.1	Modelos utilizados	66
5.1	Métricas de cada modelo	75
A.1	Perfil Usuário 1	85
A.2	Perfil Usuário 2	85
A.3	Perfil Usuário 3	85
A.4	Perfil Usuário 4	86
A.5	Perfil Usuário 5	86
A.6	Perfil Usuário 6	86
A.7	Perfil Usuário 7	86
A.8	Perfil Usuário 8	86
A.9	Perfil Usuário 9	87
A.10	Perfil Usuário 10	87
A.11	Perfil Usuário 11	87
A.12	Perfil Usuário 12	87
A.13	Perfil Usuário 13	87
B.1	Avaliação Usuário 1 (Gemini)	88
B.2	Avaliação Usuário 1 (Llama)	88
B.3	Avaliação Usuário 1 (Mistral)	89

B.4	Avaliação Usuário 2 (Gemini)	89
B.5	Avaliação Usuário 2 (Llama)	89
B.6	Avaliação Usuário 3 (Gemini)	90
B.7	Avaliação Usuário 3 (Llama)	90
B.8	Avaliação Usuário 3 (Mistral)	90
B.9	Avaliação Usuário 4 (Gemini)	91
B.10	Avaliação Usuário 4 (Llama)	91
B.11	Avaliação Usuário 4 (Mistral)	91
B.12	Avaliação Usuário 5 (Gemini)	92
B.13	Avaliação Usuário 5 (Llama)	92
B.14	Avaliação Usuário 5 (Mistral)	92
B.15	Avaliação Usuário 6 (Gemini)	93
B.16	Avaliação Usuário 6 (Llama)	93
B.17	Avaliação Usuário 6 (Mistral)	93
B.18	Avaliação Usuário 7 (Gemini)	94
B.19	Avaliação Usuário 7 (Llama)	94
B.20	Avaliação Usuário 7 (Mistral)	94
B.21	Avaliação Usuário 8 (Llama)	95
B.22	Avaliação Usuário 9 (Gemini)	95
B.23	Avaliação Usuário 9 (Llama)	95
B.24	Avaliação Usuário 9 (Mistral)	96
B.25	Avaliação Usuário 10 (Gemini)	96
B.26	Avaliação Usuário 10 (Llama)	96
B.27	Avaliação Usuário 10 (Mistral)	97
B.28	Avaliação Usuário 11 (Gemini)	97
B.29	Avaliação Usuário 11 (Llama)	97
B.30	Avaliação Usuário 12 (Gemini)	98
B.31	Avaliação Usuário 12 (Llama)	98

B.32 Avaliação Usuário 12 (Mistral)	98
B.33 Avaliação Usuário 13 (Gemini)	99
B.34 Avaliação Usuário 13 (Llama)	99
C.1 Avaliação Usuário 1 - Itens Relevantes (Gemini)	100
C.2 Avaliação Usuário 1 - Itens Relevantes (Llama)	100
C.3 Avaliação Usuário 1 - Itens Relevantes (Mistral)	101
C.4 Avaliação Usuário 2 - Itens Relevantes (Gemini)	101
C.5 Avaliação Usuário 3 - Itens Relevantes (Gemini)	101
C.6 Avaliação Usuário 3 - Itens Relevantes (Llama)	102
C.7 Avaliação Usuário 3 - Itens Relevantes (Mistral)	102
C.8 Avaliação Usuário 4 - Itens Relevantes (Gemini)	102
C.9 Avaliação Usuário 4 - Itens Relevantes (Llama)	103
C.10 Avaliação Usuário 4 - Itens Relevantes (Mistral)	103
C.11 Avaliação Usuário 5 - Itens Relevantes (Gemini)	103
C.12 Avaliação Usuário 5 - Itens Relevantes (Llama)	104
C.13 Avaliação Usuário 5 - Itens Relevantes (Mistral)	104
C.14 Avaliação Usuário 6 - Itens Relevantes (Gemini)	104
C.15 Avaliação Usuário 6 - Itens Relevantes (Llama)	105
C.16 Avaliação Usuário 6 - Itens Relevantes (Mistral)	105
C.17 Avaliação Usuário 7 - Itens Relevantes (Gemini)	105
C.18 Avaliação Usuário 7 - Itens Relevantes (Llama)	106
C.19 Avaliação Usuário 7 - Itens Relevantes (Mistral)	106
C.20 Avaliação Usuário 8 - Itens Relevantes (Gemini)	106
C.21 Avaliação Usuário 9 - Itens Relevantes (Gemini)	107
C.22 Avaliação Usuário 9 - Itens Relevantes (Llama)	107
C.23 Avaliação Usuário 9 - Itens Relevantes (Mistral)	107
C.24 Avaliação Usuário 10 - Itens Relevantes (Gemini)	108
C.25 Avaliação Usuário 10 - Itens Relevantes (Llama)	108

C.26 Avaliação Usuário 10 - Itens Relevantes (Mistral)	108
C.27 Avaliação Usuário 11 - Itens Relevantes (Gemini)	109
C.28 Avaliação Usuário 11 - Itens Relevantes (Llama)	109
C.29 Avaliação Usuário 11 - Itens Relevantes (Mistral)	109
C.30 Avaliação Usuário 12 - Itens Relevantes (Gemini)	110
C.31 Avaliação Usuário 12 - Itens Relevantes (Llama)	110
C.32 Avaliação Usuário 12 - Itens Relevantes (Mistral)	110
C.33 Avaliação Usuário 13 - Itens Relevantes (Gemini)	111
C.34 Avaliação Usuário 13 - Itens Relevantes (Llama)	111

Lista de Figuras

2.1	Sistema de recomendação como uma caixa preta (JANNACH; ZANKER; FELFERNIG, 2010)	8
2.2	Design paralelizado (JANNACH; ZANKER; FELFERNIG, 2010)	9
2.3	Design em <i>pipeline</i> (JANNACH; ZANKER; FELFERNIG, 2010)	9
2.4	Esquema de como os agentes interagem com o ambiente	15
2.5	Mensagem genérica de requisição HTTP	16
2.6	Mensagem genérica de resposta HTTP	16
2.7	Exemplo de GET	16
2.8	Exemplo de POST	17
2.9	Exemplo de PUT	17
2.10	Exemplo de DELETE	18
4.1	Nível 1: Sistema	28
4.2	Nível 2: Contêiner	29
4.3	Nível 3: Componentes	30
4.4	Forma 1 - Ferramenta, Agente e Tarefa	32
4.5	Forma 2 - Agente e Tarefa	32
4.6	Página inicial do <i>Chartsfy</i>	33
4.7	Autentificação com o <i>Spotify</i>	33
4.8	Formulário de autorização da aplicação (SPOTIFY, 2024a)	34
4.9	Esquemático da aplicação <i>Chartsfy</i>	35
4.10	<i>Collections</i> do banco de dados do <i>Chartsfy</i>	35

4.11	Representação e fluxo de uma ferramenta	38
4.12	Modelagem das <i>collections</i> do MongoDB (SPOTIFY, 2024a)	56
4.13	Fluxo Geral	61
4.14	Fluxo do Filtro Baseado em Conteúdo	61
4.15	Fluxo do Filtro Colaborativo	62
4.16	Fluxo do Filtro Híbrido	62
4.17	Fluxo do Recomendador de Músicas	63
4.18	Interface	67
4.19	Processo de avaliação do sistema	69
4.20	Inserção do email e ativação da opção <i>RelevantMode</i>	70
4.21	Modal com opções de gêneros	70
4.22	Escolha do modelo	70
4.23	Geração da recomendação de itens relevantes	71
4.24	Avaliação e envio	71
4.25	Inserção do email e sem a ativação da opção <i>RelevantMode</i>	72
4.26	Escolha do modelo	72
4.27	Geração da recomendação	73
4.28	Avaliação da recomendação	74

Capítulo 1

Introdução

Sistemas de recomendação são softwares que preveem as preferências dos usuários e recomendam itens que eles provavelmente gostarão. Esses sistemas são usados em uma ampla variedade de aplicações, incluindo comércio eletrônico, sistemas de filtragem de conteúdo e plataformas de mídia social. Existem dois tipos principais de filtragem em sistemas de recomendação: filtragem baseada em conteúdo e filtragem colaborativa. A filtragem baseada em conteúdo recomenda itens a um usuário com base na similaridade entre esses itens e os itens que o usuário já gostou no passado (JANNACH; ZANKER; FELFERNIG, 2010). Esse tipo de filtragem é muito utilizado para recomendação em serviços de streaming de músicas como Apple Music (APPLE, 2024), Spotify (SPOTIFY, 2024b), Deezer (DEEZER, 2024) e TIDAL (TIDAL, 2024).

A filtragem colaborativa recomenda itens a um usuário com base na similaridade entre o usuário e outros usuários que gostaram desses itens no passado (JANNACH; ZANKER; FELFERNIG, 2010). Essa recomendação por ser mais utilizada em redes sociais, é visível no Last.fm (LAST.FM, 2024), uma rede social com foco em músicas que os usuários ouvem. Porém existe a possibilidade de combinar diversas formas de filtragem, pois combinação de diferentes técnicas pode levar a recomendações mais precisas e relevantes (JANNACH; ZANKER; FELFERNIG, 2010) assim existindo a filtragem híbrida. O Spotify utiliza dessa abordagem em suas recomendações de músicas para os usuários (SPOTIFY, 2024a). Esta também pode utilizar desde duas técnicas mencionadas anteriormente até o uso de algoritmos de Inteligência Artificial (ELAHI et al., 2023).

Existem diversas aplicações na área de Inteligência Artificial como Machine Learning, Deep Learning, IA Generativa e Visão Computacional. Atualmente com a explosão do uso de IA Generativa é visível com aplicações de LLMs (Grandes Modelos de Linguagem) como GPT (OPENAI, 2024) e Gemini (GOOGLE, 2024). Um grande modelo de linguagem (LLM) é um modelo de linguagem treinado em um enorme conjunto de dados de texto. LLMs podem ser usados para uma variedade de tarefas, incluindo geração de texto, tradução automática e resposta a perguntas (RUSSELL; NORVIG, 2022). Recentemente, as LLMs surgiram como ferramentas poderosas e trouxeram benefícios significativos para os sistemas de recomendação tornando-os altamente vantajosos para recomendações personalizadas (HUA et al., 2023).

Além das mais conhecidas pelo público geral, existem as LLMs de código aberto como LLaMA (META, 2024a) e Mixtral (MISTRAL, 2024), nas quais apresentam desempenhos consideráveis e podem ser usadas para diversos fins. Além disso com as LLMs, é possível fazer o uso de agentes inteligentes para realizar tarefas pré-determinadas anteriormente. Também são conhecidos como um sistema que pode raciocinar, aprender e agir de forma autônoma. Agentes inteligentes são frequentemente baseados em modelos de aprendizado de máquina, como redes neurais artificiais e modelos de linguagem grande (LLMs). LLMs podem ser usados para dar aos agentes a capacidade de entender e raciocinar sobre o mundo ao seu redor, aprender com a experiência e tomar decisões autônomas (RUSSELL; NORVIG, 2022).

Este trabalho propõe um sistema de recomendação híbrido de músicas, utilizando uma arquitetura construída com agentes de inteligência artificial e LLMs. Entre os modelos selecionados estão Gemini, LLaMA e Mixtral, com o objetivo de explorar e comparar as recomendações entre modelos de código aberto com modelos de código fechado.

A organização deste trabalho está estruturada em seis capítulos. O Capítulo apresenta uma revisão da literatura, abordando os principais conceitos e fundamentos teóricos relacionados ao tema. O Capítulo 3 discute trabalhos relacionados, incluindo uma análise comparativa entre este estudo e outras abordagens existentes. O Capítulo 4 detalha a proposta de arquitetura e o processo de implementação do sistema. O Capítulo 5 expõe os resultados obtidos e promove uma análise crítica por meio de discussões. Por fim, o Capítulo 6 apresenta as considerações

finais, destacando as conclusões alcançadas e possíveis direções para trabalhos futuros.

1.1 Justificativa

Diversos aspectos são considerados na geração de recomendações musicais, tanto pela filtragem colaborativa quanto pela baseada em conteúdo. Entre esses aspectos estão: artista, álbum, gênero, duração da música, ano de lançamento, usuários que ouviram a música e similaridade entre esses usuários, porém a filtragem colaborativa requer uma grande comunidade de usuários para prover recomendações úteis (WEIHS et al., 2017) também vale mencionar a viabilidade da filtragem baseada em conteúdo, na qual diminui em grandes conjuntos de dados. Isso ocorre porque os dados precisam ser atualizados constantemente para garantir a precisão das recomendações, o que pode ser impraticável para conjuntos de dados volumosos (WEIHS et al., 2017).

Nos últimos anos várias LLMs como GPT-3 (OPENAI, 2024) e outras de código aberto, Alpaca (CRFM, 2023), emergiram como ferramentas poderosas para o processamento de linguagem natural (ACHARYA; SINGH; ONOE, 2023) e devido ao seu grande poder tornou-se possível construir ferramentas de recomendação mais efetivas, multitarefas e multimodais (HUA et al., 2023). Com essas melhorias, o uso de LLMs ganhou atenção significativa no domínio dos sistemas de recomendação (WU et al., 2024). Além disso, os sistemas de recomendação baseados em LLMs podem melhorar a acurácia e relevância das recomendações, levando a uma satisfação maior do usuário (WU et al., 2024). Logo, desenvolver um sistema de recomendação de músicas de filtragem híbrida usando uma LLM se mostra uma abordagem inovadora para a recomendação de músicas comparado as abordagens tradicionais usando filtragem colaborativa ou baseada em conteúdo.

1.2 Objetivos

Este trabalho tem como objetivo desenvolver um sistema de recomendação híbrido de músicas fazendo o uso de filtragem híbrida, LLMs de código aberto e agentes inteligentes para depois

realizar uma comparação entre desempenho e recomendações.

1.2.1 Objetivos Específicos

- Obter os dados a serem utilizados e pré-processá-los para deixá-los adequados para o desenvolvimento do sistema de recomendação;
- Criar uma API para consumo dos dados pelos agentes;
- Criar os agentes de filtragem baseada em conteúdo e colaborativa;
- Criar um agente de filtragem híbrida que una as recomendações geradas e crie a playlist com as músicas para cada usuário;
- Comparar as recomendações geradas e utilizar métricas adequadas;

1.3 Metodologia

Neste trabalho adotará a seguinte metodologia para atingir os objetivos mencionados anteriormente:

1. Realizar estudos e experimentações com sistemas de recomendação de filtragem híbrida;
2. Realizar estudos e experimentações de agentes inteligentes usando LLMs para gerar recomendações;
3. Identificar e modelar a arquitetura mais adequada ao sistema;
4. Identificar as métricas de avaliação a serem utilizadas;
5. Testes com usuários;
6. Análises e discussões dos resultados obtidos;

Capítulo 2

Fundamentação Teórica

As seções desse capítulo reúnem o resumo da literatura e abordam os conceitos de Sistemas de Recomendação, Inteligência Artificial e APIs.

2.1 Sistemas de Recomendação

Sistemas de recomendação são softwares que preveem as preferências dos usuários e recomendam itens que eles provavelmente gostarão. Esses sistemas utilizam diversos algoritmos e técnicas para analisar dados coletados sobre o comportamento e as preferências dos usuários, e assim fornecer sugestões personalizadas (JANNACH; ZANKER; FELFERNIG, 2010). Essa personalização melhora a experiência do usuário ao apresentar-lhe opções relevantes em meio a uma vasta quantidade de informações disponíveis. Esses sistemas são usados em uma ampla variedade de aplicações, incluindo comércio eletrônico, sistemas de filtragem de conteúdo e plataformas de mídia social. Logo um sistema de recomendação buscar prever a classificação ou preferência que um usuário daria a um item (como música, livros ou filmes) ou elemento social (por exemplo, pessoas ou grupos) que ele ainda não considerou. Para isso, utilizam um modelo construído a partir das características de um item (filtragem baseada em conteúdo) ou do ambiente social do usuário (filtragem colaborativa) (SHARMA; GERA, 2013).

Existem três abordagens principais de recomendação: filtragem colaborativa, baseada em conteúdo e baseada em conhecimento. Cada uma tem vantagens e desvantagens, como lidar

com dados esparsos e itens novos. Um sistema de recomendação é como uma caixa preta que transforma dados em recomendações. Dados de usuário, comunidade, produto e conhecimento podem ser usados como entrada. Sistemas híbridos combinam diferentes abordagens para superar limitações. O termo híbrido vem do latim e significa algo feito da mistura de elementos. Sistemas híbridos combinam diferentes algoritmos e componentes de recomendação (JANNACH; ZANKER; FELFERNIG, 2010).

2.1.1 Filtragem Baseada em Conteúdo

Segundo (JANNACH; ZANKER; FELFERNIG, 2010) sistemas de recomendação colaborativa filtram itens com base nas preferências de usuários similares. Isso evita a necessidade de descrever detalhadamente cada item, mas limita a recomendação baseada em características específicas. Já a recomendação baseada em conteúdo usa informações sobre os itens e o perfil do usuário para sugerir itens que combinem com as preferências do usuário. Essa abordagem necessita de informações adicionais, mas permite recomendações mesmo para usuários únicos. Geralmente, as características do item são extraídas automaticamente do próprio conteúdo do item ou de descrições textuais. Exemplos incluem sistemas que recomendam notícias com base em palavras-chave.

2.1.2 Filtragem Colaborativa

A ideia principal da abordagem de recomendação colaborativa é explorar informações sobre o comportamento passado ou as opiniões de uma comunidade de usuários existente para prever quais itens o usuário atual do sistema provavelmente gostará ou terá interesse. Esses tipos de sistemas são amplamente utilizados na indústria atualmente, principalmente como ferramenta em sites de varejo online para personalizar o conteúdo de acordo com as necessidades de um cliente específico e, assim, promover itens adicionais e aumentar as vendas. Do ponto de vista da pesquisa, esses tipos de sistemas são explorados há muitos anos, e suas vantagens, desempenho e limitações são hoje bem compreendidos. Ao longo dos anos, vários algoritmos e técnicas foram propostos e avaliados com sucesso em dados de teste reais e artificiais. As abordagens

colaborativas puras utilizam apenas uma matriz de classificações de usuário-item como entrada e normalmente produzem os seguintes tipos de saída: (a) uma previsão (numérica) indicando o quanto o usuário atual gostará ou não gostará de um determinado item e (b) uma lista de n itens recomendados. Essa lista "top N ", obviamente, não deve conter itens que o usuário atual já comprou. (JANNACH; ZANKER; FELFERNIG, 2010)

Na filtragem colaborativa existem várias recomendações diferentes como por exemplo:

- Recomendação por vizinhança baseada em usuário (*User-based Nearest Neighbor recommendation*) é uma técnica específica dentro das abordagens de recomendação colaborativa. Ela se concentra em encontrar usuários com gostos similares ao usuário atual e, então, recomendar itens que esses usuários vizinhos gostaram.
- Recomendação por vizinhança baseada em item (*Item-based Nearest Neighbor recommendation*) é outra técnica específica dentro das abordagens de recomendação colaborativa. Ao contrário da recomendação baseada em usuário, aqui o foco está em encontrar itens similares ao que o usuário já curtiu ou interagiu. Nas abordagens de recomendação baseadas em item, a similaridade de cosseno, *cosine similarity*, se estabelece como a métrica padrão, pois demonstrou produzir os resultados mais precisos. Essa métrica mede a similaridade entre dois vetores n -dimensionais com base no ângulo entre eles.

2.1.3 Filtragem Híbrida

Segundo (JANNACH; ZANKER; FELFERNIG, 2010) existem três abordagens principais de recomendação: filtragem colaborativa, baseada em conteúdo e baseada em conhecimento. Cada uma tem vantagens e desvantagens, como lidar com dados esparsos e itens novos. Um sistema de recomendação é como uma caixa preta que transforma dados em recomendações. Dados de usuário, comunidade, produto e conhecimento podem ser usados como entrada. Sistemas híbridos combinam diferentes abordagens para superar limitações. Sistemas híbridos combinam diferentes algoritmos e componentes de recomendação, com isso vários pesquisadores têm tentado combinar a filtragem colaborativa e abordagens baseadas em conteúdo para suavizar suas desvantagens e obter melhor performance nas recomendações (SHARMA; GERA, 2013).

Uma dimensão que caracteriza um algoritmo híbrido é o seu design. Existem 3 tipos base: monolítico, paralelizado ou *pipeline* híbrida.

Monolítico descreve um design de hibridização que incorpora aspectos de várias estratégias de recomendação em um único algoritmo. Conforme ilustrado na figura abaixo, vários sistemas de recomendação contribuem de forma virtual porque o híbrido usa dados de entrada adicionais específicos de outro algoritmo de recomendação, ou os dados de entrada são aumentados por uma técnica e explorados factualmente pela outra. Por exemplo, um sistema de recomendação baseado em conteúdo que também aproveita dados da comunidade para determinar similaridades entre itens se enquadra nessa categoria (JANNACH; ZANKER; FELFERNIG, 2010).

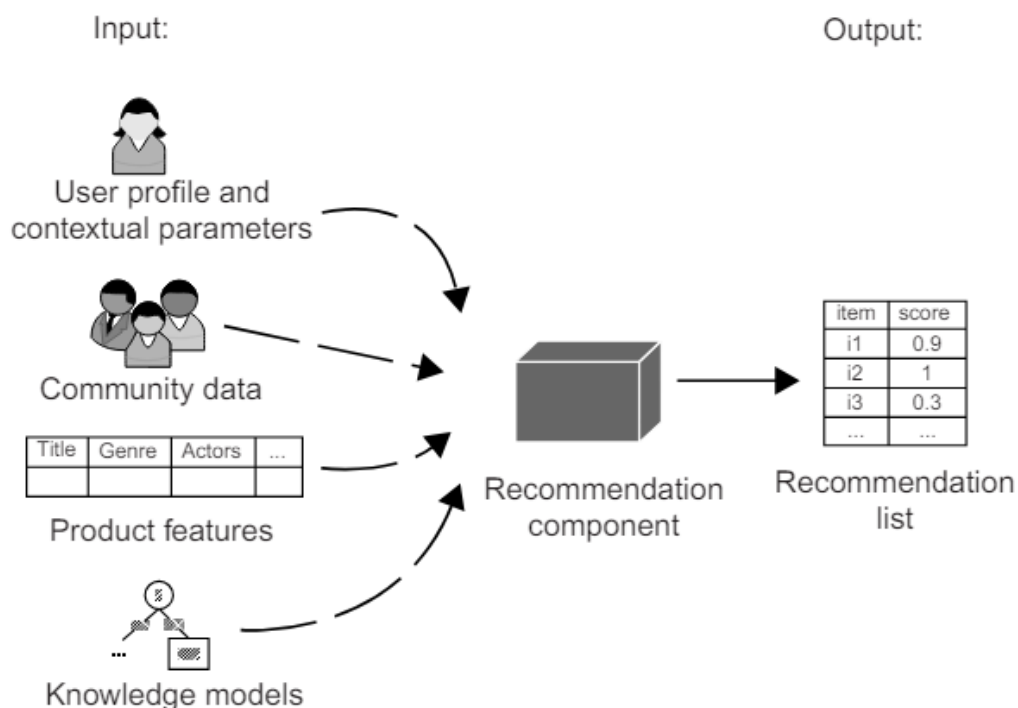


Figura 2.1: Sistema de recomendação como uma caixa preta (JANNACH; ZANKER; FELFERNIG, 2010)

Os dois projetos de hibridização restantes requerem pelo menos duas implementações separadas de sistemas de recomendação, que são então combinadas. Os sistemas de recomendação híbridos paralelizados operam independentemente uns dos outros com base em sua entrada e produzem listas de recomendação separadas. Em uma etapa subsequente de hibridização, suas saídas são combinadas em um conjunto final de recomendações (JANNACH; ZANKER;

FELFERNIG, 2010).

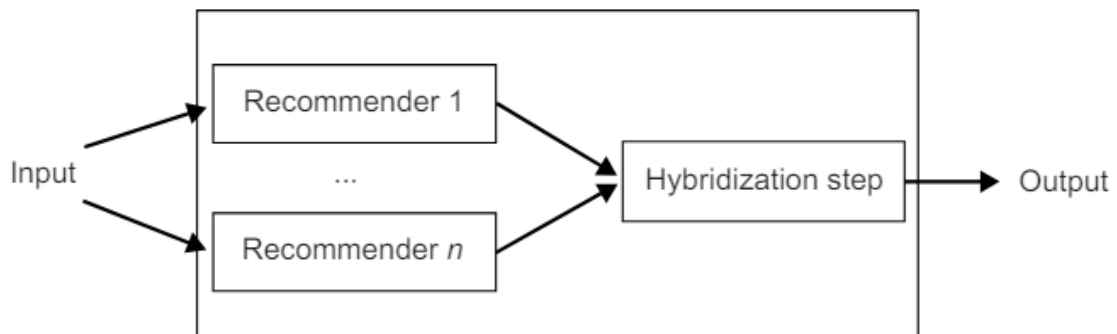


Figura 2.2: Design paralelo (JANNACH; ZANKER; FELFERNIG, 2010)

Quando diversos sistemas de recomendação são reunidos juntos em uma arquitetura em *pipeline*, a saída de um recomendador se torna parte da entrada de um subsequente (JANNACH; ZANKER; FELFERNIG, 2010).

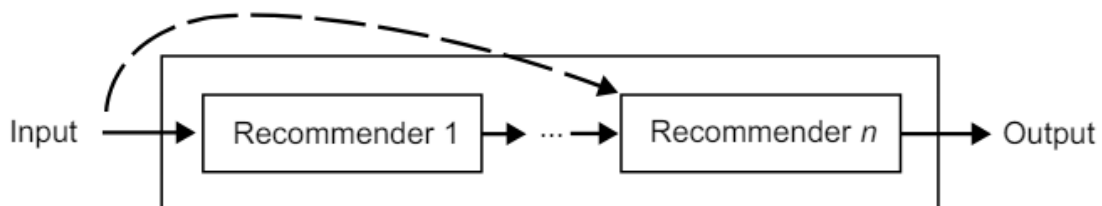


Figura 2.3: Design em *pipeline* (JANNACH; ZANKER; FELFERNIG, 2010)

2.2 Inteligência Artificial

A inteligência artificial (*Artificial Intelligence*, em inglês - AI) é um dos campos mais recentes em ciências e engenharia. O trabalho começou logo após a Segunda Guerra Mundial, e o próprio nome foi cunhado em 1956. Atualmente, a AI abrange uma enorme variedade de subcampos, do geral (aprendizagem e percepção) até tarefas específicas, como jogos de xadrez, demonstração de teoremas matemáticos, criação de poesia, direção de um carro em estrada movimentada e diagnóstico de doenças (RUSSELL; NORVIG, 2022). Logo, a AI surge com objetivo de realizar atividades humanas de forma muito mais precisa e mais rápida. Na grande variedade

de aplicações, a AI ganha destaque na área de processamento de linguagem natural (*Natural Language Processing*, em inglês - NLP) com o uso expressivo de grandes modelos de linguagem (*Large Language Models*, em inglês - LLM) como GPT-3 (OPENAI, 2024) e Gemini (GOOGLE, 2024).

2.2.1 Processamento de Linguagem Natural

Processamento de Linguagem Natural (*Natural Language Processing*, em inglês - NLP) é um termo coletivo que se refere ao processamento computacional automático de linguagens humanas. Isso inclui algoritmos que recebem texto produzido por humanos como entrada e algoritmos que produzem texto com aparência natural como saída. A necessidade desses algoritmos está cada vez maior: os humanos produzem quantidades cada vez maiores de texto a cada ano e esperam que as interfaces de computador se comuniquem com eles em sua própria língua. O processamento de linguagem natural também é muito desafiador, pois a linguagem humana é inerentemente ambígua, está sempre mudando e não é bem definida (GOLDBERG, 2017).

A linguagem natural é simbólica por natureza, e as primeiras tentativas de processamento de linguagem foram simbólicas: baseadas em lógica, regras e ontologias. No entanto, a linguagem natural também é altamente ambígua e variável, exigindo uma abordagem algorítmica mais estatística. De fato, as abordagens dominantes atuais para processamento de linguagem são todas baseadas em aprendizado de máquina estatístico. Por mais de uma década, as técnicas centrais de PNL foram dominadas por abordagens de modelagem linear para aprendizado supervisionado, centradas em torno de algoritmos como Perceptrons, Máquina de Vetor de Suporte (*Support Vector Machines*, em inglês - SVM) e Regressão Logística, treinados em vetores de recursos de dimensão muito alta, mas esparsos (GOLDBERG, 2017).

Segundo (RUSSELL; NORVIG, 2022) estes são as 3 razões primárias para os computadores fazerem o processamento de linguagem natural (PLN):

- Para aprender. Os humanos registram uma grande quantidade de conhecimento usando linguagem natural. Só a Wikipedia possui 30 milhões de páginas de fatos como "Os saguis são pequenos primatas noturnos", enquanto quase não há fontes de fatos como

esses escritas em lógica formal. Se queremos que nossos sistemas saibam muito, é melhor que eles entendam a linguagem natural;

- Para avançar a compreensão científica das linguagens e do uso da linguagem, usando as ferramentas da IA em conjunto com a linguística, psicologia cognitiva e neurociência.

2.2.2 *Large Language Models*

Segundo (RUSSELL; NORVIG, 2022) um modelo de linguagem é definido como uma distribuição de probabilidade que descreve a verossimilhança de qualquer sequência de caracteres. Com um modelo de linguagem, é possível prever quais palavras provavelmente virão a seguir em um texto e, assim, sugerir compleções para um e-mail ou mensagem de texto. É possível calcular quais alterações em um texto o tornariam mais provável e, assim, sugerir correções ortográficas ou gramaticais.

Um grande modelo de linguagem (*large language model*, em inglês - LLM) é um modelo computacional avançado projetado para entender e gerar linguagem humana, utilizando uma grande quantidade de parâmetros e capacidades de aprendizado excepcionais. Esses modelos são capazes de prever sequências de palavras ou gerar texto novo a partir de uma entrada dada. Eles são amplamente utilizados em tarefas de processamento de linguagem natural, como análise de sentimentos e geração de texto (WU et al., 2024).

Existem LLMs muito conhecidas como o GPT e o Gemini, mas também modelos *open-source* que podem ser uma alternativa interessante para demonstrar que pode-se obter um bom desempenho sem fins comerciais. (CHEN et al., 2023). Entretanto, GPT e o Gemini não são de código aberto e seus acessos são controlados por empresas privadas logo a maioria de seus detalhes técnicos permanece desconhecida. Empresas que adotam esses modelos podem se preocupar com o alto custo das chamadas de APIs, interrupções no serviço, propriedade dos dados e questões de privacidade (CHEN et al., 2023).

Por essa razão, a comunidade de pesquisa tem se esforçado ativamente para manter LLMs de alto desempenho em código aberto. Modelos como o LLaMA e o Mixtral são exemplos notáveis de LLMs open-source que podem ser utilizados para alcançar resultados competitivos,

oferecendo transparência e controle aos pesquisadores e desenvolvedores. Esses modelos permitem um maior entendimento e customização, além de mitigar preocupações relacionadas à privacidade e ao custo operacional, tornando-se alternativas viáveis e robustas para diversas aplicações (CHEN et al., 2023).

Atualmente, existem vários *benchmarks* para avaliar o desempenho de LLMs. Como por exemplo o benchmark (*Multi-task Language Understanding*, em inglês - MMLU) que fornece uma estrutura abrangente para avaliar LLMs em diversos domínios, avaliando sua capacidade de raciocinar e generalizar além dos seus dados de treinamento, tendo um propósito mais geral. (MCDONALD; PAPADOPOULOS; BENNINGFIELD, 2024). A tabela abaixo contém o desempenho desses LLMs citados segundo as fontes oficiais (META, 2024b) (GOOGLE, 2024) (MISTRAL, 2024):

Tabela 2.1: Comparação dos modelos de LLM com o benchmark MMLU

Modelo	Nota
Gemini 1.5 Pro	75.8%
LLaMA3.1-8B	73.0%
Mixtral-7B	60.1%

Com os resultados do *benchmark* é interessante ver como modelo de código fechado como o Gemini se sobressai em relação àqueles *open-source*. As próximas subseções detalham sobre estes LLMs citados.

Gemini

Gemini é uma família de modelos multimodais desenvolvidos pela Google, treinados em dados de imagem, áudio, vídeo e texto para oferecer capacidades gerais fortes e desempenho avançado em compreensão e raciocínio. A primeira versão, Gemini 1.0, tem três tamanhos: Ultra, Pro e Nano, cada um adaptado a diferentes limitações computacionais e requisitos de aplicação. Após o pré-treinamento, os modelos passam por um pós-treinamento para melhorar a qualidade, capacidades específicas e garantir critérios de segurança. Foram produzidas duas variantes pós-treinadas: Gemini Apps, para IA conversacional, e Gemini API, para uma variedade de

produtos acessíveis via Google AI Studio e Cloud Vertex AI. A performance dos modelos é avaliada em diversos benchmarks de linguagem, codificação, raciocínio e tarefas multimodais (TEAM et al., 2024a).

Em maio de 2024, a Google disponibilizou o Gemini 1.5 no qual incorpora inovações em escalabilidade esparsa e densa, além de avanços significativos em treinamento, destilação e infraestrutura de serviço, elevando a eficiência, raciocínio, planejamento, multilinguismo, chamadas de função e desempenho em contextos longos. Os modelos Gemini 1.5 podem lidar com contextos extremamente longos, lembrando e raciocinando sobre informações detalhadas de até 10 milhões de tokens, possibilitando o processamento de entradas de modalidades mistas, incluindo coleções inteiras de documentos, horas de vídeo e quase cinco dias de áudio. A série Gemini 1.5 representa um salto geracional em desempenho e eficiência de treinamento, com o 1.5 Pro superando o 1.0 Pro e o 1.0 Ultra em diversos benchmarks, necessitando significativamente menos recursos de computação para treinamento (TEAM et al., 2024b).

LLaMA

LLaMA é uma coleção de modelos de linguagem base com parâmetros variando de 7 bilhões a 65 bilhões. Os modelos são treinados em trilhões de tokens e modelos *state-of-the-art* exclusivamente com conjuntos de dados disponíveis publicamente, sem recorrer a conjuntos de dados proprietários e inacessíveis (TOUVRON et al., 2023). Ainda segundo os autores, o LLaMA-13B supera o GPT-3 (175B) na maioria dos benchmarks, e o LLaMA-65B compete com os melhores modelos, Chinchilla-70B e PaLM-540B.

Mixtral

Mixtral é um modelo de linguagem com 7 bilhões de parâmetros, projetado para alto desempenho e eficiência. O Mixtral 7B supera o melhor modelo open-source de 13 bilhões de parâmetros (LLaMA 2) em todos os benchmarks avaliados, e supera o melhor modelo lançado de 34 bilhões (LLaMA 1) em tarefas de raciocínio, matemática e geração de código. O modelo utiliza a atenção agrupada por consulta (GQA) para inferência mais rápida, combinada com a atenção de

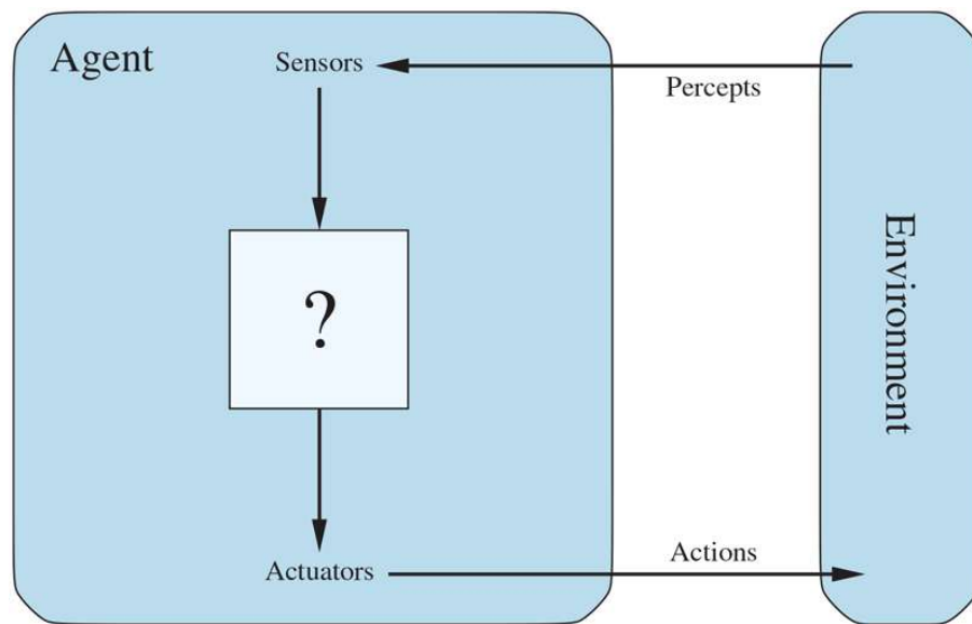
janela deslizante (SWA) para lidar efetivamente com sequências de tamanho arbitrário com um custo de inferência reduzido (JIANG et al., 2023). Segundo os autores também há um modelo ajustado para seguir instruções, o Mixtral 7B – Instruct, que supera o modelo de chat LLaMA 2 (13B) em benchmarks humanos e automatizados.

2.2.3 Agentes Inteligentes

Um agente de Inteligência Artificial é definido como um programa que utiliza técnicas de IA para realizar tarefas que normalmente requerem inteligência humana. Esses agentes podem assumir várias formas, desde chatbots simples até sistemas autônomos complexos que interagem com seu ambiente e tomam decisões em tempo real. Eles podem ser treinados usando uma variedade de técnicas de aprendizado de máquina, incluindo aprendizagem supervisionada, não supervisionada e por reforço, e podem ser programados para realizar tarefas específicas ou aprender com suas experiências para melhorar seu desempenho ao longo do tempo (RUAN et al., 2023). Os agentes também podem ser capazes de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por intermédio de atuadores. (RUSSELL; NORVIG, 2022). Por exemplo, um agente humano possui olhos, ouvidos e outros órgãos que funcionam como sensores, e mãos, pernas, cordas vocais e assim por diante como atuadores. Um agente robótico, por outro lado, pode ter câmeras e sensores infravermelhos como sensores e vários motores como atuadores.

2.3 *Application Programming Interface*

Uma interface de programação de aplicação (*Application Programming Interface*, em inglês - API) é a interface que um programa de software apresenta a outros programas, a humanos e, no caso das APIs web, ao mundo via internet. O design de uma API revela muito sobre o programa por trás dela — modelo de negócio, funcionalidades do produto, e eventuais *bugs*. APIs são os blocos de construção que permitem a interoperabilidade das principais plataformas de negócios na web (JIN; SAHIN; SHEVAT, 2018). Uma API define a maneira como os sistemas de com-



Agents interact with environments through sensors and actuators.

Figura 2.4: Esquema de como os agentes interagem com o ambiente (RUSSELL; NORVIG, 2022)

putador interagem (JJ; JON, 2021). As APIs surgiram da necessidade de trocar informações com provedores de dados que estão equipados para resolver problemas específicos, de modo que as pessoas em outras empresas não precisem gastar tempo resolvendo esses problemas elas mesmas. Ao longo dos anos, múltiplos paradigmas de API surgiram. REST, RPC, GraphQL, WebHooks e WebSockets são alguns dos padrões mais populares (JIN; SAHIN; SHEVAT, 2018).

2.3.1 HTTP

O Protocolo de Transferência de Hipertexto (*HyperText Transfer Protocol*, em inglês - HTTP) — o protocolo da camada de aplicação da Web, está no coração da Web. O HTTP é executado em dois programas: um cliente e outro servidor. Os dois, executados em sistemas finais diferentes, conversam entre si por meio da troca de mensagens HTTP. O HTTP define a estrutura dessas mensagens e o modo como o cliente e o servidor as trocam (KUROSE, 2013). As imagens abaixo mostram como são as mensagens de requisição (feita pelo lado do cliente) e de resposta (feita pelo lado do servidor):

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr
```

Figura 2.5: Mensagem genérica de requisição HTTP (KUROSE, 2013)

```
HTTP/1.1 200 OK
Connection: close
Date: Tue, 09 Aug 2011 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Tue, 09 Aug 2011 15:11:03 GMT
Content-Length: 6821
Content-Type: text/html
```

Figura 2.6: Mensagem genérica de resposta HTTP (KUROSE, 2013)

Métodos

O protocolo possui diversos métodos, sendo os principais GET, POST, PUT e DELETE. Segundo (GOURLEY et al., 2002):

- GET: É o método mais comum. Geralmente é usado para solicitar ao servidor uma requisição.



Figura 2.7: Exemplo de GET (GOURLEY et al., 2002)

- POST: Foi projetado para enviar dados de entrada ao servidor. Na prática é usado com frequência em formulários.

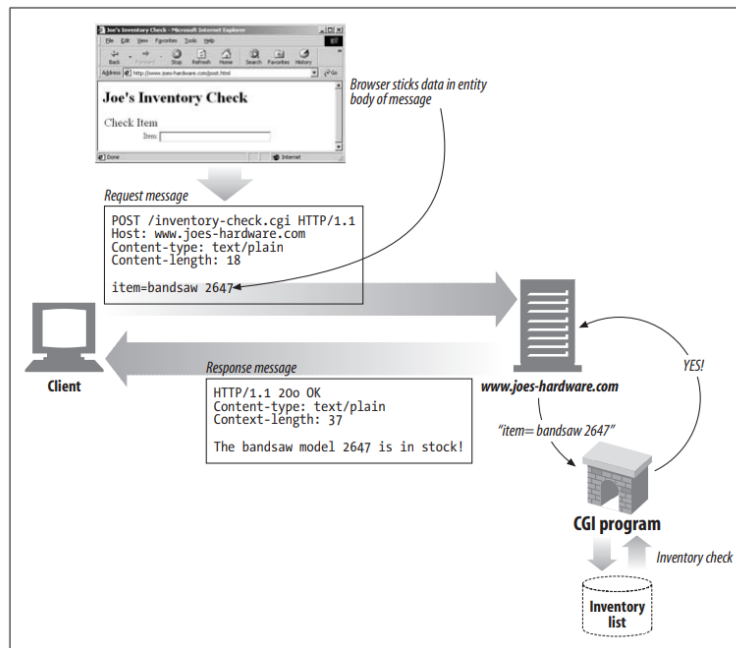


Figura 2.8: Exemplo de POST
(GOURLEY et al., 2002)

- PUT: Escreve documentos e envia ao servidor agindo de forma inversa quando comparado ao GET. A semântica deste método é atualizar documentos sem a necessidade de apagar e criar novos.

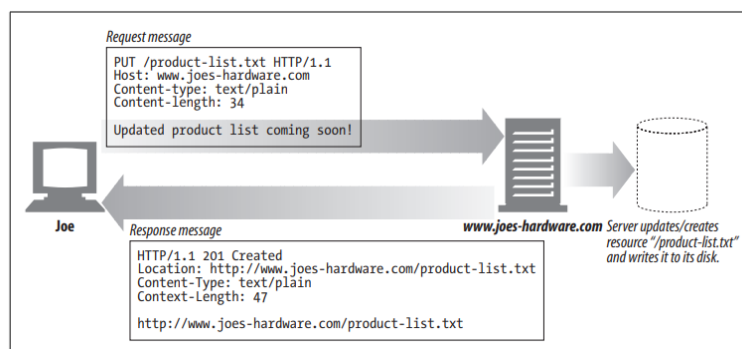


Figura 2.9: Exemplo de PUT
(GOURLEY et al., 2002)

- DELETE: Este método solicita ao servidor para deletar os recursos especificados pela URL de requisição.

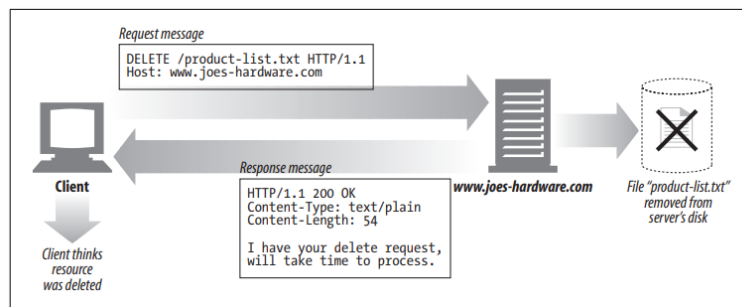


Figura 2.10: Exemplo de DELETE
(GOURLEY et al., 2002)

Códigos de Status

Os códigos de status HTTP são classificados em cinco categorias amplas. Os códigos de status fornecem uma maneira fácil para os clientes entenderem os resultados de suas transações (GOURLEY et al., 2002).

- 1XX: Representa status de informação. É usado quando uma parte inicial da requisição foi recebida e o cliente deve continuar o processo;
- 2XX: Representa status de sucesso. Indica que a requisição do cliente foi recebida, compreendida e aceita pelo servidor;
- 3XX: Representa status de redirecionamento. Informa que o recurso que o cliente está tentando acessar foi movido para uma nova localização, ou que uma mensagem alternativa deve ser fornecida ao invés do conteúdo original;
- 4XX: Representa status de erro do cliente. Ocorre quando o servidor não pode processar a requisição devido a um erro cometido pelo cliente, como uma requisição mal formatada ou acesso a uma página inexistente.
- 5XX: Representa status de erro do servidor. Indica que o servidor encontrou um erro ao tentar processar a requisição válida enviada pelo cliente.

2.3.2 REST

A transferência de estado de representação (Representational State Transfer, em inglês - REST) é a escolha mais popular para desenvolvimento de APIs atualmente. O paradigma REST é utilizado por provedores como Google, Stripe, Twitter e GitHub. REST é totalmente sobre recursos. Um recurso é uma entidade que pode ser identificada, nomeada, endereçada ou manipulada na web. APIs REST expõem dados como recursos e utilizam métodos HTTP padrão para representar transações de *Create*, *Read*, *Update* e *Delete* (CRUD) desses recursos (JIN; SAHIN; SHEVAT, 2018). A tabela abaixo mostra como o CRUD se relaciona com o HTTP:

Tabela 2.2: Relação do CRUD com os métodos HTTP

Referência	Método HTTP
<i>Create</i>	POST
<i>Read</i>	GET
<i>Update</i>	PUT
<i>Delete</i>	DELETE

As APIs REST seguem algumas regras padrão:

- Recursos fazem parte das URLs;
- Uso de métodos HTTP como visto na seção 2.3.1;
- Os códigos de status padrão de resposta HTTP são retornados pelo servidor indicando sucesso ou falha, esses status foram detalhados na seção anterior
- Retornam as resposta em formato JSON ou XML;

2.4 Conclusão

Neste capítulo foram apresentados diversos conceitos essenciais para alcançar os objetivos propostos, incluindo Sistemas de Recomendação, Inteligência Artificial, APIs e Métricas de Avaliação, todos de grande importância para o desenvolvimento deste trabalho. O próximo capítulo, o capítulo 3, abordará trabalhos que utilizam LLMs no contexto de sistemas de recomendação.

Capítulo 3

Trabalhos Relacionados

3.1 *LLM Based Generation of Item-Description for Recommendation System*

O artigo "*LLM Based Generation of Item-Description for Recommendation System*" (ACHARYA; SINGH; ONOE, 2023) explora o uso de LLMs para gerar descrições detalhadas de itens, como filmes e livros, com o objetivo de avaliar a eficácia desses modelos em comparação com as descrições obtidas por web scraping. Os pesquisadores utilizaram dois conjuntos de dados principais: o MovieLens 1M, um benchmark popular para avaliação de sistemas de recomendação que contém títulos de filmes, e o Goodreads, que inclui nomes de livros e suas descrições, obtidas via web scraping.

O principal objetivo do estudo foi automatizar a geração de descrições de itens para sistemas de recomendação, economizando tempo e reduzindo a dependência de métodos manuais. Para isso, foi utilizado o modelo Alpaca-Lora, ajustado com instruções específicas para gerar descrições baseadas em prompts contendo títulos de filmes ou livros. A metodologia seguiu o método proposto no paper GRU4RECBE, onde as descrições geradas foram utilizadas para treinar um sistema de recomendação. O treinamento envolveu o uso de embeddings de descrições de itens gerados pelo modelo BERT, que foram combinados com IDs de filmes e processados por uma camada GRU para prever o próximo item.

Os resultados foram avaliados utilizando métricas comuns em sistemas de recomendação, como Top Hits, Mean Reciprocal Rank (MRR) e Normalized Discount Cumulative Gain (NDCG). As descrições geradas pelos LLMs mostraram-se comparáveis às obtidas via web scraping, especialmente no caso do MovieLens. A combinação de descrições IMDB com LLMs (70 IMDB + 30 LLM) apresentou resultados muito próximos aos obtidos com apenas descrições do IMDB. Para o conjunto de dados do Goodreads, as descrições geradas pelos LLMs também mostraram-se eficazes, embora com algumas variações em comparação com as descrições obtidas manualmente.

A descrições de itens usando LLMs é uma alternativa viável e eficiente aos métodos tradicionais de web scraping, oferecendo uma maneira automatizada e menos tendenciosa de criar conteúdo para sistemas de recomendação. No entanto, a precisão das descrições geradas pode ser afetada pela capacidade do modelo LLM de lidar com itens menos populares ou antigos, sugerindo que modelos maiores e melhor ajustados podem proporcionar resultados ainda melhores no futuro.

3.2 TPTU: Large Language Model-based AI Agents for Task Planning and Tool Usage

O artigo *TPTU: Large Language Model-based AI Agents for Task Planning and Tool Usage* (RUAN et al., 2023) aborda o uso de LLMs para resolver tarefas complexas que requerem planejamento e uso de ferramentas externas. O principal objetivo é avaliar a capacidade dos LLMs de realizar planejamento de tarefas e uso de ferramentas (TPTU). Para isso, os autores propõem um framework estruturado para agentes de IA baseados em LLMs e exploram duas abordagens de agentes: o agente de passo único e o agente sequencial.

Para a avaliação, dois conjuntos de dados principais foram utilizados. O primeiro conjunto envolve consultas SQL simples, que contém operações básicas como SELECT, FROM, WHERE e GROUP BY, utilizadas para recuperar, filtrar, agrupar e classificar dados de uma única tabela. Exemplos incluem cálculos de média de idades e contagem de indivíduos por sexo em tabelas

como "Person" e "School". O segundo conjunto inclui consultas SQL complexas e aninhadas, que envolvem subconsultas dentro de uma consulta maior, abrangendo múltiplas operações ou cálculos através de várias tabelas, como "GoldenMelodyAwards" e "AwardNominees".

As arquiteturas desenvolvidas para os agentes foram duas. O agente de passo único realiza todo o processo de inferência de uma vez, sendo ideal para tarefas que não necessitam de múltiplas etapas de planejamento. Já o agente sequencial executa a inferência de maneira sequencial, apropriado para tarefas que exigem etapas interdependentes e planejamento contínuo. Ambas as arquiteturas foram implementadas e testadas com diferentes LLMs para avaliar suas capacidades de planejamento e uso de ferramentas.

Os resultados empíricos indicam que os agentes baseados em LLMs têm um potencial significativo, mas também enfrentam desafios. Os agentes demonstraram a capacidade de usar ferramentas externas de maneira eficaz, embora a complexidade das tarefas tenha influenciado diretamente o desempenho. No que tange ao planejamento de tarefas, o agente sequencial mostrou-se mais eficaz em tarefas que requerem múltiplas etapas e planejamento contínuo, enquanto o agente de passo único foi adequado para tarefas mais simples e diretas. No entanto, o estudo identificou áreas que necessitam de mais investigação, como a integração de novas ferramentas e a adaptação dos LLMs para lidar com dados e problemas mais atualizados e complexos.

O artigo destaca que os LLMs têm um grande potencial para serem usados como agentes de IA em aplicações reais, especialmente em tarefas que envolvem planejamento e uso de ferramentas. No entanto, há necessidade de melhorias contínuas para superar os desafios atuais e aprimorar a eficiência e a versatilidade desses modelos em cenários práticos.

3.3 Music Personalized Recommendation System Based on Hybrid Filtration

O artigo *Music Personalized Recommendation System Based on Hybrid Filtration* (WU, 2019) aborda sobre a recomendação híbrida de músicas usando as filtragens colaborativa e baseada

em conteúdo utilizando um sistema de recomendação musical personalizado. A motivação por trás do desenvolvimento deste sistema é a necessidade de melhorar a experiência do usuário em plataformas de streaming de música, oferecendo sugestões que correspondam melhor aos seus gostos pessoais.

Entre os pontos fortes do sistema proposto, destaca-se a utilização da filtragem híbrida, que permite superar as limitações presentes quando se utiliza apenas uma técnica de filtragem. A filtragem colaborativa, por exemplo, pode sofrer com o problema de dados esparsos e o *cold start*, enquanto a filtragem baseada em conteúdo pode não captar adequadamente as preferências implícitas dos usuários. Ao combinar ambas as abordagens, o sistema consegue aproveitar as vantagens de cada técnica, proporcionando recomendações mais precisas e diversificadas.

Por outro lado, um dos pontos fracos identificados no artigo é a complexidade computacional envolvida na implementação do sistema híbrido. A integração de múltiplas técnicas de filtragem exige maior poder de processamento e pode aumentar o tempo de resposta do sistema. Além disso, a eficácia do sistema depende da qualidade e da quantidade de dados disponíveis sobre as preferências dos usuários e as características das músicas, o que pode variar significativamente entre diferentes plataformas de streaming.

O artigo realiza a coleta de dados sobre as interações dos usuários com a plataforma de música, como histórico de escuta e avaliações. Esses dados são então processados e analisados para identificar padrões e preferências. O sistema aplica a filtragem colaborativa para encontrar usuários com gostos musicais similares e a filtragem baseada em conteúdo para analisar as características das músicas e fazer recomendações personalizadas.

3.4 Uncovering ChatGPT's Capabilities in Recommender Systems

O artigo *Uncovering ChatGPT's Capabilities in Recommender Systems* (DAI et al., 2023) aborda sobre as capacidades do ChatGPT como um sistema de recomendação, abordando três principais técnicas de ranking: *point-wise*, *pair-wise* e *list-wise*. Primeiramente, os autores des-

tacam o potencial dos LLMs, como o GPT3.5, para superar problemas de esparsidade de dados frequentemente encontrados nos sistemas de recomendação tradicionais. Também oferece uma visão geral sobre o uso de LLMs em diversas tarefas e levanta a questão sobre a eficácia desses modelos como sistemas de recomendação com poucas ou nenhuma amostra de treinamento prévio.

Os autores também detalham a arquitetura do GPT e como pode ser adaptado para tarefas de recomendação usando *prompts* específicos. A forma como é exemplificado lembra muito a arquitetura de agentes. As técnicas de *prompt tuning* são fundamentais para alinhar as capacidades do LLMs com as necessidades de um sistema de recomendação. Isso inclui a criação de descrições de tarefas específicas, exemplos de demonstração e consultas de entrada, que ajudam o modelo a entender o domínio específico e a realizar previsões precisas. Esse processo visa explorar as habilidades do GPT em diferentes cenários de recomendação.

Os pontos fortes do ChatGPT como sistema de recomendação incluem sua habilidade em realizar rankings *list-wise* e *pair-wise* de forma eficaz, superando até mesmo modelos tradicionais quando há dados limitados para treinamento. Além disso, o modelo mostra um bom entendimento da similaridade entre itens, o que é crucial para fornecer recomendações. No entanto, o desempenho do GPT é menos robusto em tarefas de ranking *point-wise*, o que indica uma área de melhoria necessária para otimizar todas as capacidades de recomendação do modelo.

Os autores realizaram experimentos abrangentes para avaliar o desempenho do GPT em comparação com métodos tradicionais de filtragem colaborativa. Eles utilizam conjuntos de dados de diferentes domínios para testar as capacidades de ranking do modelo, destacando tanto os pontos fortes quanto as limitações do GPT. Os resultados mostram que, embora o modelo seja promissor em muitos aspectos, há considerações de custo associadas ao uso de LLMs que precisam ser equilibradas com os benefícios de desempenho obtidos.

O artigo sugere que o ChatGPT tem um grande potencial para aprimorar sistemas de recomendação, especialmente em cenários onde os dados são escassos. Os autores recomendam a utilização de técnicas de ranking *list-wise* para obter os melhores resultados com LLMs e destacam a necessidade de mais pesquisas para otimizar as capacidades *point-wise* do ChatGPT.

Este estudo preliminar fornece novas perspectivas sobre a aplicação de LLMs em sistemas de recomendação e abre caminho para futuras investigações sobre como melhor aproveitar essa tecnologia.

3.5 *Recommender Systems in the Era of Large Language Models*

O artigo *Recommender Systems in the Era of Large Language Models* (ZHAO et al., 2024) aborda a integração de sistemas de recomendação com LLMs, analisando três paradigmas principais: pré-treinamento, ajuste fino (*fine-tuning*, em inglês) e *prompting*. Ainda analisa o uso de modelos de Deep Learning em sistemas de recomendação, destacando suas limitações em capturar conhecimento sobre os usuários através de textos. Também destaca os avanços significativos dos LLMs na personalização das recomendações ao entender melhor o contexto e as preferências dos usuários. Métodos como *LMRec* utilizam técnicas para reduzir alucinações assim mitigando impactos negativos.

A eficiência computacional é um ponto forte, alcançada através de técnicas de (*parameter-efficient Fine-tuning*, em inglês - PEFT), como LoRA. Essas técnicas permitem adaptar LLMs para tarefas de recomendação com menor custo computacional, viabilizando sua execução em GPUs de consumo único. O uso de informações textuais adicionais, como descrições de itens, é uma vantagem, melhorando a representação de usuários e itens e resultando em recomendações mais precisas.

O *fine-tuning* de modelos completos exige grandes recursos computacionais, tornando impraticável para uma única GPU manipular modelos com mais de 10 bilhões de parâmetros. Além disso, o *fine-tuning* direto de LLMs pode introduzir alucinações indesejadas, prejudicando determinados grupos ou indivíduos. A ênfase excessiva em recursos textuais pode levar à negligência de outras formas importantes de dados.

O artigo oferece uma visão abrangente das estratégias atuais e desafios na aplicação de LLMs em sistemas de recomendação. Destaca avanços significativos, mas também aponta áreas que

requerem melhorias contínuas, como a mitigação de alucinações e a eficiência computacional. A integração de LLMs nos sistemas de recomendação representa um passo importante para personalizações mais precisas e contextualmente relevantes, embora ainda existam desafios a serem superados.

3.6 Análise Comparativa

No primeiro trabalho (3.1), LLMs são utilizadas no processo de recomendação para gerar descrições de filmes, que são então comparadas com descrições feitas por humanos. Em contraste, esta monografia utiliza um LLM para recomendar diretamente, sem realizar comparações com outros sistemas ou com recomendações feitas por humanos.

O segundo trabalho (3.2) é o mais semelhante a esta monografia em termos de arquitetura proposta, ao utilizar agentes inteligentes e LLMs. No entanto, nesse trabalho, os agentes são usados para planejar tarefas e utilizar ferramentas, e não para realizar recomendações. Portanto, esta monografia se diferencia ao adotar uma arquitetura similar, mas com o foco específico em recomendações.

O terceiro trabalho (3.3) aborda o desenvolvimento de um sistema de recomendação híbrido de músicas, utilizando filtragem colaborativa e baseada em conteúdo, seguindo uma abordagem tradicional. Em contraste, esta monografia apresenta uma abordagem inovadora ao empregar LLMs para o mesmo objetivo, mitigando o problema da esparsidade de dados.

O quarto trabalho (3.4) é mais próximo desta monografia, pois também utiliza um LLM, especificamente o GPT, em um sistema de recomendação. A descrição dos autores é semelhante a uma arquitetura que usa agentes inteligentes, aproximando-se mais do que esta monografia propõe.

O quinto trabalho (3.5) oferece uma visão abrangente das estratégias e desafios ao utilizar LLMs em sistemas de recomendação. Esta monografia adota estratégias similares, como não utilizar o *fine-tuning*, para obter recomendações precisas e minimizar o risco de alucinações.

Capítulo 4

Proposta de Arquitetura e Implementação

Neste capítulo será a proposta de arquitetura para o sistema, assim como o passo a passo da implementação.

4.1 Arquitetura

A arquitetura do sistema foi modelada seguindo o modelo C4 elaborada por (BROWN, 2024), pois facilita a compreensão, dada a sua sintaxe simples e flexível (INGELMO et al., 2020). O modelo C4 possui 4 níveis de abstração: sistema, contêiner, componente e código. Para este trabalho será adotado até o nível 3, pois este nível já é bem detalhado.

4.1.1 Nível 1: Sistema

O primeiro nível mostra o sistema no nível mais externo. É importante destacar os sistemas externos a serem utilizados:

- *Chartsfy*: Provê os dados de músicas necessários para o sistema de recomendação;
- *GoogleGeminiAPI*: Permite o acesso do sistema e conseqüentemente dos agentes, ao modelo Gemini por meio de uma chave de API;
- *GroqAPI*: Permite acesso a outros modelos, como LLaMA e Mixtral, também por meio de uma chave de API;

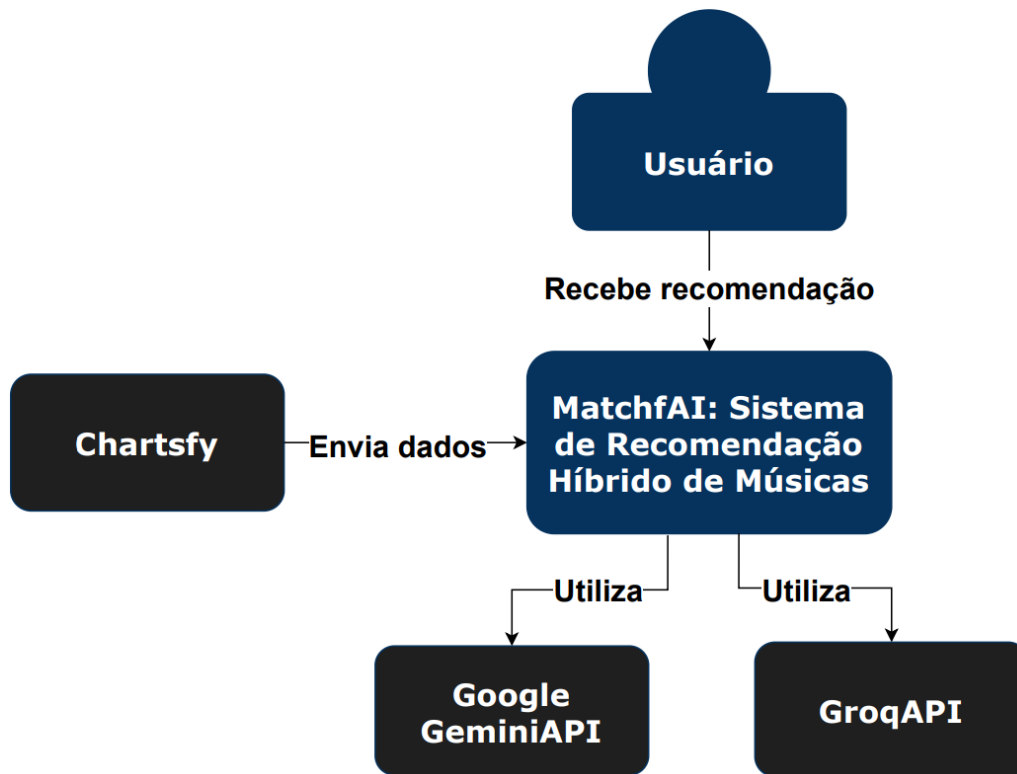


Figura 4.1: Nível 1: Sistema

4.1.2 Nível 2: Contêiner

O segundo nível já mostra um nível de detalhes maior quando comparado ao nível anterior. É possível observar a presença de contêiners, listados a seguir:

- *MatchfAI - BackEnd*: API REST encarregada de gerenciar as interações entre a interface gráfica e os dados de recomendação, fornecendo as informações necessárias para o funcionamento do sistema.
- *MatchfAI - API*: API REST responsável por executar todo o processo de recomendação, retornando os resultados ao *MatchfAI - BackEnd* e gerenciando a rotina de atualização dos dados.
- Banco de Dados: Um banco de dados não relacional, neste caso uma instância do *MongoDB* (MONGODB, 2024b) onde os dados são armazenados;
- *MatchfAI - FrontEnd*: Interface gráfica da aplicação responsável pela interação do usuário com as recomendações;

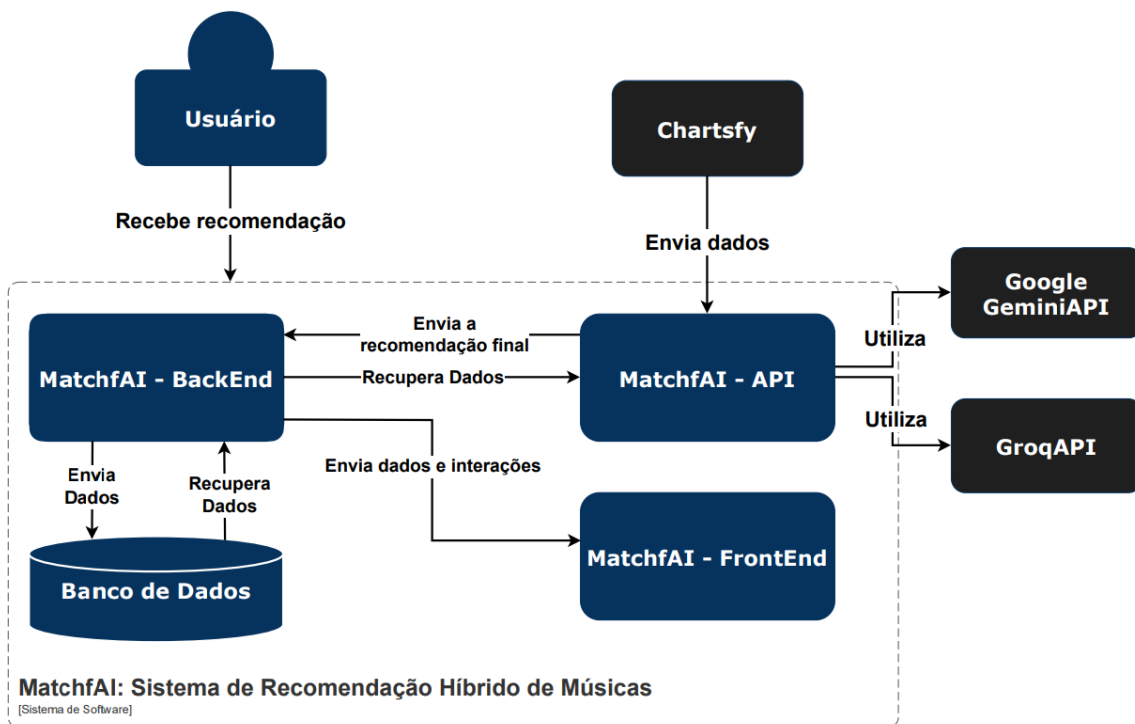


Figura 4.2: Nível 2: Contêiner

4.1.3 Nível 3: Componentes

Esta seção mostra apenas a arquitetura dos agentes, pois esta parte é o foco principal.

Foram projetados cerca de 11 agentes, sendo que 7 deles são apenas para o suporte dos agentes recomendadores:

- Leitor de Catálogo de Músicas: Retorna o catálogo de músicas;
- Leitor de Histórico de Usuário: Retorna o histórico de 1 usuário;
- Leitor de Músicas Não Curtidas do Usuário: Retorna as músicas marcadas como "não gostei" pelo usuário alvo;
- Analista de Gêneros Musicais: Recebe o histórico ou músicas curtidas de um usuário e retorna os 5 gêneros musicais que identificam aquele usuário;
- Leitor de Músicas Curtidas do Usuário: Retorna as músicas marcadas como "gostei" pelo usuário alvo;

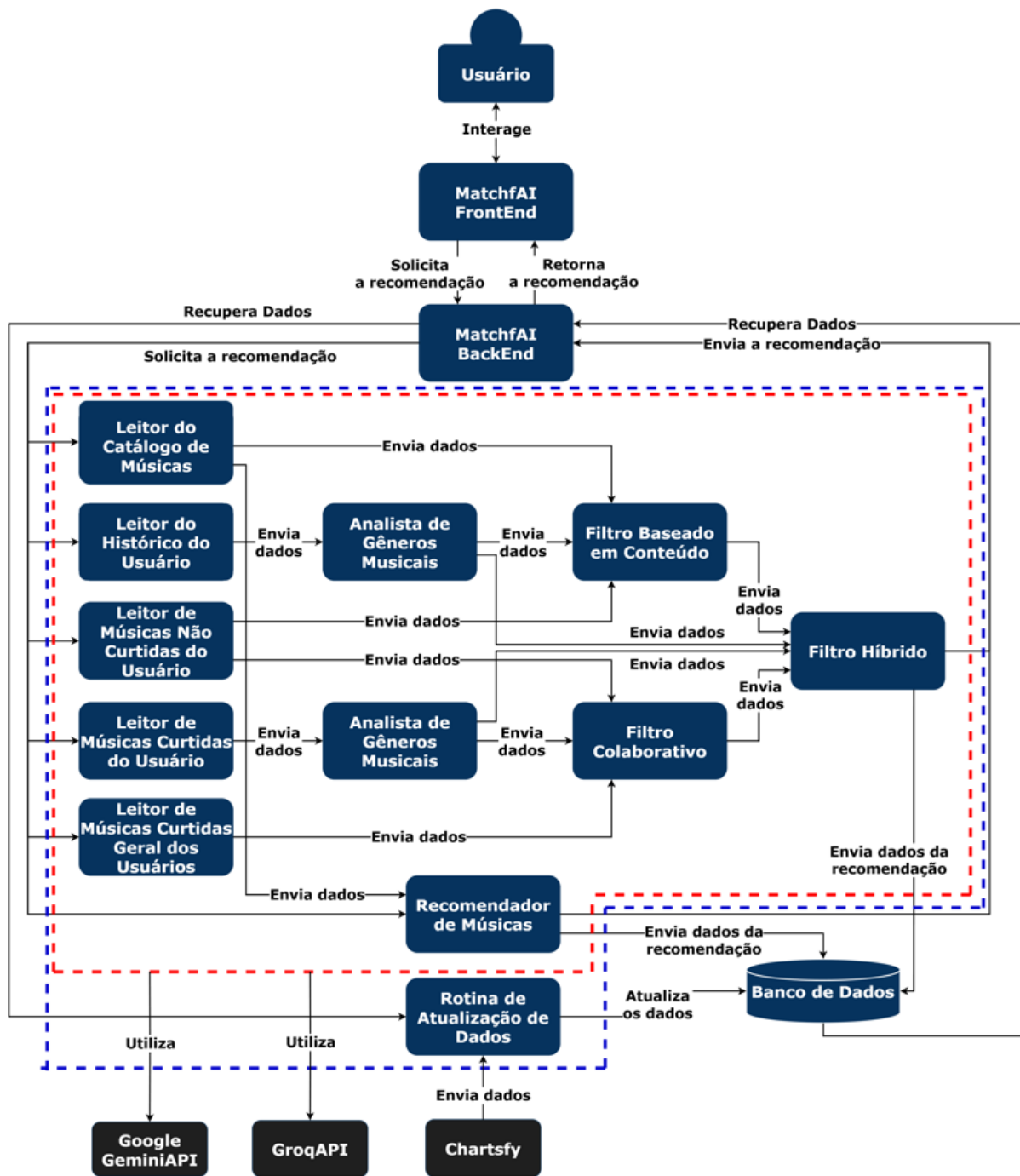


Figura 4.3: Nível 3: Componentes

- Leitor de Músicas Curtidas Geral dos Usuários: Retorna as músicas marcadas como "gostei" por todos os usuários, exceto o usuário alvo;

Os 3 agentes principais são listados a seguir:

- Filtro Baseado em Conteúdo: Ao receber os dados de seus agentes de suporte, retorna

uma lista de músicas recomendadas. Este agente recomenda baseado no gênero de cada música;

- Filtro Colaborativo: Ao receber os dados de seus agentes de suporte, retorna uma lista de músicas recomendadas. Este agente recomenda baseado em usuários com gostos musicais parecidos;
- Filtro Híbrido: Combina as duas recomendações anteriores e retorna a recomendação final;

O contêiner *MatchfAI - API* é representado pelo pontilhado azul na Figura 4.3, enquanto o pontilhado em vermelho corresponde aos componentes que englobam ferramentas, agentes e tarefas. Esses componentes, por sua vez, se conectam aos LLMs por meio de APIs, como Google Gemini e Groq.

O agente *Recomendador de Músicas* é projetado especificamente para atender novos usuários, que ainda não possuem um histórico na aplicação. É importante salientar que estes componentes são formados por agentes, ferramenta e tarefas. Na próxima subseção, um nível maior de detalhamento será apresentado e a próxima seção descreverá com mais detalhes sobre os agentes e os *prompts* utilizados.

Representação do Componente

Os componentes vistos na seção anterior podem ser representados ainda mais detalhadamente das seguintes formas:

Há agentes que dependem de uma ferramenta para obter os dados necessários à execução de suas tarefas. Por outro lado, alguns agentes não necessitam de ferramentas, pois já possuem acesso direto aos dados requeridos.

Nas seções 4.3.1, 4.3.2 e 4.3.3, serão apresentados detalhes sobre a estrutura dos agentes, ferramentas e tarefas. Já na seção 4.4, serão descritos os prompts utilizados e todos os elementos relacionados aos três mencionados anteriormente.

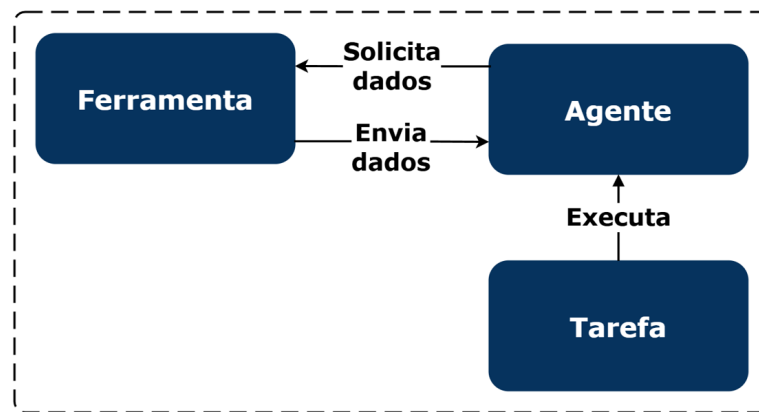


Figura 4.4: Forma 1 - Ferramenta, Agente e Tarefa

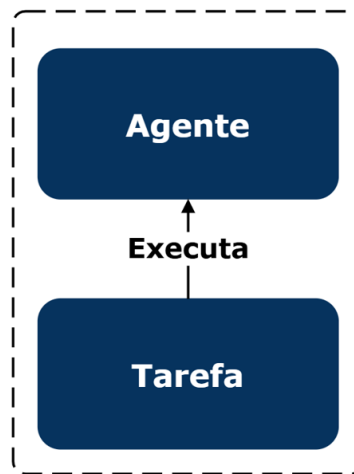


Figura 4.5: Forma 2 - Agente e Tarefa

4.2 *Chartsfy*

Conforme mencionado na seção 4.1.1, *Chartsfy* é uma aplicação desenvolvida pelo autor deste trabalho com o objetivo de registrar as músicas ouvidas pelos usuários participantes. Por meio da aplicação, os usuários conectam suas contas do *Spotify* (SPOTIFY, 2024a), permitindo que suas músicas reproduzidas sejam armazenadas em um banco de dados. O sistema desenvolvido para este trabalho utiliza os dados do *Chartsfy* para poder gerar recomendações.

4.2.1 Cadastro de usuários

1. O usuário acessa o site `https://chartsfy-web-web.app`;



Figura 4.6: Página inicial do *Chartsfy*

2. Conecta a conta do *Spotify*;

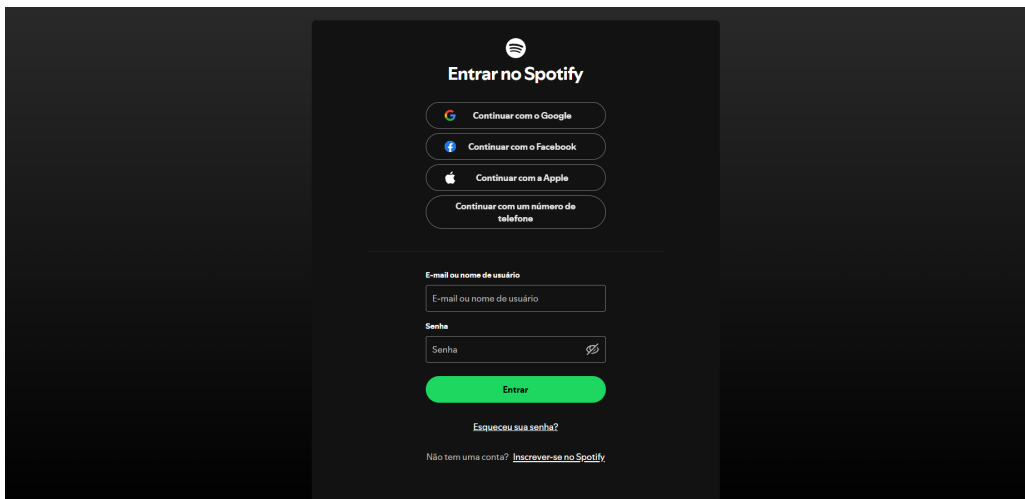


Figura 4.7: Autenticação com o *Spotify*

3. Aceita o formulário de autorização.



Figura 4.8: Formulário de autorização da aplicação (SPOTIFY, 2024a)

Seguindo os passos acima, o usuário é devidamente cadastrado e as músicas ouvidas estarão sendo armazenadas pela API do *Chartsfy*. Vale ressaltar que todos os usuários, ao se cadastrarem na aplicação, concordaram em disponibilizar esses dados por meio do formulário apresentado na figura 4.8.

4.2.2 Fluxo do Funcionamento

Após o cadastro do usuário, o fluxo de funcionamento da aplicação ocorre conforme descrito a seguir:

1. A aplicação realiza a manutenção e atualização dos tokens de autenticação, garantindo acesso contínuo aos dados do usuário;
2. Uma requisição é enviada à API do Spotify para obter a última música ouvida pelo usuário. Esse processo é executado em intervalos de 1 minuto;

3. A música recuperada é então armazenada no banco de dados da aplicação.

A Figura 4.9 ilustra esse fluxo:

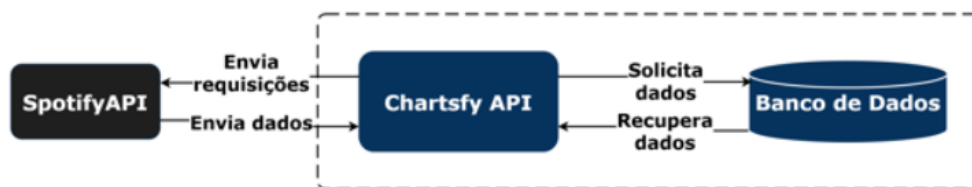


Figura 4.9: Esquemático da aplicação *Chartsfy*

4.2.3 Base de Dados

A base de dados contém aproximadamente 161462 músicas escutadas pelos 19 usuários ativos atualmente. Esta base de dados fornece todos os dados necessários para o sistema de recomendação.

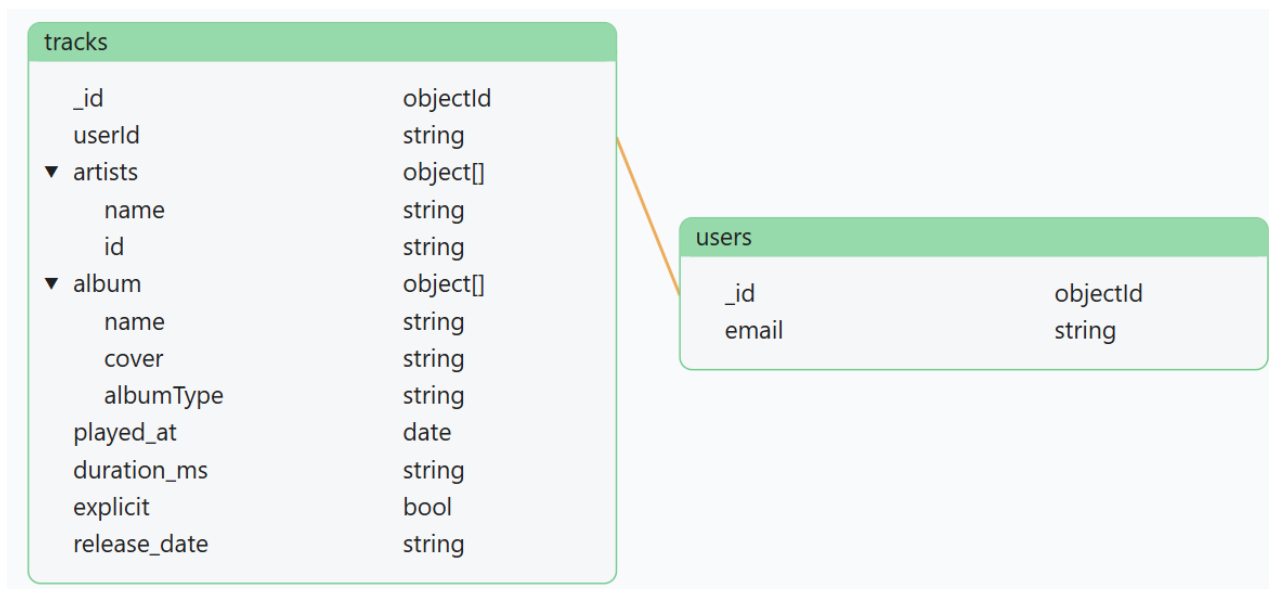


Figura 4.10: *Collections* do banco de dados do *Chartsfy*

A Figura 4.9 apresenta o esquema das coleções que armazenam os dados necessários. Nas subseções a seguir, cada item será detalhado.

tracks

- *_id*: Identificador único do objeto;

- *userId*: Identificador do usuário, proveniente da *collection* de *users*;
- *artists*: Lista que contém o nome do artista e seu identificador no *Spotify*, permitindo a consulta de informações como gênero musical e popularidade;
- *album*: Lista com o nome do álbum, URL da capa (*cover*) e o tipo de álbum (*album*, *extended play - EP* ou compilações);
- *played_at*: Data e horário em que o usuário ouviu a música;
- *duration_ms*: Duração da música em milissegundos;
- *explicit*: Indica se a música contém conteúdo explícito;
- *release_date*: Data de lançamento da música.

users

- *_id*: Identificador único do objeto;
- *userId*: Identificador do usuário.;

4.3 MatchfAI - API

Para transformar a arquitetura de agentes em um serviço consumível, foi desenvolvida uma API seguindo o padrão REST utilizando o Django, um framework web em Python de alto nível que promove o desenvolvimento rápido e um design limpo e pragmático (DJANGO, 2024). A API foi implementada com a linguagem de programação Python e inclui os seguintes *endpoints*:

- POST /recommend
 - Este é o endpoint principal da aplicação, responsável por todo o processo de geração de recomendações.
- POST /newrec

- Utilidade: Gera recomendações personalizadas com base nas preferências do usuário. É utilizado tanto para novos usuários quanto para a avaliação da métrica de revocação.

- GET /routine

- Utilidade: Executa os passos necessários para a rotina de atualização de dados.

Vale ressaltar que esta API adotou a contêinerização utilizando o Docker. A contêinerização é uma tecnologia que combina a aplicação, as dependências relacionadas e as bibliotecas do sistema organizadas na forma de um contêiner (POTDAR et al., 2020). Também com o propósito de rodar em *cloud*, em uma instância da AWS (*Amazon Web Services*, em inglês) para que garantir a disponibilidade do serviço. Além disso, a API faz uso das bibliotecas CrewAI (CREWAI, 2024a) e LangChain (LANGCHAIN, 2024), que possuem dependências específicas. Para gerenciar essas dependências de forma eficiente, foi criado um ambiente virtual utilizando o Anaconda (ANACONDA, 2024). O uso do Anaconda foi crucial para fornecer um ambiente controlado e garantir a compatibilidade das versões das bibliotecas utilizadas. Algumas dependências do CrewAI apresentam problemas de funcionamento fora de um ambiente virtual devidamente configurado, reforçando a necessidade dessa abordagem.

A instância utilizada foi a t3-small, com 2 núcleos de CPU, 2 GB de memória RAM e 8 GB de armazenamento (AWS, 2023).

4.3.1 Agentes

Os agentes foram construídos usando o *framework* CrewAI e linguagem de programação Python, pois fornece métodos para a implementação da arquitetura proposta e são feitos para o Python. Os agentes utilizam os seguintes atributos descritos abaixo, conforme a documentação (CREWAI, 2024b):

- *Role*: Define a função do agente;
- *Goal*: Objetivo individual que o agente deve buscar atingir;

- *Backstory*: Define o contexto para o agente;
- *Verbose*: Provê *logs* de execução detalhados;
- *Tools*: Define qual ferramenta o agente vai usar para trabalhar;
- *Allow-delegation*: Um agente pode delegar tarefas ou questões para outros agentes
- *LLM*: Define qual LLM vai ser usado pelo agente.

4.3.2 *Tools*

Ferramentas (*tools*, em inglês) são métodos customizáveis fornecidos pelo *Langchain* (LANGCHAIN, 2024). São muito similares as *tasks*, porém permitem que uma função feita em Python possa ser usada por um agente. Neste trabalho, foram criadas diversas funções utilizando esse método para fazer requisições ao backend e retornar os dados em *dataframes*.

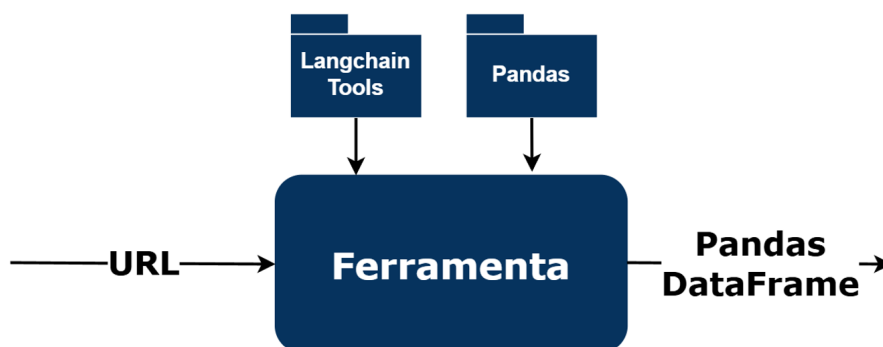


Figura 4.11: Representação e fluxo de uma ferramenta

A entrada consiste em um Localizador de Recursos Universal (Uniform Resource Locator, em inglês - URL), geralmente uma URL que representa o endpoint do backend da aplicação responsável por fornecer os dados requisitados. Conforme mencionado anteriormente, utilizando o *LangChain Tools* (LANGCHAIN, 2024) e o *Pandas* (PANDAS, 2024), a ferramenta processa a solicitação e retorna um dataframe contendo os dados solicitados. Também é necessário informar uma descrição do que a ferramenta faz para que o agente determine o seu uso (LANGCHAIN, 2024).

4.3.3 Tarefas

Tarefas (*tasks*, em inglês) são atribuições específicas realizadas por agentes. Elas fornecem todos os detalhes necessários para a execução, como uma descrição, o agente responsável, ferramentas necessárias e mais, facilitando uma ampla gama de complexidades de ação (CREWAI, 2024c). Todas as *tasks* utilizam os seguintes atributos descritos abaixo, conforme a documentação:

- *Description*: Uma declaração clara e concisa do que a tarefa envolve;
- *Agent*: O agente responsável pela tarefa, designado diretamente ou pelo processo da equipe;
- *Expected-output*: Uma descrição detalhada de como é a conclusão da tarefa.;

4.4 Estruturação dos Prompts e Ferramentas

Esta seção apresenta os prompts utilizados pelos agentes, nas tarefas e ferramentas. É importante destacar que todos os prompts foram utilizados em inglês, mas cada um contará com sua respectiva tradução para o português ao lado.

4.4.1 Ferramentas

GetMusicCatalogue

- *Description*: *Fetches the music data and preprocesses content from a given URL. Return a pandas dataframe with data from URL.* (em português, "Busca os dados musicais e pré-processa o conteúdo a partir de uma URL fornecida. Retorna um DataFrame do Pandas com os dados obtidos da URL.");
- URL: (<https://matchfai-backend.onrender.com/getAllData>)

GetUserHistoryData

- URL: (<https://matchfai-backend.onrender.com/getUserData/:userId>)

GetUserDislikedSongs

- *Description: Fetches the user listening history and preprocesses content from a given URL. Return a pandas dataframe with data from URL, or an empty DataFrame if no data is available. (em português, "Busca o histórico de reprodução do usuário e pré-processa o conteúdo a partir de uma URL fornecida. Retorna um DataFrame do Pandas com os dados obtidos da URL ou um DataFrame vazio caso nenhum dado esteja disponível.");*
- URL: (<https://matchfai-backend.onrender.com/getUserDislikedSongs?userId=:userId>)

GetUserLikedSongs

- *Description: Fetches the user listening history and preprocesses content from a given URL. Return a pandas dataframe with data from URL, or an empty DataFrame if no data is available. (em português, "Busca o histórico de reprodução do usuário e pré-processa o conteúdo a partir de uma URL fornecida. Retorna um DataFrame do Pandas com os dados obtidos da URL ou um DataFrame vazio caso nenhum dado esteja disponível.");*
- URL: (<https://matchfai-backend.onrender.com/getUserLikedSongs?userId=:userId>)

GetAllUsersLikedSongs

- *Description: Fetches the music data and preprocesses content from a given URL. Returns a pandas DataFrame with data from the URL, or an empty DataFrame if no data is available. (em português, "Busca os dados musicais e pré-processa o conteúdo a partir de uma URL fornecida. Retorna um DataFrame do Pandas com os dados obtidos da URL ou um DataFrame vazio caso nenhum dado esteja disponível.");*
- URL: (<https://matchfai-backend.onrender.com/getOtherUserLikedSongs?userId=:userId>)

4.4.2 Agentes - Filtro Baseado em Conteúdo

Leitor do Catálogo de Músicas

- *Role: Song Catalogue Reader* (em português, "Leitor do Catálogo de Músicas");
- *Goal: Read all the songs from a catalogue* (em português, "Lê todas as músicas de um catálogo");
- *Backstory: Specializes in handling and returning a song catalogue* (em português, "Especializa-se em gerenciar e retornar um catálogo de músicas.");
- *Verbose: True*;
- *Tools: GetMusicCatalogue* (seção 4.4.1 página 39);
- *Allow-delegation: False*;
- *LLM: crewai.LLM*.

Leitor de Histórico de Usuário

- *Role: User Music History Reader* (em português, "Leitor de Histórico de Usuário");
- *Goal: Read all a music history from an user* (em português, "Lê todo o histórico musical de um usuário.");
- *Backstory: Specializes in handling and returning a music history* (em português, "Especializa-se em gerenciar e retornar o histórico musical.");
- *Verbose: True*;
- *Tools: GetUserHistoryData* (seção 4.4.1 página 39);
- *Allow-delegation: False*;
- *LLM: crewai.LLM*.

Analista de Gêneros Musicais

- *Role: User Music Genres* (em português, "Analista de Gêneros Musicais");
- *Goal: Inferring the user's favorite music genre* (em português, "Inferindo o gênero musical favorito do usuário.");
- *Backstory: Specializes in analyse the user music history to inferring what is their 5 favorites music genres* (em português, "Especializa-se em analisar o histórico musical do usuário para inferir quais são seus 5 gêneros musicais favoritos.");
- *Verbose: True*;
- *Allow-delegation: False*;
- *LLM: crewai.LLM*.

Leitor de Músicas Não Curtidas do Usuários

- *Role: DisLiked Songs Reader* (em português, "Leitor de Músicas Não Curtidas");
- *Goal: Read all disliked songs from an user* (em português, "Lê todas as músicas não curtidas de um usuário.");
- *Backstory: Specializes in handling and returning the user disliked songs* (em português, "Especializa-se em gerenciar e retornar as músicas não curtidas pelo usuário");
- *Verbose: True*;
- *Tools: GetUserDisLikedSongs* (seção 4.4.1 página 40);
- *Allow-delegation: False*;
- *LLM: crewai.LLM*.

Filtro Baseado em Conteúdo

- *Role: Content-Based Music Recommender* (em português, "Recomendador de Músicas Baseado em Conteúdo");
- *Goal: Recommend songs to a user using their listening histories* (em português, "Recomenda músicas a um usuário utilizando seu histórico de reprodução.");
- *Backstory: You are a personalized music recommender, you have high technical competence to use information filtering techniques to recommend music to a user, you must analyze the genres of the songs in the catalog and check which songs are most suitable for a user based on their history* (em português, "Você é um recomendador de músicas personalizado, com alta competência técnica para aplicar técnicas de filtragem de informações a fim de sugerir músicas para um usuário. Sua tarefa é analisar os gêneros das músicas no catálogo e identificar quais são as mais adequadas para o usuário com base em seu histórico de reprodução.");
- *Verbose: True;*
- *Allow-delegation: False;*
- *LLM: crewai.LLM.*

4.4.3 Agentes - Filtro Colaborativo

Analista de Gêneros Musicais

- *Role: User Music Genres* (em português, "Analista de Gêneros Musicais");
- *Goal: Inferring the user's favorite music genre* (em português, "Inferindo o gênero musical favorito do usuário.");
- *Backstory: Specializes in analyse the user liked songs list to inferring what is their 5 favorites music genres* (em português, "Especializa-se em analisar a lista de músicas curtidas do usuário para inferir quais são seus 5 gêneros musicais favoritos.");

- *Verbose: True;*
- *Tools: GetAllUsersLikedSongs* (seção 4.4.1 página 40);
- *Allow-delegation: False;*
- *LLM: crewai.LLM.*

Leitor de Músicas Curtidas do Usuário

- *Role: Liked Songs Reader* (em português, "Leitor de Músicas Curtidas do Usuário");
- *Goal: Read all the liked songs from an user* (em português, "Lê todas as músicas curtidas por 1 usuário");
- *Backstory: Specializes in handling and returning a music history* (em português, "Especializa-se em gerenciar e retornar o histórico musical.");
- *Verbose: True;*
- *Tools: GetUserLikedSongs* (seção 4.4.1 página 40);
- *Allow-delegation: False;*
- *LLM: crewai.LLM.*

Leitor de Músicas Não Curtidas do Usuários

- *Role: DisLiked Songs Reader* (em português, "Leitor de Músicas Não Curtidas");
- *Goal: Read all disliked songs from an user* (em português, "Lê todas as músicas não curtidas de um usuário.");
- *Backstory: Specializes in handling and returning the user disliked songs* (em português, "Especializa-se em gerenciar e retornar as músicas não curtidas pelo usuário");
- *Verbose: True;*

- *Tools: GetUserDisLikedSongs* (seção 4.4.1 página 40);
- *Allow-delegation: False*;
- *LLM: crewai.LLM*.

Leitor de Músicas Curtidas Geral dos Usuários

- *Role: Liked Songs Reader* (em português, "Leitor de Músicas Curtidas do Usuário");
- *Goal: Read all the liked songs from users* (em português, "Lê todas as músicas curtidas pelos usuários.");
- *Backstory: Specializes in handling and returning a list with all users liked songs* (em português, "Especializa-se em gerenciar e retornar uma lista com todas as músicas curtidas pelos usuários.");
- *Verbose: True*;
- *Tools: GetAllUserLikedSongs* (seção 4.4.1 página 40);
- *Allow-delegation: False*;
- *LLM: crewai.LLM*.

Filtro Colaborativo

- *Role: Collaborative Music Recommender* (em português, "Recomendador de Músicas Colaborativo");
- *Goal: Recommend songs to a user using the KNN algorithm* (em português, "Recomenda músicas a um usuário utilizando o algoritmo KNN.");
- *Backstory: You are a personalized music recommender, you have high technical competence to use information filtering techniques to recommend music to a user, you have to use the KNN algorithm about user's liked songs profile to recommend songs to a target*

user, you cannot repeat the songs (em português, "Você é um recomendador de músicas personalizado, com alta competência técnica para aplicar técnicas de filtragem de informações a fim de sugerir músicas para um usuário. Sua tarefa é usar o algoritmo KNN com o perfil de músicas curtidas do usuário para recomendar novas músicas a um usuário-alvo, sem repetir as músicas já recomendadas.");

- *Verbose: True;*
- *Allow-delegation: False;*
- *LLM: crewai.LLM.*

4.4.4 Agentes - Filtro Híbrido

Leitor de Músicas Curtidas do Usuário

- *Role: Liked Songs Reader* (em português, "Leitor de Músicas Curtidas do Usuário");
- *Goal: Read all the liked songs from an user* (em português, "Lê todas as músicas curtidas por 1 usuário");
- *Backstory: Specializes in handling and returning a music history* (em português, "Especializa-se em gerenciar e retornar o histórico musical.");
- *Verbose: True;*
- *Tools: GetUserLikedSongs* (seção 4.4.1 página 40);
- *Allow-delegation: False;*
- *LLM: crewai.LLM.*

Leitor de Músicas Não Curtidas do Usuários

- *Role: DisLiked Songs Reader* (em português, "Leitor de Músicas Não Curtidas");

- *Goal: Read all disliked songs from an user* (em português, "Lê todas as músicas não curtidas de um usuário.");
- *Backstory: Specializes in handling and returning the user disliked songs* (em português, "Especializa-se em gerenciar e retornar as músicas não curtidas pelo usuário");
- *Verbose: True;*
- *Tools: GetUserDisLikedSongs* (seção 4.4.1 página 40);
- *Allow-delegation: False;*
- *LLM: crewai.LLM.*

Filtro Híbrido

- *Role: Hybrid Music Recommender* (em português, "Recomendador de Músicas Híbrido");
- *Goal: Recommend songs to a user by combining two strategies: collaborative and content-based filtering.* (em português, "Recomenda músicas a um usuário combinando duas estratégias: filtragem colaborativa e filtragem baseada em conteúdo.");
- *Backstory: You receive two lists of song recommendations: one from collaborative filtering and one from content-based filtering. Your goal is to recommend the most relevant songs to the user by using the best hybrid recommendation strategy.* (em português, "Você recebe duas listas de recomendações de músicas: uma proveniente da filtragem colaborativa e outra da filtragem baseada em conteúdo. Seu objetivo é recomendar as músicas mais relevantes para o usuário utilizando a melhor estratégia de recomendação híbrida.");
- *Verbose: True;*
- *Allow-delegation: False;*
- *LLM: crewai.LLM.*

4.4.5 Agentes - Nova recomendação

Recomendador de Músicas por Preferências

- *Role: Music Recommender* (em português, "Recomendador de Músicas");
- *Goal: Recommend songs to a user using their preferences* (em português, "Recomenda músicas a um usuário usando as preferências dele");
- *Backstory: You are a personalized music recommender, uses the user's preferences to recommend music.* (em português, "Você é um recomendador de música personalizado, usa as preferências do usuário para recomendar músicas.");
- *Verbose: True;*
- *Allow-delegation: False;*
- *LLM: crewai.LLM.*

4.4.6 Tarefas - Filtro Baseado em Conteúdo

SongCatalogueTask

- *Description: Read and returns all the song catalogue at this URL: <https://matchfai-backend.onrender.com/getAllData>* (em português, "Lê e retorna todo o catálogo de músicas");
- *Agent: Leitor do Catálogo de Músicas* (seção 4.4.2 página 41);
- *Expected-output: Song Catalogue* (em português, "Catálogo de Músicas");

UserMusicHistoryTask

- *Description: Read and returns the user music history at this URL: <https://matchfai-backend.onrender.com/getUserData/:userId>* (em português, "Lê e retorna o histórico de reprodução do usuário");

- *Agent*: Leitor do Histórico do Usuário (seção 4.4.2 página 41);
- *Expected-output*: *User listening history* (em português, "Histórico de reprodução do usuário");

UserDislikedSongsTask

- *Description*: *Read and returns the user's disliked songs at this URL: <https://matchfai-backend.onrender.com/getUserDislikedSongs?userId=:userId>* (em português, "Lê e retorna as músicas não curtidas do usuários");
- *Agent*: Leitor de Músicas Não Curtidas do Usuário (seção 4.4.2 página 42);
- *Expected-output*: *User DisLiked Songs* (em português, "Músicas Não Curtidas do Usuário");

UserFavoriteGenreTask

- *Description*: *Inferring the user's favorite music genre. For this task, use the user's listening history to infer their 5 most preferred music genre* (em português, "Inferindo o gênero musical favorito do usuário. Para esta tarefa, utilize o histórico de reprodução do usuário para inferir os 5 gêneros musicais que ele mais prefere.");
- *Agent*: Analista de Gêneros Musicais (seção 4.4.2 página 42);
- *Expected-output*: *User favorite genre* (em português, "Gêneros favoritos do usuário");

FilteringContentAgentTask

- *Description*: *Generate a list of 10 songs, containing only the songs that have been selected from the music catalog. For this task, use the user's music history and genres to create the user's music profile. Select from the music catalog only the songs that are closest to the user's profile and that are not in the user's music history. Select 2 songs for each genre you found, you cannot repeat the songs and return these songs in a JSON.*

The JSON object must have the genre, song name and artist name, you must have to return like this: [{ "genre": "genre name", "songname": "song name", "artistname": "artist name"}, { "genre": "genre name", "song name": "song name", "artistname": "artist name"}] only this, don't add a \on JSON, don't add any text before, just the return don't add any note or comment above Don't forget to remove songs which are in user disliked songs list. (em português, "Gere uma lista de 10 músicas, contendo apenas as músicas selecionadas do catálogo musical. Para esta tarefa, use o histórico musical e os gêneros do usuário para criar o perfil musical do mesmo. Selecione do catálogo musical apenas as músicas mais próximas do perfil do usuário e que não estejam no histórico musical do usuário. Selecione 2 músicas para cada gênero identificado, sem repetir as músicas, e retorne essas músicas em um JSON. O objeto JSON deve conter o gênero, o nome da música e o nome do artista, e deve ser retornado no seguinte formato: [{ "genre": "genre name", "songname": "song name", "artistname": "artist name"}, { "genre": "genre name", "songname": "song name", "artistname": "artist name"}] Somente isso, sem adicionar uma \no JSON, sem nenhum texto antes, e sem nenhuma nota ou comentário acima. Não se esqueça de remover as músicas que estão na lista de músicas descurtidas do usuário.");

- *Agent*: Filtro Baseado em Conteúdo (seção 4.4.2 página 43);
- *Expected-output*: *Recommended Songs for the user* (em português, "Músicas Recomendadas para o usuário");

4.4.7 Tarefas - Filtro Colaborativo

UserDislikedSongsTask

- *Description*: *Read and returns the user's disliked songs at this URL: <https://matchfail-backend.onrender.com/getUserDislikedSongs?userId=:userId>* (em português, "Lê e retorna as músicas não curtidas do usuários");

- *Agent*: Leitor de Músicas Não Curtidas do Usuário (seção 4.4.3 página 44);
- *Expected-output*: *User DisLiked Songs* (em português, "Músicas Não Curtidas do Usuário");

UserLikedSongsTask

- *Description*: *Read and returns the user's liked songs at this URL: <https://matchfaibackend.onrender.com/getUserLikedSongs?userId=:userId>* (em português, "Lê e retorna as músicas curtidas do usuários");
- *Agent*: Leitor de Músicas Curtidas do Usuário (seção 4.4.3 página 44);
- *Expected-output*: *User Liked Songs* (em português, "Músicas Curtidas do Usuário");

AllUsersLikedSongsTask

- *Description*: *Fetch all the user's liked songs at this URL: <https://matchfaibackend.onrender.com/getOtherUserLikedSongs?userId=:userId>* (em português, "Lê e retorna as músicas curtidas de todos usuários");
- *Agent*: Leitor de Músicas Curtidas Geral dos Usuários (seção 4.4.3 página 45);
- *Expected-output*: *All users' liked songs* (em português, "Todas as músicas curtidas de todos os Usuários");

UserFavoriteGenreTask

- *Description*: *Inferring the user's favorite music genre. For this task, use the user's listening history to infer their 5 most preferred music genre* (em português, "Inferindo o gênero musical favorito do usuário. Para esta tarefa, utilize o histórico de reprodução do usuário para inferir os 5 gêneros musicais que ele mais prefere.");
- *Agent*: Analista de Gêneros Musicais (seção 4.4.3 página 43);
- *Expected-output*: *User favorite genre* (em português, "Gêneros favoritos do usuário");

CollaborativeAgentTask

- *Description: For this task, use the list of songs liked by all users to infer the musical taste of each user, based on the music genres of the liked songs identified by their respective userIds. From this inference, calculate the Euclidean distance between the target user's musical profile and the profiles of other users to find the 3 closest neighbors. The recommendation should include 10 songs, selected only from the profiles of the closest neighbors. The recommended songs: Must not be in the target user's liked songs list, Must not be repeated within the recommendation, Must belong to the genres identified in the neighbors' profiles, Must not include songs the target user has marked as disliked. Return the final recommendatio list in a JSON with this format: [{ "genre": "genre name", "songname": "song name", "artistname": "artist name"}, { "genre": "genre name", "song name": "song name", "artist-name": "artist name"}] Do not add backslashes \to the JSON, and do not include any additional messages in the output. Also, ensure to remove disliked songs from the recommendation, excluding those that are in the user's disliked songs list. (em português, "Para esta tarefa, use a lista de músicas curtidas por todos os usuários para inferir o gosto musical de cada usuário, considerando os gêneros musicais das músicas curtidas identificadas por seus respectivos userIds. A partir dessa inferência, calcule a distância Euclidiana entre o perfil musical do usuário-alvo e os demais usuários para encontrar os 3 vizinhos mais próximos. A recomendação deve conter 10 músicas, selecionadas apenas dos perfis dos vizinhos mais próximos. As músicas recomendadas, Não podem estar na lista de músicas curtidas do usuário-alvo. Não podem ser repetidas dentro da recomendação. Devem pertencer aos gêneros identificados nos perfis dos vizinhos. Não podem ser músicas que o usuário-alvo já tenha marcado como "disliked". Caso não haja músicas suficientes dos gêneros encontrados nos vizinhos próximos, recomende apenas as músicas que encontrou, sem preencher a quantidade restante com músicas de outros usuários fora dos vizinhos próximos. Divida a recomendação em 2 músicas para cada gênero identificado, garantindo que não haja repetições entre os gêneros. Retorne a lista*

final de músicas recomendadas em um JSON com o seguinte formato: [{ "genre": "genre name", "songname": "song name", "artistname": "artist name"}, { "genre": "genre name", "song name": "song name", "artistname": "artist name"}] Não adicione barras invertidas (\\) ao JSON e não inclua nenhuma mensagem adicional no retorno. E não esqueça de remover as musicas que o usuario nao gostou da recomendação que estão na lista de musicas que ele não gostou.”)

- *Agent*: Filtro Colaborativo (seção 4.4.3 página 45);
- *Expected-output*: *Recommended Songs for the user* (em português, "Músicas Recomendadas para o usuário");

4.4.8 Tarefas - Filtro Híbrido

HybridAgentTask

- *Description*: *Use the raw outputs from the collaborative and content-based filtering agents. Select the 10 most relevant songs: 5 from collaborative and 5 from content-based. Ensure no duplicates, and choose based on the user's favorite genres inferred from genreanalyser and genreanalyserlike. Select 2 songs per genre. Return a JSON in the following format [{ "genre": "genre name", "songname": "song name", "artistname": "artist name"}, { "genre": "genre name", "song name": "song name", "artistname": "artist name"}] No extra notes, comments, or \in the JSON.*(em português, "Use as saídas brutas dos agentes de filtragem colaborativa e baseada em conteúdo. Selecione as 10 músicas mais relevantes: 5 da filtragem colaborativa e 5 da filtragem baseada em conteúdo. Garanta que não haja duplicatas e escolha com base nos gêneros favoritos do usuário, inferidos a partir de genreanalyser e genreanalyserlike. Selecione 2 músicas por gênero. Retorne um JSON no seguinte formato: [{ "genre": "genre name", "songname": "song name", "artistname": "artist name"}, { "genre": "genre name", "song name": "song name", "artistname": "artist name"}] Nenhuma nota extra, comentário ou \no JSON.”)

- *Agent*: Filtro Híbrido (seção 4.4.4 página 47);
- *Expected-output*: *Recommended Songs for the user* (em português, "Músicas Recomendadas para o usuário");

4.4.9 Tarefas - Nova recomendação

SongCatalogueTask

- *Description*: *Read and returns all the song catalogue at this URL: <https://matchfai-backend.onrender.com/getAllData>* (em português, "Lê e retorna todo o catálogo de músicas");
- *Agent*: Leitor do Catálogo de Músicas (seção 4.4.2 página 41);
- *Expected-output*: *Song Catalogue* (em português, "Catálogo de Músicas");

ExecuteRecommendation

- *Description*: *Generate a list of 10 songs, you have to know that the user prefers these genres: preferences and only recommends the songs which exists on catalogue. Select 2 songs for each genre you found, the maximum of 2 songs per artist you have to select the songs using the plays, give some preferences to songs which have more plays, you cannot repeat the songs and return these songs in a JSON* (em português, "Gere uma lista com 10 músicas, sabendo que o usuário prefere esses gêneros: preferences e recomende apenas as músicas que existem no catálogo. Selecione 2 músicas para cada gênero encontrado, com no máximo 2 músicas por artista. Você deve selecionar as músicas com base no número de reproduções, dando preferência às músicas com mais reproduções. Não repita as músicas e retorne essas músicas em um JSON.")
- *Agent*: Recomendador de Músicas por Preferências (seção 4.4.5 página 48);
- *Expected-output*: *Recommended Songs for the user in a JSON and the songs need to be of one genre each and you cannot repeat: The JSON object must have the*

genre, song name and artist name, you must have to return like this [{ "genre": "genre name", "songname": "song name", "artistname": "artist name"}, { "genre": "genre name", "song name": "song name", "artistname": "artist name"}] only this, don't add a \on JSON, don't add any text before, just the return don't add any note or comment above (em português, "Músicas recomendadas para o usuário em um JSON, e as músicas precisam ser de um gênero cada, sem repetições: O objeto JSON deve conter o gênero, o nome da música e o nome do artista, e você deve retornar dessa forma: [{ "genre": "genre name", "songname": "song name", "artistname": "artist name"}, { "genre": "genre name", "song name": "song name", "artistname": "artist name"}] apenas isso, não adicione uma \no JSON, não adicione nenhum texto depois, apenas retorne e não adicione numa nota ou comentário abaixo.");

4.5 Banco de Dados

O banco de dados utilizado neste trabalho foi o MongoDB, um banco de dados orientado a documentos. Sua flexibilidade permite que os dados sejam armazenados de acordo com a estrutura retornada pela aplicação, eliminando a necessidade de junções complexas e, conseqüentemente, melhorando o desempenho (MONGODB, 2024a). E a escolha deste foi devido à sua fácil integração com o Node.js por meio do Mongoose, um modelador de dados (*Object Data Modeling* - ODM) baseado em Node.js, que facilita o gerenciamento e a validação de dados. A versão utilizada foi a 7.0.12.

A subseção abaixo descreverá as *collections* criadas.

4.5.1 *genres*

- Função: Armazena todos os gêneros musicais identificados no *Chartsfy*;
- Utilidade: Listagem dos gêneros para a escolha dos usuários na interface gráfica.

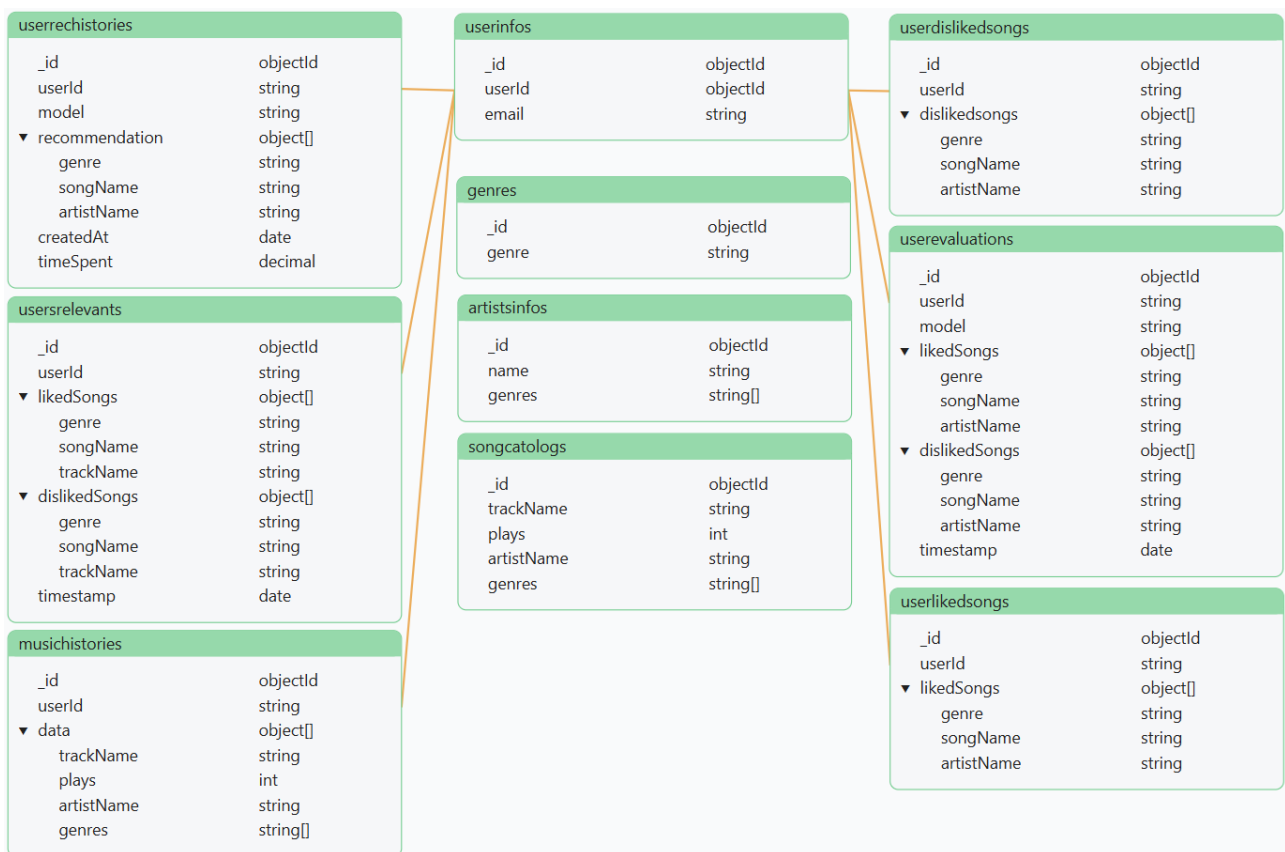


Figura 4.12: Modelagem das *collections* do MongoDB (SPOTIFY, 2024a)

4.5.2 *artistinfos*

- Função: Armazena todos os artistas e seus respectivos gêneros musicais identificados no *Chartsfy*;
- Utilidade: Atualização da *collection* de *songscatologs* para adicionar os gêneros musicais às músicas.

4.5.3 *songscatologs*

- Função: Armazena o catálogo musical com todas as músicas identificadas no *Chartsfy*;
- Utilidade: Essencial para Filtro Baseado em Conteúdo (seção 4.4.2 pg.43) e (seção 4.4.3 pg.45) recomendarem as músicas para os usuários.

4.5.4 *userinfos*

- Função: Armazena os identificadores dos usuários e seus emails;
- Utilidade: Para simplificar o uso da aplicação, os usuários podem inserir seus e-mails para que seus identificadores sejam recuperados automaticamente. Esses identificadores são utilizados nos *endpoints* para acessar e retornar os dados correspondentes.

4.5.5 *userrecherhistories*

- Função: Armazena todas as recomendações geradas, o modelo utilizado e o tempo que foi necessário para gerar;
- Utilidade: Registro do histórico de recomendações e o tempo de cada uma, sendo este último utilizado para a comparação dos modelos.

4.5.6 *musichistories*

- Função: Armazena os históricos musicais dos usuários;
- Utilidade: Essencial para Filtro Baseado em Conteúdo (seção 4.4.2 pg.43) identificar o perfil do usuário e recomendar corretamente.

4.5.7 *userdislikedsongs*

- Função: Armazena as músicas marcadas como **não gostei** pelos usuários após as avaliações;
- Utilidade: Fundamental para o funcionamento do Filtro Baseado em Conteúdo (Seção 4.4.2, pg.43) e do Filtro Colaborativo (seção 4.4.3 pg.45), permitindo que músicas marcadas como "não curtidas" sejam evitadas durante o processo de recomendação.

4.5.8 *userlikedsongs*

- Função: Armazena as músicas marcadas como **gostei** pelos usuários após as avaliações;

- Utilidade: Fundamental para o funcionamento Filtro Colaborativo (Seção 4.4.3, pg.45), pois estas músicas são utilizadas para identificar o perfil do usuário e realizar as recomendações.

4.5.9 *userevaluations*

- Função: Armazena as músicas avaliadas pelos usuários após a recomendação;
- Utilidade: Os dados são utilizados para a métrica de precisão.

4.5.10 *usersrelevants*

- Função: Armazena as músicas avaliadas como relevantes pelos usuários após a recomendação;
- Utilidade: Os dados são utilizados para a métrica de revocação.

4.6 Rotina de Atualização de Dados

Para garantir que os dados dos usuários e o catálogo de músicas estejam sempre atualizados, foi implementada uma rotina de atualização, conforme ilustrado pelo componente apresentado na arquitetura em nível de componentes na Figura 4.3 (página 30). Esta seção também abordará o processo de pré-processamento de dados em detalhes. A rotina é realizada pelo *MatchfAI - Backend* por meio da chamada ao endpoint */routine* do *MatchfAI - API*. Esse processo é automatizado utilizando o NodeCron (NPM, 2024), que foi configurado para ser executado diariamente em um horário específico.

4.6.1 Pré-Processamento dos Dados

Os dados são extraídos diretamente do banco de dados do *Chartsfy* e pré-processados para o banco de dados do *MatchfAI*.

Obtenção de Dados dos Usuários

Os dados da coleção de *users* (seção 4.2.3) são exportados diretamente para a coleção de *usersinfos*. As próximas etapas utilizam a coleção de *tracks* (seção 4.2.3)

Obtenção dos Artistas e Gêneros Musicais

A coleção foi gerada agrupando os artistas pelo nome e separando seus identificadores. Vale destacar que esses identificadores são fornecidos diretamente pelo *Spotify*, e foram essenciais para obter um dado crucial a ser utilizado pelo agente de filtragem baseado em conteúdo: o gênero musical do artista. É importante ressaltar que o *Spotify* atribui um ou mais gêneros a cada artista, destacando o gênero principal, que representa a maior parte de sua discografia. No total, foram obtidos aproximadamente 5633 artistas.

Com os identificadores dos artistas em mãos, foram realizadas diversas requisições à API do *Spotify* para recuperar os gêneros musicais associados a cada um deles. Esse processo resultou em cerca de 1763 gêneros musicais distintos. Com os gêneros disponíveis, a coleção *genres* (seção 4.5.1) foi atualizada. Em seguida, foi realizada uma junção com a tabela de artistas, permitindo o salvamento dos dados na coleção *artistsinfos* (seção 4.5.2).

Obtenção do Catálogo de Músicas

Com os dados de gêneros musicais dos artistas, agora é possível processar as informações para construir o catálogo de músicas, que será utilizado tanto pelo agente de filtragem baseado em conteúdo quanto pelo colaborativo. Os seguintes passos foram realizados:

1. As músicas foram agrupadas pela coluna *trackName*, gerando uma nova coluna chamada *plays*, que contém a soma dos valores agrupados;
2. As músicas foram ordenadas pela coluna *plays* em ordem decedente;
3. O nome do artista foi extraído de cada música;
4. Foi realizada uma junção com a coleção *artistsinfos* (seção 4.5.2).

Após o agrupamento e processamento dos dados, foram obtidas 22.178 músicas, e a coleção *songscatalogs* (seção 4.5.3) foi devidamente atualizada.

Todas as etapas mencionadas são executadas na ordem apresentada, uma vez que algumas dependem da disponibilidade de dados de etapas anteriores. Vale ressaltar que, para todas as coleções, é realizada uma consulta nos dados atuais, e apenas os dados novos são processados e inseridos, garantindo assim maior agilidade no processo.

Obtenção do Histórico dos Usuários

Para esta etapa foram realizados os seguintes passos:

1. Filtragem pela coluna *userId* para obter as músicas de um usuário específico;
2. As músicas foram agrupadas pela coluna *trackName* assim gerando uma nova coluna *plays*, contendo o valor agrupado;
3. Ordenação por *plays* de forma descendente;
4. Extração do nome do artista;
5. Foi realizada uma junção com a coleção *artistsinfos* (seção 4.5.2).

Com os passos anteriores concluídos, a coleção *musichistories* (seção 4.5.6) foi atualizada com os dados de históricos de todos os usuários.

4.7 Fluxos de Geração da Recomendação

Esta seção vai descrever os fluxos de geração da recomendação no sistema.

4.7.1 Fluxo Geral

1. O processo se inicia na interface da aplicação que pode ser acessada pelo endereço <https://matchfai.web.app>
2. O usuário insere o email cadastrado e escolhe um dos modelos;

3. O *MatchfAI - FrontEnd* envia uma requisição ao *MatchfAI - BackEnd* que por sua vez envia a requisição ao *MatchfAI - API* que realiza o processo de recomendação.

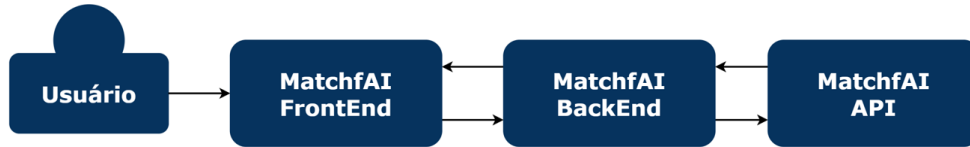


Figura 4.13: Fluxo Geral

4. O processo começa com a execução do Filtro Baseado em Conteúdo, que utiliza tarefas e ferramentas para reunir os dados necessários aos agentes.
5. O agente de Filtro Baseado em Conteúdo recebe os dados e retorna a lista de músicas recomendadas.

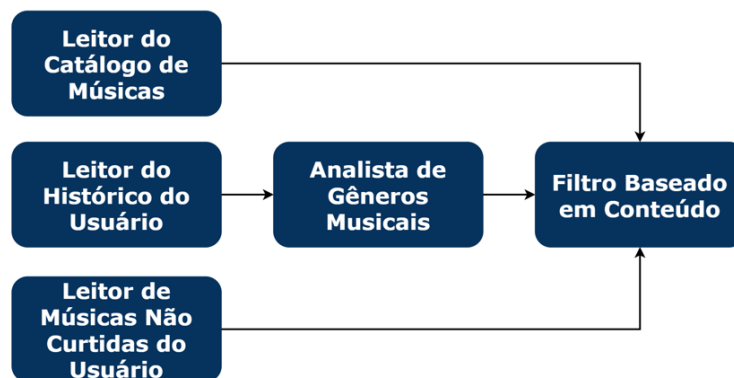


Figura 4.14: Fluxo do Filtro Baseado em Conteúdo

6. Após esta etapa, o filtro colaborativo é iniciado e assim como o anterior, executa todas as tarefas e ferramentas necessárias para obter os dados para os agentes;
7. O agente de Filtro Colaborativo recebe os dados e retorna a lista de músicas recomendadas.
8. Por fim, o agente híbrido recebe as músicas recomendadas dos filtros anteriores e retorna as 10 músicas mais relevantes.

Porém o sistema possui outros fluxos possíveis.

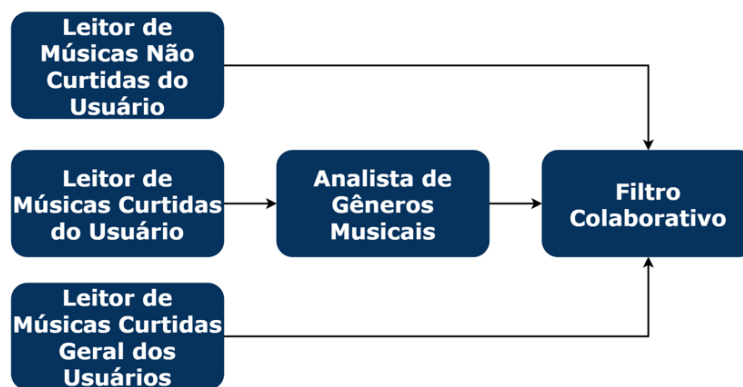


Figura 4.15: Fluxo do Filtro Colaborativo



Figura 4.16: Fluxo do Filtro Híbrido

4.7.2 Fluxo - Sem Músicas Avaliadas

Este fluxo ocorre quando o usuário ainda não avaliou nenhuma música, o que pode impactar o funcionamento do filtro colaborativo. Esse método depende de dados de músicas curtidas para realizar o processo de recomendação de forma eficaz.

1. O processo se inicia na interface da aplicação que pode ser acessada pelo endereço <https://matchfai.web.app>
2. O usuário insere o email cadastrado e escolhe um dos modelos;
3. O *MatchfAI - FrontEnd* envia uma requisição ao *MatchfAI - BackEnd* que por sua vez envia a requisição ao *MatchfAI - API* que realiza o processo de recomendação.
4. O processo começa com a execução do Filtro Baseado em Conteúdo, que utiliza tarefas e ferramentas para reunir os dados necessários aos agentes.

5. O agente de Filtro Baseado em Conteúdo processa os dados e retorna uma lista de músicas recomendadas.
6. Como o usuário ainda não avaliou nenhuma música, a lista de recomendações é gerada e retornada imediatamente.

4.7.3 Fluxo - Sem Histórico de Músicas Ouvidas

Este fluxo ocorre quando o usuário é novo e ainda não possui um histórico armazenado. Como o sistema depende desses dados para funcionar de maneira ideal, uma abordagem alternativa é aplicada.



Figura 4.17: Fluxo do Recomendador de Músicas

1. O processo se inicia na interface da aplicação que pode ser acessada pelo endereço <https://matchfai.web.app>
2. O usuário insere o email cadastrado;
3. Como não há dados disponíveis, o sistema exibe um modal solicitando que o usuário selecione 5 gêneros que considera mais relevantes;
4. O usuário escolhe um modelo para recomendar;
5. O *MatchfAI - FrontEnd* envia uma requisição ao *MatchfAI - BackEnd* que por sua vez envia a requisição ao *MatchfAI - API* que realiza o processo de recomendação;
6. O recomendador de músicas inicia o processo de sugestão com base nos gêneros indicados como relevantes pelo usuário, gerando uma lista personalizada de músicas recomendadas.

4.8 MatchfAI - Backend

Para facilitar o consumo dados pré-processados pelos agentes, foi criado uma API com o seguindo o padrão REST com o *framework* NodeJS e a linguagem de programação TypeScript. Esta API oferece um melhor suporte e facilita o uso dos dados pelos agentes. Esta API é conectada ao banco de dados apresentado na seção 4.5, logo quase todos os endpoints vão recuperar ou inserir dados neste. Contém os seguintes *endpoints*:

- GET /getAllData
 - Entrada: Identificador do usuário;
 - Retorno: Retorna o catálogo de músicas.
- GET /getUserData/:userId
 - Entrada: Identificador do usuário;
 - Retorno: Retorna o histórico de músicas do usuário.
- GET /getUserInfo
 - Entrada: Email do usuário;
 - Retorno: Retorna o identificador do usuário.
- GET /getUserDislikedSongs
 - Entrada: Email do usuário;
 - Retorno: Retorna as músicas marcadas como ***não gostei*** pelo usuário.
- GET /getOtherUserLikedSongs
 - Entrada: Email do usuário;
 - Retorno: Retorna as músicas marcadas como ***gostei*** de todos os usuários, exceto do usuário que está solicitando.
- GET /getUserLikedSongs

- Entrada: Email do usuário;
- Retorno: Retorna as músicas marcadas como *gostei* pelo usuário.
- GET /getGenres
 - Entrada: Não possui;
 - Retorno: Retorna todos os gêneros musicais presentes no banco de dados.
- POST /addRelevantEvaluation
 - Entrada: Lista de músicas curtidas e não curtidas, identificador do usuário e modelo utilizado;
 - Retorno: Mensagem de sucesso ou erro.
 - Observação: Este endpoint apenas envia as avaliações das músicas relevantes/não relevantes.
- POST /recommendation
 - Entrada: Email do usuário e modelo a ser utilizado na recomendação;
 - Retorno: Retorna a lista de músicas geradas na recomendação.
 - Observação: Este endpoint envia os dados para o *MatchfAI - API* que por sua vez gera a recomendação e retorna os dados.
- POST /addUserReview
 - Entrada: Lista de músicas curtidas e não curtidas, identificador do usuário e modelo utilizado;
 - Retorno: Mensagem de sucesso ou erro.
- POST /newUserRec
 - Entrada: Email do usuário, modelo a ser utilizado na recomendação e lista com as preferências;

- Retorno: Retorna a lista de músicas geradas na recomendação.
- Observação: Este endpoint envia os dados para o *MatchfAI - API* que por sua vez gera a recomendação e retorna os dados.

4.9 Modelos Utilizados

Forma utilizados os seguintes modelos e suas respectivas versões:

Tabela 4.1: Modelos utilizados

Modelo
Gemini 1.5 Pro
LLaMA3.1-8B-Instant
Mixtral-8x7B-32768

É importante ressaltar que a temperatura (*temperature*, em inglês) adotada para os modelos foi de 0.1, para diminuir a aleatoriedade do modelo e buscar evitar a recomendação de músicas que não existem. Segundo (PEEPERKORN et al., 2024), temperaturas mais altas aumentam a entropia e a perplexidade, levando a mais aleatoriedade e incerteza no processo generativo.

4.10 *MatchfAI - FrontEnd*

A fim de mostrar ao usuário em uma interface gráfica mais agradável, foi desenvolvido uma aplicação front-end simples utilizando o ReactJS, TypeScript e Vite. A interface possui o objetivo principal de ser a forma como o usuário interage com o sistema, mostra as recomendações geradas e também permite ao usuário, avaliar as recomendações recebidas. Foi utilizado o Figma (FIGMA, 2024) para criar o protótipo antes de ser feito.

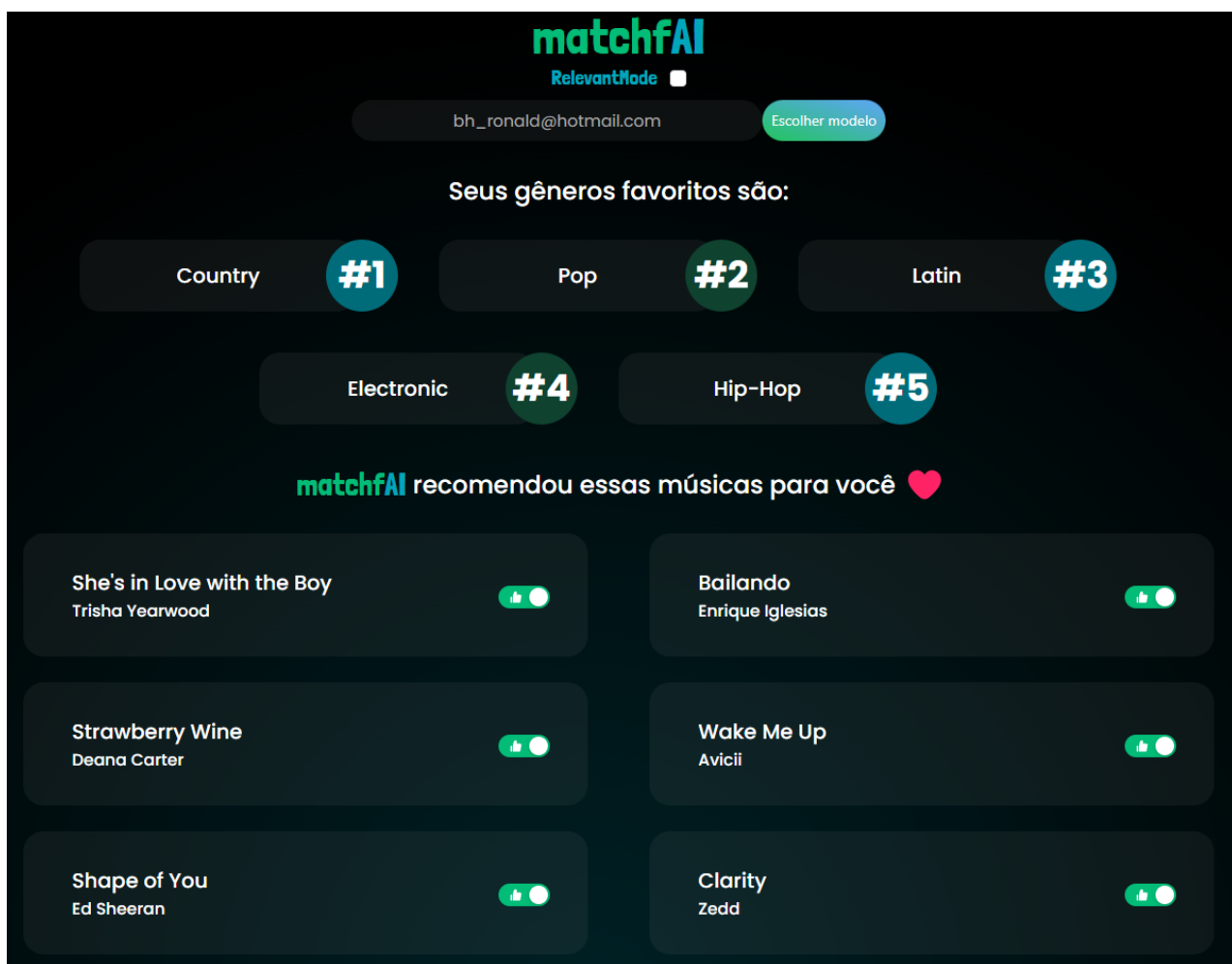


Figura 4.18: Interface

4.11 Métricas de Avaliação

O objetivo de uma tarefa de classificação no contexto de recomendação de produtos é identificar os n itens mais relevantes para um determinado usuário (JANNACH; ZANKER; FELFERNIG, 2010). As seguintes métricas serão utilizadas neste trabalho: precisão, revocação e *F1-Score*.

A precisão mede a proporção de recomendações corretas em relação ao total de recomendações feitas pelo sistema. Ela indica a qualidade das recomendações, ou seja, quantas das músicas recomendadas foram realmente apreciadas pelos usuários.

$$\text{Precisão} = \frac{\text{Número de Itens Recomendados Corretamente}}{\text{Número Total de Itens Recomendados}} \quad (4.1)$$

A revocação mede a capacidade do sistema de recomendação em recuperar todos os itens relevantes disponíveis.

$$\text{Revocação} = \frac{\text{Número de Itens Relevantes Recomendados Corretamente}}{\text{Número Total de Itens Relevantes Recomendados}} \quad (4.2)$$

O F1-score é uma métrica que combina precisão e revocação em uma única medida, proporcionando uma visão mais abrangente do desempenho de um sistema de recomendação.

$$F1\text{-Score} = 2 \times \frac{\text{Precisão} \times \text{Revocação}}{\text{Precisão} + \text{Revocação}} \quad (4.3)$$

A obtenção das avaliações necessárias para estas métricas, serão apresentadas nas próximas seções.

4.12 Avaliação das Recomendações

Conforme discutido na seção 4.11, as métricas mencionadas serão essenciais para avaliar o desempenho dos modelos Gemini, LLaMA e Mixtral em recomendações. A avaliação será fundamentada no feedback dos usuários, que desempenha um papel crucial nessa validação. A figura abaixo ilustra o processo de obtenção das avaliações.

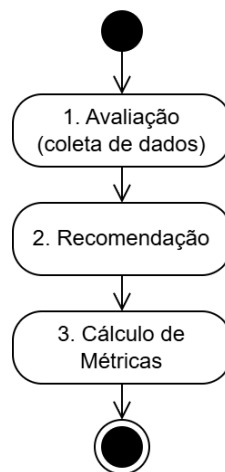


Figura 4.19: Processo de avaliação do sistema

1. Um grupo de usuários avalia as músicas como relevantes/não relevantes;
2. O mesmo grupo de usuários avalia os itens recomendados;
3. O cálculo das métricas (precisão, revocação e f1-score) é feito;

4.12.1 Processo de Testes

Para coletar os dados necessários às métricas de avaliação, foram elaboradas duas etapas de testes envolvendo os usuários.

Etapa 1: Músicas Relevantes

Esta etapa foi projetada para avaliar a métrica de revocação, permitindo que os usuários identifiquem as músicas que consideram relevantes. Durante esse processo, a funcionalidade chamada ***RelevantMode*** é ativada na interface. Essa opção permite que o usuário insira os gêneros musicais que considera alinhados com seu gosto pessoal. Com base nessas informações, o sistema retorna uma lista de músicas que melhor se adequam ao contexto indicado, proporcionando ao usuário a oportunidade de classificá-las como relevantes ou não relevantes. O procedimento seguiu os seguintes passos:

1. Na interface da aplicação, o usuário seleciona a opção ***RelevantMode*** e insere o email cadastrado no *Chartsfy*;

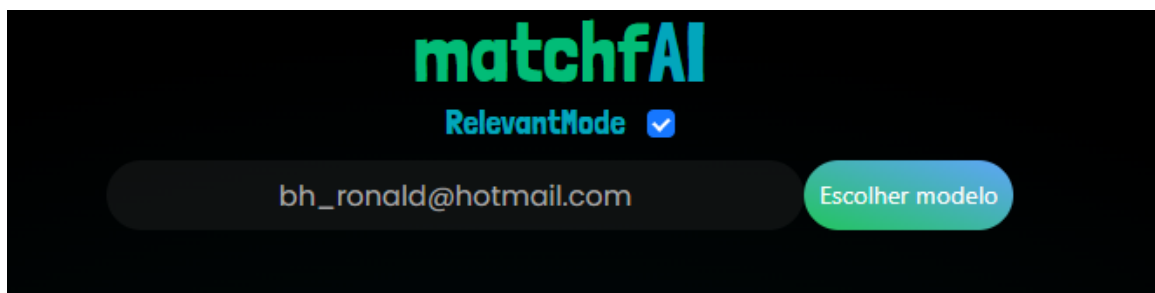


Figura 4.20: Inserção do email e ativação da opção *RelevantMode*

- Um modal é apresentado para o usuário para escolher os 5 gêneros mais relevantes;

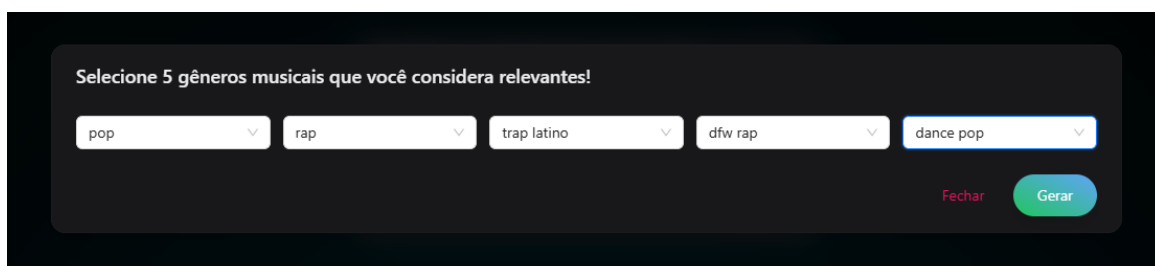


Figura 4.21: Modal com opções de gêneros

- Escolhe o modelo;

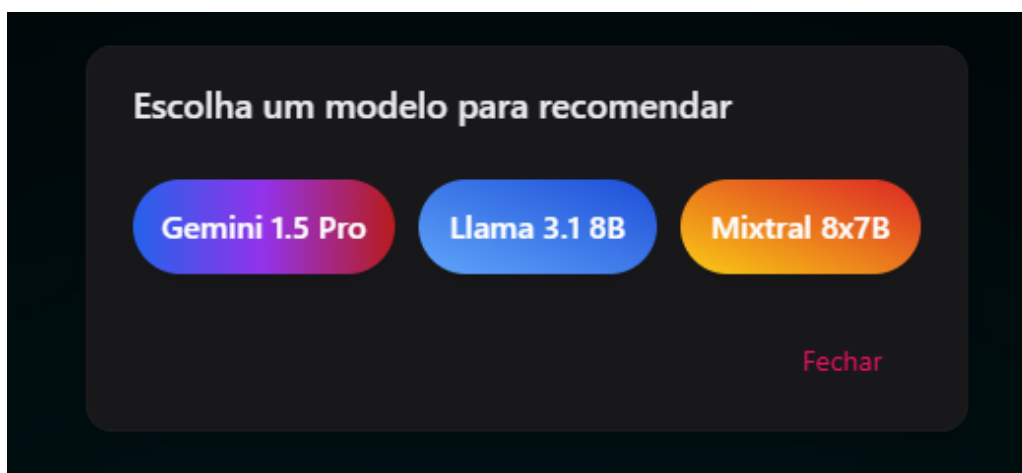


Figura 4.22: Escolha do modelo

- A recomendação é gerada.
- O usuário avalia as músicas e envia a avaliação.

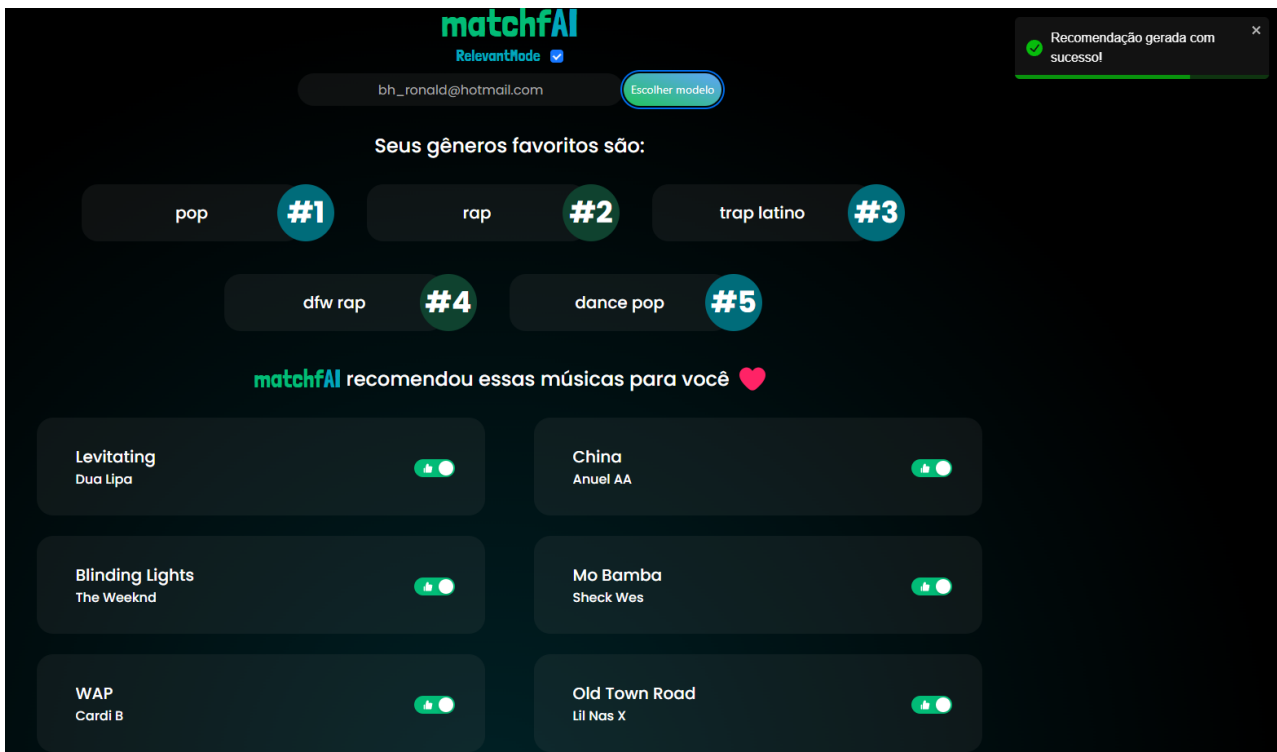


Figura 4.23: Geração da recomendação de itens relevantes

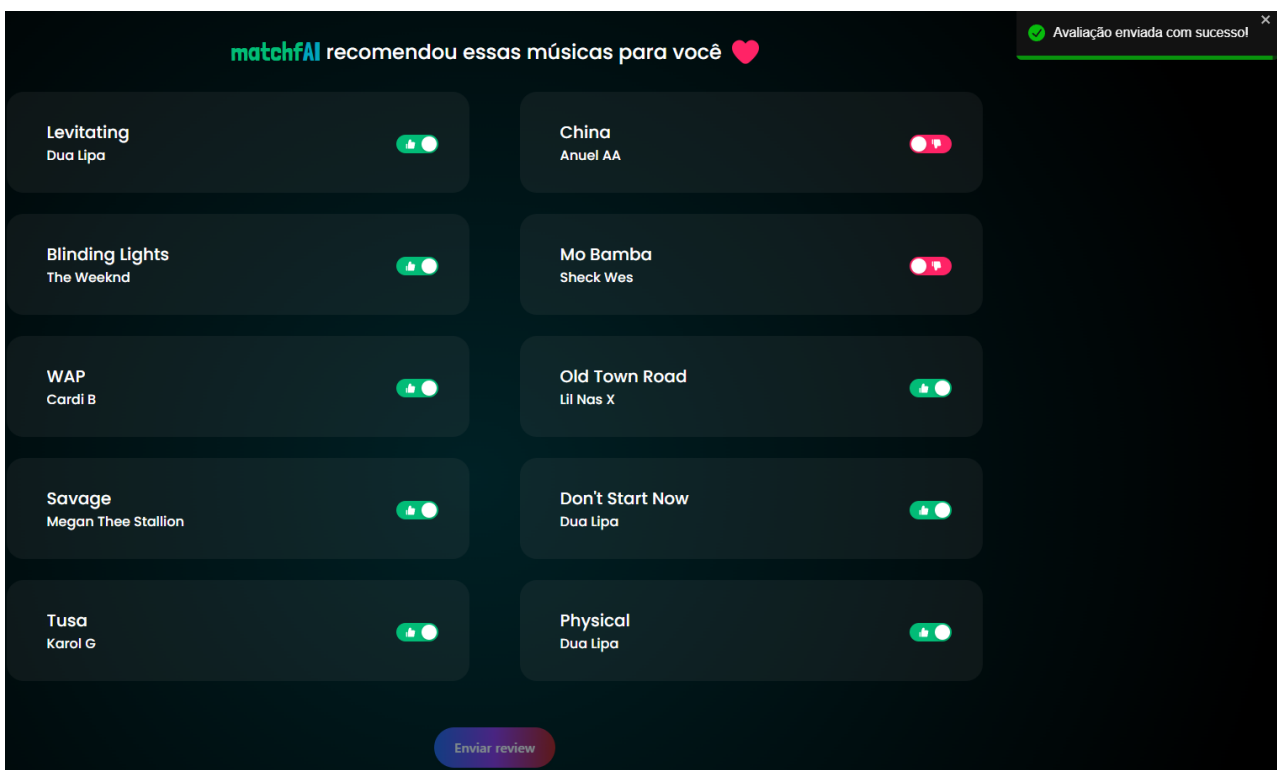


Figura 4.24: Avaliação e envio

Etapa 2: Músicas Recomendadas

Esta etapa foi direcionada à avaliação da métrica de precisão, permitindo que os usuários avaliassem diretamente as músicas recomendadas pelo sistema. Os passos foram os seguintes:

1. O usuário insere o email cadastrado no *Chartsfy*;

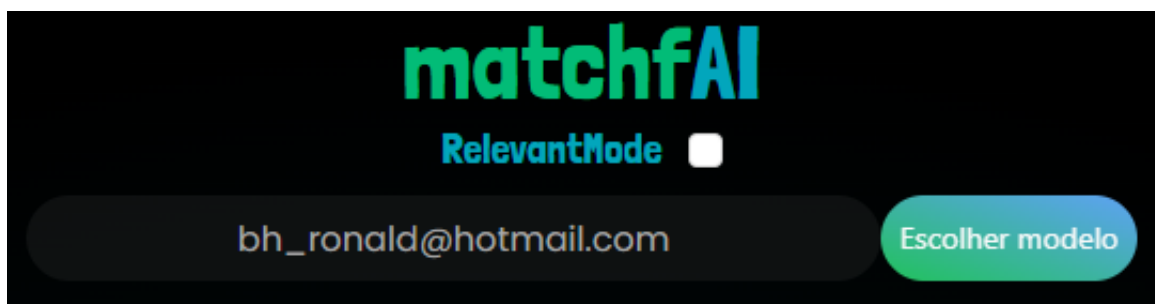


Figura 4.25: Inserção do email e sem a ativação da opção *RelevantMode*

2. Escolhe o modelo;

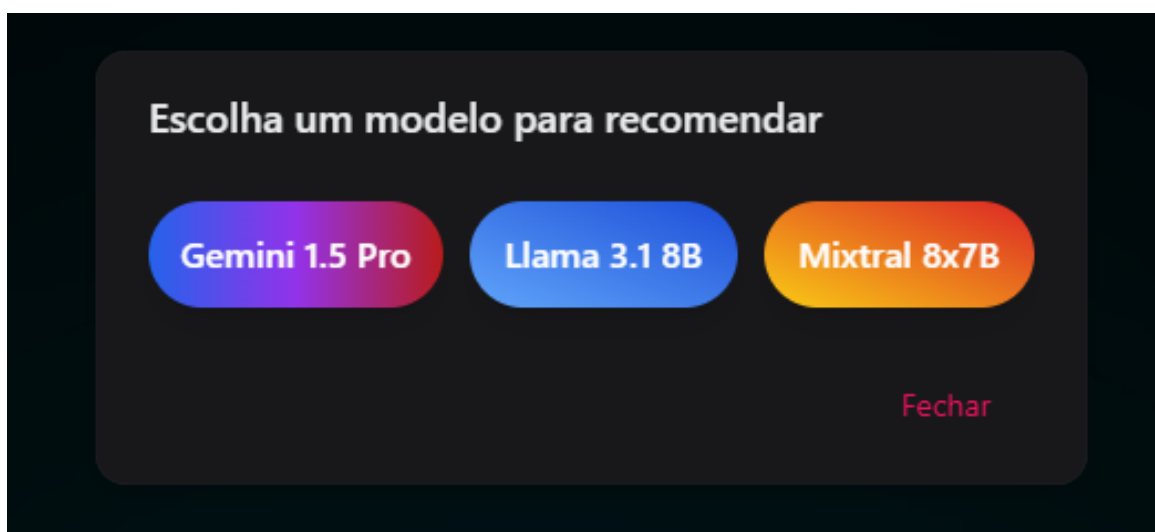


Figura 4.26: Escolha do modelo

3. A recomendação é gerada.

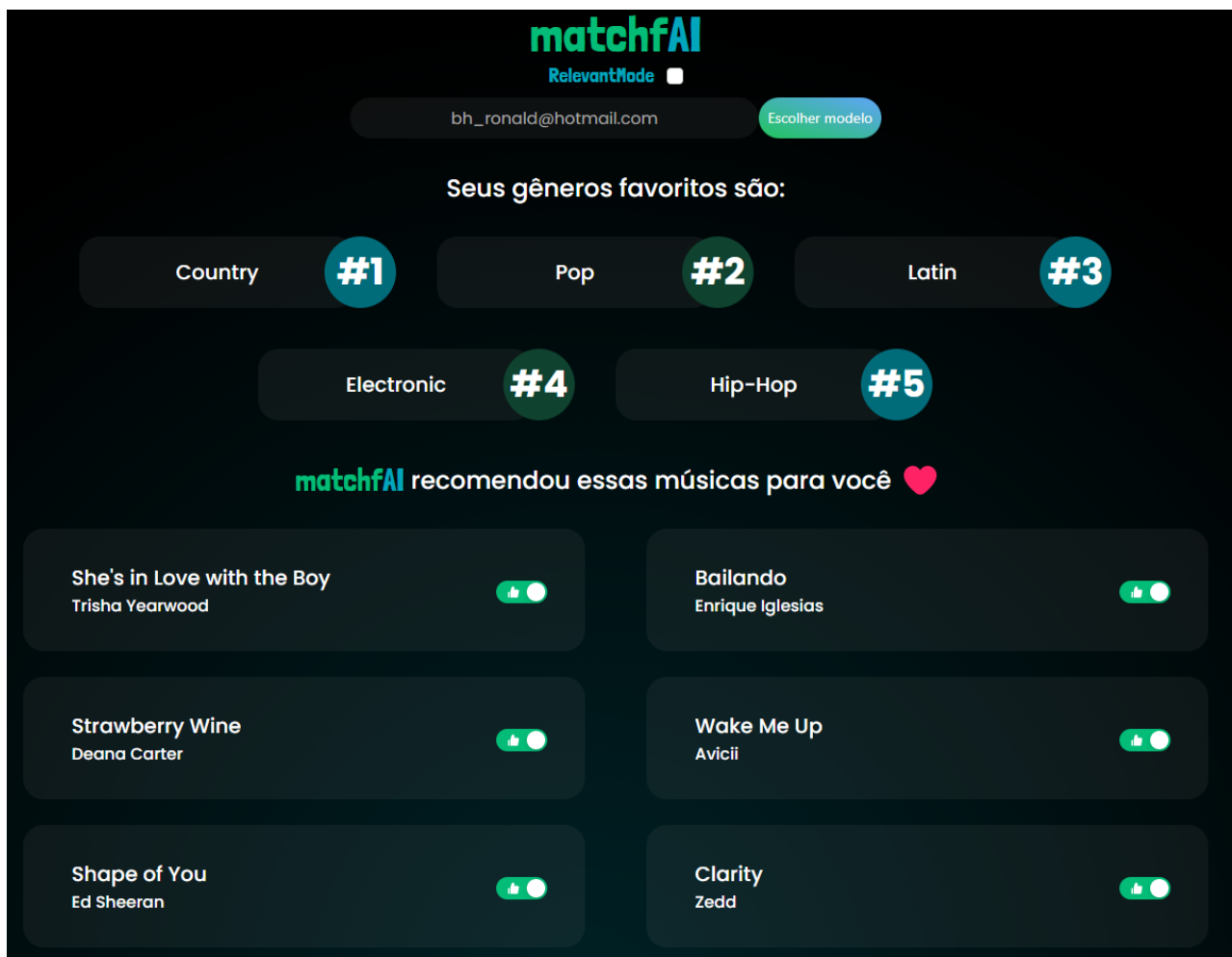


Figura 4.27: Geração da recomendação

4. Avaliação do usuário dos itens recomendados.

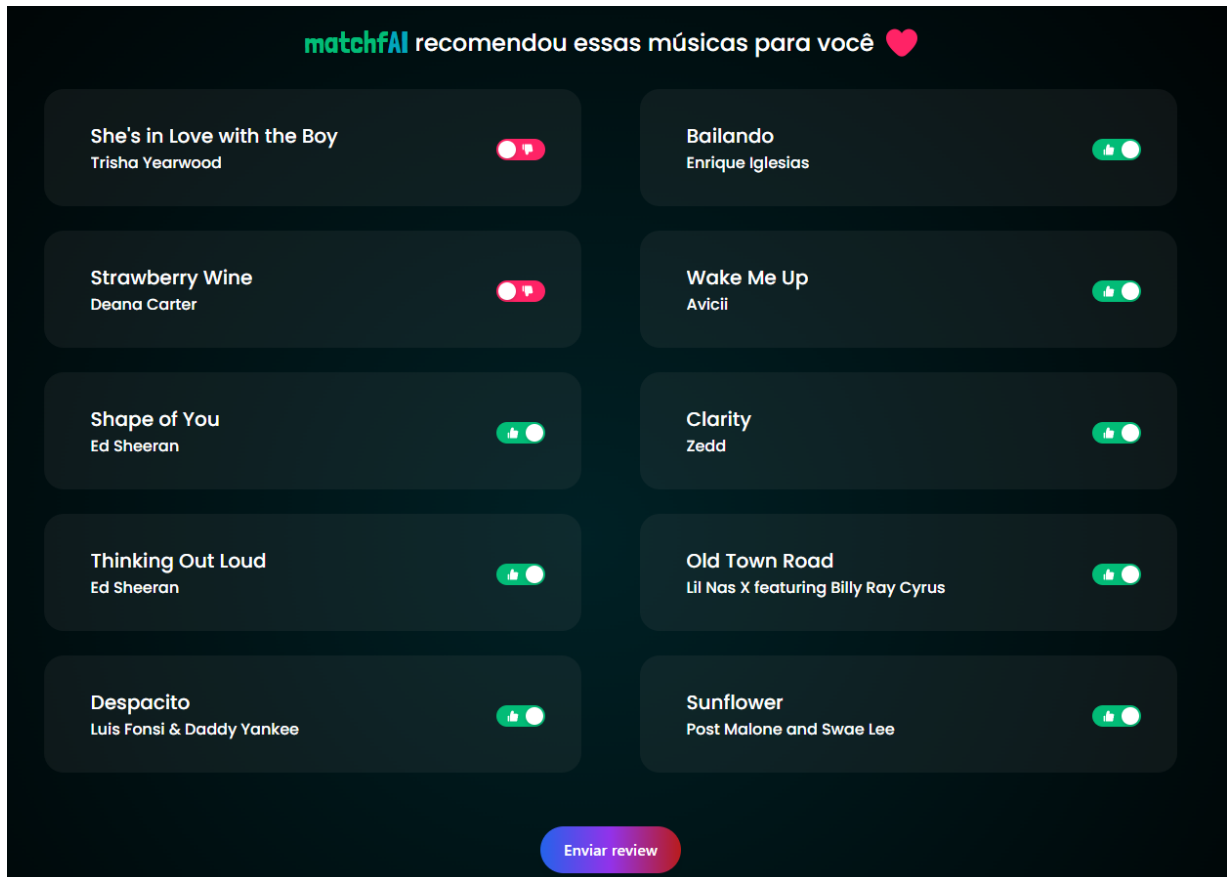


Figura 4.28: Avaliação da recomendação

5. Envio da avaliação.

4.13 Conclusão

Neste capítulo foi apresentado a proposta de arquitetura e implementação deste trabalho. Foram abordados tópicos como base de dados, pré-processamento dos dados, banco de dados, desenvolvimento da API, agentes e modelos utilizados. O Capítulo 5 discutirá os resultados obtidos e discussões sobre este trabalho.

Capítulo 5

Resultados e Discussões

Neste capítulo serão apresentados os resultados deste trabalho, obtidos pela aplicação dos métodos descritos no Capítulo 4. Esses resultados incluem todos os passos necessários para o desenvolvimento completo do sistema.

5.1 Comparação dos Modelos

Os testes foram realizados no período de 13/11/2024 a 19/11/2024 com 13 usuários. Os resultados experimentais estão expostos na tabela 5.1 em termos de média e desvio padrão. A tabela também apresenta as métricas dos três modelos utilizados neste trabalho (Gemini, LLaMA e Mixtral). Os resultados indicam que o modelo **Gemini 1.5 Pro** obteve o melhor desempenho em todas as métricas. Observa-se que o desvio padrão foi elevado em todos os modelos, o que pode ser atribuído à presença de extremos nas recomendações, com dados variando entre 100% de acertos e apenas 10%. Essa variabilidade contribui significativamente para o aumento do desvio padrão.

Tabela 5.1: Métricas de cada modelo

Modelo	Tempo (s)	Precisão	Revocação	F1-Score
LLaMA 3.1 8B	131.21	0.630 ± 0.297	0.751 ± 0.242	0.685 ± 0.283
Mixtral 7x8B	250.92	0.644 ± 0.310	0.786 ± 0.225	0.708 ± 0.277
Gemini 1.5 Pro	96.45	0.704 ± 0.229	0.790 ± 0.174	0.745 ± 0.196

Os apêndices A, B e C possuem as informações de perfis de cada usuário que participou do processo de avaliação do sistema, bem como as avaliações feitas para as recomendações. Ao comparar os modelos pelo tempo de execução do processo de recomendação, observa-se que o **Gemini 1.5 Pro** obteve o menor tempo de processamento entre os modelos, destacando-se como o de melhor desempenho em todas as métricas e também o mais rápido. A maior velocidade pode ser explicada pelo fato de que a API de acesso ao Gemini é desenvolvida e mantida pelo próprio Google. Em contraste, os modelos LLaMA e Mixtral utilizam o Groq como solução para acessar modelos acelerados, o que implica que esses dois modelos não possuem um acesso proprietário, como o do Google, e dependem de uma solução de terceiros para essa integração. No entanto, o tempo de execução não foi um fator determinante na avaliação dos usuários. Embora o modelo Mixtral tenha superado o LLaMA em todas as métricas, seu tempo de execução foi mais de 90% superior ao do LLaMA.

5.2 Custos

Para disponibilizar o serviço de forma acessível aos usuários, foi necessário investir em serviços de nuvem. O *MatchfAI - Backend* foi hospedado no Render, enquanto a *MatchfAI - API* utilizou a infraestrutura da AWS. Durante os 15 dias de disponibilidade do sistema, o custo total foi de US\$ 16,39, sendo US\$ 12,39 destinados à AWS e US\$ 4,00 ao Render. Já o *MatchfAI - FrontEnd* foi hospedado gratuitamente no Firebase.

Capítulo 6

Considerações Finais

Este trabalho teve como objetivo desenvolver um sistema de recomendação híbrido de músicas, utilizando LLMs e agentes inteligentes, com um conjunto de dados composto por 22.178 músicas, 1.763 gêneros musicais e 5.633 artistas. Foram empregados três LLMs: *Gemini 1.5 Pro*, *LLaMA 3.1-8B-Instant* e *Mixtral-8x7B-32768*. A arquitetura proposta foi genérica para os três modelos, ou seja, o mesmo código e os mesmos prompts foram utilizados para todos, garantindo uma comparação justa em um ambiente comum. Durante o desenvolvimento do sistema, observou-se que a combinação dos LLMs com uma arquitetura de agentes inteligentes foi eficaz na geração de recomendações musicais, conforme observado nos resultados no capítulo anterior. A partir dos resultados obtidos, constatou-se que o modelo *Gemini 1.5 Pro* apresentou um bom desempenho, com valores de precisão, revocação e *F1-Score* superiores a 0.7, além de ter o menor tempo de execução entre os modelos avaliados.

Durante o desenvolvimento do sistema, diversos desafios foram enfrentados como:

- **Estruturação dos Prompts:** Como o sistema incluía um front-end e um back-end, houve dificuldade em lidar com a subjetividade nas respostas dos LLMs após o processo de recomendação. Ambos os componentes esperavam uma estrutura de dados fixa, mas os LLMs frequentemente geravam respostas variáveis a cada interação. Isso exigiu inúmeros ajustes nos prompts dos agentes para alcançar estabilidade nos três modelos, dado que o mesmo código era utilizado para todos. Apesar disso, o modelo Mixtral algumas vezes

não seguia corretamente, e assim forçava que a recomendação fosse realizada novamente.

- **Contêinerização da API:** Esse foi um dos processos mais desafiadores, pois foi necessário criar um ambiente virtual específico para o *CrewAI*, que requer várias dependências complexas. A instalação dessas dependências no sistema operacional Windows mostrou-se problemática e muitas vezes falhava. Diversos ajustes foram feitos nos arquivos Docker para inicializar corretamente o contêiner. Contudo, como a imagem final tinha cerca de 3 GB e demandava alto poder de processamento para ser criada, foi necessário utilizar uma instância da AWS com maior capacidade de processamento e memória RAM. O Render foi considerado como alternativa, mas o uso de RAM ultrapassava os 512 MB, causando a interrupção do serviço.
- **Construção da Rotina de Atualização de Dados:** Essa rotina era essencial para manter os dados dos usuários atualizados. Inicialmente, foi planejado incorporá-la ao *MatchfAI - Backend*, mas isso exigia uma imagem Docker customizada com Node.js e Python, cuja estabilidade no Render não era garantida. Por isso, o script foi transferido para dentro do *MatchfAI - API*, o que aumentou o uso de RAM durante sua execução, tornando necessária a utilização de uma instância mais robusta.
- **Recomendação de itens inexistentes:** Esse problema foi observado exclusivamente com o modelo LLaMA, que, em algumas situações, recomendava músicas inexistentes ou atribuía nomes incorretos aos artistas. Além disso, o modelo apresentava dificuldades em realizar recomendações em determinados momentos.
- **Congelamento do Mixtral:** Durante os testes e a execução, o modelo *Mixtral* frequentemente congelava, obrigando a reinicialização da API para restabelecer o funcionamento. Esse problema ocorreu várias vezes ao longo do processo de desenvolvimento e testes.
- **Transição do Langchain para o LLM próprio do CrewAI:** Durante o desenvolvimento, o Langchain (LANGCHAIN, 2024a) e (LANGCHAIN, 2024b) eram utilizados para conectar os agentes aos modelos. No entanto, foi identificado um problema ao tentar executar

ferramentas nos modelos LLaMA e Mixtral, onde um loop infinito era gerado, fazendo com que o modelo tentasse executar repetidamente até atingir o limite diário de tokens no Groq. A solução para esse problema foi a adoção da instância de LLM nativa do CrewAI. Essa abordagem exigiu extensiva pesquisa em fóruns da plataforma CrewAI até que a solução definitiva fosse encontrada.

Por fim, este trabalho teve como objetivo principal avaliar a capacidade dos LLMs em realizar recomendações de músicas, sem se aprofundar em comparações diretas com outros sistemas de recomendação. Essa análise comparativa está planejada como uma etapa futura a ser explorada em trabalhos posteriores.

6.1 Trabalhos Futuros

Como propostas para trabalhos futuros, sugerem-se as seguintes melhorias e extensões:

- Desenvolvimento de um sistema de recomendação tradicional: Implementar um sistema baseado em abordagens tradicionais, utilizando os mesmos dados e usuários, para realizar uma comparação direta com os resultados obtidos neste trabalho.
- Recomendações mais personalizadas: Permitir a escolha de períodos específicos para analisar o histórico do usuário e gerar recomendações com base nesses intervalos de tempo.
- Exploração de outros LLMs: Investigar o desempenho de diferentes modelos de linguagem, como o GPT (OPENAI, 2024), ampliando o conjunto de LLMs avaliados.
- Análise de hiperparâmetros: Testar diferentes valores para o hiperparâmetro de temperatura e avaliar seu impacto na qualidade das recomendações.
- Integração com plataformas externas: Adicionar links diretos para as músicas, permitindo que os usuários sejam redirecionados ao *Spotify* para facilitar a experiência de reprodução.
- Exploração de novos dados: Há informações ainda não exploradas no processo de recomendação, como a data de lançamento da música, a presença de conteúdo explícito e a duração da música.

Referências Bibliográficas

ACHARYA, A.; SINGH, B.; ONOE, N. Llm based generation of item-description for recommendation system. *RecSys '23: Proceedings of the 17th ACM Conference on Recommender Systems*, 2023.

ANACONDA. *Anaconda | The Operating System for AI*. 2024. <<https://www.anaconda.com/>>. Acesso em: 24/11/2024.

APPLE. *Apple Music - Apple (BR)*. 2024. <<https://www.apple.com/br/apple-music/>>. Acesso em: 05/05/2024.

AWS. *Amazon EC2 tipos de instâncias - AWS*. 2023. <<https://aws.amazon.com/pt/ec2/instance-types/>>. Acesso em: 13/07/2024.

BROWN, S. *The C4 model for visualising software architecture*. 2024. <<https://c4model.com/>>. Acesso em: 07/05/2024.

CHEN, H. et al. Chatgpt's one-year anniversary: are open-source large language models catching up? *arXiv preprint arXiv:2311.16989*, 2023.

CREWAI. *crewAI - Platform for Multi AI Agents Systems*. 2024. <<https://www.crewai.com/>>. Acesso em: 13/07/2024.

CREWAI. *crewAI Agents - crewAI*. 2024. <<https://docs.crewai.com/core-concepts/Agents/#what-is-an-agent>>. Acesso em: 13/07/2024.

CREWAI. *crewAI Tasks - crewAI*. 2024. <<https://docs.crewai.com/core-concepts/Tasks/>>. Acesso em: 13/07/2024.

CRFM, S. *Stanford CRFM*. 2023. <<https://crfm.stanford.edu/2023/03/13/alpaca.html>>. Acesso em: 07/05/2024.

DAI s. et al. Uncovering chatgpt's capabilities in recommender systems. *RecSys '23: Proceedings of the 17th ACM Conference on Recommender Systems*, 2023.

DEEZER. *Deezer | Ouvir musica online | Aplicativo para escutar música*. 2024. <<https://www.deezer.com/br/>>. Acesso em: 05/05/2024.

DJANGO. *The web framework for perfectionists with deadlines | Django*. 2024. <<https://www.djangoproject.com/>>. Acesso em: 24/11/2024.

- ELAHI, M. et al. Hybrid recommendation by incorporating the sentiment of product reviews. *Informatics and Computer Science Intelligent Systems Applications An International Journal*, 2023.
- FIGMA. *Figma: a ferramenta de design de interface colaborativa*. 2024. <<https://www.figma.com/pt-br/>>. Acesso em: 25/11/2024.
- GOLDBERG, Y. *Neural network methods for natural language processing*. [S.l.]: Synthesis Lectures on Human Language Technologies, 2017.
- GOOGLE. *Gemini - Google DeepMind*. 2024. <<https://deepmind.google/technologies/gemini/#introduction>>. Acesso em: 07/05/2024.
- GOURLEY, D. et al. *HTTP: The Definitive Guide*. 1st. ed. [S.l.]: O'Reilly Media, 2002.
- HUA, W. et al. Tutorial on large language models for recommendation. *RecSys '23: Proceedings of the 17th ACM Conference on Recommender Systems*, 2023.
- INGELMO, V. et al. C4 model in a software engineering subject to ease the comprehension of uml and the software. In: *2020 IEEE Global Engineering Education Conference (EDUCON)*. [S.l.: s.n.], 2020. p. 919–924.
- JANNACH, D.; ZANKER, M.; FELFERNIG, A. *Recommender Systems: An Introduction*. 1st. ed. [S.l.]: Cambridge University Press, 2010.
- JIANG, A. Q. et al. Mistral 7B. arXiv, 2023.
- JIN, B.; SAHIN, S.; SHEVAT, A. *Designing Web APIs: Building APIs that Developers Love*. [S.l.]: O'Reilly Media, 2018.
- JJ, G.; JON, S. *API Design Patterns*. [S.l.]: Manning Publications Co., 2021.
- KUROSE, K. R. J. *Redes de computadores e a Internet: uma abordagem top-down*. 6th. ed. [S.l.]: Pearson Education do Brasil, 2013.
- LANGCHAIN. *Defining Custom Tools - LangChain*. 2024. <https://python.langchain.com/v0.1/docs/modules/tools/custom_tools/>. Acesso em: 13/07/2024.
- LANGCHAIN. *Defining Custom Tools | Langchain*. 2024. <https://python.langchain.com/v0.1/docs/modules/tools/custom_tools/>. Acesso em: 20/11/2024.
- LANGCHAIN. *GoogleAI - LangChain*. 2024. <https://python.langchain.com/v0.2/docs/integrations/llms/google_ai/>. Acesso em: 13/07/2024.
- LANGCHAIN. *Groq - LangChain*. 2024. <<https://python.langchain.com/v0.2/docs/integrations/chat/groq/>>. Acesso em: 13/07/2024.
- LAST.FM. *Página Inicial | Last.fm*. 2024. <<https://www.last.fm/pt/home>>. Acesso em: 05/05/2024.
- MCDONALD, D.; PAPADOPOULOS, R.; BENNINGFIELD, L. Reducing llm hallucination using knowledge distillation: A case study with mistral large and mmlu benchmark. *Authorea Preprints*, Authorea, 2024.

- META. *Meta Llama*. 2024. <<https://llama.meta.com/>>. Acesso em: 07/05/2024.
- META. *Meta Llama 3*. 2024. <<https://llama.meta.com/llama3/>>. Acesso em: 15/07/2024.
- MISTRAL. *Mistral AI | Frontier AI in your hands*. 2024. <<https://mistral.ai/>>. Acesso em: 07/05/2024.
- MONGODB. *Modelagem de dados | Manual do MongoDB v8.0*. 2024. <<https://www.mongodb.com/pt-br/docs/manual/data-modeling/#std-label-manual-data-modeling-intro>>. Acesso em: 21/11/2024.
- MONGODB. *MongoDB: a plataforma de dados para desenvolvedores | MongoDB*. 2024. <<https://www.mongodb.com/pt-br>>. Acesso em: 17/11/2024.
- NPM. *node-cron - npm*. 2024. <<https://www.npmjs.com/package/node-cron>>. Acesso em: 25/11/2024.
- OPENAI. *GPT3 powers the next generation of apps | OpenAI*. 2024. <<https://openai.com/index/gpt-3-apps>>. Acesso em: 07/05/2024.
- PANDAS. *pandas - Python Data Analysis Library*. 2024. <<https://pandas.pydata.org/>>. Acesso em: 13/07/2024.
- PEEPERKORN, M. et al. *Is Temperature the Creativity Parameter of Large Language Models?* 2024. Disponível em: <<https://arxiv.org/abs/2405.00492>>.
- POTDAR, A. M. et al. Performance evaluation of docker container and virtual machine. *Procedia Computer Science*, Elsevier, v. 171, p. 1419–1428, 2020.
- RUAN, J. et al. TPTU: Large language model-based AI agents for task planning and tool usage. arXiv, 2023.
- RUSSELL, J.; NORVIG, P. *Artificial Intelligence - A Modern Approach*. 4th. ed. [S.l.]: GEN LTC, 2022.
- SHARMA, L.; GERA, A. A survey of recommendation system: Research challenges. *International Journal of Engineering Trends and Technology (IJETT)*, v. 4, n. 5, p. 1989–1992, 2013.
- SPOTIFY. *Entenda as recomendações no Spotify*. 2024. <<https://www.spotify.com/br-pt/safetyandprivacy/understanding-recommendations>>. Acesso em: 05/05/2024.
- SPOTIFY. *Spotify Web Player*. 2024. <<https://open.spotify.com/intl-pt>>. Acesso em: 05/05/2024.
- TEAM, G. et al. *Gemini: A Family of Highly Capable Multimodal Models*. 2024. Disponível em: <<https://arxiv.org/abs/2312.11805>>.
- TEAM, G. et al. *Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context*. 2024. Disponível em: <<https://arxiv.org/abs/2403.05530>>.
- TIDAL. *TIDAL - High Fidelity Music Streaming*. 2024. <<https://tidal.com/>>. Acesso em: 05/05/2024.

TOUVRON, H. et al. LLaMA: Open and efficient foundation language models. arXiv, 2023.

WEIHS, C. et al. *Music Data Analysis: Foundations and Applications*. 1st. ed. [S.l.]: CRC Press, 2017.

WU, D. Music personalized recommendation system based on hybrid filtration. In: *2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*. [S.l.]: IEEE, 2019.

WU, L. et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 2024.

ZHAO, Z. et al. Recommender systems in the era of large language models (LLMs). *IEEE Trans. Knowl. Data Eng.*, Institute of Electrical and Electronics Engineers (IEEE), p. 1–20, 2024.

Apêndice A

Perfis de Usuários

Tabela A.1: Perfil Usuário 1

Gênero
Pop
Reggaeton
Art Pop
Funk Pop
Funk RJ

Tabela A.2: Perfil Usuário 2

Gênero
Pop
Dfw Rap
Art Pop
R&b Brasileiro
K-Pop

Tabela A.3: Perfil Usuário 3

Gênero
K-Pop
K-Pop Boy Group
Anime
Art Pop
K-Rap

Tabela A.4: Perfil Usuário 4

Gênero
Alternative Rock
Garage Rock
Hip hop
Alternative Metal
Funk Meta

Tabela A.5: Perfil Usuário 5

Gênero
Brooklyn Indie
Bedroom Pop
Pov: indie
Pagode
Pagode Novo

Tabela A.6: Perfil Usuário 6

Gênero
Baroque Pop
British Invasion
Pop
Indie Pop
R&b

Tabela A.7: Perfil Usuário 7

Gênero
Funk Carioca
Pagode
Brazilian Gospel
Axé
Arrocha

Tabela A.8: Perfil Usuário 8

Gênero
Dfw Rap
Melodic Rap
Trap Cristão
Forró
R&b brasileiro

Tabela A.9: Perfil Usuário 9

Gênero
Art Pop
Pop
Alternative Metal
Rap
Atl Hip Hop

Tabela A.10: Perfil Usuário 10

Gênero
Chicago Rap
R&b brasileiro
Rap Cearense
Conscious Hip Hop
Detroit Trap Brasileiro

Tabela A.11: Perfil Usuário 11

Gênero
Deathgrass
Conscious Hip Hop
Alternative Hip Hop
Rock-and-roll
Rockabilly

Tabela A.12: Perfil Usuário 12

Gênero
Album Rock
Rock-and-roll
Classic Rock
British Invasion
Grunge

Tabela A.13: Perfil Usuário 13

Gênero
Alternative Metal
Nu Metal
Pop
Post-grunge
Indie Folk

Apêndice B

Avaliações dos Usuários

Tabela B.1: Avaliação Usuário 1 (Gemini)

Música	Gênero	Artista	Avaliação
Blinding Lights	Pop	The Weeknd	Curtida
Watermelon Sugar	Pop	Harry Styles	Curtida
Born This Way	Art Pop	Lady Gaga	Curtida
Applause	Art Pop	Lady Gaga	Curtida
Sweater Weather	Alternative Pop Rock	The Neighbourhood	Curtida
Riptide	Alternative Pop Rock	Vance Joy	Curtida
Chop Suey!	Alternative Metal	System of a Down	Curtida
Toxicity	Alternative Metal	System of a Down	Curtida
Down with the Sickness	Nu Metal	Disturbed	Não Curtida
Freak on a Leash	Nu Metal	Korn	Não Curtida

Tabela B.2: Avaliação Usuário 1 (Llama)

Música	Gênero	Artista	Avaliação
Happy	Pop	Pharrell Williams	Curtida
Can't Stop the Feeling!	Pop	Justin Timberlake	Curtida
Uptown Funk	Dance Pop	Mark Ronson ft. Bruno Mars	Curtida
SexyBack	Dance Pop	Justin Timberlake	Curtida
Mr. Brightside	Alternative Rock	The Killers	Curtida
Best Of You	Alternative Rock	Foo Fighters	Curtida
Mr. Tambourine Man	Indie Pop	The Byrds	Curtida
Here Comes the Sun	Indie Pop	The Beatles	Curtida
Tsunami	Electronic Dance Music	Dash Berlin	Não Curtida
One	Electronic Dance Music	Veracocho	Não Curtida

Tabela B.3: Avaliação Usuário 1 (Mistral)

Música	Gênero	Artista	Avaliação
Blue Suede Shoes	Rock	Elvis Presley	Curtida
Hey Jude	Rock	The Beatles	Curtida
Forget You	Pop	CeeLo Green	Curtida
Hit the Road Jack	Jazz	Ray Charles	Curtida
Elevators (Me & You)	Hip Hop/Rap	OutKast	Não Curtida
B.O.B.	Hip Hop/Rap	OutKast	Não Curtida
Green Onions	R&B/Soul	Booker T. & the MG's	Não Curtida
Hold On, I'm Comin'	R&B/Soul	Sam & Dave	Não Curtida
Bright Lights Bigger City	Pop	CeeLo Green	Não Curtida
Georgia on My Mind	Jazz	Ray Charles	Não Curtida

Tabela B.4: Avaliação Usuário 2 (Gemini)

Música	Gênero	Artista	Avaliação
Seven Nation Army	Rock	The White Stripes	Curtida
Smells Like Teen Spirit	Rock	Nirvana	Curtida
Shape of You	Pop	Ed Sheeran	Curtida
Bad Romance	Pop	Lady Gaga	Curtida
Lose Yourself	Hip-Hop	Eminem	Curtida
Till I Collapse	Hip-Hop	Eminem	Curtida
Levels	Electronic	Avicii	Curtida
Wake Me Up	Electronic	Avicii	Curtida
Friends in Low Places	Country	Garth Brooks	Não Curtida
Man! I Feel Like a Woman!	Country	Shania Twain	Não Curtida

Tabela B.5: Avaliação Usuário 2 (Llama)

Música	Gênero	Artista	Avaliação
Happy	pop	Pharrell Williams	Curtida
Uptown Funk	pop	Mark Ronson ft. Bruno Mars	Curtida
Can't Stop the Feeling!	funk pop	Justin Timberlake	Curtida
Mr. Brightside	alternative rock	The Killers	Curtida
SexyBack	funk pop	Justin Timberlake	Curtida
Best of You	alternative rock	Foo Fighters	Curtida
Song 3	pop	Artist X	Não Curtida
Song 11	edm	Artist X	Não Curtida
Tsunami	edm	DVBBS & Borgeous	Não Curtida
Scary Monsters and Nice Sprites	edm	Skrillex	Não Curtida

Tabela B.6: Avaliação Usuário 3 (Gemini)

Música	Gênero	Artista	Avaliação
Love Dive	K-pop	IVE	Curtida
Tomboy	K-pop	(G)I-DLE	Curtida
As It Was	Pop	Harry Styles	Curtida
Stay	Pop	The Kid LAROI, Justin Bieber	Curtida
Blinding Lights	Art pop	The Weeknd	Curtida
Levitating	Art pop	Dua Lipa	Curtida
GANADARA	K-rap	Jay Park	Curtida
Celebrity	K-rap	IU	Curtida
Chop Suey!	Alternative metal	System of a Down	Não Curtida
Toxicity	Alternative metal	System of a Down	Não Curtida

Tabela B.7: Avaliação Usuário 3 (Llama)

Música	Gênero	Artista	Avaliação
High Hopes	art pop	Panic! At The Disco	Curtida
7 Rings	art pop	Ariana Grande	Curtida
Bad Guy	pop	Billie Eilish	Curtida
Yes or Yes	pop	TWICE	Curtida
Kill This Love	k-pop	BLACKPINK	Curtida
Fancy	k-pop	TWICE	Curtida
Airplane Pt. 2	k-pop boy group	BTS	Curtida
Not Today	k-pop boy group	BTS	Curtida
Mo Bamba	rap	Sheck Wes	Não Curtida
Sicko Mode	rap	Travis Scott	Não Curtida

Tabela B.8: Avaliação Usuário 3 (Mistral)

Música	Gênero	Artista	Avaliação
Bad Guy	Pop	Billie Eilish	Curtida
Bohemian Rhapsody	Rock	Queen	Curtida
Sweet Child O' Mine	Rock	Guns N' Roses	Curtida
God's Plan	Hip Hop	Drake	Curtida
So What	Jazz	Miles Davis	Curtida
Take Five	Jazz	Dave Brubeck	Curtida
Fur Elise	Classical	Ludwig van Beethoven	Curtida
Moonlight Sonata	Classical	Ludwig van Beethoven	Curtida
Dance Monkey	Pop	Tones and I	Não Curtida
Sicko Mode	Hip Hop	Travis Scott	Não Curtida

Tabela B.9: Avaliação Usuário 4 (Gemini)

Música	Gênero	Artista	Avaliação
I Wanna Be Your Dog	garage rock	The Stooges	Curtida
Louie Louie	garage rock	The Kingsmen	Curtida
Smells Like Teen Spirit	modern rock	Nirvana	Curtida
Basket Case	modern rock	Green Day	Curtida
Maps	permanent wave revival	Yeah Yeah Yeahs	Curtida
Take Me Out	permanent wave revival	Franz Ferdinand	Curtida
Creep	alternative rock	Radiohead	Curtida
Wonderwall	alternative rock	Oasis	Curtida
Bohemian Rhapsody	rock	Queen	Curtida
Sweet Child O' Mine	rock	Guns N' Roses	Curtida

Tabela B.10: Avaliação Usuário 4 (Llama)

Música	Gênero	Artista	Avaliação
Lose Yourself	Hip hop	Eminem	Curtida
Juicy	Hip hop	The Notorious B.I.G.	Curtida
SexyBack	R&B	Justin Timberlake	Curtida
Crazy in Love	R&B	Beyoncé ft. Jay-Z	Curtida
Song 3	Electronic	Artist 3	Não Curtida
Song 9	Electronic	Artist 9	Não Curtida
Uptown Funk	Pop	Mark Ronson ft. Bruno Mars	Não Curtida
Happy	Pop	Pharrell Williams	Não Curtida
Sandstorm	Electronic	Darude	Não Curtida
One	Electronic	Veracocha	Não Curtida

Tabela B.11: Avaliação Usuário 4 (Mistral)

Música	Gênero	Artista	Avaliação
Dance Monkey	Pop	Tones and I	Curtida
Circles	Pop	Post Malone	Curtida
Don't Start Now	R&B	Dua Lipa	Curtida
Goosebumps	R&B	Travis Scott	Curtida
Heat Waves	Rock	Glass Animals	Curtida
All The Good Girls Go To Hell	Rock	Billie Eilish	Curtida
Blinding Lights	Dance	The Weeknd	Curtida
God's Plan	Hip-Hop	Drake	Não Curtida
Sicko Mode	Hip-Hop	Travis Scott	Não Curtida
Rain On Me	Dance	Lady Gaga, Ariana Grande	Não Curtida

Tabela B.12: Avaliação Usuário 5 (Gemini)

Música	Gênero	Artista	Avaliação
Seven Nation Army	Rock	The White Stripes	Curtida
Smells Like Teen Spirit	Rock	Nirvana	Curtida
Shape of You	Pop	Ed Sheeran	Curtida
Lose Yourself	Hip-Hop	Eminem	Curtida
Till I Collapse	Hip-Hop	Eminem	Curtida
Man! I Feel Like a Woman!	Country	Shania Twain	Curtida
Levels	Electronic	Avicii	Curtida
Wake Me Up	Electronic	Avicii	Curtida
Despacito	Pop	Luis Fonsi	Não Curtida
Friends in Low Places	Country	Garth Brooks	Não Curtida

Tabela B.13: Avaliação Usuário 5 (Llama)

Música	Gênero	Artista	Avaliação
The National - Bloodbuzz Ohio	brooklyn indie	The National	Curtida
The Strokes - Last Nite	brooklyn indie	The Strokes	Curtida
Arctic Monkeys - Do I Wanna Know?	pov: indie	Arctic Monkeys	Curtida
The 1975 - Give Yourself a Try	pov: indie	The 1975	Curtida
Clairo - Pretty Girl	bedroom pop	Clairo	Curtida
Lana Del Rey - Video Games	bedroom pop	Lana Del Rey	Curtida
M83 - Midnight City	dream pop	M83	Curtida
Cocteau Twins - Cherry-Coloured Funk	dream pop	Cocteau Twins	Curtida

Tabela B.14: Avaliação Usuário 5 (Mistral)

Música	Gênero	Artista	Avaliação
7 Rings	Pop	Ariana Grande	Curtida
Thank U, Next	Pop	Ariana Grande	Curtida
Senorita	R&B	Shawn Mendes, Camila Cabello	Curtida
Havana	R&B	Camila Cabello	Curtida
Sicko Mode	Hip Hop	Travis Scott	Curtida
Nice for What	Hip Hop	Drake	Curtida
God's Plan	Hip Hop	Drake	Curtida
I Don't Care	Dance	Ed Sheeran, Justin Bieber	Curtida
Sunflower	Dance	Post Malone, Swae Lee	Curtida
Bad Guy	Pop	Billie Eilish	Curtida

Tabela B.15: Avaliação Usuário 6 (Gemini)

Música	Gênero	Artista	Avaliação
The Less I Know the Better	Indie pop	Tame Impala	Curtida
Closer	Pop	The Chainsmokers	Curtida
Radioactive	Alternative pop rock	Imagine Dragons	Curtida
Counting Stars	Alternative pop rock	OneRepublic	Curtida
Crossroads	Blues	Robert Johnson	Curtida
Hound Dog	Blues	Elvis Presley	Curtida
Louie Louie	Classic garage rock	The Kingsmen	Curtida
Sweater Weather	Indie pop	The Neighbourhood	Não Curtida
Shape of You	Pop	Ed Sheeran	Não Curtida
Gloria	Classic garage rock	The Shadows	Não Curtida

Tabela B.16: Avaliação Usuário 6 (Llama)

Música	Gênero	Artista	Avaliação
Electric Feel	indie pop	MGMT	Curtida
Mr. Brightside	indie pop	The Killers	Curtida
Skinny Love	pov: indie	Bon Iver	Curtida
Home	pov: indie	Edward Sharpe & The Magnetic Zeros	Curtida
Wagon Wheel	pov: indie	Old Crow Medicine Show	Curtida
I Want You Back	classic garage rock	The Jackson 5	Curtida
Song 1	classic garage rock	Artist 1	Não Curtida
Song 11	classic garage rock	Artist 11	Não Curtida

Tabela B.17: Avaliação Usuário 6 (Mistral)

Música	Gênero	Artista	Avaliação
Pride and Joy	Blues Rock	Stevie Ray Vaughan	Curtida
The Sky is Crying	Blues	Stevie Ray Vaughan	Curtida
Knockin' on Heaven's Door	Folk Rock	Bob Dylan	Curtida
Mr. Tambourine Man	Folk Rock	Bob Dylan	Curtida
I Can't Quit You Baby	Blues Rock	Led Zeppelin	Curtida
You Shook Me	Blues	Led Zeppelin	Curtida
Don't Think Twice, It's All Right	Folk	Bob Dylan	Curtida
Tears in Heaven	Blues Rock	Eric Clapton	Curtida
Stormy Monday	Blues	The Allman Brothers Band	Curtida
The Weight	Folk Rock	The Band	Curtida
Maggie's Farm	Folk Rock	Bob Dylan	Curtida
Going Down Slow	Blues	Howlin' Wolf	Curtida

Tabela B.18: Avaliação Usuário 7 (Gemini)

Música	Gênero	Artista	Avaliação
I Want It That Way	Pop	Backstreet Boys	Curtida
Sweet Child O' Mine	Rock	Guns N' Roses	Não Curtida
Enter Sandman	Rock	Metallica	Não Curtida
Lose Yourself	Hip-Hop	Eminem	Não Curtida
Toxic	Pop	Britney Spears	Não Curtida
Wake Me Up	Electronic	Avicii	Não Curtida
Levels	Electronic	Avicii	Não Curtida
Man! I Feel Like a Woman!	Country	Shania Twain	Não Curtida
Friends in Low Places	Country	Garth Brooks	Não Curtida
Juicy	Hip-Hop	The Notorious B.I.G.	Não Curtida

Tabela B.19: Avaliação Usuário 7 (Llama)

Música	Gênero	Artista	Avaliação
SexyBack	pop	Justin Timberlake	Curtida
We Found Love	pop	Rihanna ft. Calvin Harris	Não Curtida
Shut Up and Dance	pop	Walk the Moon	Não Curtida
All About That Bass	pop	Meghan Trainor	Não Curtida
Blank Space	pop	Taylor Swift	Não Curtida
Bad Guy	pop	Billie Eilish	Não Curtida
Senorita	pop	Shawn Mendes and Camila Cabello	Não Curtida
Perfect	pop	Ed Sheeran	Não Curtida
Sorry	pop	Justin Bieber	Não Curtida
Love Yourself	pop	Justin Bieber	Não Curtida

Tabela B.20: Avaliação Usuário 7 (Mistral)

Música	Gênero	Artista	Avaliação
Bad Guy	Pop	Billie Eilish	Curtida
Ai Se Eu Te Pego	Sertanejo	Michel Teló	Curtida
Shape of You	Pop	Ed Sheeran	Não Curtida
Bola de Meia, Balaca	Funk carioca	Tuta & MC Cabelinho	Não Curtida
Vou Ficar na Lua	Funk carioca	Djonga & Mc MM	Não Curtida
O Xote das Meninas	Forro	Jackson do Pandeiro	Não Curtida
Meu Menino	Arrocha	Gaby Amarantos	Não Curtida
Forró do Muíño	Forro	Luiz Gonzaga	Não Curtida
Tudo Que Eu Quero	Sertanejo	Zé Neto & Cristiano	Não Curtida
Amor Perfeito	Arrocha	Beth Carvalho	Não Curtida

Tabela B.21: Avaliação Usuário 8 (Llama)

Música	Gênero	Artista	Avaliação
Stairway to Heaven	piano rock	Led Zeppelin	Curtida
Lose Yourself	uk alternative hip hop	Eminem	Curtida
Guren no Yumiya	anime	Linked Horizon	Curtida
Samba de Uma Nota Só	forro	Baden Powell	Não Curtida
Aquarela do Brasil	forro	Antônio Carlos Jobim	Não Curtida
Funk Carioca	funk carioca	MC Fioti	Não Curtida
Vai Malandra	funk carioca	Pablo Vittar	Não Curtida
Hotel California	piano rock	Eagles	Não Curtida
Ultralight Beam	uk alternative hip hop	Kanye West	Não Curtida
Crossing Field	anime	LisA	Não Curtida

Tabela B.22: Avaliação Usuário 9 (Gemini)

Música	Gênero	Artista	Avaliação
The Sound of Silence	Hardwave	Disturbed	Curtida
Radioactive	Hardwave	Imagine Dragons	Curtida
Weightless	Ambient lo-fi	Marconi Union	Curtida
Clair de Lune	Ambient lo-fi	Claude Debussy	Curtida
Strawberry Swing	Art pop	Coldplay	Curtida
Holocene	Art pop	Bon Iver	Curtida
Shape of You	Pop	Ed Sheeran	Curtida
Hedwig's Theme	Orchestral soundtrack	John Williams	Curtida
Star Wars Main Theme	Orchestral soundtrack	John Williams	Curtida
Despacito	Pop	Luis Fonsi	Não Curtida

Tabela B.23: Avaliação Usuário 9 (Llama)

Música	Gênero	Artista	Avaliação
Weightless	ambient lo-fi	Marconi Union	Curtida
Get You	art pop	Daniel Caesar	Curtida
Breaking the Habit	alternative metal	Linkin Park	Curtida
Rainy Days	ambient lo-fi	Jinsang	Não Curtida
Electric Feel	art pop	MGMT	Não Curtida
Tetris	hardwave	Laserdance	Não Curtida
Song Title 1	Genre 1	Artist Name 1	Não Curtida
Space Age Whiz Kids	hardwave	Laserdance	Não Curtida
Samba de Uma Nota Só	musica capixaba	Baden Powell	Não Curtida
Last Resort	alternative metal	Papa Roach	Não Curtida

Tabela B.24: Avaliação Usuário 9 (Mistral)

Música	Gênero	Artista	Avaliação
Leave The Door Open	Jazz	Silk Sonic	Curtida
Head & Heart	Jazz	Joel Corry, MNEK	Curtida
Don't Start Now	Pop	Dua Lipa	Curtida
Dance Monkey	Pop	Tones and I	Curtida
Up	Hip-Hop	Cardi B	Curtida
Last Christmas	Rock	Wham!	Não Curtida

Tabela B.25: Avaliação Usuário 10 (Gemini)

Música	Gênero	Artista	Avaliação
Me Sinto Abençoado	Trap brasileiro	MC Cabelinho	Curtida
Rei Lacoste	Trap brasileiro	MC Cabelinho	Curtida
Flor de Lis	R&b brasileiro	Djavan	Curtida
All Caps	Chicago rap	Madlib	Curtida
The World Is Yours	Chicago rap	Nas	Curtida
Rap do Silva	Rap cearense	Don L	Não Curtida
O Sol	Rap cearense	Don L	Não Curtida
Me Espera	R&b brasileiro	Djavan	Não Curtida
The Message	Hip hop	Grandmaster Flash and The Furious Five	Não Curtida
Rapper's Delight	Hip hop	The Sugarhill Gang	Não Curtida

Tabela B.26: Avaliação Usuário 10 (Llama)

Música	Gênero	Artista	Avaliação
Lose Yourself	Hip-Hop	Eminem	Curtida
God's Plan	Hip-Hop	Drake	Curtida
Lose Yourself	Rap	Eminem	Curtida
Not Afraid	Rap	Eminem	Curtida
Happy	Pop	Pharrell Williams	Curtida
Can't Stop the Feeling!	Pop	Justin Timberlake	Curtida
Sweet Child O' Mine	Rock	Guns N' Roses	Curtida
Hotel California	Rock	Eagles	Curtida
Uptown Funk	Electronic	Mark Ronson ft. Bruno Mars	Curtida
Pumped Up Kicks	Electronic	Foster the People	Curtida

Tabela B.27: Avaliação Usuário 10 (Mistral)

Música	Gênero	Artista	Avaliação
Shape of You	Pop	Ed Sheeran	Curtida
Bad Guy	Pop	Billie Eilish	Curtida
Bohemian Rhapsody	Rock	Queen	Curtida
Sweet Child O' Mine	Rock	Guns N' Roses	Curtida
Uptown Funk	Funk	Mark Ronson ft. Bruno Mars	Curtida
Get Lucky	Funk	Daft Punk ft. Pharrell Williams	Curtida
Encore une fois	Electronic	Daft Punk	Curtida
Take Me Out	Alternative	Franz Ferdinand	Curtida
Ai Se Eu Te Pego	Sertanejo Universitário	Michel Teló	Não Curtida
Tudo Que Você Quiser	Sertanejo Universitário	Henrique e Juliano	Não Curtida
Take Five	Jazz	Dave Brubeck	Não Curtida
So What	Jazz	Miles Davis	Não Curtida

Tabela B.28: Avaliação Usuário 11 (Gemini)

Música	Gênero	Artista	Avaliação
Seven Nation Army	Rock	The White Stripes	Curtida
Smells Like Teen Spirit	Rock	Nirvana	Curtida
Shape of You	Pop	Ed Sheeran	Curtida
Lose Yourself	Hip-Hop	Eminem	Curtida
Till I Collapse	Hip-Hop	Eminem	Curtida
Wake Me Up	Electronic	Avicii	Curtida
Despacito	Pop	Luis Fonsi	Não Curtida
Friends in Low Places	Country	Garth Brooks	Não Curtida
Man! I Feel Like a Woman!	Country	Shania Twain	Não Curtida
Levels	Electronic	Avicii	Não Curtida

Tabela B.29: Avaliação Usuário 11 (Llama)

Música	Gênero	Artista	Avaliação
All the Small Things	pop punk	Blink-182	Curtida
First Date	pop punk	Blink-182	Curtida
Sweet Child O' Mine	rock	Guns N' Roses	Curtida
Livin' on a Prayer	rock	Bon Jovi	Curtida
Juicy	hip hop	The Notorious B.I.G.	Curtida
Song 3	Alternative Rock	Artist 3	Não Curtida
Song 8	Alternative Rock	Artist 8	Não Curtida
I'm Not Okay (I Promise)	emo	My Chemical Romance	Não Curtida
The Middle	emo	Jimmy Eat World	Não Curtida
Regulate	hip hop	Warren G ft. Nate Dogg	Não Curtida

Tabela B.30: Avaliação Usuário 12 (Gemini)

Música	Gênero	Artista	Avaliação
Enter Sandman	Rock	Metallica	Curtida
Lose Yourself	Hip-Hop	Eminem	Curtida
Sweet Child O' Mine	Rock	Guns N' Roses	Não Curtida
I Want It That Way	Pop	Backstreet Boys	Não Curtida
Toxic	Pop	Britney Spears	Não Curtida
Juicy	Hip-Hop	The Notorious B.I.G.	Não Curtida
Friends in Low Places	Country	Garth Brooks	Não Curtida
Man! I Feel Like a Woman!	Country	Shania Twain	Não Curtida
Levels	Electronic	Avicii	Não Curtida
Wake Me Up	Electronic	Avicii	Não Curtida

Tabela B.31: Avaliação Usuário 12 (Llama)

Música	Gênero	Artista	Avaliação
Stairway to Heaven	Classic Rock	Led Zeppelin	Curtida
Hotel California	Classic Rock	Eagles	Curtida
Livin' on a Prayer	Rock	Bon Jovi	Curtida
Smells Like Teen Spirit	Alternative Rock	Nirvana	Curtida
Creep	Alternative Rock	Radiohead	Curtida
Imagine	Album Rock	John Lennon	Curtida
Barracuda	Hard Rock	Heart	Curtida
Sweet Child O' Mine	Rock	Guns N' Roses	Não Curtida
Bohemian Rhapsody	Album Rock	Queen	Não Curtida
Back in Black	Hard Rock	AC/DC	Não Curtida

Tabela B.32: Avaliação Usuário 12 (Mistral)

Música	Gênero	Artista	Avaliação
I Wanna Dance with Somebody	Pop	Whitney Houston	Curtida
Poker Face	Electronic	Lady Gaga	Curtida
Sweet Child O' Mine	Rock	Guns N' Roses	Não Curtida
Electricity	Electronic	Silk City, Dua Lipa	Não Curtida
Take Five	Jazz	Dave Brubeck	Não Curtida
Dance Monkey	Pop	Tones and I	Não Curtida
Bad Guy	Pop	Billie Eilish	Não Curtida
God's Plan	Hip Hop	Drake	Não Curtida
So What	Jazz	Miles Davis	Não Curtida
Bohemian Rhapsody	Rock	Queen	Não Curtida

Tabela B.33: Avaliação Usuário 13 (Gemini)

Música	Gênero	Artista	Avaliação
Shape of You	Pop	Ed Sheeran	Curtida
Despacito	Pop	Luis Fonsi	Curtida
Chop Suey!	Alternative metal	System of a Down	Curtida
Toxicity	Alternative metal	System of a Down	Curtida
Down with the Sickness	Nu metal	Disturbed	Curtida
Freak on a Leash	Nu metal	Korn	Curtida
Black Hole Sun	Post-grunge	Soundgarden	Curtida
Zombie	Post-grunge	The Cranberries	Curtida
DDU-DU DDU-DU	K-pop girl group	Blackpink	Curtida
Kill This Love	K-pop girl group	Blackpink	Curtida

Tabela B.34: Avaliação Usuário 13 (Llama)

Música	Gênero	Artista	Avaliação
Enter Sandman	rock	Metallica	Curtida
Bulls on Parade	rock	Rage Against the Machine	Curtida
Last Resort	alternative metal	Papa Roach	Curtida
Breaking the Habit	alternative metal	Linkin Park	Curtida
Uptown Funk	pop	Mark Ronson ft. Bruno Mars	Curtida
Can't Stop the Feeling!	pop	Justin Timberlake	Curtida
Nekomata	anime lo-fi	Yume1000	Curtida
Hollow	anime lo-fi	Yume1000	Curtida
Kill This Love	k-pop girl group	BLACKPINK	Curtida
Fancy	k-pop girl group	TWICE	Curtida

Apêndice C

Avaliações dos Usuários - Itens Relevantes

Tabela C.1: Avaliação Usuário 1 - Itens Relevantes (Gemini)

Música	Gênero	Artista	Avaliação
Shape of You	pop	Ed Sheeran	Curtida
Despacito	pop	Luis Fonsi	Curtida
Ginza	reggaeton	J Balvin	Curtida
Sweater Weather	indie pop	The Neighbourhood	Curtida
Holocene	indie pop	Bon Iver	Curtida
Counting Stars	pop rock	OneRepublic	Curtida
Radioactive	pop rock	Imagine Dragons	Curtida
Call Me Maybe	electropop	Carly Rae Jepsen	Curtida
I Gotta Feeling	electropop	The Black Eyed Peas	Curtida
Dura	reggaeton	Daddy Yankee	Não Curtida

Tabela C.2: Avaliação Usuário 1 - Itens Relevantes (Llama)

Música	Gênero	Artista	Avaliação
Blank Space	pop	Taylor Swift	Curtida
Shake It Off	pop	Taylor Swift	Curtida
Despacito	reggaeton	Luis Fonsi ft. Daddy Yankee	Curtida
Mr. Brightside	indie pop	The Killers	Curtida
Pumped Up Kicks	indie pop	Foster the People	Curtida
Sugar	pop rock	Maroon 5	Curtida
Best Day of My Life	pop rock	American Authors	Curtida
Tik Tok	electropop	Ke\$ha	Curtida
We Found Love	electropop	Rihanna ft. Calvin Harris	Curtida
Vaina Loca	reggaeton	Ozuna	Não Curtida

Tabela C.3: Avaliação Usuário 1 - Itens Relevantes (Mistral)

Música	Gênero	Artista	Avaliação
Shape of You	pop	Ed Sheeran	Curtida
Bad Guy	pop	Billie Eilish	Curtida
Despacito	reggaeton	Luis Fonsi, Daddy Yankee	Curtida
Dákiti	reggaeton	Bad Bunny, Jhay Cortez	Curtida
Viva la Vida	pop rock	Coldplay	Curtida
High Hopes	pop rock	Panic! at the Disco	Curtida
Electric Feel	electropop	MGMT	Curtida
Sweet Dreams (Are Made of This)	electropop	Eurythmics	Curtida
Supalonely	indie pop	BENEE, Gus Dapperton	Não Curtida
Pretty Girl	indie pop	Clairo	Não Curtida

Tabela C.4: Avaliação Usuário 2 - Itens Relevantes (Gemini)

Música	Gênero	Artista	Avaliação
Shape of You	pop	Ed Sheeran	Curtida
Despacito	pop	Luis Fonsi	Curtida
Sicko Mode	dfw rap	Travis Scott	Curtida
God's Plan	hip hop	Drake	Curtida
HUMBLE.	hip hop	Kendrick Lamar	Curtida
Closer	dance pop	The Chainsmokers	Curtida
Starboy	dance pop	The Weeknd	Curtida
Old Town Road	rap	Lil Nas X	Curtida
Mo Bamba	dfw rap	Sheck Wes	Não Curtida
Bodak Yellow	rap	Cardi B	Não Curtida

Tabela C.5: Avaliação Usuário 3 - Itens Relevantes (Gemini)

Música	Gênero	Artista	Avaliação
Dynamite	k-pop	BTS	Curtida
Butter	k-pop	BTS	Curtida
Holocene	indie pop	Bon Iver	Curtida
Shallow	soundtrack	Lady Gaga	Curtida
Bohemian Rhapsody	soundtrack	Queen	Curtida
Wonderwall	pop rock	Oasis	Curtida
Yellow	pop rock	Coldplay	Curtida
Shape of You	pop	Ed Sheeran	Não Curtida
Perfect	pop	Ed Sheeran	Não Curtida
Skinny Love	indie pop	Bon Iver	Não Curtida

Tabela C.6: Avaliação Usuário 3 - Itens Relevantes (Llama)

Música	Gênero	Artista	Avaliação
Blank Space	pop	Taylor Swift	Curtida
Shake It Off	pop	Taylor Swift	Curtida
Boy With Luv	k-pop	BTS ft. Halsey	Curtida
Kill This Love	k-pop	BLACKPINK	Curtida
Mr. Brightside	indie pop	The Killers	Curtida
A Thousand Years	soundtrack	Christina Perri	Curtida
Let It Go	soundtrack	Idina Menzel	Curtida
We Found Love	pop rock	Rihanna ft. Calvin Harris	Curtida
SexyBack	pop rock	Justin Timberlake	Curtida
Pumped Up Kicks	indie pop	Foster the People	Não Curtida

Tabela C.7: Avaliação Usuário 3 - Itens Relevantes (Mistral)

Música	Gênero	Artista	Avaliação
Bad Guy	pop	Billie Eilish	Curtida
Dynamite	k-pop	BTS	Curtida
Gangnam Style	k-pop	PSY	Curtida
Titanium	indie pop	David Guetta ft. Sia	Curtida
Work	indie pop	Rihanna ft. Drake	Curtida
Let It Go	soundtrack	Idina Menzel	Curtida
Shallow	soundtrack	Lady Gaga & Bradley Cooper	Curtida
Sweet Child O' Mine	pop rock	Guns N' Roses	Curtida
Don't Stop Believin'	pop rock	Journey	Curtida
Shape of You	pop	Ed Sheeran	Não Curtida

Tabela C.8: Avaliação Usuário 4 - Itens Relevantes (Gemini)

Música	Gênero	Artista	Avaliação
Do I Wanna Know?	indie rock	Arctic Monkeys	Curtida
Mr. Brightside	indie rock	The Killers	Curtida
Smells Like Teen Spirit	alternative rock	Nirvana	Curtida
Wonderwall	alternative rock	Oasis	Curtida
Lucy in the Sky with Diamonds	psychedelic rock	The Beatles	Curtida
Strawberry Fields Forever	psychedelic rock	The Beatles	Curtida
O Tempo Não Para	brazilian rock	Cazuza	Curtida
Pais e Filhos	brazilian rock	Legião Urbana	Curtida
Lose Yourself	rap	Eminem	Curtida
Stan	rap	Eminem	Curtida

Tabela C.9: Avaliação Usuário 4 - Itens Relevantes (Llama)

Música	Gênero	Artista	Avaliação
Mr. Brightside	indie rock	The Killers	Curtida
Best Of You	indie rock	Foo Fighters	Curtida
Smells Like Teen Spirit	alternative rock	Nirvana	Curtida
Everlong	alternative rock	Foo Fighters	Curtida
White Rabbit	psychedelic rock	Jefferson Airplane	Curtida
Break On Through (To the Other Side)	psychedelic rock	The Doors	Curtida
Lose Yourself	rap	Eminem	Curtida
Juicy	rap	The Notorious B.I.G.	Curtida
Tangerina	brazilian rock	Titãs	Não Curtida
Samba Pa Ti	brazilian rock	Raul Seixas	Não Curtida

Tabela C.10: Avaliação Usuário 4 - Itens Relevantes (Mistral)

Música	Gênero	Artista	Avaliação
Creep	alternative rock	Radiohead	Curtida
High and Dry	alternative rock	Radiohead	Curtida
Lucy in the Sky with Diamonds	psychedelic rock	The Beatles	Curtida
Strawberry Fields Forever	psychedelic rock	The Beatles	Curtida
Aguas de Março	brazilian rock	Elis Regina	Curtida
Rap God	rap	Eminem	Curtida
Lose Yourself	rap	Eminem	Curtida
Holocene	indie rock	Bon Iver	Não Curtida
Flume	indie rock	Bon Iver	Não Curtida
Tropicália	brazilian rock	Os Mutantes	Não Curtida

Tabela C.11: Avaliação Usuário 5 - Itens Relevantes (Gemini)

Música	Gênero	Artista	Avaliação
Take Five	jazz	Dave Brubeck	Curtida
So What	jazz	Miles Davis	Curtida
Stormy Monday	jazz blues	T-Bone Walker	Curtida
The Thrill Is Gone	jazz blues	B.B. King	Curtida
Mr. Brightside	indie rock	The Killers	Curtida
Float On	indie rock	Modest Mouse	Curtida
I Gotta Feeling	pop	The Black Eyed Peas	Curtida
Bad Romance	pop	Lady Gaga	Curtida
Don't Stop Believin'	pop rock	Journey	Curtida
Livin' on a Prayer	pop rock	Bon Jovi	Curtida

Tabela C.12: Avaliação Usuário 5 - Itens Relevantes (Llama)

Música	Gênero	Artista	Avaliação
Take Five	jazz	The Dave Brubeck Quartet	Curtida
My Funny Valentine	jazz	Chet Baker	Curtida
Stormy Monday	jazz blues	T-Bone Walker	Curtida
The Thrill Is Gone	jazz blues	B.B. King	Curtida
Mr. Brightside	indie rock	The Killers	Curtida
Viva la Vida	indie rock	Coldplay	Curtida
Happy	pop	Pharrell Williams	Curtida
Uptown Funk	pop	Mark Ronson ft. Bruno Mars	Curtida
We Are Young	pop rock	Fun. ft. Janelle Monáe	Curtida
Counting Stars	pop rock	OneRepublic	Curtida

Tabela C.13: Avaliação Usuário 5 - Itens Relevantes (Mistral)

Música	Gênero	Artista	Avaliação
So What	jazz	Miles Davis	Curtida
All Of Me	jazz	Johnny Hodges	Curtida
St. Louis Blues	jazz blues	Louis Armstrong	Curtida
The Thrill Is Gone	jazz blues	B.B. King	Curtida
Season 2 Episode 3	indie rock	Glass Animals	Curtida
Chocolate	indie rock	The 1975	Curtida
Shape Of You	pop	Ed Sheeran	Curtida
Bad Guy	pop	Billie Eilish	Curtida
Mr. Brightside	pop rock	The Killers	Curtida
High Hopes	pop rock	Panic! At The Disco	Curtida

Tabela C.14: Avaliação Usuário 6 - Itens Relevantes (Gemini)

Música	Gênero	Artista	Avaliação
I Get Around	vocal harmony group	The Beach Boys	Curtida
God Only Knows	vocal harmony group	The Beach Boys	Curtida
Mr. Brightside	indie rock	The Killers	Curtida
Somebody Told Me	indie rock	The Killers	Curtida
Mais Uma Vez	pop rock brasileiro	Lulu Santos	Curtida
Tempo Perdido	pop rock brasileiro	Legião Urbana	Curtida
Smells Like Teen Spirit	alternative rock	Nirvana	Curtida
Lithium	alternative rock	Nirvana	Curtida
Can't Stop the Feeling!	dance pop	Justin Timberlake	Não Curtida
Uptown Funk	dance pop	Mark Ronson ft. Bruno Mars	Não Curtida

Tabela C.15: Avaliação Usuário 6 - Itens Relevantes (Llama)

Música	Gênero	Artista	Avaliação
The Nearness of You	vocal jazz	Billie Holiday	Curtida
My Funny Valentine	vocal jazz	Chet Baker	Curtida
Uptown Funk	pop	Mark Ronson ft. Bruno Mars	Curtida
Boy With Luv	k-pop	BTS ft. Halsey	Curtida
Mas Que Nada	samba	Sergio Mendes & Brasil '66	Curtida
The Girl from Ipanema	samba	Stan Getz & Astrud Gilberto	Curtida
White Rabbit	psychedelic rock	Jefferson Airplane	Curtida
Purple Haze	psychedelic rock	Jimi Hendrix	Curtida
Kill This Love	k-pop	BLACKPINK	Não Curtida
Can't Stop the Feeling!	pop	Justin Timberlake	Não Curtida

Tabela C.16: Avaliação Usuário 6 - Itens Relevantes (Mistral)

Música	Gênero	Artista	Avaliação
Alaska	indie pop	Maggie Rogers	Curtida
Supalonely (feat. Gus Dapperton)	indie pop	BENEE	Curtida
Hotel California	classic rock	Eagles	Curtida
Stairway to Heaven	classic rock	Led Zeppelin	Curtida
Dynamite	k-pop	BTS	Curtida
Gangnam Style	k-pop	PSY	Curtida
Somebody to Love	vocal harmony group	Queen	Curtida
Hallelujah	vocal harmony group	Pentatonix	Curtida
Shape of You	pop	Ed Sheeran	Curtida
Bad Guy	pop	Billie Eilish	Curtida

Tabela C.17: Avaliação Usuário 7 - Itens Relevantes (Gemini)

Música	Gênero	Artista	Avaliação
Evidências	sertanejo	Chitãozinho & Xororó	Curtida
Evidências	sertanejo	Chitãozinho & Xororó	Curtida
Deixa a Vida Me Levar	pagode	Zeca Pagodinho	Curtida
Despacito	pop	Luis Fonsi	Não Curtida
Shape of You	pop	Ed Sheeran	Não Curtida
Deus é Fiel	gospel	Fernandinho	Não Curtida
Ousado Amor	gospel	Gabriela Rocha	Não Curtida
Tempo Perdido	pop rock brasileiro	Legião Urbana	Não Curtida
Pais e Filhos	pop rock brasileiro	Legião Urbana	Não Curtida
Samba do Arnesto	pagode	Zeca Pagodinho	Não Curtida

Tabela C.18: Avaliação Usuário 7 - Itens Relevantes (Llama)

Música	Gênero	Artista	Avaliação
Vai Malandra	pop rock brasileiro	Anitta	Curtida
Só Eu Sei de Você	sertanejo	Bruno & Marrone	Não Curtida
Eu Sei de Coração	sertanejo	Bruno & Marrone	Não Curtida
Vaina Loca	pop	Ozuna	Não Curtida
Con Calma	pop	Daddy Yankee	Não Curtida
Aqui Estou	gospel	Tarcísio de Almeida	Não Curtida
Eu Vou Te Amar	gospel	Tarcísio de Almeida	Não Curtida
Só Depois do Amor	pop rock brasileiro	Anitta	Não Curtida
Samba de Uma Nota Só	pagode	Sivuca	Não Curtida
Aquarela do Brasil	pagode	Sivuca	Não Curtida

Tabela C.19: Avaliação Usuário 7 - Itens Relevantes (Mistral)

Música	Gênero	Artista	Avaliação
Ai Se Eu Te Pego	sertanejo	Michel Teló	Curtida
Bad Guy	pop	Billie Eilish	Curtida
What a Beautiful Name	gospel	Hillsong Worship	Curtida
Reckless Love	gospel	Cory Asbury	Curtida
Tico-Tico no Fubá	sertanejo	Zezé Di Camargo & Luciano	Não Curtida
Shape of You	pop	Ed Sheeran	Não Curtida
Eu Quero Mais	pop rock brasileiro	Skank	Não Curtida
Ainda Gosto de Você	pop rock brasileiro	Jota Quest	Não Curtida
Fala Mal de Mim	pagode	Só Pra Contrariar	Não Curtida
Vem Ni Mim	pagode	Banda Eva	Não Curtida

Tabela C.20: Avaliação Usuário 8 - Itens Relevantes (Gemini)

Música	Gênero	Artista	Avaliação
Deus é Fiel	brazilian gospel	Aline Barros	Curtida
Ousado Amor	brazilian gospel	Gabriela Rocha	Curtida
Shape of You	pop	Ed Sheeran	Curtida
Chega de Saudade	mpb	João Gilberto	Curtida
Thinking Out Loud	acoustic pop	Ed Sheeran	Curtida
Perfect	acoustic pop	Ed Sheeran	Curtida
Garota de Ipanema	mpb	Tom Jobim	Curtida
Rap da Felicidade	funk consciente	Cidinho & Doca	Não Curtida
Sou Negoinho	funk consciente	Racionais MC's	Não Curtida
Despacito	pop	Luis Fonsi	Não Curtida

Tabela C.21: Avaliação Usuário 9 - Itens Relevantes (Gemini)

Música	Gênero	Artista	Avaliação
Blinding Lights	rap	The Weeknd	Curtida
Bohemian Rhapsody	rock	Queen	Curtida
Stairway to Heaven	rock	Led Zeppelin	Curtida
Shape of You	pop	Ed Sheeran	Curtida
Perfect	pop	Ed Sheeran	Curtida
Can't Help Falling in Love	soundtrack	Elvis Presley	Curtida
Moon River	soundtrack	Audrey Hepburn	Curtida
Save Your Tears	rap	The Weeknd	Curtida
Bum Bum Tam Tam	trap brasileiro	MC Fioti	Não Curtida
Envolvimento	trap brasileiro	MC Loma e as Gêmeas Lacração	Não Curtida

Tabela C.22: Avaliação Usuário 9 - Itens Relevantes (Llama)

Música	Gênero	Artista	Avaliação
Blank Space	pop	Taylor Swift	Curtida
Shake It Off	pop	Taylor Swift	Curtida
God's Plan	rap	Drake	Curtida
One Dance	rap	Drake	Curtida
Lose Yourself	hip hop	Eminem	Curtida
Uptown Funk	dance pop	Mark Ronson ft. Bruno Mars	Curtida
Can't Stop the Feeling!	dance pop	Justin Timberlake	Curtida
Let It Go	soundtrack	Idina Menzel	Não Curtida
A Dream Is a Wish Your Heart Makes	soundtrack	Ilene Woods	Não Curtida
Juicy	hip hop	The Notorious B.I.G.	Não Curtida

Tabela C.23: Avaliação Usuário 9 - Itens Relevantes (Mistral)

Música	Gênero	Artista	Avaliação
God's Plan	rap	Drake	Curtida
Sicko Mode	rap	Travis Scott	Curtida
Shape of You	dance pop	Ed Sheeran	Curtida
One Kiss	dance pop	Calvin Harris, Dua Lipa	Curtida
The Sound of Silence	soundtrack	Disturbed	Curtida
Humble	hip hop	Kendrick Lamar	Curtida
Bad Guy	hip hop	Billie Eilish	Curtida
Shallow	soundtrack	Lady Gaga, Bradley Cooper	Não Curtida
Bola Rebola	trap brasileiro	Tropkillaz, J Balvin, Anitta	Não Curtida
Solta a Batida	trap brasileiro	MC Lan, Anitta	Não Curtida

Tabela C.24: Avaliação Usuário 10 - Itens Relevantes (Gemini)

Música	Gênero	Artista	Avaliação
Lose Yourself	rap	Eminem	Curtida
Till I Collapse	rap	Eminem	Curtida
Shape of You	pop	Ed Sheeran	Curtida
Perfect	pop	Ed Sheeran	Curtida
Sicko Mode	trap	Travis Scott	Curtida
Highest in the Room	trap	Travis Scott	Curtida
God's Plan	hip hop	Drake	Curtida
Hotline Bling	hip hop	Drake	Curtida
Love Sosa	chicago rap	Chief Keef	Não Curtida
Faneto	chicago rap	Chief Keef	Não Curtida

Tabela C.25: Avaliação Usuário 10 - Itens Relevantes (Llama)

Música	Gênero	Artista	Avaliação
God's Plan	rap	Drake	Curtida
Lose Yourself	rap	Eminem	Curtida
Happy	pop	Pharrell Williams	Curtida
Uptown Funk	pop	Mark Ronson ft. Bruno Mars	Curtida
Sicko Mode	trap	Travis Scott	Curtida
Bad and Boujee	trap	Migos ft. Lil Uzi Vert	Curtida
Dançar Kuduro	trap brasileiro	Don Omar ft. Lucenzo	Não Curtida
Vai Malandra	trap brasileiro	Pabllo Vittar	Não Curtida
U Can't Touch This	hip hop	MC Hammer	Não Curtida
Juicy	hip hop	The Notorious B.I.G.	Não Curtida

Tabela C.26: Avaliação Usuário 10 - Itens Relevantes (Mistral)

Música	Gênero	Artista	Avaliação
God's Plan	rap	Drake	Curtida
Sicko Mode	rap	Travis Scott	Curtida
Taste	trap	Tyga	Curtida
Shape of You	pop	Ed Sheeran	Curtida
Lucid Dreams	hip hop	Juice WRLD	Curtida
SAD!	hip hop	Xxxtentacion	Curtida
Clarity	pop	Zedd, Foxes	Curtida
Narcos	trap	Bad Bunny	Não Curtida
Bola Rebola	trap brasileiro	Tropkillaz, J Balvin, Anitta	Não Curtida
Soltinha	trap brasileiro	MC Lan, Anitta	Não Curtida

Tabela C.27: Avaliação Usuário 11 - Itens Relevantes (Gemini)

Música	Gênero	Artista	Avaliação
Scooby-Doo, Where Are You!	cartoon	Larry Marks	Curtida
I Want It That Way	pop	Backstreet Boys	Curtida
Wannabe	pop	Spice Girls	Curtida
Hotel California	classic rock	Eagles	Curtida
Evidências	pop rock brasileiro	Chitãozinho & Xororó	Curtida
Garota de Ipanema	pop rock brasileiro	Tom Jobim	Curtida
The Flintstones Theme	cartoon	The B-52's	Curtida
The Thrill Is Gone	jazz blues	B.B. King	Não Curtida
Stairway to Heaven	classic rock	Led Zeppelin	Não Curtida
Summertime	jazz blues	Ella Fitzgerald	Não Curtida

Tabela C.28: Avaliação Usuário 11 - Itens Relevantes (Llama)

Música	Gênero	Artista	Avaliação
Blank Space	pop	Taylor Swift	Curtida
Shake It Off	pop	Taylor Swift	Curtida
Hotel California	classic rock	Eagles	Curtida
Só Depois Tudo Bem	pop rock brasileiro	Anitta	Não Curtida
Vai Malandra	pop rock brasileiro	Anitta	Não Curtida
Stairway to Heaven	classic rock	Led Zeppelin	Não Curtida
Sitting on the Dock of the Bay	jazz blues	Otis Redding	Não Curtida
Purple Rain	jazz blues	Prince	Não Curtida
The Entertainer	cartoon	Scott Joplin	Não Curtida
The Chicken Dance	cartoon	Werner Thomas	Não Curtida

Tabela C.29: Avaliação Usuário 11 - Itens Relevantes (Mistral)

Música	Gênero	Artista	Avaliação
Let It Go	cartoon	Idina Menzel	Curtida
How Far I'll Go	cartoon	Auli'i Cravalho	Curtida
Bohemian Rhapsody	classic rock	Queen	Curtida
Stairway to Heaven	classic rock	Led Zeppelin	Não Curtida
Shallow	pop	Lady Gaga, Bradley Cooper	Não Curtida
The Thrill Is Gone	jazz blues	B.B. King	Não Curtida
St. James Infirmary Blues	jazz blues	Louis Armstrong	Não Curtida
O Pulso	pop rock brasileiro	Skank	Não Curtida
Bola Vai	pop rock brasileiro	Titãs	Não Curtida
A Thousand Years	pop	Christina Perri	Não Curtida

Tabela C.30: Avaliação Usuário 12 - Itens Relevantes (Gemini)

Música	Gênero	Artista	Avaliação
Highway to Hell	hard rock	AC/DC	Curtida
Back in Black	hard rock	AC/DC	Curtida
The Trooper	power metal	Iron Maiden	Curtida
Fear of the Dark	power metal	Iron Maiden	Curtida
Close to the Edge	progressive rock	Yes	Curtida
Stairway to Heaven	classic rock	Led Zeppelin	Curtida
Bohemian Rhapsody	classic rock	Queen	Curtida
O Tempo Não Para	brazilian rock	Cazuza	Curtida
Pro Dia Nascer Feliz	brazilian rock	Barão Vermelho	Curtida
Roundabout	progressive rock	Yes	Não Curtida

Tabela C.31: Avaliação Usuário 12 - Itens Relevantes (Llama)

Música	Gênero	Artista	Avaliação
Barracuda	hard rock	Heart	Curtida
Crazy Train	hard rock	Ozzy Osbourne	Curtida
Powerslave	power metal	Iron Maiden	Curtida
Stairway to Heaven	progressive rock	Led Zeppelin	Curtida
Close to the Edge	progressive rock	Yes	Curtida
Hotel California	classic rock	Eagles	Curtida
Samba Pa Ti	brazilian rock	Raul Seixas	Curtida
Aquele Abraço	brazilian rock	Raul Seixas	Curtida
Symphony of Enchanted Lands	power metal	Rhapsody	Curtida
Sweet Emotion	classic rock	Aerosmith	Não Curtida

Tabela C.32: Avaliação Usuário 12 - Itens Relevantes (Mistral)

Música	Gênero	Artista	Avaliação
Back in Black	hard rock	AC/DC	Curtida
Whole Lotta Love	hard rock	Led Zeppelin	Curtida
The Final Countdown	power metal	Europe	Curtida
Hotel California	classic rock	Eagles	Curtida
Stairway to Heaven	classic rock	Led Zeppelin	Curtida
Rock and Roll	brazilian rock	Os Paralamas do Sucesso	Curtida
Flores e Espinas	brazilian rock	Barão Vermelho	Curtida
I Want Out	power metal	Helloween	Não Curtida
Bohemian Rhapsody	progressive rock	Queen	Não Curtida
Roundabout	progressive rock	Yes	Não Curtida

Tabela C.33: Avaliação Usuário 13 - Itens Relevantes (Gemini)

Música	Gênero	Artista	Avaliação
Dynamite	k-pop	BTS	Curtida
Butter	k-pop	BTS	Curtida
Smells Like Teen Spirit	alternative rock	Nirvana	Curtida
Basket Case	alternative rock	Green Day	Curtida
Céu Azul	rock alternativo brasileiro	Charlie Brown Jr.	Curtida
Dias Melhores	rock alternativo brasileiro	Jota Quest	Curtida
Shape of You	pop	Ed Sheeran	Curtida
Despacito	pop	Luis Fonsi	Curtida
O Tempo Não Para	rock nacional brasileiro	Cazuza	Não Curtida
Pais e Filhos	rock nacional brasileiro	Legião Urbana	Não Curtida

Tabela C.34: Avaliação Usuário 13 - Itens Relevantes (Llama)

Música	Gênero	Artista	Avaliação
Boy With Luv	k-pop	BTS ft. Halsey	Curtida
Kill This Love	k-pop	BLACKPINK	Curtida
Mighty Long Fall	j-rock	ONE OK ROCK	Curtida
Blank Space	pop	Taylor Swift	Curtida
Shake It Off	pop	Taylor Swift	Curtida
Ho Hey	folk rock	The Lumineers	Curtida
Stubborn Love	folk rock	The Lumineers	Curtida
Mr. Brightside	alternative rock	The Killers	Curtida
Somebody Told Me	alternative rock	The Killers	Curtida
Gamble	j-rock	ONE OK ROCK	Não Curtida