



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FLAVIANO DIAS FONTES

Deejai: Uma plataforma open source para recomendação de músicas para grupos de usuários de serviços de streaming

Recife

2022

FLAVIANO DIAS FONTES

Deejai: Uma plataforma open source para recomendação de músicas para grupos de usuários de serviços de streaming

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Mestre em Ciência da Computação

Área de Concentração: Mídia e Interação

Orientador: Prof. Dr. Giordano Ribeiro Eulálio Cabral

Recife

2022

Catálogo na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

F683d Fontes, Flaviano Dias
 Deejai: uma plataforma open source para recomendação de músicas para grupos de usuários de serviços de streaming / Flaviano Dias Fontes. – 2022.
 80 f.: il., fig., tab.

 Orientador: Giordano Ribeiro Eulálio Cabral.
 Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2022.
 Inclui referências e apêndices.

 1. Mídia e Interação. 2. Filtro colaborativo. I. Cabral, Giordano Ribeiro Eulálio (orientador). II. Título.

 006.7 CDD (23. ed.) UFPE - CCEN 2023-20

Flaviano Dias Fontes

**“Deejai: Uma plataforma open source para recomendação de músicas
para grupos de usuários de serviços de streaming”**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação. Área de Concentração: Mídia e Interação.

Aprovado em: 15 de agosto de 2022.

BANCA EXAMINADORA

Prof. Dr. Filipe Carlos de Albuquerque Calegário
Centro de Informática / UFPE

Prof. Dr. Gilberto Amado de Azevedo Cysneiros Filho
Departamento de Estatística e Informática / UFRPE

Prof. Dr. Giordano Ribeiro Eulalio Cabral
Centro de Informática / UFPE
(Orientador)

Dedico este trabalho a todos que estiveram comigo durante essa jornada em especial a minha mãe Maria José, minha avó Marcolina, minha esposa Jessyca Lorena e meus filhos Maria Heloísa e Pedro. Obrigado a todos por serem meu alicerce todos os dias.

AGRADECIMENTOS

Agradeço primeiramente a Deus, que ao longo da minha trajetória esteve comigo me iluminando nas escolhas e que não me deixou desistir da jornada mesmo com as adversidades que apareceram. A minha mãe Maria José, que sempre fez o possível e o impossível para que não me faltasse nada e por me incentivar todos os dias a seguir pelo caminho da educação, por sua bondade comigo e seu amor que nunca me faltou. A minha avó Marcolina que sempre esteve ao meu lado e me incentivou em todos os aspectos durante a minha vida. Aos meus orientadores Giordano Cabral e Geber Ramalho por confiarem em mim e me incentivarem durante toda essa jornada. A minha esposa Jessyca Lorena que apesar de todas as adversidades que o mundo passou, nunca largou a minha mão e sempre me apoiou com muito amor e carinho me fazendo acreditar que seria possível terminar esse ciclo. Aos meus filhos Maria Heloísa e Pedro Antônio por diariamente me iluminarem com os seus sorrisos. Aos familiares da minha esposa que me acolheram na reta final para que pudesse defender esse trabalho, seu Jefferson e dona Sônia, além da Kelly, da Karla, do Enzo e do Júnior. Aos meus tios (in memorian) Marcelino e Milton que desde cedo me mostraram que com perseverança e trabalho duro podemos alcançar nossos objetivos. A meu primo que considero como tio Edvaldo e sua esposa Eralda que sempre me acolheram e aconselharam quando preciso. Aos meus eternos amigos Delando, Karina, Linaldo, Givaldo, Guilherme, Anderson, Davi, Celina, Júnior Lima, Erick, Jessica, Ananda, Deborah, Rafael, Humberto, Antonio, Ceíça, Davi, Sarah e Pedro. Aos meus companheiros do LaCETI Prof Edna, Marvson, Vanessa, Victor, Karine, Rogério, Gabriel, Tulio, Luan, Gonça, Ryan, Alice, Paulo e Biu. Ao pessoal do Mustic, que em todas reuniões sempre estiveram dispostos a me ajudar, ao corpo docente e técnico do CIn e ao CNPq.

RESUMO

Consumir música é algo que fazemos tanto de forma individual quanto coletiva. Quando realizamos em conjunto é nos dada a possibilidade de criar uma relação emocional forte, pois associamos as pessoas com o ambiente e criamos uma experiência diferenciada. A definição se a experiência será boa ou não acaba ficando nas mãos de poucas pessoas, o artista que está tocando, o DJ da festa, ou o responsável pela playlist. Com o intuito de aprimorar a experiência de se consumir músicas em ambiente coletivo, nos propusemos a utilizar de estudos sobre sistemas de recomendação focado em música, aproveitando suas técnicas já estudadas para construir uma plataforma open source com a possibilidade de integrar dois serviços de streaming a fim de facilitar a colaboração dos usuários levando em consideração os seus gostos e principalmente seus desgostos para tentar melhorar o resultado da playlist gerada. Para a seleção das músicas foram utilizadas técnicas de recomendação híbrida, associando as músicas mais ouvidas por cada usuário individualmente nas plataformas de streaming, a fim de mitigar o problema de partida a frio que é quando não se sabe o que recomendar ao usuário inicialmente, associado a um filtro colaborativo dentro da plataforma desenvolvida. Para a definição de contexto foi criado uma estrutura de sala dentro da plataforma para associar os usuários com as músicas que serão ouvidas. O experimento consistiu na utilização de plataforma em dois momentos por três grupos focais onde a cada rodada o grupo avaliava individualmente as músicas que foram recomendadas dando notas de 1 a 5 para cada músicas, no primeiro momento a playlist foi gerada contendo apenas as músicas que os usuários individualmente gostam. Já num segundo momento, espaçado de 7 dias, a playlist gerada levou em consideração filtrar as músicas em que a médias das avaliações fosse menor que 4 adicionando novas músicas que estavam disponíveis e ainda sem votos dos usuários. As playlists geradas utilizam do EveryNoiseAtOnce onde os estilos são mapeados como pontos cartesianos, a fim de aproximar estilos próximos é um bom auxílio para poder criar a playlist para usuário que possuem gostos muito distantes, pois é possível construir um caminho entre esses pontos, buscando o vizinho mais próximo. Para a avaliação da plataforma foi utilizado o questionário System Usability Scale (SUS) o questionário foi aplicado durante uma apresentação pública e presencial no Rec'n'Play no Recife, onde foi apresentado a plataforma e realizado uma demonstração com o público presente.

Palavras-chaves: sistemas de recomendação para grupos; filtro colaborativo; Deezer; Spotify;

ABSTRACT

Consuming music is something we do both individually and collectively. When we play music together, we are given the possibility of creating a strong emotional relationship, as we associate people with the environment and create a differentiated experience. The definition of whether the experience will be good or not ends up being in the hands of a few people, the playing artist, the DJ at the party, or the person responsible for the playlist. To improve the experience of consuming music in a collaborative environment, we proposed to use studies on music-focused recommendation systems, taking advantage of their already studied techniques to build an open-source platform with the possibility of integrating two streaming services to facilitate the collaboration of users taking into account their likes and especially their dislikes to try to improve the result of the generated playlist. For the selection of songs, we used hybrid recommendation techniques, associating the songs most listened to by each user on streaming platforms, to mitigate the cold start problem, which is when you do not know what you recommend to the user initially, associated to a collaborative filter within the developed platform. For context definition, a room structure was created within the platform to associate users with the songs that will be listened. The experiment consisted of the use of the platform in two moments by three focus groups. In each round, the group individually evaluated the recommended songs giving scores from 1 to 5 for each music. In the first moment, the playlist was generated containing only the songs liked by users. In a second moment, after seven days, the playlist generated took into account filtering the music in which the average of the estimates was less than four, adding new songs that were available and still without user votes. The generated playlists use EveryNoiseAtOnce where the styles are mapped as Cartesian points, in order to approximate close styles it is a good help to be able to create the playlist for users who have very distant tastes, as it is possible to build a path between these points, seeking the nearest neighbor. To evaluate the platform, the System Usability Scale (SUS) questionnaire was applied during a public and face-to-face presentation at Rec'n'Play in Recife, where the platform was presented and a demonstration was carried out with the public present.

Keywords: group recommender systems; colaborative filter; Deezer; Spotify;

LISTA DE FIGURAS

Figura 1 – Classificação dos Sistemas de Recomendação	20
Figura 2 – Arquitetura do sistema	32
Figura 3 – Resposta no formato JSON	33
Figura 4 – Detalhamento do backend	34
Figura 5 – Estrutura de pastas do frontend	36
Figura 6 – Diagrama ER da plataforma	37
Figura 7 – Construção da Entidade User no Schema do Prisma	39
Figura 8 – Tabela usuário gerada no banco de dados	40
Figura 9 – Tela de login com Spotify	41
Figura 10 – Tela de login com Deezer	44
Figura 11 – Tela inicial da aplicação (Login)	51
Figura 12 – Listagem de músicas do usuário	52
Figura 13 – Reprodução Every Noise At Once	52
Figura 14 – Criação de uma sala na plataforma	53
Figura 15 – Sala para técnica de filtro colaborativo	54
Figura 16 – Listagem de músicas que o usuário não gosta	54
Figura 17 – Gráfico tipo de experimento	55
Figura 18 – Gráfico de distribuição do gênero	58
Figura 19 – Gráfico de distribuição da idade	58
Figura 20 – Gráfico sobre plataforma de consumo	59
Figura 21 – Respostas da primeira questão do SUS	60
Figura 22 – Respostas da segunda questão do SUS	60
Figura 23 – Respostas da terceira questão do SUS	61
Figura 24 – Respostas da quarta questão do SUS	61
Figura 25 – Respostas da quinta questão do SUS	62
Figura 26 – Respostas da sexta questão do SUS	62
Figura 27 – Respostas da sétima questão do SUS	63
Figura 28 – Respostas da oitava questão do SUS	63
Figura 29 – Respostas da nona questão do SUS	64
Figura 30 – Respostas da décima questão do SUS	64

LISTA DE TABELAS

Tabela 1 – Comparação entre técnicas de recomendação	23
Tabela 2 – Comparação entre técnicas de filtragem colaborativa	23
Tabela 3 – Distribuição das preferências individuais	25
Tabela 4 – Modelo de Agregação dos sistemas de recomendação para grupos com música	29
Tabela 5 – Escopos disponibilizados pela API do Spotify	42
Tabela 6 – Permissões disponibilizados pela API do Deezer	44
Tabela 7 – Endpoints utilizados da API do Deezer	45
Tabela 8 – Endpoints utilizados da API do Spotify	45
Tabela 9 – Avaliação dos usuários na primeira rodada do grupo 1	56
Tabela 10 – Avaliação dos usuários na primeira rodada do grupo 2	56
Tabela 11 – Avaliação dos usuários na primeira rodada do grupo 3	57

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CBF	Content-based filtering
CF	Collaborative filtering
CRA	Create React App
CSS	Cascading Style Sheets
DJ	Disc Jockey
HTTP	Hypertext Transfer Protocol
IA	Inteligência artificial
ISRC	International Standard Recording Code
JSON	Javascript Object Notation
JSX	JavaScript XML
MIME	Multipurpose Internet Mail Extensions
MRS	Music Recommender System
ORM	Object-relational mapping
REST	Representational state transfer
RWR	Random Walks with Restarts
SPA	Single-page application
SQL	Structured Query Language
SSL	Secure Sockets Layer
SUS	System Usability Scale
SVM	Support Vector Machine
TLS	Transport Layer Security
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locators

UX	User Experience
XML	Extensible Markup Language
XSS	Cross-Site Scripting

SUMÁRIO

1	INTRODUÇÃO	14
1.1	CONTEXTUALIZAÇÃO	14
1.2	PROBLEMA DE PESQUISA	16
1.3	OBJETIVOS	17
1.3.1	Objetivos gerais	17
1.3.2	Objetivos específicos	17
2	REVISÃO DA LITERATURA	18
2.1	SISTEMAS DE RECOMENDAÇÃO	18
2.1.1	Mudanças de preferências	19
2.1.2	Dados esparsos	19
2.1.3	Escalabilidade	19
2.1.4	Sinonímia	19
2.1.5	Privacidade	19
2.2	TÉCNICAS DE RECOMENDAÇÃO	19
2.2.1	Filtragem baseada em conteúdo	20
2.2.2	Filtro colaborativo	21
2.2.2.1	<i>Filtragem baseada em memória</i>	21
2.2.2.2	<i>Filtragem baseada em modelo</i>	22
2.2.3	Filtragem híbrida	22
2.3	SISTEMAS DE RECOMENDAÇÃO MUSICAL	24
2.4	SISTEMAS DE RECOMENDAÇÃO PARA GRUPOS	25
2.4.1	Estratégias baseadas na maioria	26
2.4.2	Estratégias baseadas em consenso	26
2.4.3	Estratégias limítrofes (borderline)	27
2.5	SISTEMAS DE RECOMENDAÇÃO PARA GRUPOS PARA MÚSICA	27
2.5.1	Every Noise At Once	29
3	METODOLOGIA	30
3.1	ARQUITETURA GERAL DO SISTEMA	31
3.1.1	Back-End	31
3.1.2	Front-end	35

3.1.3	Banco de dados	37
3.1.3.1	<i>Prisma</i>	39
3.2	FONTE DE DADOS	40
3.2.1	Spotify	40
3.2.2	Deezer	43
3.2.3	Funcionalidades utilizadas	43
3.3	TRATAMENTO DOS DADOS	46
3.4	GRUPO FOCAL	46
3.5	EXPERIMENTO	47
3.6	QUESTIONÁRIOS	48
3.6.1	Avaliação da playlist	48
3.6.2	System Usability Scale	48
3.7	LIMITAÇÕES	49
4	DISCUSSÃO DOS RESULTADOS	51
5	CONCLUSÃO	65
5.1	PRINCIPAIS CONTRIBUIÇÕES	66
5.2	TRABALHOS FUTUROS	67
	REFERÊNCIAS	68
	APÊNDICE A – FORMULÁRIO DE PESQUISA	73
	APÊNDICE B – FORMULÁRIO DE PESQUISA	74
	APÊNDICE C – FORMULÁRIO DE PESQUISA	75
	APÊNDICE D – FORMULÁRIO DE PESQUISA	76
	APÊNDICE E – FORMULÁRIO DE PESQUISA	77
	APÊNDICE F – FORMULÁRIO DE PESQUISA	78
	APÊNDICE G – FORMULÁRIO DE PESQUISA	79
	APÊNDICE H – RESPOSTAS SUS	80

1 INTRODUÇÃO

Neste capítulo iremos apresentar de forma sucinta e objetiva a pesquisa, informando os motivos dentro da área de sistema de recomendação de música para grupo, que levaram à sua realização, da sua natureza e quais método foram escolhidos para realizar. Este capítulo possui uma estruturação com os seguintes pontos: contextualização, motivação e objetivos.

1.1 CONTEXTUALIZAÇÃO

Consumir música é uma parte muito importante para a vida das pessoas. Músicas estão presentes na vida do indivíduo durante diversos momentos, são músicas selecionadas para trabalhar, praticar esporte ou para momentos de lazer. A experiência musical inclui tanto o uso individual quanto o uso coletivo, mas quando o consumo musical é tido de forma coletiva a pergunta que fazemos é quais serão as músicas selecionadas e quem ficará responsável por essa seleção. (CUNNINGHAM; NICHOLS, 2009)

O consumo de música em ambientes coletivos nos dá a possibilidade de criar uma conexão emocional forte, maior do que um *download* assíncrono de uma *playlist* para consumir a distância. (BASSOLI; MOORE; AGAMANOLIS, 2006)

A forma como consumimos música mudou bastante ao longo do tempo. No século XIX, por exemplo, a música só poderia ser consumida em ambientes privados ou apresentações ao vivo. Ao longo do século XX a música se tornou mais ampla e disponível, com a ampliação dos contextos em que se consumia música tornando-se muito mais variada e diversificada: o valor cultural colocado na experiência musical tornou-se dependente do contexto em questão. Por exemplo, a interação entre um Disc Jockey (DJ) e pista de dança em uma boate ou aquela entre a orquestra e o público em uma sala de concertos, de tal forma que a distinção entre música “séria” e “popular” tornou-se muito menos clara. (NORTH; HARGREAVES; HARGREAVES, 2004)

O mercado musical não iniciou com a invenção do fonógrafo ou do disco, mas sim com a distribuição em massa e o uso comercial da música. Podemos considerar seu início a partir do momento em que a produção e o consumo musical romperam os laços com o contexto da corte feudal e da igreja. Não é possível definir uma data pois se trata de um processo de desenvolvimento caracterizado por uma transição entre uma cultura de corte feudal para uma

cultura burgo-capitalista. (TSCHMUCK, 2006)

A indústria da música passou por uma transição de comercializar mídias físicas, como CDs, discos de vinil, fitas cassete, para adotar um modelo de assinatura, onde vendem o acesso a uma quantidade extremamente diversa de músicas por um preço mais acessível, e com muitos serviços com conteúdo gratuito sendo financiado por meio de propagandas, com a disponibilidade de uma quantidade significativa de plataformas. Atualmente a liberdade do usuário em consumir música se tornou instantânea. (MORRIS; POWERS, 2015)

Essa forma de consumir tem como base a disponibilidade de músicas na ordem dos milhões de exemplares, para os serviços de *streaming* alguns parâmetros intrínsecos ao usuário possuem uma importância para que seja possível inferir o gosto musical e assim recomendar músicas com uma maior assertividade, como seu nível de experiência, sua localidade e seus traços de personalidade. Apesar de toda esta coleta de dados, para o usuário, comumente é apenas disponibilizado a lista de artistas e músicas que ele gosta, seja através da listagem de músicas mais ouvidas ou a listagem de faixas, álbuns e artistas que ele assinalou que gosta.

É facilmente encontrado nas Application Programming Interface (API)s *endpoints* para listarmos as músicas mais ouvidas daquela conta ou os estilos dos artistas ouvidos, porém as músicas que o usuário não gosta não são disponibilizadas, mesmo que marcadas com o não gostei disponibilizado por algumas plataformas.

Para recomendar itens, frequentemente necessitamos fazer escolhas sem um conhecimento prévio suficiente das alternativas disponíveis. (RESNICK; VARIAN, 1997) Ao longo da história a nossa sociedade atravessou diversas mudanças, a internet contribuiu para essas mudanças acontecerem e mais recentemente com a pandemia de Covid-19 essas mudanças foram ainda mais permeadas na sociedade. As compras pela web que já estávamos acostumados ficou ainda mais abrangente, coletamos e cedemos dados através de pesquisas na internet, redes sociais, *e-commerce*, até o trabalho e o relacionamento entre pessoas se tornou mais digital. (ALMEIDA; SANTOS; MONTEIRO, 2020).

Essas interações são registradas eletronicamente, o que possibilita para os engenheiros e pesquisadores desses serviços analisarem de forma detalhada o comportamento do usuário. (LÜ et al., 2012) E a década de 2010 possuiu como característica uma proliferação de plataformas digitais sob demanda que têm o intuito de oferecer serviços ao consumidor. (BISSELL, 2020) Um dos que tiveram forte crescimento foi a indústria da música, através de plataformas de *streaming*, como *Spotify*, *Apple Music*, *Amazon Music*, *TIDAL* e *Deezer*. (MAASØ; HAGEN, 2020) Sendo o *Spotify* o *streaming* líder e que respondeu por 32% do mercado global no ano

de 2020. (KUMAR, 2020)

1.2 PROBLEMA DE PESQUISA

Sistemas de recomendação é uma área de estudo em constante expansão, com aplicações em diversas áreas como *e-commerce*, filmes, música e outros com o intuito de promover sugestões para o usuário. Quando falamos de música, temos mais de 20 anos de estudos tentando resolver o problema de uma boa geração de *playlists* que maximize a satisfação de um número maior de ouvintes. (FONTES; CABRAL; RAMALHO, 2021)

O objetivo de um sistema de recomendação é gerar recomendações significativas para uma coleção de usuários para itens ou produtos que possam interessá-los. Sugestões para livros na Amazon ou filmes na Netflix são exemplos do mundo real da operação de sistemas de recomendação de força do setor. O design de tais mecanismos de recomendação depende do domínio e das características particulares dos dados disponíveis. (MELVILLE; SINDHWANI, 2010) O primeiro sistema de recomendação comercial foi o Tapestry, que foi projetado para recomendar documentos extraídos de grupos de notícias para uma coleção de usuários. (GOLDBERG et al., 1992)

Com a disponibilidade de opções atuais o processo de tomada de decisão para personalização torna-se um problema a ser combatido. (MELVILLE; SINDHWANI, 2010)

As técnicas utilizadas por sistemas de recomendações são variadas como filtro colaborativo, recomendação baseada em confiança, recomendação baseada em conteúdo, entre outras. Para as técnicas de filtragem colaborativa não possui a necessidade de conhecimento prévio a respeito dos usuários ou itens pois a recomendação é feita através da interação entre eles. Problemas comuns nessas abordagens são dados por problema de "partida a frio", a precisão da previsão e a incapacidade de capturar as interações complexas entre os usuários e os itens. (NASSAR; JAFAR; RAHHAL, 2020)

Para recomendações musicais para grupos diversas soluções foram criadas, como o *Smart Radio*, que em 1996 já considerava as informações de *dislikes* dos usuários, para o filtro colaborativo além de fazer uma extração automática de características com o intuito de fazer recomendação baseada em conteúdo.

Em 1998, o *MusicFX* abordou a classificação da preferência do usuário através de uma escala que os valores são amar, gostar, não se importar, não gostar e detestar, o ambiente de validação foi uma academia de ginástica.

Já o *JMusicGroupRecommender* levava em consideração as músicas que o usuário reproduzia no computador para sincronizar essas informações e adicionava avaliações pessoais dos usuários para, por fim, gerar as recomendações. (FONTES; CABRAL; RAMALHO, 2021)

Com os serviços de *streaming*, não temos acesso a informação do que o usuário não gosta, nem informação de pular faixa, para gerar um serviço de recomendação para grupos, que podem utilizar serviços de *streaming* diferentes, precisamos de uma meta-ferramenta para capturar essas interações entre indivíduos e armazenar as informações de *dislikes*.

A hipótese trabalhada neste trabalho se dá a partir da captura da informação do não gostei para usuários de plataformas diferentes para gerar uma *playlist* única para recomendação de músicas irá trazer algum benefício para os usuários.

1.3 OBJETIVOS

1.3.1 Objetivos gerais

O presente trabalho tem como objetivo geral a avaliação da geração automática de uma playlist com a informação de dislike é mais assertiva para a experiência do usuário User Experience (UX) do que simplesmente levar em consideração as suas preferências.

1.3.2 Objetivos específicos

1. Desenvolver uma plataforma para recomendar músicas coletivamente baseadas nas músicas de usuários de serviços de streaming que não gostam.
2. Construir a ferramenta capaz de recomendar músicas com base em um contexto.
3. Construir uma base de dados que armazene as informações das músicas capturadas.
4. Aplicar técnicas de sistemas de recomendação para utilizar a informação de dislike do usuário.
5. Desenvolver um sistema para agrupar e recomendar músicas para determinados usuários.

2 REVISÃO DA LITERATURA

Neste capítulo descrevemos a literatura na qual esse estudo utilizou como base, incluindo os principais conceitos relacionados. O objetivo é a apresentação do estado da arte, através da literatura pré-existente sobre sistemas de recomendação, sistemas de recomendação para grupos, sistemas de recomendação musical e sistemas de recomendação para grupos envolvendo música.

2.1 SISTEMAS DE RECOMENDAÇÃO

Comumente um indivíduo, durante sua vida, se vê na necessidade de realizar escolhas sem um conhecimento prévio sobre as alternativas. Para auxiliar nessa tomada de decisão, recomendações de terceiros como resenhas de filmes, livros ou restaurantes, são comumente utilizadas. Sistemas de recomendação aumentam a chance de sucesso neste processo, tendo como entrada as recomendações de outras pessoas para serem agregadas e destinadas a outro indivíduo interessado. (RESNICK; VARIAN, 1997)

A área de sistemas de recomendações necessita geralmente de um conjunto de dados públicos para que possam ser testados novos métodos e algoritmos. A existência desses *datasets*, como o *MovieLens*, *FilmTrust* e *Netflix* para vídeos e a *LastFM* para músicas torna possível o desenho e melhoria dos algoritmos de Inteligência artificial (IA), por possibilitar a comparação de resultados e métodos. (ORTEGA et al., 2018)

Um dos desafios do mercado é oferecer produtos e serviços que agradem as necessidades de um determinado cliente. Os serviços digitais customizados ajudam a resolver o problema da sobrecarga de informações, tornando o processo de tomada de decisão mais rápido e fácil, aprimorando assim a UX de utilizar aquele determinado serviço. Os sistemas de recomendações utilizados na personalização foram criados tendo como base o uso de técnicas e teóricas de outros campos da IA para perfis de usuário e descoberta de preferências. (ZHANG; LU; JIN, 2021)

Segundo (RICCI; ROKACH; SHAPIRA, 2011) e (LAKSHMI; LAKSHMI, 2014) na área de sistemas de recomendação existem os seguintes desafios:

2.1.1 Mudanças de preferências

Sistemas de recomendação tem como base o interesse e o perfil do indivíduo, porém esses valores mudam depois de um tempo e alterar as preferências de um usuário é um grande desafio.

2.1.2 Dados esparsos

Usuários tendem a avaliar apenas alguns itens, quando são utilizadas técnicas de recomendação através da vizinhança, caso o usuário tenha avaliados poucos itens avaliados torna-se difícil inferir as suas preferências e pode ser relacionado a uma falsa vizinhança.

2.1.3 Escalabilidade

À medida que a quantidade de usuário e itens cresce, o sistema precisa de mais recursos para processar a informação.

2.1.4 Sinonímia

Sinonímia é a probabilidade de itens muito próximos terem nomes ou entradas diferentes. A maioria dos sistemas de recomendação acha difícil fazer distinção entre itens relacionados mais próximos, como a diferença entre a roupa infantil e a roupa de bebê.

2.1.5 Privacidade

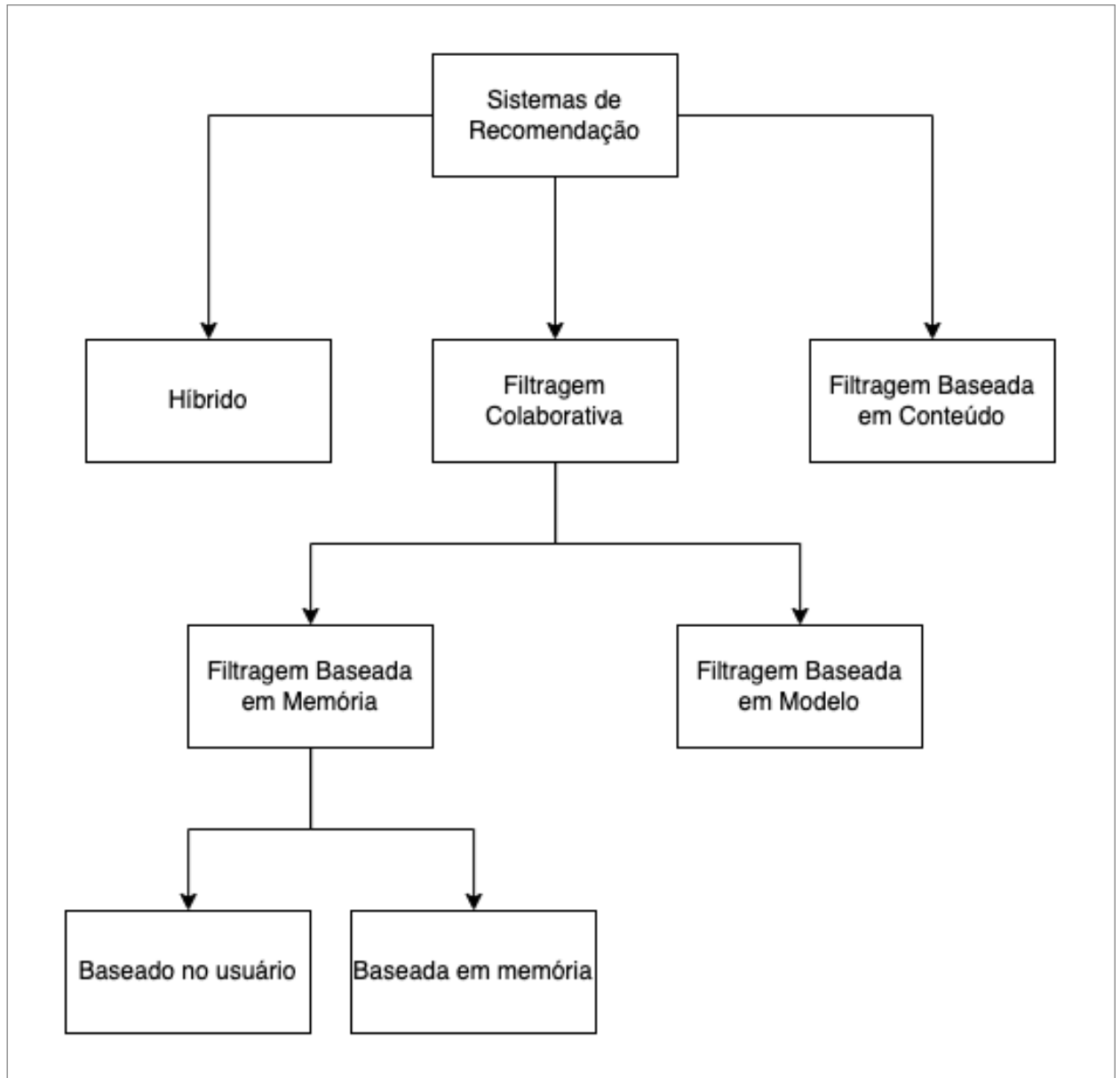
Para receber as recomendações mais precisas e corretas, o sistema deve adquirir a maior quantidade possível de informações sobre o usuário, incluindo dados demográficos e dados sobre a localização de um determinado usuário.

2.2 TÉCNICAS DE RECOMENDAÇÃO

Segundo o estudo de (PATEL; DESAI; PANCHAL, 2017), que levantou que para analisar e compreender as principais aplicações de sistemas de recomendação em diferentes áreas,

utilizam os seguintes métodos, que podem ser visualizados na Figura 1:

Figura 1 – Classificação dos Sistemas de Recomendação



Fonte: Do Autor. (2022)

2.2.1 Filtragem baseada em conteúdo

A técnica de filtragem baseada em conteúdo, do inglês Content-based filtering (CBF), tem a recomendação aos usuários dada apenas pelo comportamento e pelos dados de um único usuário. (THORAT; GOUDAR; BARVE, 2015) Esta técnica é utilizada ao recomendar documentos como páginas da internet, notícias, artigos e restaurantes, por exemplo. (ORTEGA et al., 2013)

O conceito por trás de recomendações baseadas em conteúdos é realizado pela análise da descrição dos itens preferidos de um determinado indivíduo, tendo essas preferências armazenadas e vinculando-as a um perfil, a fim descrever esses itens. De posse dessas informações é realizada a comparação dos atributos de cada item com o perfil do usuário a fim de filtrar os itens relacionados que tenham um alto grau de similaridade com o perfil do indivíduo e que assim possam ser feitas as recomendações. (THORAT; GOUDAR; BARVE, 2015)

2.2.2 Filtro colaborativo

Na técnica de filtragem colaborativa, do inglês Collaborative filtering (CF), a recomendação será dada ao usuário com base no comportamento dele e no comportamento de um outro usuário compatível. São construídos conjuntos de dados das preferências do usuário e de acordo com essa recomendação, os outros usuários também serão recomendados. (ORTEGA et al., 2013) Os sistemas de filtragem colaborativa revisam mais de um item comum para gerar um conjunto de usuários. Os aplicativos de filtragem colaborativa contêm conjuntos de dados muito grandes, o algoritmo de filtragem colaborativa é classificado principalmente em três tipos, como métodos baseados em memória, baseados em modelos e híbridos. (PATEL; DESAI; PANCHAL, 2017)

2.2.2.1 Filtragem baseada em memória

Segundo (ISINKAYE; FOLAJIMI; OJOKOH, 2015) , os itens previamente avaliados pelo indivíduo são relevantes para busca em um par que compartilha a mesma opinião, dado que exista um vizinho, a partir do descobrimento dele é possível utilizar diversos algoritmos para a combinação de interesses entre os vizinhos a fim de encontrar recomendações. E segundo (CACHEDA et al., 2011) esses algoritmos utilizam um conjunto de dados para calcular as recomendações, utilizando medidas de similaridades para encontrar usuários ou itens semelhantes ao indivíduo que está requisitando a informação, para em seguida calcular a recomendação tendo como fonte as classificações dos grupos ou usuários próximos.

1. **Filtragem baseada no item:** A técnica de filtragem baseada em itens calcula a similaridade entre os itens, a partir de uma matriz usuário-item, são recuperados os itens similares em relação ao item de entrada, baseando-se nos K itens mais semelhantes.

2. **Filtragem baseada no usuário:** A técnica de filtragem baseada no usuário calcula a similaridade entre usuários e não entre itens. Ao encontrar a semelhança entre os usuários, comparando suas avaliações no mesmo item, em seguida é realizada a classificação prevista para aquele item, a fim de fornecer a recomendação de acordo com esta semelhança. (LAKSHMI; LAKSHMI, 2014)

2.2.2.2 *Filtragem baseada em modelo*

Esta técnica utiliza as classificações anteriores para aprender um modelo a fim de melhorar o desempenho. (LAKSHMI; LAKSHMI, 2014) O processo de construção do modelo pode ser feito usando técnicas de aprendizado de máquina ou mineração de dados. Essa técnica usa a classificação anterior dos usuários para aprender um modelo para melhorar o desempenho da técnica de filtragem colaborativa. (PATEL; DESAI; PANCHAL, 2017) O processo de aprendizado de modelo pode ser feito usando técnicas de mineração de dados ou aprendizado de máquina. Exemplos dessas técnicas são decomposição de valor singular, técnica de redução de dimensionalidade, agrupamento, árvore de decisão e mineração de regras de associação. Ele resolve os problemas de esparsidade associados aos sistemas de recomendação. (LAKSHMI; LAKSHMI, 2014)

2.2.3 **Filtragem híbrida**

A técnica de filtragem híbrida é utilizada combinando dois métodos para dar a recomendação aos usuários. Para combinar as técnicas, existem diferentes maneiras, como por exemplo, a implementação individual das técnicas e depois combiná-las na saída, a combinação da técnica baseada em modelo com a técnica de filtragem baseada em memória. (PATEL; DESAI; PANCHAL, 2017) Esta técnica mescla diferentes técnicas de recomendação para maior otimização do sistema para evitar algumas limitações e problemas. O seu funcionamento se dá a partir da combinação das duas técnicas que fornecerão recomendações melhores e mais eficazes do que o algoritmo único, porque a desvantagem de um algoritmo será superada por outro algoritmo. A abordagem comum nas técnicas de recomendação híbridas existentes é mesclar a técnica baseada em conteúdo com outras técnicas de recomendação para evitar problemas de partida a frio, esparsidade e escalabilidade. Existem várias maneiras de combinar dois ou mais métodos na técnica de filtragem híbrida. (ISINKAYE; FOLAJIMI; OJOKOH, 2015)

Tabela 1 – Comparação entre técnicas de recomendação

Método	Usuários	Vantagens	Desvantagens
Filtragem baseada em conteúdo	Baseado em um único usuário	Transparência e independência do usuário	Análise limitada de conteúdo novos usuários
Filtragem colaborativa	Baseado em múltiplos usuários com interesse similares	Melhoria na performance de recomendação	Escassez de dados, Escalabilidade, Sinonímia
Filtragem Híbrida	Combinação entre os interesses de um usuário e o grupo	Evitar esparsividade de problema de partida a frio	Aumento de complexidade, implementação custosa

Fonte: Elaborada pelo autor (2022)

Tabela 2 – Comparação entre técnicas de filtragem colaborativa

Método	Vantagens	Desvantagens
Filtragem baseada em memória	Fácil implementação	O desempenho diminui quando os dados são escassos, exigem muita memória e tempo de CPU
Filtragem baseada em modelo	Melhor esparsidade e escalabilidade, Melhora o desempenho da recomendação, Não requer tanta memória e tempo de CPU	Informações úteis soltas para a técnica de redução de dimensionalidade. Caro.
Filtragem Híbrida	Melhora a esparsidade	Maior complexidade, caro na implementação

Fonte: Elaborada pelo autor (2022)

Segundo o estudo de (PATEL; DESAI; PANCHAL, 2017), o qual realizou uma comparação entre as diferentes técnicas de recomendação tendo a técnica de filtragem como a mais comum em sistemas de recomendação é possível observar na tabela 1 as vantagens e desvantagens de cada técnica.

Ainda de acordo com o estudo de (PATEL; DESAI; PANCHAL, 2017), a técnica de filtragem colaborativa tem mais vantagens sobre outras técnicas. A principal vantagem sobre a técnica de filtragem baseada em conteúdo é que ela melhora o desempenho do sistema de recomendação e fornece melhores recomendações pois também considera os interesses de outros usuários semelhantes e o histórico desses usuários semelhantes para fornecer as recomendações. Na tabela 2 é possível verificar o comparativo entre as técnicas de filtragem colaborativa, baseada em modelo, baseada em memória e no formato híbrido.

2.3 SISTEMAS DE RECOMENDAÇÃO MUSICAL

Normalmente, os sistemas de recomendação consistem em três componentes principais: usuários, itens e processos de correspondência de itens de usuário. Essa estrutura geral também se aplica à sistemas de recomendação musicais, do inglês Music Recommender System (MRS), onde os usuários são os ouvintes e os itens são as músicas. (SONG; DIXON; PEARCE, 2012)

O sistema coleta dados relevantes sobre seus usuários e aplica modelos matemáticos e técnicas diversas para encontrar e propor itens que possam ser de interesse do ouvinte. Para recomendar itens que provavelmente se encaixam nas preferências dos ouvintes, o sistema precisa gerenciar dados sobre ouvintes e itens de música, o sistema gera perfis para cada um deles e determina “boas” correspondências comparando esses perfis. (BAUER; KHOLODYLO; STRAUSS, 2017)

Os algoritmos de correspondência de item de usuário em sistemas de recomendação são tipicamente baseados em uma das seguintes abordagens: filtragem baseada em conteúdo CBF, filtragem colaborativa CF ou abordagens híbridas combinando técnicas de CBF e CF. As abordagens da CBF para MRS exploram descritores de conteúdo de itens, por exemplo, ritmo, andamento, instrumentação, letra, gênero ou estilo de uma música a fim de construir um perfil de usuário. (BAUER; SCHEDL, 2019)

Segundo o estudo de (SCHEDL et al., 2015) , os sistemas de recomendação, de modo geral, tornaram-se atualmente um componente essencial para a navegação aprimorada de grande volume de dados, ajudando assim os usuários a descobrirem “o que eles precisam” dentro do oceano cada vez mais amplo de informação digital. No nosso dia a dia, os recomendadores podem sugerir que tipo de música ouvir no seu serviço de *streaming*. A área de sistemas de recomendação voltada para a música obteve um ganho substancial de interesse tanto da academia quanto através da indústria. Os serviços de *streaming* de música atualmente permitem acesso a dezenas de milhões de músicas.

O tempo em que um usuário leva para consumir e formar a opinião sobre música é muito menor do que o tempo que leva em outras mídias. Este item pode ser consumido repetidamente diversas vezes enquanto outros são consumidos no máximo algumas vezes.

Ao realizar uma filtragem das músicas, com o intuito de facilitar a escolha, os sistemas de recomendação musicais, tendem a ter um sucesso significativo, porém ainda estão longe de serem perfeitos, por frequentemente recomendar itens que não agrada ao usuário. O que implica que um usuário pode não simplesmente tolerar mas apreciar recomendações de itens

previamente conhecidos.

2.4 SISTEMAS DE RECOMENDAÇÃO PARA GRUPOS

O aumento das atividades sociais aumentou o uso de sistemas de recomendação em geral e particularmente aqueles voltados para grupos. Um sistema de recomendação para grupos é um sistema que recomenda itens para um determinado grupo de usuários de forma coletiva, de acordo com suas preferências individuais. Além das preferências do usuário, usar aspectos sociais e comportamentais dos membros do grupo para gerar recomendações de grupo melhora a qualidade do conteúdo recomendado em grupos heterogêneos. Os sistemas de recomendação de grupo também abordam o problema de partida a frio que surge em um sistema de recomendação individual. (DARA; CHOWDARY; KUMAR, 2020)

De acordo com (KADAM, 2020), um sistema de recomendação de grupo analisa os interesses de cada membro do grupo e fornece uma decisão final que seria aceita por todos os membros. Criar recomendações para grupos não é uma tarefa fácil, pois é necessário levar em consideração os gostos e desgostos de cada usuário no grupo antes de fazer qualquer recomendação.

As técnicas de recomendação em grupo são geralmente divididas em três grupos de estratégias, as baseadas na maioria, que utilizam os itens mais populares para a agregação, as baseadas em consenso, onde as preferências de todos os indivíduos do grupo são consideradas e as por borderline que consideram apenas um subconjunto do grupo original.

Tabela 3 – Distribuição das preferências individuais

	A	B	C	D	E	F	G	H	I	J
Maria	3	8	10	8	6	10	4	6	9	10
Enzo	8	9	7	8	9	1	9	6	9	3
Pedro	2	6	9	6	7	10	5	5	8	7

Fonte: Elaborada pelo autor (2022)

Considerando a distribuição de preferências individuais da Tabela 3 sobre um conjunto de 10 itens.

2.4.1 Estratégias baseadas na maioria

1. **Votação de Aprovação:** Apenas as classificações de um item acima do limite de aprovação são contadas. Supondo que o limite seja 5. (KADAM, 2020) A tarefa é determinar o conjunto de itens que aumentam a pontuações individuais de satisfação de todos os usuários do grupo. O objetivo é maximizar a satisfação geral do grupo. (DARA; CHOWDARY; KUMAR, 2020) Portanto, a classificação para o item A será 1, pois apenas uma classificação está acima do nível do limite, enquanto a classificação do item C será 3.
2. **Contagem de Borda:** Os pontos são contados a partir das classificações atribuídas a um item na lista de preferências de um indivíduo, com o item de baixo recebendo 0 pontos, o próximo recebendo um ponto, etc. (KADAM, 2020) A classificação do grupo para o item F será 17; 0 (último para Enzo) + 9 (primeiro para Maria) + 8 (top 3 compartilhado para Pedro).
3. **Regra de Copeland:** O número de vezes que o item supera outros itens (usando votação majoritária) menos o número de vezes que ele perde. (COLOMER, 2013) A classificação do grupo para I é 5; pois vence 7 itens (A,B,D,E,G,H,J) e perde de 2 (C,F).
4. **Votação de pluralidade:** “First Past The Post” é usado; os itens com votos altos são escolhidos todas as vezes. (SALEHI-ABARI; BOUTILIER, 2015) F será escolhido primeiro, pois exibe a classificação mais alta, seguido por C (que tem a segunda classificação mais alta para a maioria).

2.4.2 Estratégias baseadas em consenso

1. **Utilitário Aditivo:** Soma as classificações dos itens após o item com a maior soma ser escolhido. (KADAM, 2020) Item A $\rightarrow 3+8+2 = 13$; Item C $\rightarrow 10+7+9 = 26$
2. **Média:** Obtém a média das classificações individuais. (KADAM, 2020) B terá uma classificação de grupo de 7,67 $\rightarrow (8+9+6)/3$
3. **Média sem miséria:** obtém uma média das classificações individuais após excluir as classificações abaixo de algum nível de limite assumido. (KADAM, 2020) Assumindo limiar igual a 5. E terá uma classificação de grupo de 7,3 (média de 6,9,7), enquanto F é excluído porque Enzo odeia.

4. **Justiça:** As classificações serão dadas a um item como se os usuários os estivessem escolhendo um após o outro. (KAŠŠÁK; KOMPAN; BIELIKOVÁ, 2016) O item F pode ser escolhido primeiro (mais alto para Pedro), seguido por C (mais alto para Maria) e depois B (mais alto para Enzo).
5. **Multiplicativo:** Multiplica todas as avaliações dos usuários. (KADAM, 2020) O item A terá uma classificação de grupo de 48 ($3 \cdot 8 \cdot 2$).

2.4.3 Estratégias limítrofes (borderline)

1. **Least Misery:** Considera a classificação mínima de um item. (DARA; CHOWDARY; KUMAR, 2020) A classificação do grupo para A será 2, pois é a menor entre (3,8,2).
2. **Maior satisfação:** Considera a classificação máxima de um item. (DARA; CHOWDARY; KUMAR, 2020) A classificação do grupo para A será 8, pois é a maior entre (3,8,2).
3. **Pessoa Mais Respeitada (Ditadura):** Considera as classificações dadas pela pessoa mais respeitada do grupo. (KADAM, 2020) Se Pedro for a pessoa mais respeitada, a nota de A será 2. Se Enzo for a pessoa mais respeitada, a nota será 8.

2.5 SISTEMAS DE RECOMENDAÇÃO PARA GRUPOS PARA MÚSICA

Segundo o estudo de (DARA; CHOWDARY; KUMAR, 2020) , voltado para sistemas de recomendação para grupos envolvendo música, o MusicFX (MCCARTHY; ANAGNOST, 1998) é um sistema especializado de recomendação de música para ambientes compartilhados, tais quais as academias de ginástica.

No estudo foram mapeadas as preferências musicais de todos os clientes da academia para realizar o estudo. A aplicação fornece duas interfaces para membros e funcionários, onde os membros podem atualizar suas preferências e dar feedbacks de forma anônima, enquanto a equipe de funcionários pode monitorar o sistema com a permissão de alterar os parâmetros a fim de mudar a estratégia de recomendação. Dado que existe a mudança dinâmica dos usuários monitorada através de um sistema de controle de acesso, o sistema consegue alterar as recomendações de forma dinâmica.

O Flytrap (CROSSEN; BUDZIK; HAMMOND, 2002) é um sistema de recomendação que re-produz músicas com o intuito de satisfazer os usuários ao redor do sistema, acompanhando a

música que está sendo reproduzida no computador do usuário e consegue armazenar os dados, como gênero, artista e etc. Os metadados das músicas são fornecidos de forma automática com a informação de gênero o sistema consegue evitar que os mesmos gêneros e artistas se repitam de forma contínua.

Enquanto o jMusicRecommender (CHRISTENSEN; SCHIAFFINO, 2011), que é um sistema para gerar recomendações para grupos através da mesclagem das recomendações individuais, foram utilizadas uma técnica de mesclagem, uma técnica para gerar preferência de grupo, quatro técnicas de agregação é uma técnica híbrida que inclui mesclagem e agregação são propostas neste trabalho. Cada técnica é avaliada com base na satisfação individual de todos os membros do grupo. Cada música contém alguns atributos, como artista, álbum, idioma, gênero, etc.

Segundo (DARA; CHOWDARY; KUMAR, 2020) dentre todos os algoritmos propostos neste artigo, relata-se que a técnica híbrida apresenta alta satisfação individual entre os usuários de um grupo. Em contraste com a seleção automática de consenso, este modelo dá aos usuários a opção de selecionar o consenso no início.

No estudo de (DARA; CHOWDARY; KUMAR, 2020) mostra que o sistema *SparseRec*, proposto em (KIM; SADDIK, 2015) é um sistema de recomendação de grupo que recomenda música para grupos no sistema de rede social. Aqui, é proposta uma nova abordagem estocástica baseada em grafos que analisa a estrutura do link de forma probabilística. Esse modelo utiliza as tags do sistema de rede social para a recomendação de itens aos grupos. Este modelo suporta dois tipos de recomendações de grupo: a agregação de preferências individuais e cálculo da classificação e o através dos cálculos das classificações individuais e depois combinar as classificações obtidas.

Este método emprega Random Walks with Restarts (RWR) que tem sido amplamente utilizado nas áreas de recuperação de informação, classificação de busca na web, etc. As semelhanças entre um par de usuários ou entre um par de itens são computadas usando similaridade de cosseno. O modelo proposto supera muitos métodos de linha de base quando avaliado usando um conjunto de dados padrão.

As técnicas tradicionais de recomendação de grupo não produzem melhores recomendações quando a matriz usuário-item é esparsa e as preferências do usuário são desconhecidas. (GHAZARIAN; NEMATBAKHS, 2015) O problema de esparsidade de dados é tratado em sistemas de recomendação de grupo de diferentes maneiras. O modelo do *SparseRec* emprega similaridade item-item para preencher a matriz de classificação usuário-item. (DERY et al., 2010)

O modelo de *SparseRec* calcula a similaridade item-item e as similaridades computadas são

usadas para aprimorar as técnicas de recomendação baseadas em memória existentes. O grau de similaridade é um atributo decimal que fica entre 0 e 1, que é um atributo extra adicionado neste modelo a cada par item-item. Inicialmente, a técnica de aprendizado Support Vector Machine (SVM) empregada neste modelo é comparada com outros modelos de regressão, e verifica-se que o SVM supera outros modelos. O modelo proposto supera o modelo de fator latente, que em geral é considerado o melhor método a ser utilizado entre os modelos baseados em memória. (GHAZARIAN; NEMATBAKHS, 2015)

Tabela 4 – Modelo de Agregação dos sistemas de recomendação para grupos com música

Sistema	Modelo de agregação	Preferência do usuário
MusicFX	Voto agregado	Implícito
Flytrap	Voto agregado	Implícito
jMusicRecommender	Multiplicativo, Voto agregado, least misery, justiça	Feedback negativo, Implícito
SparseRec	Least misery, Voto agregado	Implícito
MusicRec	Voto agregado	Implícito

Fonte: Elaborada pelo autor (2022)

2.5.1 Every Noise At Once

Every Noise at Once é “uma tentativa contínua de um gráfico de dispersão gerado por algoritmos e ajustado à legibilidade de o gênero-espaço musical, com base em dados rastreados e analisados para 5.917 gêneros pelo *Spotify*”. (ECHONEXT, 2022)

Segundo o estudo (ORD, 2020) o qual define o *Every Noise At Once* como a abrangência deste campo musical, é espetacularmente representada na página inicial do site como uma vasta constelação de gêneros. Embora “a calibração seja difusa [...] em geral, para baixo é mais orgânico, para cima é mais mecânico e elétrico; a esquerda é mais densa e atmosférica, a direita é mais espetada e saltitante”. O site oferece três métodos principais de organização: o já mencionado “*scatter plot*”, uma visualização de lista e uma lista de reprodução do *Spotify* (“*The Sound of Everything*”). Estas são complementadas por uma pluralidade de outras opções de pesquisa derivadas de algoritmos.

3 METODOLOGIA

Neste capítulo iremos destrinchar a plataforma desenvolvida durante a pesquisa, o deejai, a fim de apresentar em detalhes sua construção passando pelos tópicos de arquitetura utilizada, modelagem das informações, algoritmos implementados, soluções de usabilidade, legislação, segurança e o processo de deploy.

Segundo (COLLIS; HUSSEY, 2005) uma pesquisa exploratória é desenvolvida sobre uma questão ou problema o qual a quantidade de informação disponível previamente sobre o tema é pouca ou indisponível, tendo como objetivo buscar padrões, ideias ou hipóteses. Seu objetivo, como pesquisa exploratória não é avaliar ou afirmar uma determinada hipótese, mas testar quais teorias ou conceitos podem ser aplicados para um problema.

Segundo o estudo de (GIL, 2002) esta pesquisa, por ter como objetivo aumentar a familiaridade com o problema de recomendação para grupos através da filtragem colaborativa, método já utilizado, mas com a adição da informação de dislike do usuário além do pré-carregamento dos dados a fim de evitar um problema na partida a frio do usuário.

Para alcançar o objetivo dessa pesquisa foi desenvolvido uma meta-plataforma para capturar as preferencias musicais do usuário advindas do serviço de streaming que está utilizando como também as informações de quais faixas musicais o usuário não gosta, pois essa informação de dislike não é informada pelos serviços de streaming.

No momento em que um usuário faz login com a sua plataforma, obtemos para usuários do Spotify as 50 músicas mais ouvidas, já no Deezer utilizamos o método de recomendação que a API fornece para preenchermos essa informação do usuário e posteriormente formatamos para uma estrutura de dados que possa ser comum, utilizando como identificador o código ISRC. O ISRC é um código internacional de gravação que é definido pela ISO 3901:2001 e é utilizado para rastrear os direitos autorais de artistas e gravadoras através de metadados de áudio. (TOULSON; GRINT; STAFF, 2013)

Na pesquisa de (DENORA, 2000) a música é vista mais como um processo do que como um objeto, pois auxilia na reconstrução contínua de objetivos durante a realização de atividades baseadas em um contexto, fazendo com que a música seja mais do que um plano de fundo. Para adaptar a criação de contexto para as recomendações foi criada uma estrutura de salas, onde os usuários entram e a média que entram ou saem das músicas são novamente recomendada para os grupos levando em consideração os usuários da sala e seus dislikes individuais. Dentro

de uma sala é possível dois tipos de interação, uma onde o sistema realiza a recomendação com base nos gostos e gêneros e a outra onde é possível que o usuário faça um filtro colaborativo, colocando ele mesmo as músicas para a playlist.

3.1 ARQUITETURA GERAL DO SISTEMA

Segundo (BASS; CLEMENTS; KAZMAN, 2003) , a arquitetura de software de um programa é a estrutura ou estruturas do sistema, que incluem componentes de software, as propriedades visíveis desses componentes, e as relações entre eles. A plataforma foi desenvolvida utilizando a arquitetura cliente-servidor, como descrito na Figura 2, na qual foram desenvolvidos em duas partes uma API que é responsável por se comunicar com os serviços do *Deezer* e do *Spotify* além do processamento das informações advindas da aplicação web e realiza a comunicação com o banco de dados e uma aplicação web para que os usuários possam interagir. O desenvolvimento foi realizado utilizando a linguagem de programação *Typescript*, que é um *superset* da linguagem *JavaScript*, o que significa que as funções para *JavaScript* funcionam no *Typescript*. A vantagem de utilizar o *Typescript* é que além da sintaxe ser a mesma é adicionado a possibilidade de utilizar tipagem das variáveis, onde é possível definir qual tipo um determinado objeto espera ou irá retornar quando chamado. (NODEJS, 2022a)

3.1.1 Back-End

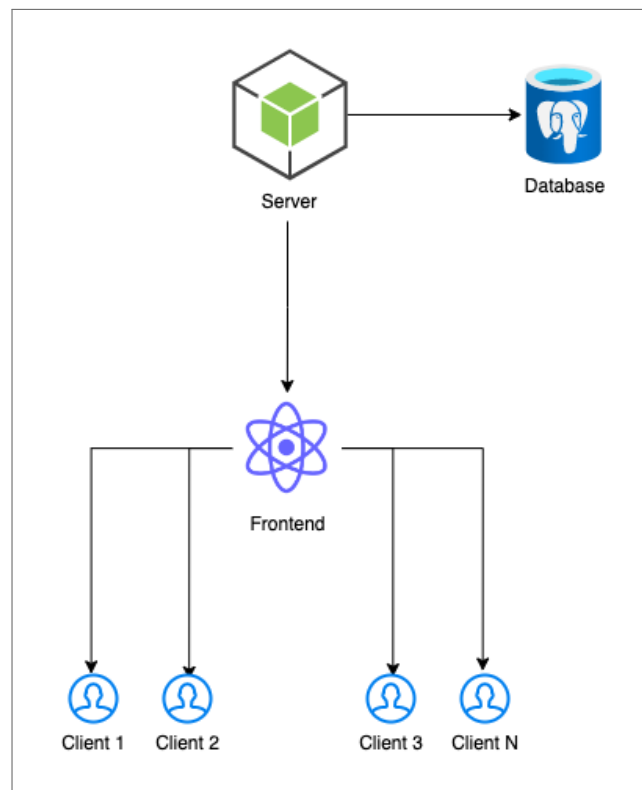
O *back-end* é a parte responsável pelo processamento das informações e foi desenvolvido seguindo a abordagem Representational state transfer (REST), que é um estilo arquitetural para criação de sistemas distribuídos na web.

Segundo (BELQASMI et al., 2012) a abordagem REST é uma API arquitetural que fornece comunicações cliente-servidor para aplicações Web sobre o protocolo Hypertext Transfer Protocol (HTTP). Os três principais princípios de design REST são:

- Endereçamento
- Interface uniforme
- Ausência de estado

O REST aborda a aceitabilidade definindo *endpoints* em uma estrutura de diretório por

Figura 2 – Arquitetura do sistema



Fonte: Do autor. (2022)

meio de Uniform Resource Identifier (URI)s diferentes para extrair os dados. (CHOI, 2012) A API funciona com o princípio de CRUD (Create, Read, Update, Delete), que correspondem às funções Structured Query Language (SQL) de *INSERT*, *SELECT*, *UPDATE* e *DELETE* respectivamente. Chamar um serviço *RESTful* pelo protocolo HTTP pode ser feito em várias linguagens de programação.

Uma API REST pode ser chamada a partir de uma consulta utilizando os verbos HTTP *GET*, para obter dados, *POST*, para adicionar novos elementos, *PUT*, para atualizar todo um recurso, *PATCH*, para atualizar alguns elementos de um recurso e *DELETE* que é usado com o intuito de remover algum recurso. Uma resposta comum de dados está na forma de dados Javascript Object Notation (JSON). A resposta típica do JSON pode ser vista na Figura 3. Os dados também podem ser retornados na forma de Extensible Markup Language (XML). (BELQASMI et al., 2012).

Para desenvolver a API foi utilizado o *NodeJS*, que é uma tecnologia de servidor web baseada em *JavaScript* e que utiliza o *V8 Javascript Engine* com complementos de módulo integrados. (NODEJS, 2022b) Foi utilizado o *Express* como *framework* pela sua referência na comunidade *JavaScript* e por fornecer um conjunto robusto de recursos para aplicações web

Figura 3 – Resposta no formato JSON

```
{
  "tracks": {
    "href": "https://api.spotify.com/v1/search?query=isrc%3ABXZAW2100023&type=track&offset=0&limit=20",
    "items": [
    ],
    "limit": 20,
    "next": null,
    "offset": 0,
    "previous": null,
    "total": 0
  }
}
```

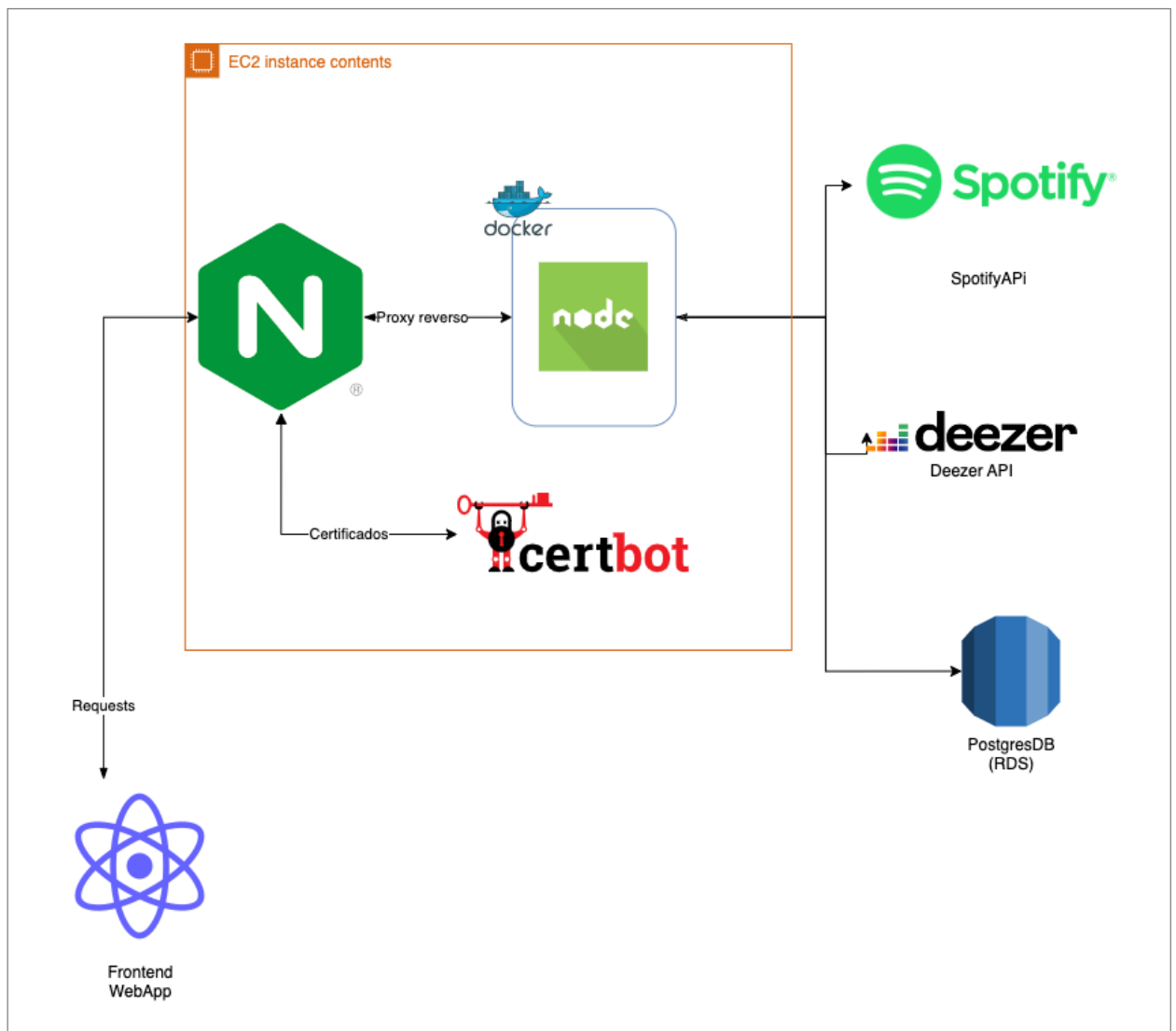
Fonte: Do autor. (2022)

e móvel. (EXPRESS, 2022)

Uma das vantagens do uso do *Express* se dá pela ampla variedade de *middlewares* disponíveis para entrega do software para ambiente de produção. Existe uma lista com boas práticas para implementação de APIs REST. Com o *Express*, foram implementados os seguintes itens.

- **Transport Layer Security (TLS)** A fim de sanar o problema levantado pelo *OWASP Top 10* (OWASP, 2022a) para falhas de criptografia é recomendado que utilize Secure Sockets Layer (SSL)/TLS para comunicação. Implementamos o serviço do *Let's Encrypt* para geração automática de certificados SSL e usamos o *NGINX* como *proxy* reverso para fazer a mudança de porta das requisições HTTPS na porta 443 e encaminhar para a porta que o *container* Docker está executando, como descrito na Figura 4.
- **Helmet** O uso da biblioteca ajuda a proteger a aplicação de algumas vulnerabilidades já conhecidas na web.
- **X-Powered-By** a remoção deste cabeçalho dificulta atacantes a identificar a tecnologia utilizada e realizar ataques direcionados.
- Configuração dos cabeçalhos *Cache-Control* e *Pragma* para desativar o armazenamento em *cache* no lado do cliente.

Figura 4 – Detalhamento do backend



Fonte: Do autor. (2022)

- **X-Content-Type-Options** para evitar que os navegadores procurem por Multipurpose Internet Mail Extensions (MIME) uma resposta a partir do *content-type* declarado.
- **X-Frame-Options** a configuração do cabeçalho para fornecer proteção contra *clickjacking*, que é uma técnica maliciosa que induz um usuário a clicar em algo diferente do que ele vê.
- **X-XSS-Protection** para ativar o filtro de Cross-Site Scripting (XSS) nos navegadores da web mais recentes.

3.1.2 Front-end

A aplicação que permite o acesso do usuário foi desenvolvida utilizando o *React*, que é uma biblioteca *JavaScript* que é implantada para desenvolver componentes de User Interface (UI) reutilizáveis. De acordo com documentação oficial do *React*, é uma biblioteca para construir interfaces de usuário modulares. O que significa que cada parte de uma interface do usuário mais complexa é composta de partes isoladas menores chamadas de componentes. Isso torna seu código mais previsível e fácil de depurar, em outras palavras, mais fácil de manusear. (FACEBOOK, 2021)

React basicamente permite o desenvolvimento de aplicações grandes e complexas na web e que tem a possibilidade de alterar seus dados sem atualizações das páginas. (AGGARWAL, 2018)

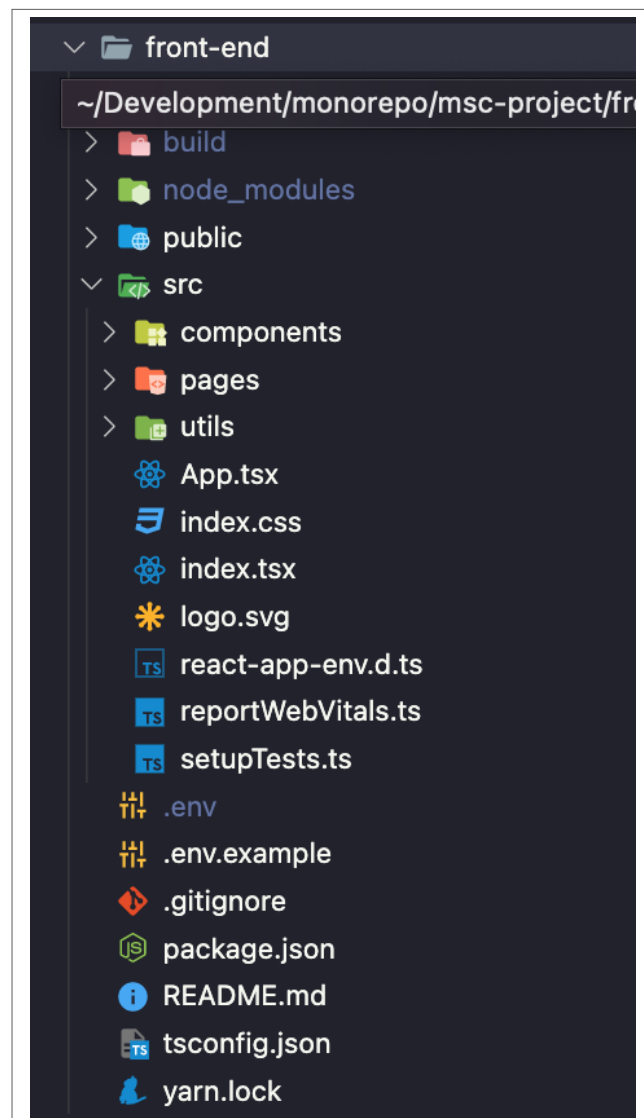
JavaScript XML (JSX) é uma extensão para JavaScript com sintaxe semelhante a XML. *React* utiliza JSX para produzir componentes *React*. Não é uma necessidade ao usar o *React*, mas é uma ferramenta útil em uma perspectiva visual e também permite que o *React* exiba mensagens de aviso e erro mais detalhadas e funcionais. O JSX também tem outras vantagens, como evitar ataques de injeção. (JSX, 2022)

Neste projeto foi utilizado a biblioteca *styled-components* para a criação dos componentes, pois existe com ela a possibilidade de escrever Cascading Style Sheets (CSS) no formato padrão sendo ela é responsável por adicionar esse componente dentro do *React*.

Ao utilizar a biblioteca *styled-components* os componentes serão renderizados em uma página e será adicionado apenas o estilo que será utilizado de forma automática e essa operação tem o intuito de diminuir a quantidade de código gerado. Um problema que pode acontecer, quando não utiliza essa biblioteca, é com a nomenclatura das classes que no JSX padrão pode haver choque na nomenclatura da classe, com o *styled-components* é gerado um nome único para o componente dessa forma não haverá sobreposição de nome das classes e assim ocorreria problemas com os estilos que seriam carregado em uma página.

A aplicação foi iniciada utilizando o Create React App (CRA) que é uma estrutura com suporte oficial que é usada para criar aplicativos de página única, do inglês Single-page application (SPA) sem a necessidade de fornecer sua própria configuração. Ao desenvolver um aplicativo que é inicializado com o CRA, os pacotes que estão sendo implantados serão otimizados automaticamente. O *Create React App* suporta todos os navegadores modernos. (FACEBOOK, 2022)

Figura 5 – Estrutura de pastas do frontend

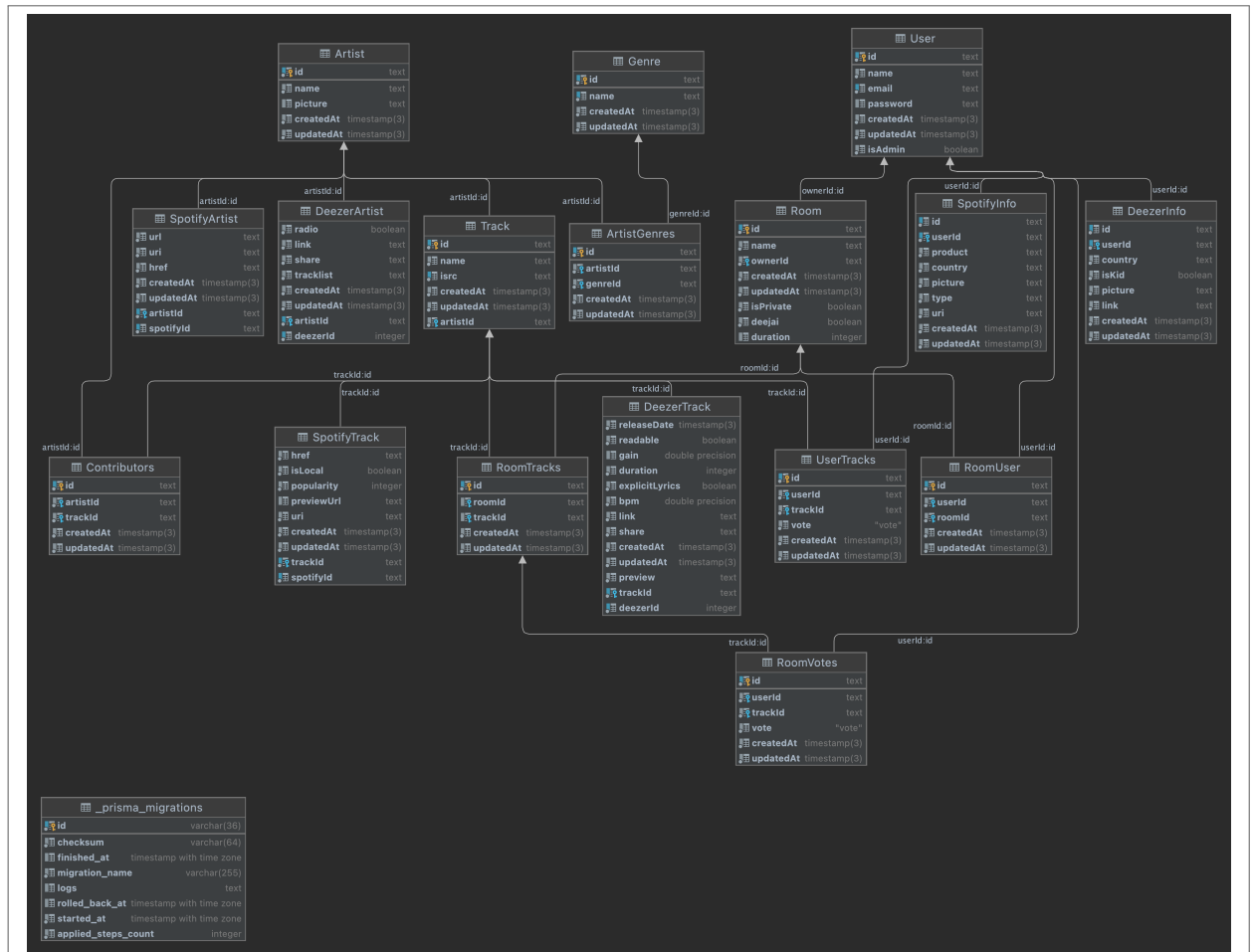


Fonte: Do autor. (2022)

O CRA requer um diretório chamado *public* com um arquivo chamado *index.html* e um diretório chamado *src* com o ponto de entrada *JavaScript index.js*. Qualquer outro arquivo é livre para ser modificado ou removido. (JOHANSSON, 2021) A estrutura do projeto ficou como exibido na Figura 5.

3.1.3 Banco de dados

Figura 6 – Diagrama ER da plataforma



Fonte: Do autor. (2022)

O banco de dados escolhido foi o *PostgreSQL* por ser um sistema de banco de dados relacional de código aberto e que utiliza a linguagem SQL, por ter uma reputação forte sobre sua arquitetura, confiabilidade, integridade, além de conseguir ser executado em diversos sistemas operacionais. (POSTGRESQL, 2022) A modelagem foi feita separando as seguintes entidades e pode ser observada na Figura 6.

- **Artist** Informações sobre o artista como, por exemplo nome e foto.
- **ArtistGenres** Tabela ternária para relacionar os gêneros do artista.
- **Contributors** Tabela para relacionar a música com os artistas que fizeram participação na música.

- **DeezerArtist** Informação do artista no *Deezer* como id na plataforma, *link* para Uniform Resource Locators (URL) do perfil e a playlist do artista, que é chamada de rádio.
- **DeezerInfo** Tabela que contém informações do usuário no *Deezer* para vincular com o perfil da plataforma.
- **DeezerTrack** Informação sobre a música na plataforma do *Deezer*, como id e a URL para preview.
- **Genre** Tabela com informações sobre os gêneros, fornecida pelo *Spotify* e vinculada com a plataforma do *Every Noise At Once*.
- **Room** Informação sobre a sala, com o usuário que criou, o dono, se é pública ou privada e a duração da recomendação (limite de tempo em minutos).
- **RoomTracks** Tabela ternária para vincular as faixas que estão na sala.
- **RoomUser** Tabela ternária para vincular quais usuários são membros da sala para computar a recomendação.
- **RoomVotes** Tabela para armazenar quando a sala não for de geração automática, mas sim um filtro colaborativo, informar quais músicas os usuários votaram positivo ou negativamente.
- **SpotifyArtist** Informação sobre o artista advinda da plataforma *Spotify*, como id e URL.
- **SpotifyInfo** Informação do usuário na plataforma *Spotify*.
- **SpotifyTrack** Informação sobre a música na plataforma do *Spotify*, como a URL para *preview*.
- **Track** Tabela para armazenar a informação sobre o nome e o International Standard Recording Code (ISRC) da música.
- **User** Tabela para armazenar as informações básicas do usuário como o nome, email que é usado como chave entre os serviços de *streaming*.
- **UserTracks** Tabela ternária para armazenar as músicas que o usuário interagiu seja de forma positiva curtindo, negativa dando *dislike* ou que teve interação em uma sala ou recomendação do *streaming*, armazenada como neutra.

3.1.3.1 Prisma

Com o intuito de evitar falhas de segurança listadas na *OWASP Top 10*, na camada de dados foi utilizado uma biblioteca de Object-relational mapping (ORM), onde ela é responsável por se comunicar com o banco de dados de forma confiável, além de auxiliar na criação e manipulação de tabelas, com o uso de sanitizadores de dados e rastreabilidade de ações. (OWASP, 2022b) Na Figura 7 é possível verificar como uma tabela é criada no *schema* e na Figura 8 como ela é gerada no banco de dados. A utilização de um ORM é a facilidade de migrar de banco de dados caso necessite, pois as chamadas dentro do código são abstraídas de códigos SQL, utilizando apenas a própria sintaxe da biblioteca. (PRISMA, 2022)

Figura 7 – Construção da Entidade User no Schema do Prisma

```
You, 2 weeks ago | 1 author (You)
// This is your Prisma schema file,      You, 2 months ago • ✨ login with deezer
// learn more about it in the docs: https://pris.ly/d/prisma-schema

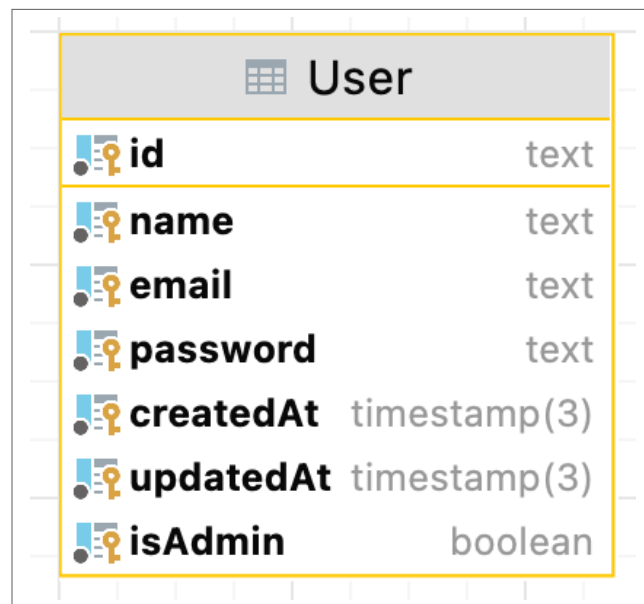
generator client {
  provider = "prisma-client-js"
}

datasource db {
  provider = "postgresql"
  url      = env("DATABASE_URL")
}

model User {
  id          String      @id @default(cuid())
  name        String
  email       String      @unique
  password    String?
  spotify     SpotifyInfo?
  deezer      DeezerInfo?
  rooms       Room[]      @relation("owner")
  memberRooms RoomUser[]  @relation("member")
  votes       RoomVotes[] @relation("userVotes")
  tracks      UserTracks[]
  isAdmin     Boolean      @default(false)
  createdAt   DateTime     @default(now())
  updatedAt   DateTime     @default(now())
}
```

Fonte: Do autor. (2022)

Figura 8 – Tabela usuário gerada no banco de dados



User	
id	text
name	text
email	text
password	text
createdAt	timestamp(3)
updatedAt	timestamp(3)
isAdmin	boolean

Fonte: Do autor. (2022)

3.2 FONTE DE DADOS

Para a elaboração desta pesquisa, utilizamos como fonte de dados as músicas que cada indivíduo mais escuta no Spotify, através do endpoint que retorna as músicas que o usuário tem consumido, com um limite de 50 faixas de áudio no Spotify, mas no Deezer não existe um endpoint similar para buscar as faixas mais ouvidas pelo usuário, porém existe um endpoint que retorna 5 recomendações para aquele usuário com base nas músicas que ele já ouviu.

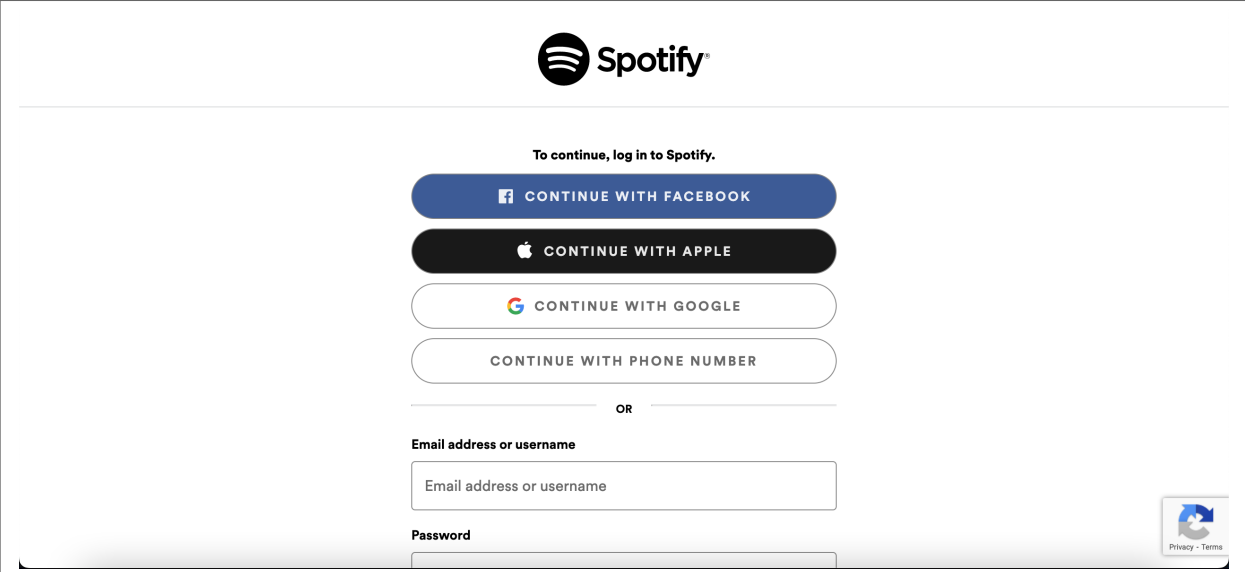
A fim de evitar o problema da partida a frio, comum em sistemas de recomendações, utilizamos esses dois endpoints para criar uma base de dados inicial para aquele usuário. Os dados são complementados com as informações de gênero vindas do Every Noise At Once, o site que faz uma correlação entre os gêneros por proximidade, desta forma quando sincronizamos as informações musicais, pegamos também as informações dos artistas da música, como o gênero musical e relacionamos a informação do genero com os gêneros mapeados pelo Every Noise At Once. (ECHONEXT, 2022)

3.2.1 Spotify

Os dados disponibilizados pelos serviços de streaming precisaram ser tratados para serem armazenados pela plataforma. Para usar a plataforma do Spotify, é necessário a configura-

ção dos escopos de autenticação. Os escopos fornecem aos usuários do Spotify que desejam utilizar aplicativos de terceiros a garantia de que apenas as informações que eles escolherem compartilhar serão compartilhadas e nada mais. Os escopos necessários para cada caso podem ser vistos na Tabela 5. Para acessar esses recursos é necessário que o usuário autorize a aplicação, durante o processo de autenticação. Esta autorização pode ser visualizada na Figura 9 (SPOTIFY, 2021)

Figura 9 – Tela de login com Spotify

The image shows the Spotify login interface. At the top center is the Spotify logo. Below it, the text "To continue, log in to Spotify." is displayed. There are four large, rounded buttons for social login: "CONTINUE WITH FACEBOOK" (blue), "CONTINUE WITH APPLE" (black), "CONTINUE WITH GOOGLE" (white with Google logo), and "CONTINUE WITH PHONE NUMBER" (white). Below these buttons is a horizontal line with the word "OR" in the center. Underneath the line, there are two input fields: "Email address or username" and "Password". In the bottom right corner, there is a small icon for "Privacy - Terms".

Fonte: Do autor. (2022)

Tabela 5 – Escopos disponibilizados pela API do Spotify

Escopo	Entidade	Descrição
ugc-image-upload	Imagens	Permissão de envio de imagens para capas de playlists
user-modify-playback-state	Aplicativo	Controle a reprodução em seus clientes Spotify e dispositivos Spotify Connect.
user-read-playback-state	Aplicativo	Obter a informação do que está sendo reproduzido no momento
user-read-currently-playing	Aplicativo	Obter a informação do que está sendo reproduzido no momento além das informações de quais dispositivos estão reproduzindo no momento
user-follow-modify	Seguidores	Modificar quem está seguindo usuários ou artistas
user-follow-read	Seguidores	Acessar seguidores e lista de quem está seguindo
user-read-recently-played	Histórico	Acessar itens tocados recentemente
user-read-playback-position	Histórico	Ler a informação de qual posição o player está tocando no momento
user-top-read	Histórico	Obter a listagem de músicas mais tocadas pelo usuário
playlist-read-collaborative	Playlists	Acessar as playlists colaborativas
playlist-modify-public	Playlists	Gerenciar playlists públicas
playlist-read-private	Playlists	Acessar playlists privadas
playlist-modify-private	Playlists	Gerenciar playlists privadas
app-remote-control	Playback	Comunicação com o dispositivo
streaming	Playback	Reproduzir o conteúdo em outro dispositivo (web) apenas premium
user-read-email	Usuários	Ler a informação do email do usuário
user-read-private	Usuários	Ler a informação do perfil do usuário
user-library-modify	Biblioteca	Gerenciar conteúdo salvo
user-library-read	Biblioteca	Ler conteúdo salvo

Fonte: Elaborada pelo autor (2022)

Para coletar os dados do Spotify é necessário entender a modelagem dos relacionamentos e seus domínios. Os domínios que o Spotify utiliza na sua modelagem são:

- Albums
- Artists

- Browse
- Episodes
- Follow
- Library
- Markets
- Personalization
- Player
- Playlists
- Search
- Tracks
- Shows
- Users Profile

3.2.2 Deezer

O serviço de streaming Deezer, possui uma modelagem diferente do Spotify, para acessar os dados de uma conta é necessário requisitar um token de acesso e posteriormente validar esse token, após a validação um novo token é criado e será válido por 24 horas. Esse processo é dividido em 3 etapas Autenticação do usuário, a fim de verificar a identidade do usuário, Autorização do aplicativo, com o intuito de mostrar qual aplicação está requisitando aqueles dados e Autenticação do aplicativo que tem o intuito de apresentar para o usuário o que a aplicação solicita de permissão, esta solicitação pode ser vista na Figura 10. (DEEZER, 2021a) A listagem de permissões pode ser vista na Tabela 6. (DEEZER, 2021b)

3.2.3 Funcionalidades utilizadas

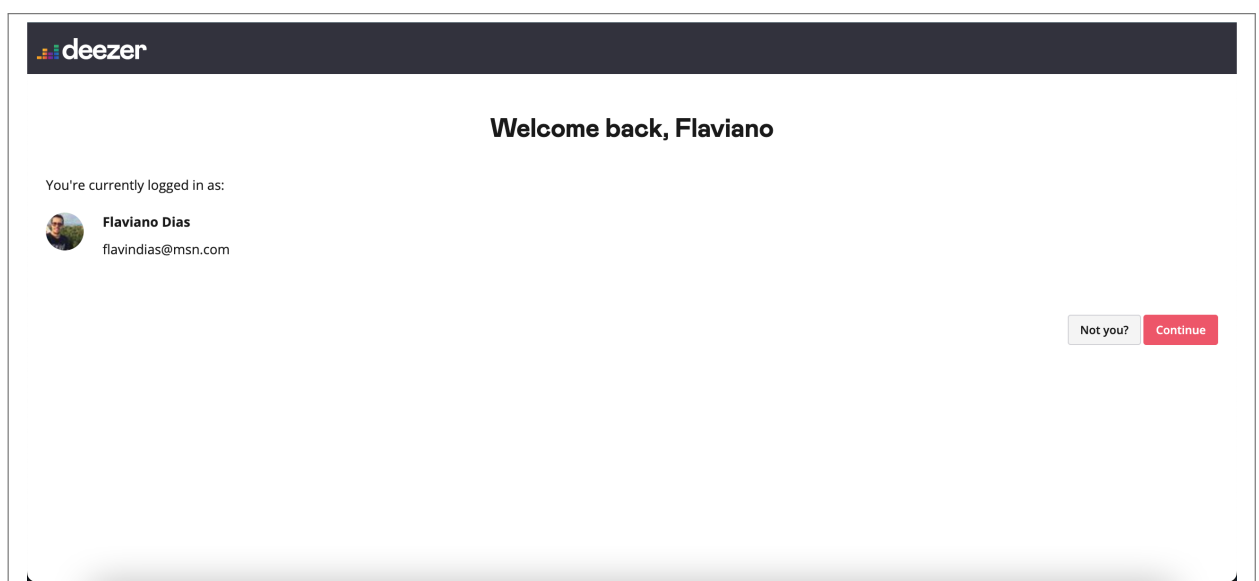
Para o desenvolvimento da plataforma foi necessário realizar um levantamento dos endpoints disponíveis para obter as informações necessárias, para realizar as recomendações, esses endpoints podem ser vistos e detalhados na Tabela 7 e na Tabela 8.

Tabela 6 – Permissões disponibilizados pela API do Deezer

Nome	Título	Descrição
basic_access	Acesso a informações básicas	Nome, sobrenome e foto
email	Acesso ao email do usuário	
offline_access	Acessar os dados do usuário quando não conectado a rede	A aplicação pode acessar os dados a qualquer momento
manage_library	Gerenciar a biblioteca do usuário	Adicionar e remover ou mudar a ordem de uma playlist
manage_community	Gerenciar lista de amigos do usuário	Adicionar e remover um outro usuário
delete_library	Remover itens da biblioteca	Permissão para remover item da biblioteca do usuário
listening_history	Permite que acesse o histórico de reproduções do usuário	

Fonte: Elaborada pelo autor (2022)

Figura 10 – Tela de login com Deezer



Fonte: Do autor. (2022)

Tabela 7 – Endpoints utilizados da API do Deezer

Método	Endpoint	Descrição
GET	/user/me	Obter informação do usuário logado
GET	/user/me/recommendations /tracks	Obter a listagem de músicas recomenda- das pelo Deezer para o usuário logado.
GET	/track/\${id}	Obter informação de uma determinada música
GET	/2.0/track/isrc:\${isrc}	Buscar informação de uma música base- ado no ISRC
POST	/user/me/playlists	Cria uma playlist
POST	/playlist/\${playlistId}/tracks	Adiciona itens a uma playlist

Fonte: Elaborada pelo autor (2022)

Tabela 8 – Endpoints utilizados da API do Spotify

Método	Endpoint	Descrição
GET	/v1/me/top/tracks	Obter as músicas mais ouvidas
GET	/v1/artists/{id}	Obter a informação de um determinado artista
GET	/v1/search?type=track &q=isrc:\${isrc}	Buscar informação de uma música base- ado no ISRC
GET	/v1/me	Obter informação do usuário logado
POST	/v1/playlists/\${playlistId} /tracks	Adiciona itens a uma playlist
POST	/v1/users/\${userId}/playlists	Cria uma playlist

Fonte: Elaborada pelo autor (2022)

3.3 TRATAMENTO DOS DADOS

A fim de eliminar a possibilidade de existir música repetidas, o código ISRC foi utilizado como chave única, desta forma, antes de registrar uma música nova é consultado no banco se existe uma música já armazenada e só irá salvar apenas se esta música não exista. Para as informações sobre os usuários o campo de e-mail também foi utilizado como chave única, logo caso um usuário possua acesso aos dois serviços de *streaming* os dois serviços serão vinculados ao mesmo usuário.

Para as informações específicas de um artista, música ou usuário são gerados durante a criação identificadores únicos para a plataforma *deejai* e armazenados nas tabelas respectivas do *Deezer* ou *Spotify*, como chave estrangeira.

As informações exclusivas desses *streamings* mas associando com o identificador gerado pela plataforma com o intuito de verificar se a informação já foi adicionada através das operações SQL de *JOIN*.

As informações relacionadas ao gênero do artista são checadas com a tabela *Genres* que possui a informação de posicionamento espacial X e Y advindas do *Every Noise At Once*.

Quando um usuário faz *login* são buscadas as suas músicas, as 50 mais ouvidas caso autenticação tenha sido com o *Spotify* ou 15 músicas recomendadas caso o usuário se autentique através do *Deezer* baseado no seu perfil, para cada música é checado se a mesma já está cadastrada na base, checando os artistas e as informações exclusivas da plataforma.

Por exemplo, uma música caso esteja associada apenas a usuários que possuem conta no *Spotify* não terá informação do *Deezer*, até que um usuário que possua conta no *Deezer* entre em contato com ela, seja ao compartilhar uma sala ou que ela seja recomendada para ele pelo *Deezer*.

3.4 GRUPO FOCAL

Segundo (CAPLAN, 1990), grupos focais são um pequeno agrupamentos de pessoas reunidas com o intuito de avaliar conceitos ou identificar problemas. Já segundo (VAUGHN; SCHUMM; SINAGUB, 1996) grupo focal é uma técnica qualitativa que pode ser usada em combinação com outras técnicas qualitativas ou quantitativas, com o intuito de aprofundar o conhecimento das necessidades de usuários.

Segundo (DIAS, 2000) o objetivo central do grupo focal é identificar percepções, sentimen-

tos, atitudes e idéias dos participantes a respeito de um determinado assunto. Já os objetivos específicos podem variar de acordo com a abordagem da pesquisa. O grupo focal é iniciado com o conjunto de 6 a 10 pessoas reunidas com base em suas características, homogêneas ou heterogêneas, em relação ao assunto a ser tratado. É importante destacar que grupos com menos de 6 pessoas as ideias e interações tendem a ser mais esparsas e existe uma maior probabilidade de algumas pessoas se sentirem intimidadas pelos mais extrovertidos e quando grupos possuem mais de 10 pessoas são mais difíceis de serem gerenciados quanto ao foco da discussão e à distribuição do tempo disponível para a participação efetiva de todos.

3.5 EXPERIMENTO

O experimento consistiu em separar três grupos de 6 pessoas que possuísem contas ativas no Spotify ou Deezer com o intuito de gerar uma playlist de 60 minutos e verificar a aceitação para cada indivíduo do grupo com as músicas recomendadas. Tendo como característica na playlist a impossibilidade de pular qualquer faixa recomendada. Durante a primeira interação a fim de evitar o problema da partida a frio, foi definida uma técnica de recomendação híbrida, onde ao entrarem na plataforma a depender da forma de login são sincronizadas as 50 músicas mais ouvidas, no caso do Spotify, ou 15 recomendações do Deezer. A interação dos membros do grupos é dada através de uma sala, ao entrar na sala as recomendações individuais são somadas, nessa etapa não há filtragem dos itens que o usuário não gosta, após entrar na sala é possível avaliar as músicas recomendadas, para que em futuras interações que o usuário participar essas músicas não entrem na recomendação. Para a segunda interação do grupo focal, separada por pelo menos 7 dias, a fim de minimizar a interferência do conhecimento prévio gerado pela playlist da primeira interação. Durante a segunda interação as recomendações são geradas para o grupo em uma nova sala, para haver a separação entre as playlists da primeira e segunda interação. O experimento na segunda interação, diferente da primeira, levou em consideração as músicas que os usuários não gostam para gerar a playlist para todo o grupo, mantendo as mesmas 6 pessoas do grupo e os 60 minutos de música. A recomendação nessa etapa levou em consideração a sala anterior das músicas que foram ouvidas removendo as que tiveram uma avaliação média abaixo de 4. As músicas que entraram na seleção levaram em consideração as músicas previamente aprovadas a fim de maximizar a aceitação dos usuários, a técnica de recomendação foi a combinação das músicas previamente aprovadas pelo grupo com outras músicas que os usuários gostam que possuem o estilo musical semelhante ou próximo,

segundo a classificação espacial do EveryNoiseAtOnce. Foram aplicados três questionários para o grupo focal, um ao término da primeira interação, outro para avaliar a usabilidade do sistema é um terceiro para avaliar a segunda interação e tentar apurar se levar em consideração se a informação de músicas que não são apreciadas individualmente pelos usuários melhora a experiência de consumo musical em grupo.

3.6 QUESTIONÁRIOS

3.6.1 Avaliação da playlist

Com o grupo focal foi realizado para cada música da playlist a pergunta do quanto aquela música agradava o usuário individualmente. Os 6 membros de cada grupo focal precisaram ouvir todas as músicas da playlist, sem a opção de pular, juntos e em sequência e para cada música avaliar quanto aquela música com uma nota de 1 a 5 essa informação foi preenchida em uma tabela do Google planilhas. A seleção inicial se deu a partir das concatenação das músicas que mais ouviam (caso usuário Spotify) ou da recomendação do Deezer (para usuários dessa plataforma) totalizando 60 minutos. Após 7 dias as músicas que obtiveram uma avaliação média abaixo de 4 pontos foram removidas e adicionadas novas músicas para completar a playlist. Com a nova playlist gerada o grupo focal ouviu novamente a playlist completa, novamente sem a opção de pular, e reavaliou todas as músicas com uma nota de 1 a 5. Com o intuito de identificar se a qualidade geral da playlist melhoraria ao remover as músicas que os usuários não gostaram.

3.6.2 System Usability Scale

A Escala de Usabilidade do Sistema, ou System Usability Scale (SUS), foi criada em 1986 por John Brooke e tem sido amplamente utilizada para testar vários aplicativos e sistemas. O SUS é uma ferramenta agnóstica de tecnologia composta por dez perguntas com cinco respostas para cada pergunta, variando de "concordo totalmente" a "discordo totalmente". (KLUG, 2017)

As dez perguntas são as seguintes:

- Acho que gostaria de usar este sistema com frequência.

- Achei o sistema desnecessariamente complexo.
- Achei o sistema fácil de usar.
- Acho que precisaria do apoio de um técnico para poder utilizar este sistema.
- Achei que as várias funções neste sistema estavam bem integradas.
- Achei que havia muita inconsistência neste sistema.
- Eu imagino que a maioria das pessoas aprenderia a usar este sistema muito rapidamente.
- Achei o sistema muito complicado de usar.
- Eu me senti muito confiante ao usar o sistema.
- Eu precisei aprender muitas coisas antes de poder usar este sistema.

O SUS gera no final uma nota de 0 a 100. (GRIER et al., 2013) De acordo com (BANGOR; KORTUM; MILLER, 2008), uma pontuação de 85 ou superior representa usabilidade excepcional e uma pontuação abaixo de 70 representa usabilidade inaceitável. Vários estudos demonstraram que o SUS é altamente confiável.

O cálculo do score do SUS é dado através da soma da contribuição de cada item. Para valores ímpares deve subtrair 1 da resposta, para valores pares deve subtrair 5. Após obter o escore de cada item soma-se os valores e multiplica o resultado por 2,5. Este resultado serra o índice de satisfação do usuário e varia de 0 à 100. (GERALDES; MARTINS; AFONSECA, 2019)

3.7 LIMITAÇÕES

A API do Spotify possui limites de requisições por aplicativo e por IP, algumas requisições precisaram ser feitas do lado do frontend para obter informações das faixas como as características do audio (AudioFeatures), os estilos que o artista de cada música que o usuário possui na sua listagem de top 50 músicas mais ouvidas e informações mais detalhadas do próprio artista, como a foto do artista. Existe um limite de assinaturas para requisitar um token, pois o comportamento esperado é que o usuário peça a cada requisição um escopo bem definido de informações que serão consumidas por um determinado ponto de acesso da API. A recuperação das informações no Spotify possuem um limite de requisições por segundo, desta forma desenvolvemos um banco de dados para manter esses dados armazenados realizando a

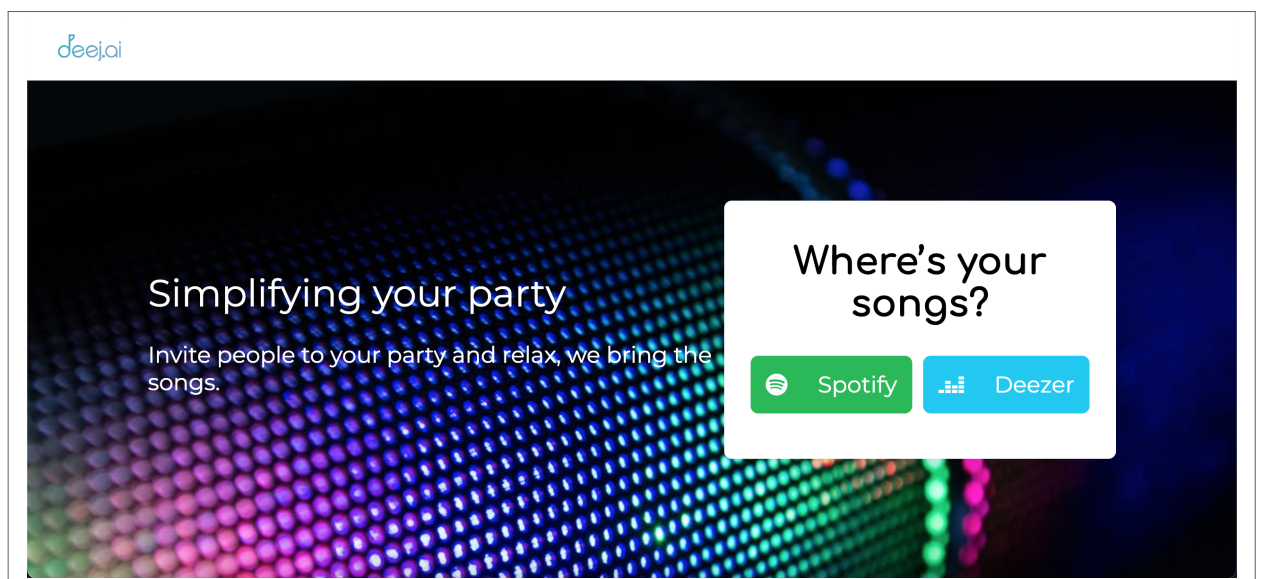
busca na API apenas quando a informação não é encontrada dentro do banco da aplicação. Para gerar a recomendação utilizamos o mapeamento cartesiano do Every Noise At Once, que utiliza uma aproximação dos estilos com base nas informações das características de audio das faixas, porém essa classificação é realizada agrupando os estilos dos artistas, que por sua vez possuem a liberdade de adicionar os estilos que mais se encaixam com a estratégia de divulgação. Uma solução para melhor recomendações é vincular os estilos as faixas ao invés da associação com o artista, pois não é incomum um artista a medida que vai avançando o tempo de carreira transitar em diferentes estilos musicais.

4 DISCUSSÃO DOS RESULTADOS

Nesta seção abordaremos os resultados das implementações das técnicas de sistemas de recomendação para grupos, como descritas no capítulo 2.

Os desafios destacados que são comuns em sistemas de recomendação foram a falta de dados para a recomendação inicial (partida a frio), a mudança de preferências de um usuário ao longo do tempo, e com o intuito de superar esses dois desafios, foi implementado na plataforma um sincronizador, onde todas as vezes que um usuário acessa a plataforma o sistema é responsável por buscar as informações do seu serviço de *streaming* a fim de diminuir a chance de não haver dados iniciais ou dados desatualizados, para realizar esta ação basta o usuário selecionar de qual *streaming* ele deseja importar as suas músicas, como pode ser visto na Figura 11. Além da sincronização automática é possível que o usuário, a qualquer

Figura 11 – Tela inicial da aplicação (Login)

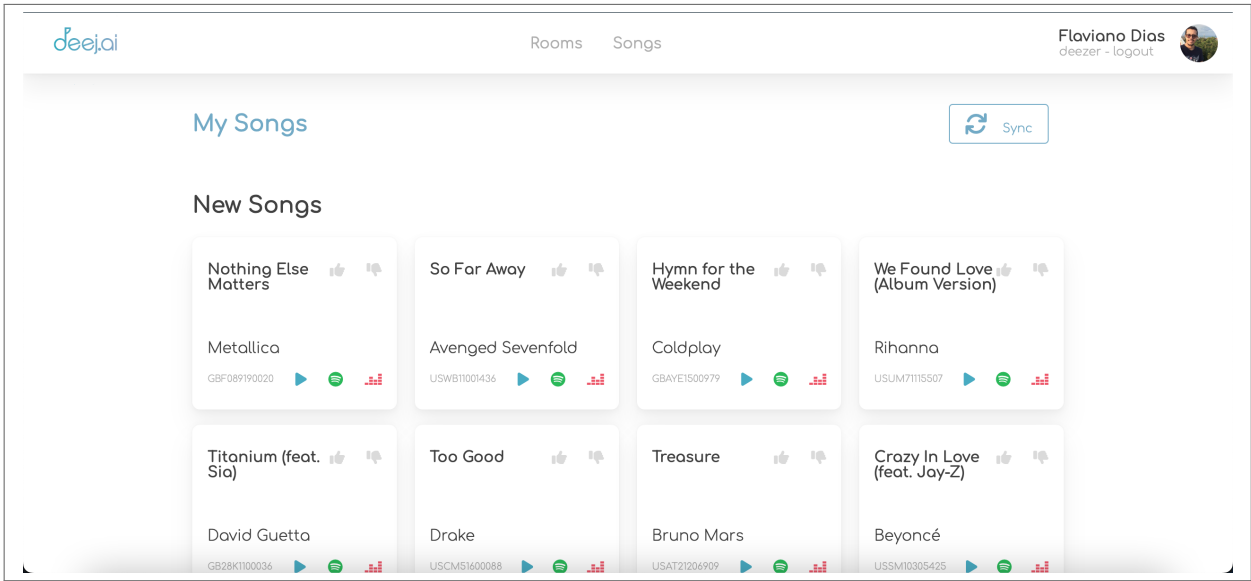


Fonte: Do autor. (2022)

momento, sincronize as suas músicas na aba de *Songs* clicando no botão Sync, como pode ser observado na Figura 12. O algoritmo é diferente para usuários do *Deezer* e do *Spotify*, pois a API do *Deezer* conta com um sistema de recomendação que a cada iteração recomenda novos itens para aquele usuário, tendo como base o comportamento recente na plataforma. Para usuários do *Spotify* a API possui um *endpoint* que retorna apenas a listagem das músicas mais ouvidas pelo usuário, tendo como limite máximo 50 faixas.

A plataforma foi desenvolvida utilizando a tecnologia de *containers Docker*, desta forma

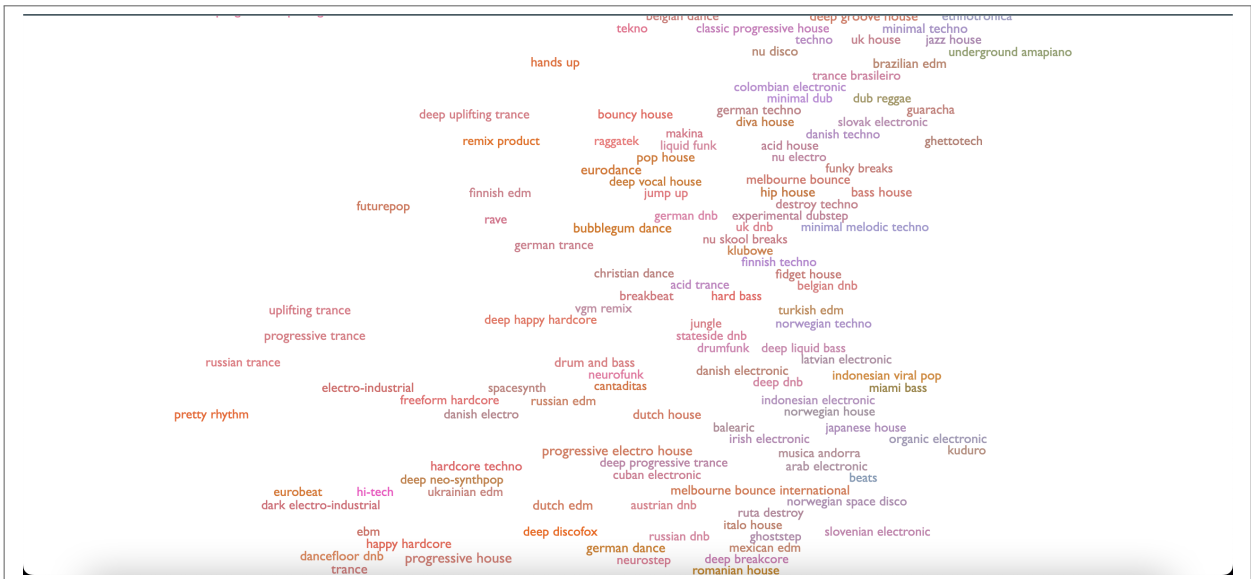
Figura 12 – Listagem de músicas do usuário



Fonte: Do autor. (2022)

podemos escalar a plataforma apenas aumentando a quantidade de *containers* responsáveis pelo *backend* da aplicação, evitando assim o problema comum em sistemas de recomendação que é a escalabilidade. Para o desafio da sinonímia utilizamos o estudo do *Every Noise At Once* onde mapeia a proximidade dos gêneros de forma espacial atribuindo um deslocamento superior e esquerdo para posicionar os estilos como pode ser visualizado na Figura 13. Ao

Figura 13 – Reprodução Every Noise At Once

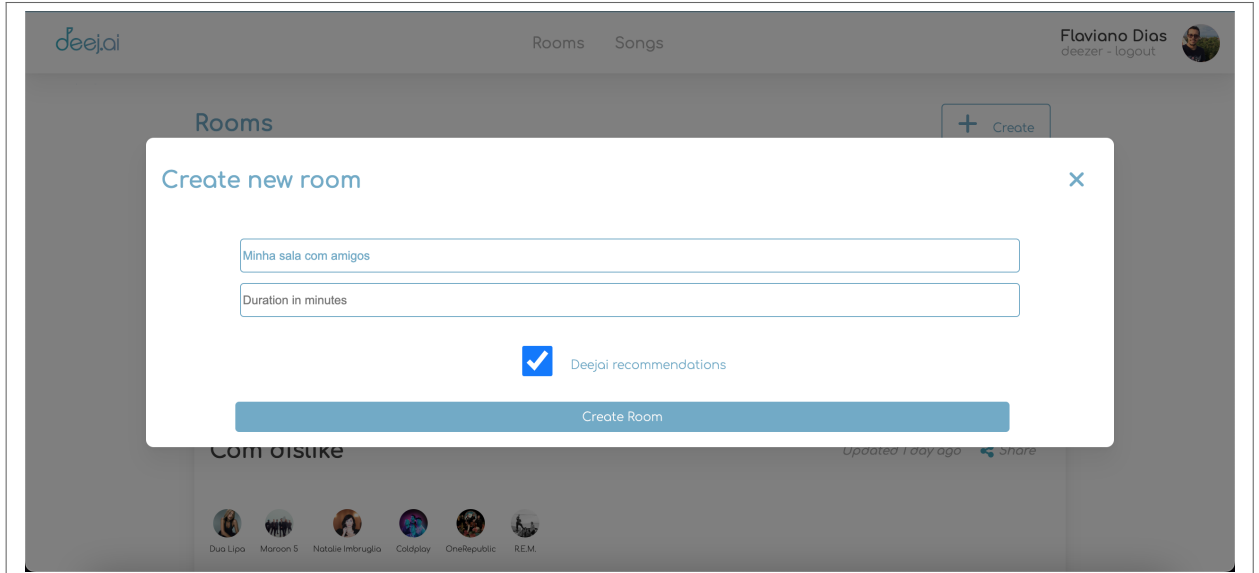


Fonte: Every Noise At Once. (2022)

sincronizar a informação de uma música é capturado também as informações dos artistas que participam daquela música e os gêneros que aquele artista tem, vinculando assim a tabela de

gêneros dos artistas (*ArtistGenre*) com a tabela de gêneros (*Genre*) que foi criada com base nos gêneros coletados do *ENAO*.

Figura 14 – Criação de uma sala na plataforma

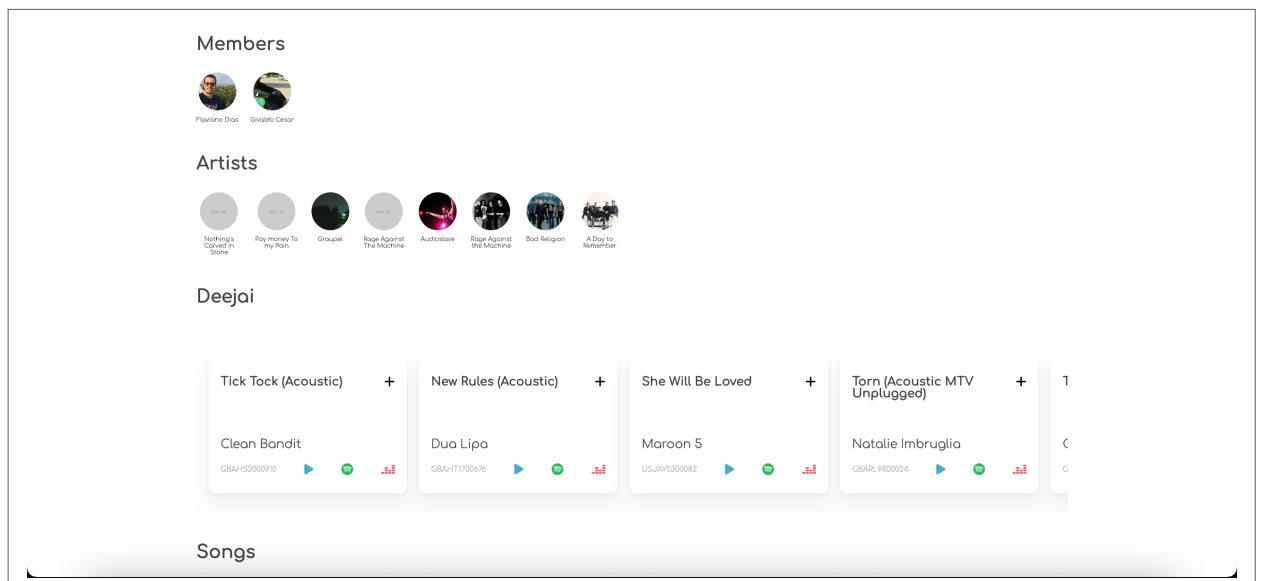


Fonte: Do Autor. (2022)

De posse desta informação o usuário pode criar uma sala, como visto na Figura 14, com uma recomendação automática ou com um filtro colaborativo que conta com o aprendizado prévio dos *dislikes* daquele usuário, com o intuito de filtrar a listagem inicial de músicas mais ouvidas ou recomendadas pelo serviço de *streaming*.

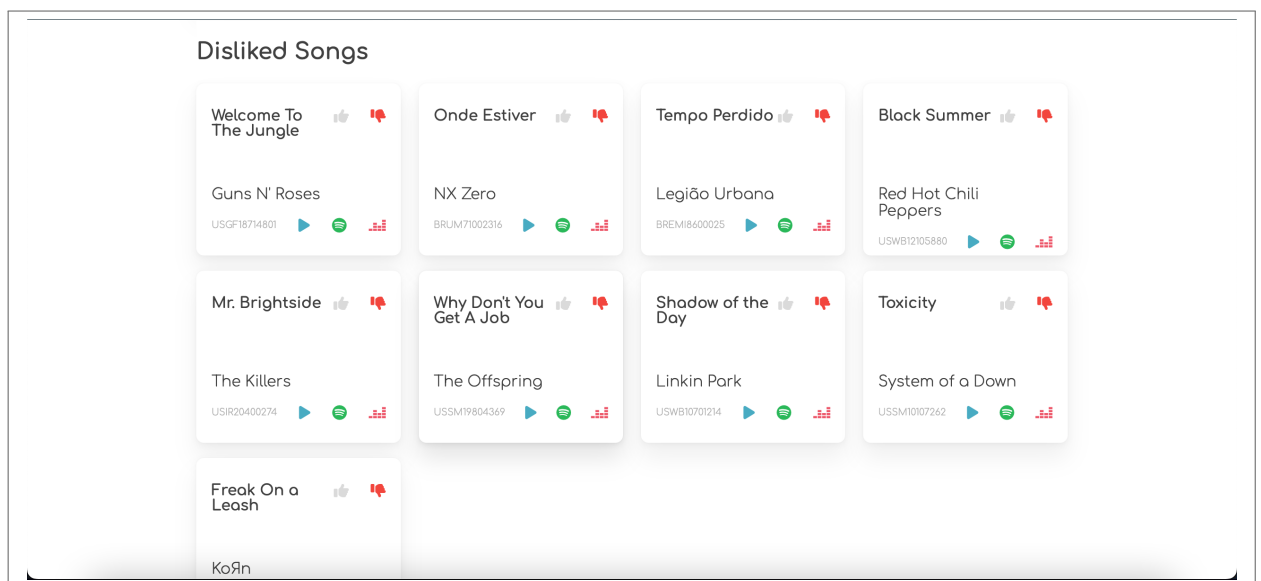
Ao efetivamente criar uma sala sem a assistência do *deejai*, o algoritmo vai buscar as músicas que o criador da sala gosta ou interagiu ao longo da plataforma e apresentá-las como opções para serem adicionadas à *playlist* final. À medida que novos usuários participam da sala as listas de recomendações passam a levar em consideração a informação individual de *dislike*, não retornando essas músicas para a lista de músicas que podem ser adicionadas na *playlist*. A interação dos usuários com as faixas para serem adicionadas pode ser conferido de acordo com a Figura 15. Quando o usuário cria uma sala com a assistência do *deejai*, a *playlist* é modificada automaticamente à medida que entram ou saem pessoas daquela sala, as músicas são buscadas pela união dos *likes* de todos os membros subtraída da lista de *dislikes* de todos os membros e limitada pela informação, caso exista, da duração da *playlist*. O resultado gerado é exibido para todos e caso algum usuário não goste da lista gerada pode de forma simples dar um *deslike* na faixa, como exibido na Figura 16. Caso a quantidade de músicas recomendadas não atinja o limite da duração da *playlist* novas faixas serão recomendadas com

Figura 15 – Sala para técnica de filtro colaborativo



Fonte: Do Autor. (2022)

Figura 16 – Listagem de músicas que o usuário não gosta

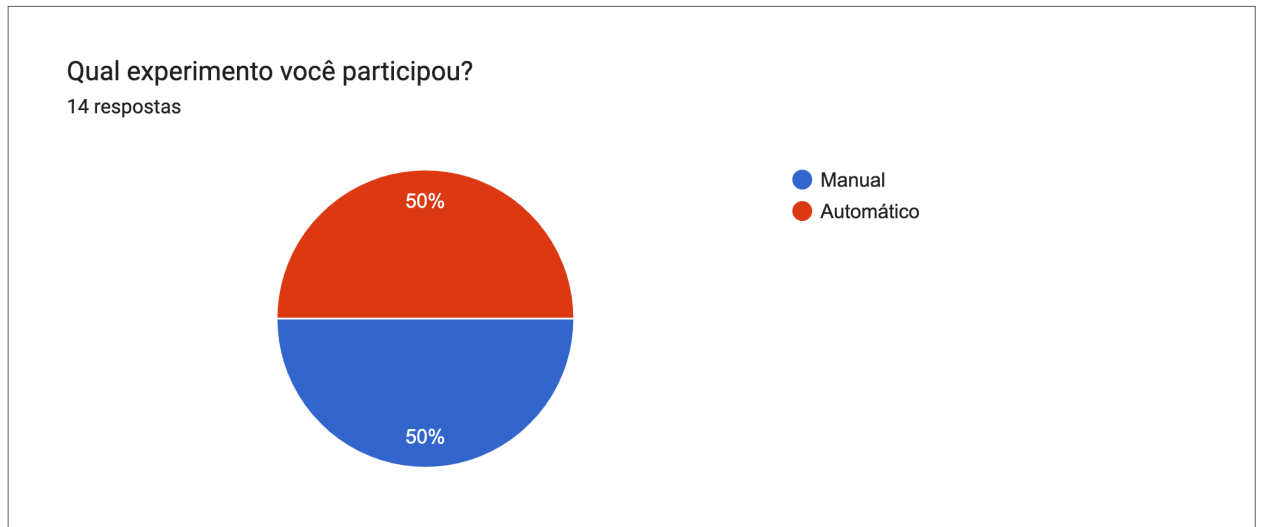


Fonte: Do autor. (2022)

base no contexto das músicas já inseridas pelos membros, criando uma recomendação híbrida utilizando o gênero mais comum das faixas inicialmente recomendadas para buscar outras faixas de gênero próximas espacialmente.

O experimento foi executado através da metodologia de grupo focal, com 3 grupos de 6 pessoas cada, com um total de 18 usuários distintos, tendo como requisito do participante do grupo a utilização dos serviços de streaming Deezer ou Spotify. Os grupos entraram na plataforma e uma pessoa de cada grupo foi responsável por criar uma sala, uma sala utilizando

Figura 17 – Gráfico tipo de experimento



Fonte: Do Autor. (2022)

a recomendação automática, foram selecionadas de forma aleatória músicas que estavam na listagem de recomendações do Deezer, ou de mais ouvidas, no caso do Spotify. Após a seleção automática das músicas, as faixas foram ordenadas com base no principal estilo do artista que está interpretando aquela música. De posse dos estilos foi buscado os estilos mais distantes entre si segundo a organização espacial do EveryNoiseAtOnce. A playlist ordenada possuía aproximadamente 60 minutos, tempo mínimo definido pela metodologia de grupo focal, e foi executada todas as faixas sem a possibilidade de pular a faixa que estava sendo reproduzida no momento, enquanto a faixa era executada cada membro da sala deveria avaliar a música em uma escala de 1 a 5, esses votos foram salvos através de um formulário para que o voto de um usuário não influenciasse no voto do outro participante do experimento. Após a execução de toda a playlist obtivemos os seguintes resultados.

No primeiro grupo, 3 músicas foram classificadas com média abaixo de 4, como pode ser visto na tabela 9.

No segundo grupo, 3 músicas foram classificadas com média abaixo de 4, como pode ser observado na tabela 10.

Já no terceiro grupo, 5 músicas acabaram com média inferior a 4 e foram marcadas para serem substituídas na próxima rodada, como pode ser conferido na tabela 11.

O experimento foi repetido com os mesmos usuários e seus respectivos grupos após sete dias, esse espaçamento se fez necessário a fim de evitar a influência da playlist ouvida na primeira rodada. Durante a segunda rodada do experimento as músicas recomendadas para

Tabela 9 – Avaliação dos usuários na primeira rodada do grupo 1

	user1	user2	user3	user4	user5	user6	Avg
GBAHS2000910	5	2	5	2	3	4	3,5
GBAHT1700676	3	5	4	5	5	5	4,5
USJAY0300082	4	3	5	3	4	5	4
GBARL9800024	5	3	2	3	5	4	3,666666667
GBAYE0200772	2	4	5	4	4	4	3,833333333
USUM70757102	5	3	5	3	3	5	4
USUM71400278	5	5	5	5	5	5	5
GBCAD1800940	1	4	4	5	5	5	4
USJAY0400210	5	1	5	3	5	3	3,666666667
USCA20905185	5	5	3	5	5	3	4,333333333
USUM71705516	5	5	5	4	5	2	4,333333333
USWB10600700	5	4	5	5	5	4	4,666666667
USSM11205433	3	5	4	3	5	5	4,166666667
USIR20000145	5	5	4	4	4	4	4,333333333
GBFFP0300054	5	5	5	4	3	5	4,5

Fonte: Elaborada pelo autor (2022)

Tabela 10 – Avaliação dos usuários na primeira rodada do grupo 2

	user1	user2	user3	user4	user5	user6	avg
USRC10800301	5	4	5	4	3	4	4,166666667
USUM70757102	3	5	4	5	4	5	4,333333333
BRDEP0600434	4	4	5	5	4	5	4,5
USSM11205433	5	4	5	3	3	2	3,666666667
USWB10600700	5	4	4	5	4	3	4,166666667
USRE10400992	5	5	5	3	4	5	4,5
USAR10200229	5	5	4	5	5	4	4,666666667
GBFTG0300127	3	4	4	3	5	3	3,666666667
USVIC1060302	5	4	5	5	5	3	4,5
GBCEL0501181	5	5	5	5	5	5	5
BREMI0200152	3	4	3	5	5	5	4,166666667
TCJPL2092009	3	4	5	5	5	5	4,5
USSM19919500	3	5	3	3	3	4	3,5
USSM10211584	3	5	3	5	5	4	4,166666667
USEP40407105	3	5	3	4	3	4	3,666666667
USRC10100765	5	5	4	5	5	5	4,233333333
GBCEL1300370	5	4	4	4	3	5	4,166666667

Fonte: Elaborada pelo autor (2022)

Tabela 11 – Avaliação dos usuários na primeira rodada do grupo 3

	user1	user2	user3	user4	user5	user6	Avg
GBAHS2000910	5	4	5	5	3	4	4,333333333
GBAHT1700676	4	5	4	5	5	3	4,333333333
USJAY0300082	4	3	5	2	5	4	3,833333333
GBARL9800024	5	5	4	5	2	3	4
GBAYE0200772	3	3	3	5	3	3	3,333333333
USUM70757102	4	5	5	3	4	5	4,333333333
USUM71400278	3	5	4	4	4	5	4,166666667
GBCAD1800940	4	3	5	4	3	4	3,833333333
USJAY0400210	4	4	4	3	5	3	3,833333333
USCA20905185	5	5	4	5	5	3	4,5
USUM71705516	3	4	3	5	5	3	3,833333333
USWB10600700	5	5	5	5	4	5	4,833333333
USSM11205433	4	5	3	5	5	4	4,333333333
USIR20000145	3	5	4	5	5	4	4,333333333
GBFFP0300054	4	5	3	5	3	4	4

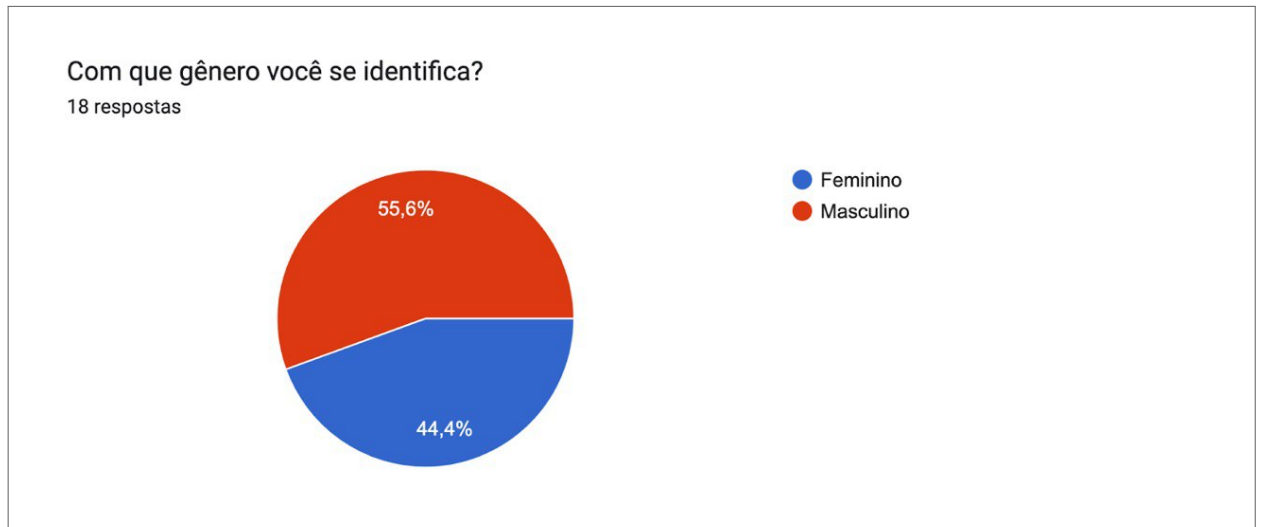
Fonte: Elaborada pelo autor (2022)

os usuários foram as mesmas da primeira rodada, porém filtrando as músicas que tiveram uma avaliação média inferior a 4 na rodada anterior. Para complementar o tempo de 60 minutos necessário para o experimento, foram adicionadas músicas baseadas no estilo das que estavam com média acima de 4, realizando uma recomendação baseada do item, utilizando a informação dos estilos principais dos artistas que tiveram uma avaliação positiva, filtrando assim os dislikes dos usuários envolvidos com aquele determinado grupo.

A distribuição de gênero dos usuários foi de 55,6% do gênero masculino e 44,4% do gênero feminino, como visto na Figura 18. A faixa etária dos participantes do experimento se concentrou entre 25 e 35 anos com 44,4% e 33,3% entre 35-50 anos como pode ser observado na Figura 19, como o experimento colocou como filtro para o grupo focal utilizar os serviços de streaming Deezer e Spotify, a distribuição ficou 72,2% para o Spotify, como pode ser visto na Figura 20.

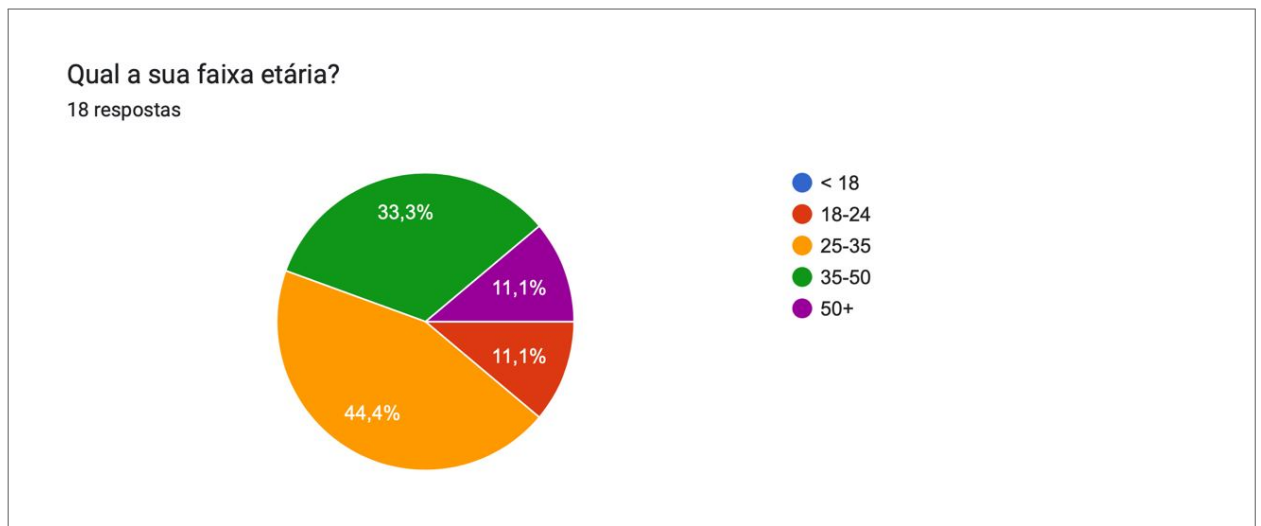
A utilização da média como estratégia baseada em consenso possui uma desvantagem pois quando os gostos individuais diferem bastante, por exemplo, um usuário dá nota 5 enquanto outro dá nota 2 e os outros derem uma nota 4, aquele item que o usuário que deu a menor nota ainda continuará na playlist mesmo o sistema levando em consideração os dislikes dos usuários. Ao votar com gostar ou não em um determinado contexto a chance de um item que

Figura 18 – Gráfico de distribuição do gênero



Fonte: Do Autor. (2022)

Figura 19 – Gráfico de distribuição da idade



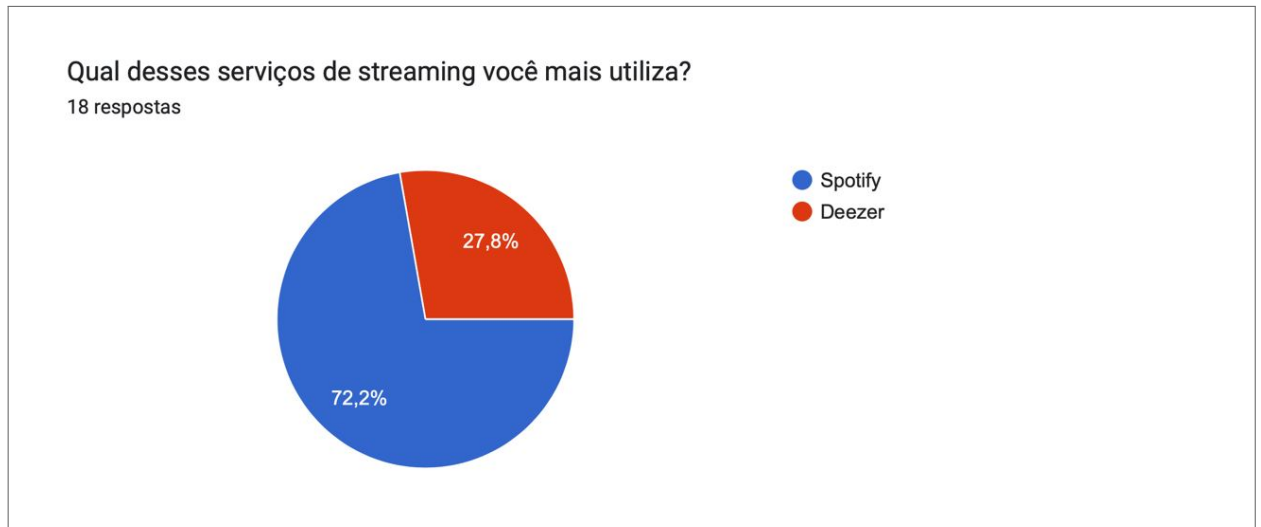
Fonte: Do Autor. (2022)

desagrade um usuário manter na playlist é bastante reduzida.

Outro ponto importante é que ao repetir as músicas com o espaço de uma semana, ainda assim as avaliações individuais a respeito daquela determinada música mudaram, como já levantando no capítulo 2 sobre as mudanças de preferencias dos usuários, a medida que o usuário é mais exposto a uma determinada música ele, a medida que o tempo passa, tende a deixar de gostar daquela música, vai “ enjoando”.

Um ponto a destacar desse experimento é que quanto mais músicas o usuário tem contato, maior será a lista disponível para ele selecionar ao criar uma sala com filtro colaborativo e isso tem um impacto direto na rapidez que a seleção será finalizada.

Figura 20 – Gráfico sobre plataforma de consumo



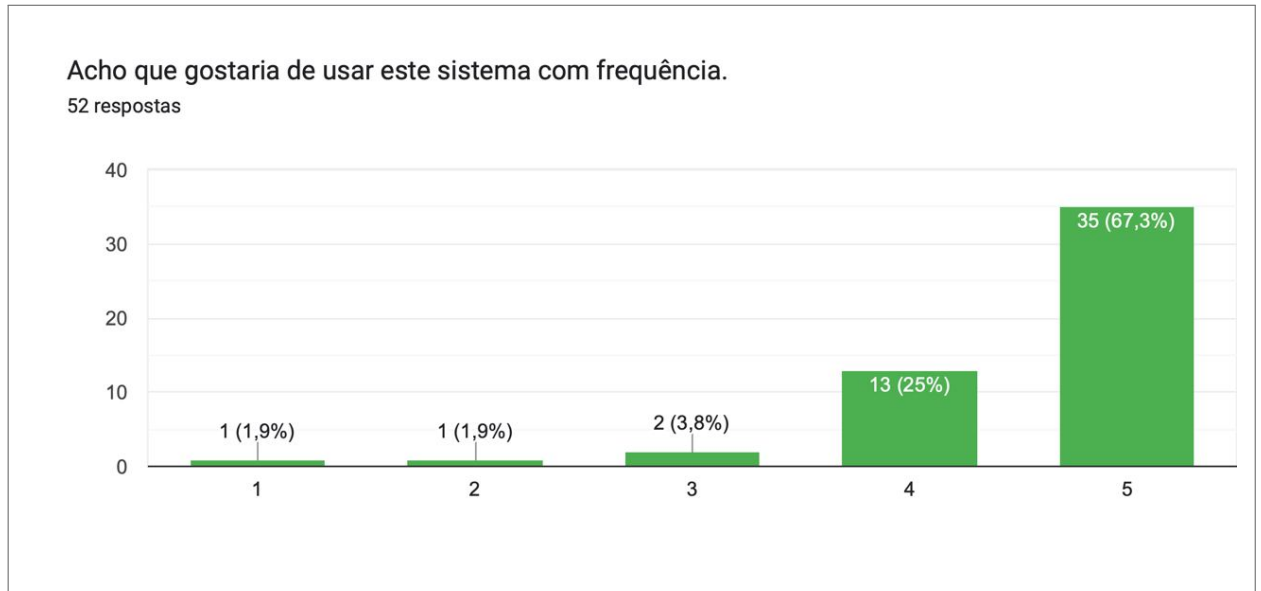
Fonte: Do Autor. (2022)

A utilização do SUS para avaliação da plataforma se deu em um evento durante o Rec'n'Play onde houve uma apresentação de como sistemas de recomendação funcionam e uma demonstração da plataforma com os presentes, após a utilização da plataforma, foi apresentado um questionário com as perguntas do SUS a fim de avaliar a usabilidade do sistema.

O resultado do SUS foi de 84,90, onde valores acima de 70 indicam que o sistema possui uma usabilidade aceitável. (BANGOR; KORTUM; MILLER, 2008) As questões ímpares avaliam o critério positivo para um bom sistema precisam que as notas concentrem-se em valores altos, próximo a 5, esta distribuição pode ser conferida nas imagens 21, 23, 25, 27 e 29.

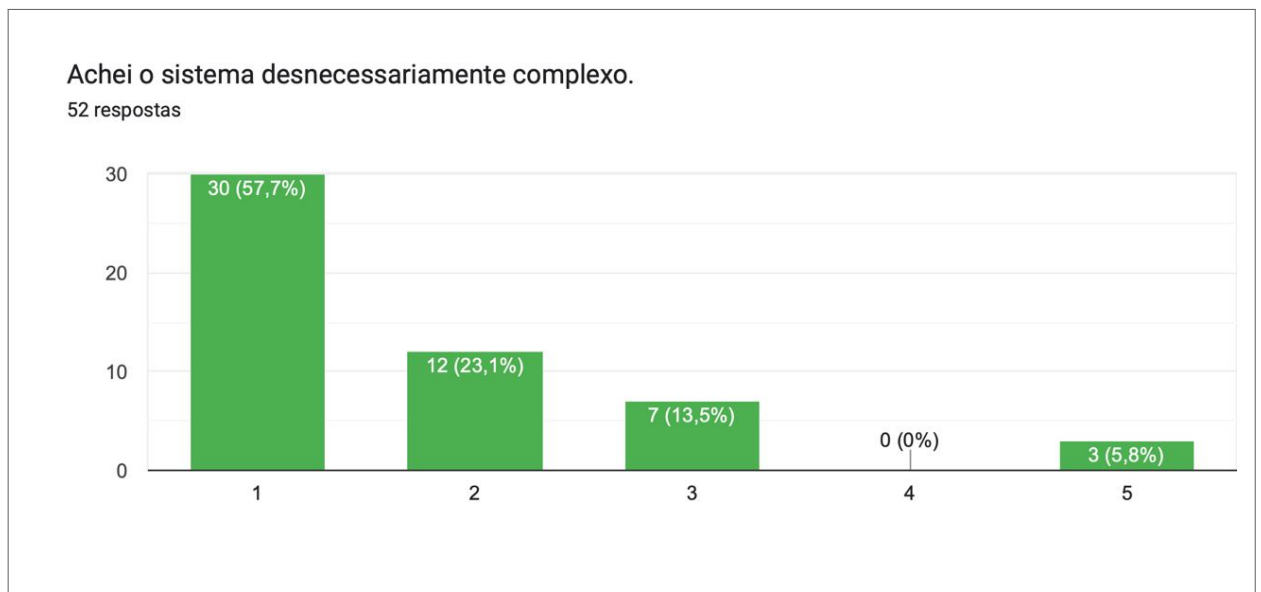
Já as questões pares avaliam a experiência negativa no uso da aplicação e para um bom sistema os valores devem concentrar-se nos valores próximos a 1. A distribuição dessas perguntas pode ser vista nas imagens 22, 24, 26, 28 e 30. Os dados com o resultado da pesquisa encontra-se no Apendice H.

Figura 21 – Respostas da primeira questão do SUS



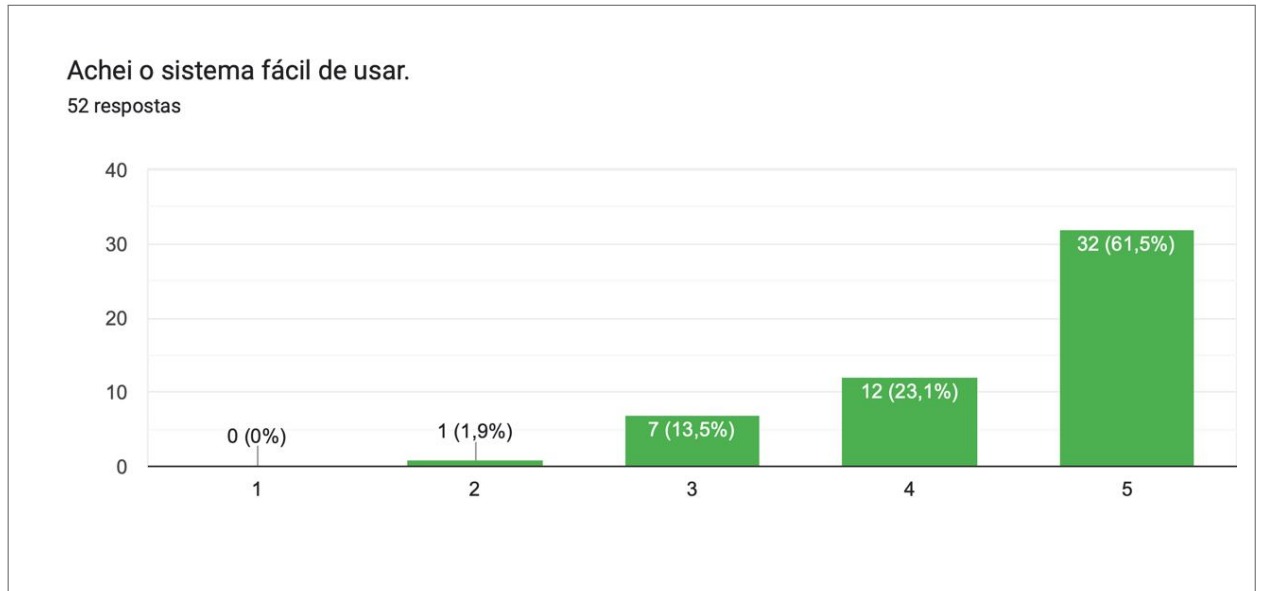
Fonte: Do Autor. (2022)

Figura 22 – Respostas da segunda questão do SUS



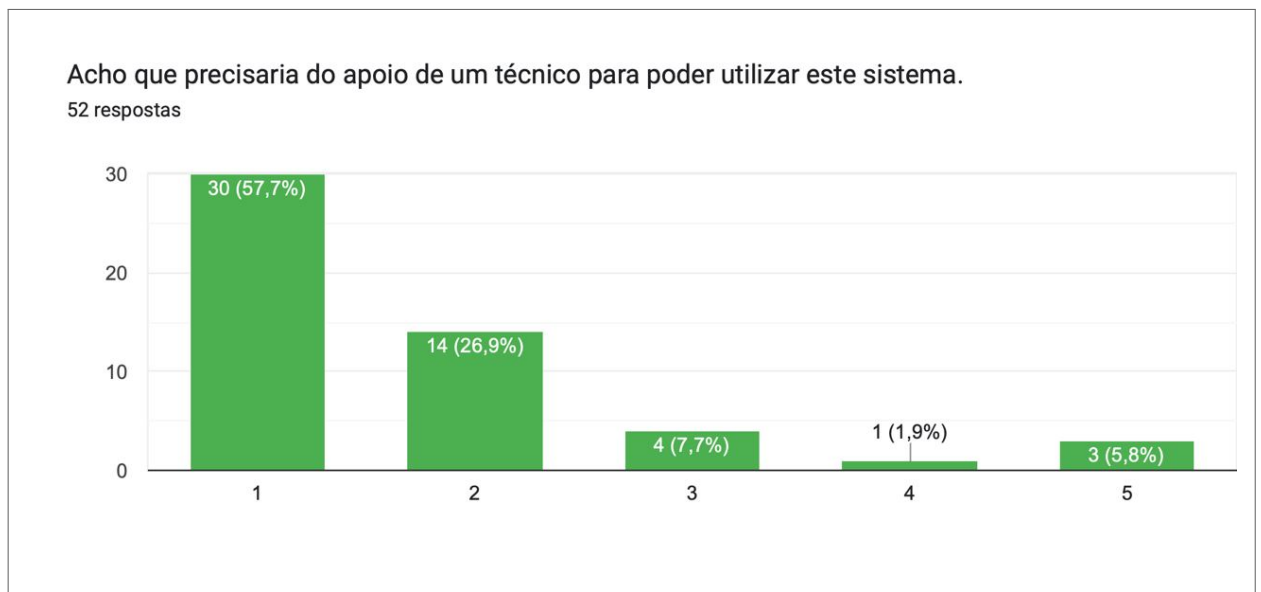
Fonte: Do Autor. (2022)

Figura 23 – Respostas da terceira questão do SUS



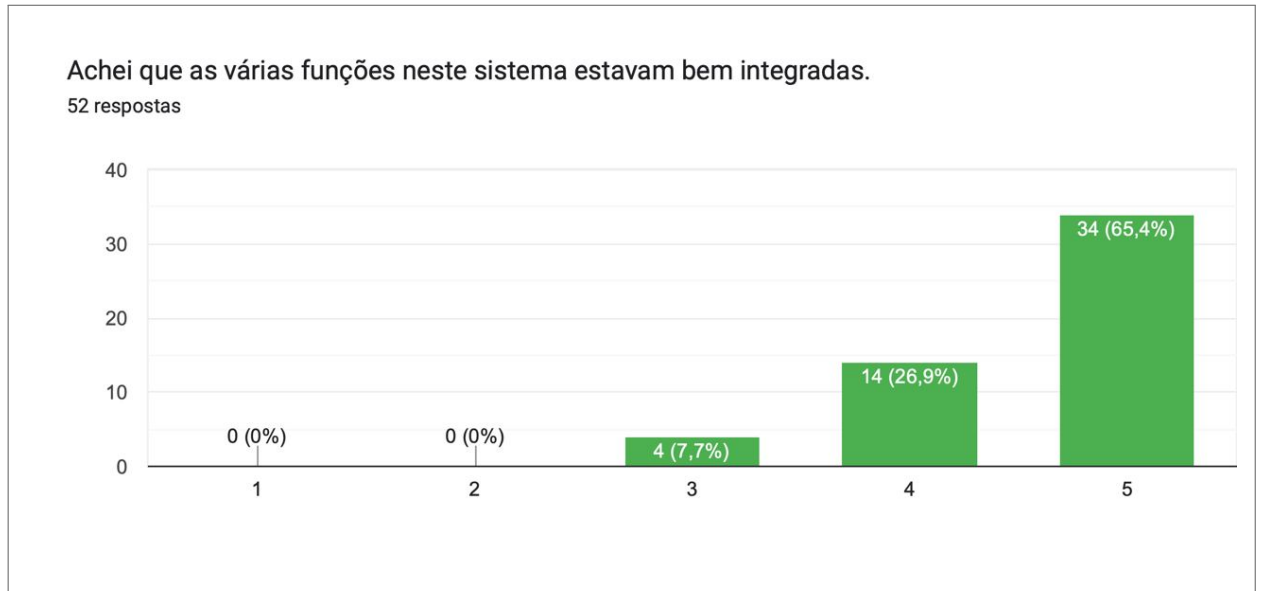
Fonte: Do Autor. (2022)

Figura 24 – Respostas da quarta questão do SUS



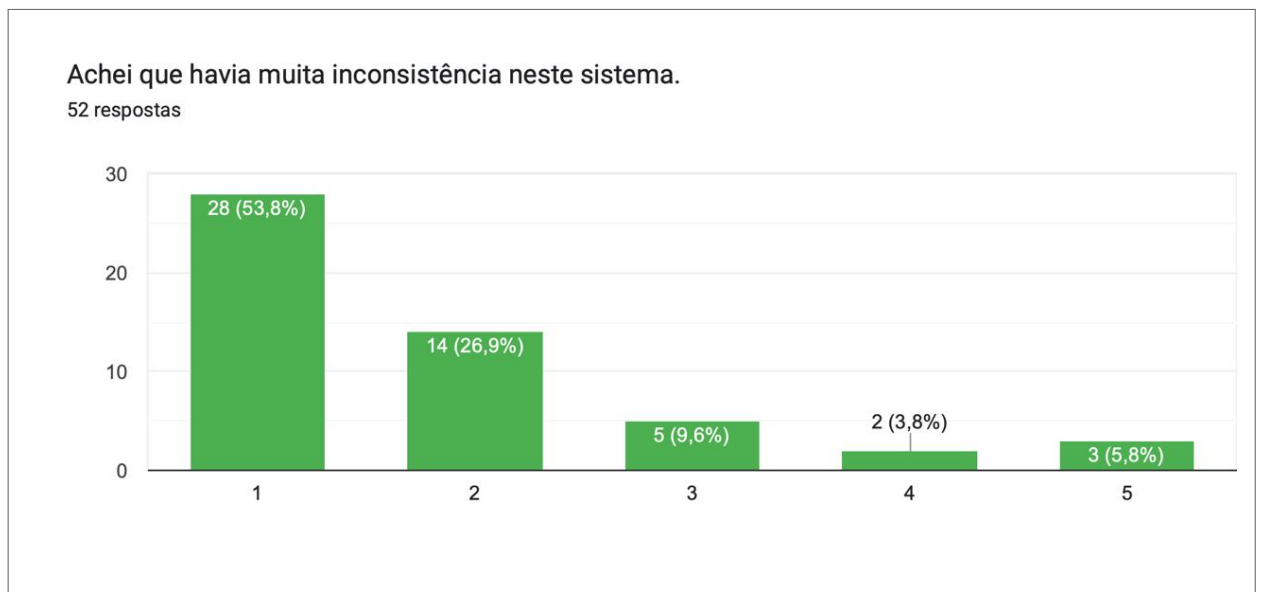
Fonte: Do Autor. (2022)

Figura 25 – Respostas da quinta questão do SUS



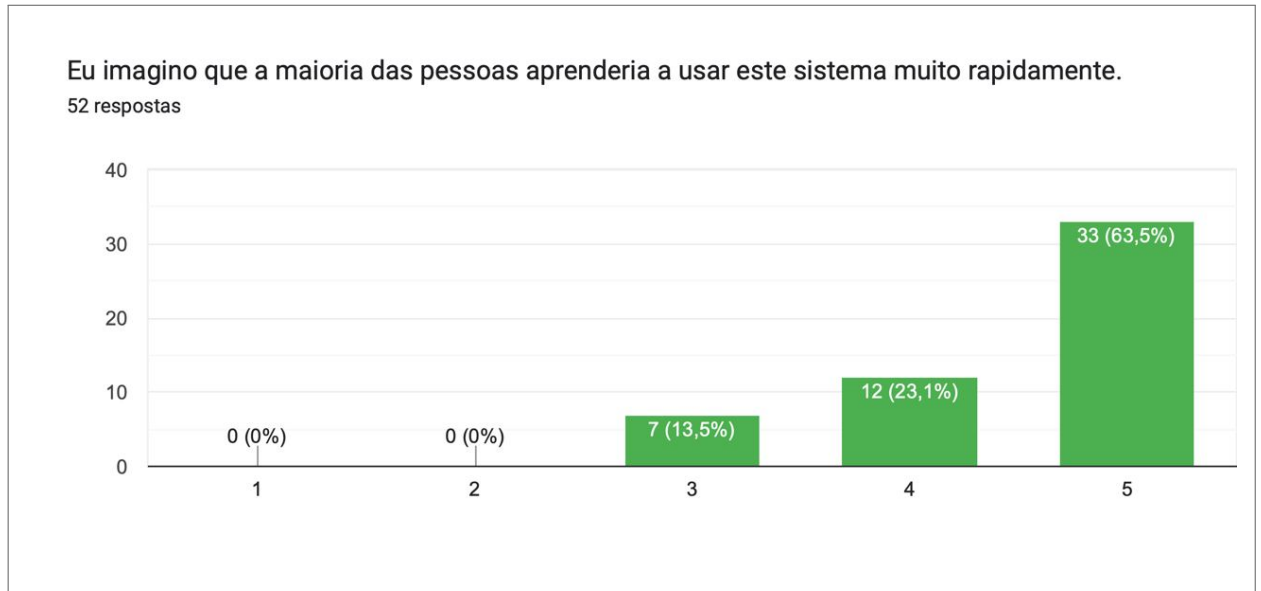
Fonte: Do Autor. (2022)

Figura 26 – Respostas da sexta questão do SUS



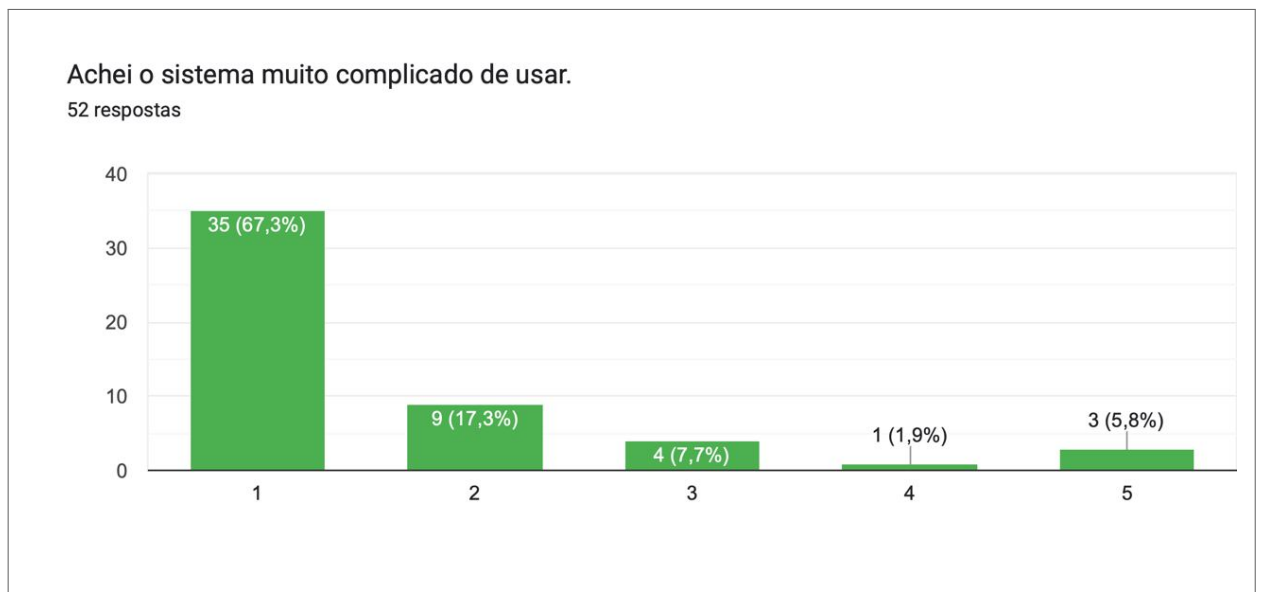
Fonte: Do Autor. (2022)

Figura 27 – Respostas da sétima questão do SUS



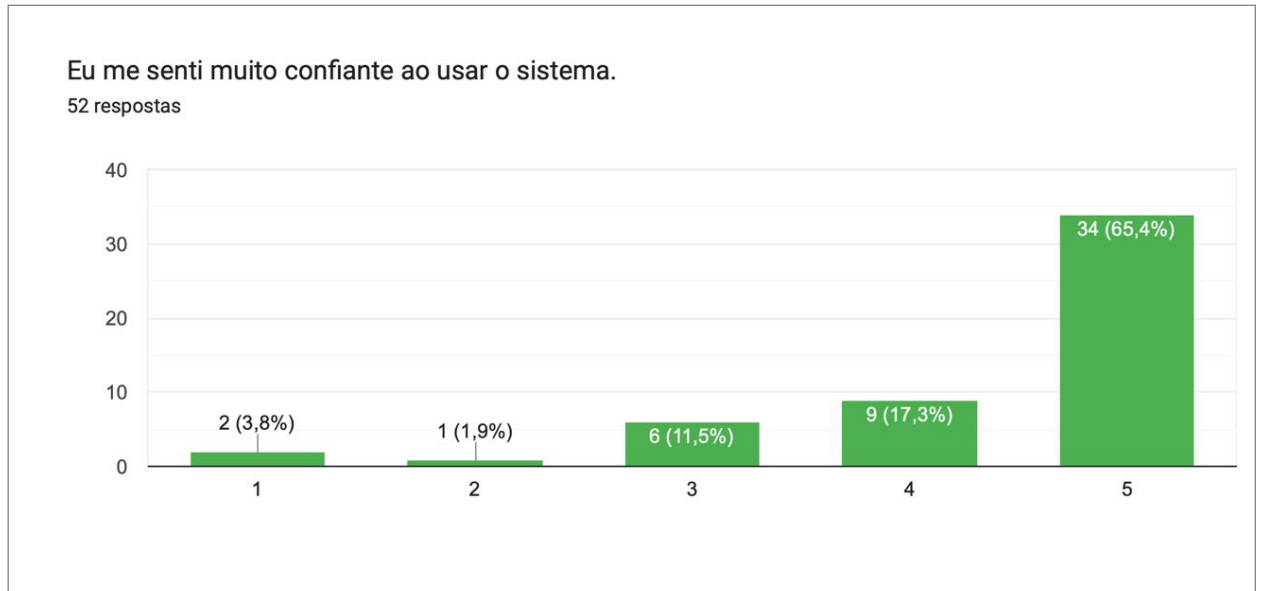
Fonte: Do Autor. (2022)

Figura 28 – Respostas da oitava questão do SUS



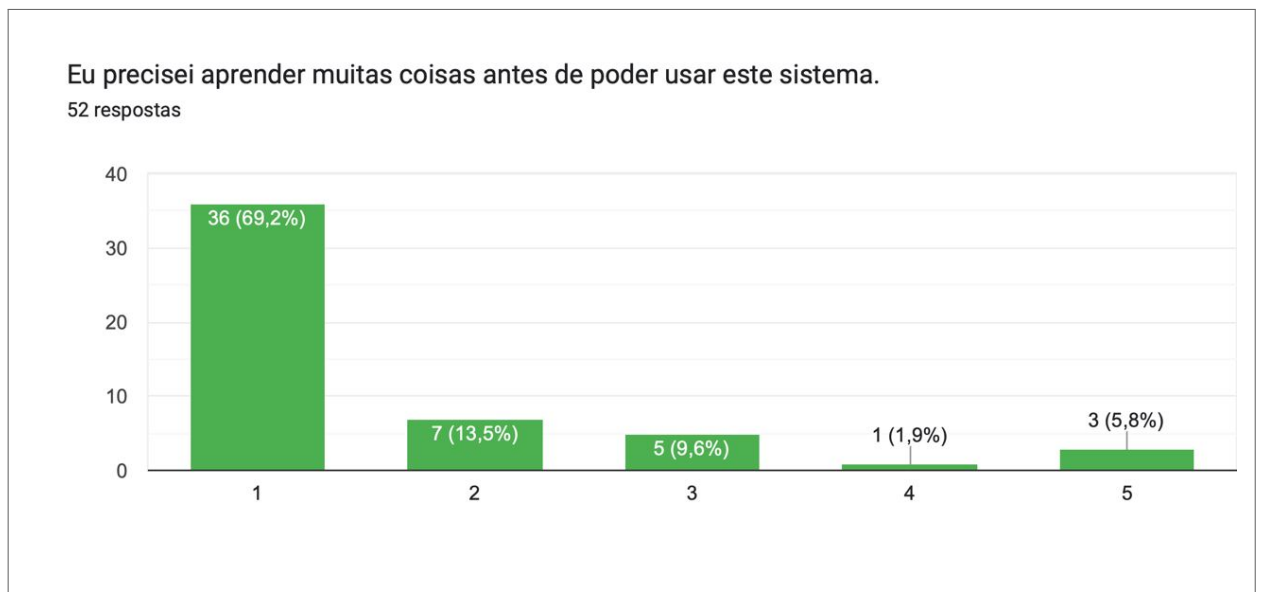
Fonte: Do Autor. (2022)

Figura 29 – Respostas da nona questão do SUS



Fonte: Do Autor. (2022)

Figura 30 – Respostas da décima questão do SUS



Fonte: Do Autor. (2022)

5 CONCLUSÃO

Nesta seção apresentaremos as considerações finais sobre os principais pontos abordados nesta dissertação, expondo as contribuições alcançadas e apontando tópicos para trabalhos futuros.

Estudos sobre sistemas de recomendação para grupos envolvendo música são apresentados desde a década de 1990, levando em consideração a criação de um contexto, ambiente no qual os indivíduos se relacionam, seja uma academia de ginástica ou uma sala virtual, como abordamos no desenvolvimento do *deejai*.

Com o advento do mercado de *streaming* de música, foi modificado a forma do usuário de consumir música, porém um dos elementos principais abordados pelos estudos de sistemas de recomendação é a informação do que o usuário não gosta, para que possa ser feito uma filtragem, ou o quanto o usuário gosta de algo, afinal foi observado que o usuário pode gostar da música A para um contexto e da música B para outro contexto, pois o consumo de música é algo efêmero e com constante mudança.

Ao longo do desenvolvimento deste trabalho foi observado que para a definição de contexto quanto mais rica a informação sobre a motivação da criação daquela sala virtual mais assertiva é a recomendação.

Os serviços de streaming não disponibilizam as informações do que um determinado usuário não gosta o que dificulta a criação de uma lista de reprodução, a fim de sanar esse problema foi criada a meta-plataforma onde as informações dos usuários a respeito de músicas que ele não gosta era armazenada, deste modo a realização de recomendações para grupos utilizando uma técnica híbrida se tornou possível nos ambientes onde as músicas são importadas dos serviços de *streaming*.

Ao agrupar os usuários em salas tornou-se possível a criação de um contexto de forma virtual, ter uma sala para criar uma *playlist* para uma festa ou para um churrasco e terem recomendações distintas.

Uma dificuldade de sistemas de recomendação para grupos era a necessidade de submissão das suas músicas, seja através de um cliente instalado na máquina do usuário ou um formulário para responder sobre as músicas que o usuário gosta, com o auxílio das APIs dos serviços de *streaming* foi possível importar essas informações necessitando apenas que o usuário permita o acesso da aplicação a sua biblioteca de músicas através de uma simples etapa de auten-

ticação. Dessa forma os desafios como a mudança de preferências de um usuário pode ser contornado realizando essa checagem de músicas mais ouvidas a cada vez que o indivíduo entra na aplicação.

A esparsidade dos dados de um usuário é outro fator que ao desenvolvermos que existam três estados, onde um é quando efetivamente ele gostou de algo, outro quando ele não gosta de algo e um terceiro que é quando ele interage com aquele item de alguma forma, por estar em uma sala e a recomendação ser de terceiros, caso o indivíduo não goste ele irá assinalar que aquele item não o agrada. A aplicação foi construída utilizando o método de *deploy* baseado em *container* Docker, onde caso haja a necessidade de escalar o app, é possível adicionar novos *containers* e realizar um balanceamento de carga.

A utilização da classificação elaborada pelo *Every Noise At Once* auxilia na sinonímia, pois os artistas são agrupados pelos seus estilos musicais que são representados por pontos em um gráfico e não por terem grafia similar.

O sistema utiliza de dados dos usuários advindos de serviços de terceiros, para receber informações de outros usuários é necessário que os mesmos interajam em um determinado contexto, mas essas informações são restritas a fim de respeitar as legislações específicas, como LGPD para o Brasil e GDPR para a União Européia.

5.1 PRINCIPAIS CONTRIBUIÇÕES

Em face aos resultados conquistados, acreditamos que este trabalho contribuiu para um entendimento mais atualizado sobre sistemas de recomendações para músicas além dos fatores que fazem uma recomendação de música ser aceita para um determinado grupo. Além disso, é possível destacar outras contribuições importantes como:

- Um melhor entendimento a respeito da utilização de técnicas de recomendação baseada em consenso para recomendar músicas para grupos.
- Um processo definido para utilizar informações dos *streamings* de músicas com o intuito de melhorar a experiência de recomendação.
- Uma base de dados que consegue interligar serviços de *streaming* distintos, contendo uma abordagem de gênero musical por proximidade, desenvolvida pelo *Every Noise At Once* para facilitar a recomendação.

5.2 TRABALHOS FUTUROS

Neste trabalho integramos a aplicação com dois serviços de streaming, o *Spotify* e o *Deezer*. Para os trabalhos futuros, temos o desejo de acrescentar a quantidade desses serviços, adicionando o suporte ao *Tidal*, ao *Amazon Music*, ao *Apple Music*, ao *Resso* e ao *Youtube Music* a fim de aumentar a abrangência de pessoas utilizando a plataforma. Um outro ponto de melhoria é levantar as características únicas de análise de áudio que essas plataformas podem oferecer para assim utilizar outros métodos de recomendação tendo como base, por exemplo, as informações de características de áudio disponíveis. Outro ponto importante seria mudar a classificação atual da informação a respeito do gênero está ligada ao artista e não a faixa de áudio, dado que artistas têm a possibilidade de se reinventarem ao longo da carreira, o estilo de uma faixa do seu primeiro álbum pode ser diferente de outra faixa do mesmo álbum ou de um outro álbum. A partir da informação de gênero realizar um estudo tendo como motivação de pesquisa a avaliação de que ao centrar a recomendação de músicas no gênero da faixa ocorra uma possível melhoria. Abordar em trabalhos futuros a utilização das *Audio Features* disponibilizadas pelo *Spotify* com técnicas de filtragem baseada em modelo, utilizando como modelo os recursos calculados fornecidos, por exemplo, dança, energia e valência.

REFERÊNCIAS

- AGGARWAL, S. Modern web-development using reactjs. *International Journal of Recent Research Aspects*, v. 5, n. 1, p. 133–137, 2018.
- ALMEIDA, F.; SANTOS, J. D.; MONTEIRO, J. A. The challenges and opportunities in the digitalization of companies in a post-covid-19 world. *IEEE Engineering Management Review*, IEEE, v. 48, n. 3, p. 97–103, 2020.
- BANGOR, A.; KORTUM, P.; MILLER, J. The system usability scale (sus): An empirical evaluation. *International Journal of Human-Computer Interaction*, v. 24, n. 6, p. 574–594, 2008.
- BASS, L.; CLEMENTS, P.; KAZMAN, R. *Software architecture in practice*. [S.l.]: Addison-Wesley Professional, 2003.
- BASSOLI, A.; MOORE, J.; AGAMANOLIS, S. tuna: Socialising music sharing on the move. In: *Consuming music together*. [S.l.]: Springer, 2006. p. 151–172.
- BAUER, C.; KHOLODYLO, M.; STRAUSS, C. Music recommender systems challenges and opportunities for non-superstar artists. *Bled eConference*, p. 48, 2017.
- BAUER, C.; SCHEDL, M. Global and country-specific mainstreamness measures: Definitions, analysis, and usage for improving personalized music recommendation systems. *PloS one*, Public Library of Science San Francisco, CA USA, v. 14, n. 6, p. e0217389, 2019.
- BELQASMI, F.; SINGH, J.; MELHEM, S. Y. B.; GLITHO, R. H. Soap-based vs. restful web services: a case study for multimedia conferencing. *IEEE internet computing*, IEEE, v. 16, n. 4, p. 54–63, 2012.
- BISSELL, D. Affective platform urbanism: Changing habits of digital on-demand consumption. *Geoforum*, Elsevier, v. 115, p. 102–110, 2020.
- CACHEDA, F.; CARNEIRO, V.; FERNÁNDEZ, D.; FORMOSO, V. Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Transactions on the Web (TWEB)*, ACM New York, NY, USA, v. 5, n. 1, p. 1–33, 2011.
- CAPLAN, S. Using focus group methodology for ergonomic design. *Ergonomics*, Taylor & Francis, v. 33, n. 5, p. 527–533, 1990.
- CHOI, M. A performance analysis of restful open api information system. In: SPRINGER. *International Conference on Future Generation Information Technology*. [S.l.], 2012. p. 59–64.
- CHRISTENSEN, I. A.; SCHIAFFINO, S. Entertainment recommender systems for group of users. *Expert systems with applications*, Elsevier, v. 38, n. 11, p. 14127–14135, 2011.
- COLLIS, J.; HUSSEY, R. *Pesquisa em administração: um guia prático para alunos de graduação e pós-graduação*. [S.l.]: Bookman, 2005.
- COLOMER, J. M. Ramon llull: from ‘ars electionis’ to social choice theory. *Social Choice and Welfare*, Springer, v. 40, n. 2, p. 317–328, 2013.

- CROSSEN, A.; BUDZIK, J.; HAMMOND, K. J. Flytrap: intelligent group music recommendation. In: *Proceedings of the 7th international conference on Intelligent user interfaces*. [S.l.: s.n.], 2002. p. 184–185.
- CUNNINGHAM, S. J.; NICHOLS, D. M. Exploring social music behaviour: An investigation of music selection at parties. In: ISMIR. [S.l.], 2009.
- DARA, S.; CHOWDARY, C. R.; KUMAR, C. A survey on group recommender systems. *Journal of Intelligent Information Systems*, Springer, v. 54, n. 2, p. 271–295, 2020.
- DEEZER. *oAuth*. 2021. <<https://developers.deezer.com/api/oauth>>. [Online; accessed 03-March-2021].
- DEEZER. *Permissions*. 2021. <<https://developers.deezer.com/api/permissions>>. [Online; accessed 03-March-2021].
- DENORA, T. *Music in everyday life*. [S.l.]: Cambridge University Press, 2000.
- DERY, L. N.; KALECH, M.; ROKACH, L.; SHAPIRA, B. Iterative voting under uncertainty for group recommender systems. In: *Proceedings of the fourth ACM conference on Recommender systems*. [S.l.: s.n.], 2010. p. 265–268.
- DIAS, C. A. Grupo focal: técnica de coleta de dados em pesquisas qualitativas. *Informação & Sociedade*, Universidade Federal da Paraíba-Programa de Pós-Graduação em Ciência da ... , v. 10, n. 2, 2000.
- ECHONEXT. *Every Noise At Once*. 2022. <<https://everynoise.com/engenremap.html>>. [Online; accessed 30-July-2022].
- EXPRESS. *Express*. 2022. <<http://expressjs.com/>>. [Online; accessed 20-July-2022].
- FACEBOOK. *React Framework*. 2021. <reactjs.org>. [Online; accessed 10-July-2022].
- FACEBOOK. *Create React App*. 2022. <<https://create-react-app.dev/>>. [Online; accessed 01-July-2022].
- FONTES, F. D.; CABRAL, G. R. E.; RAMALHO, G. L. An open source platform to assist the creation of group playlists through artificial intelligence algorithms. In: SBC. *Anais do XVIII Simpósio Brasileiro de Computação Musical*. [S.l.], 2021. p. 166–169.
- GERALDES, W. B.; MARTINS, E. R.; AFONSECA, U. R. Avaliação da usabilidade do scratch utilizando o método system usability scale (sus). In: SBC. *Anais da X Escola Regional de Informática de Mato Grosso*. [S.l.], 2019. p. 25–30.
- GHAZARIAN, S.; NEMATBAKHS, M. A. Enhancing memory-based collaborative filtering for group recommender systems. *Expert systems with applications*, Elsevier, v. 42, n. 7, p. 3801–3812, 2015.
- GIL, A. C. *Como elaborar projetos de pesquisa*. [S.l.]: Atlas São Paulo, 2002. v. 4.
- GOLDBERG, D.; NICHOLS, D.; OKI, B. M.; TERRY, D. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, ACM New York, NY, USA, v. 35, n. 12, p. 61–70, 1992.

- GRIER, R. A.; BANGOR, A.; KORTUM, P.; PERES, S. C. The system usability scale: Beyond standard usability testing. In: SAGE PUBLICATIONS SAGE CA: LOS ANGELES, CA. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. [S.l.], 2013. v. 57, n. 1, p. 187–191.
- ISINKAYE, F. O.; FOLAJIMI, Y. O.; OJOKOH, B. A. Recommendation systems: Principles, methods and evaluation. *Egyptian informatics journal*, Elsevier, v. 16, n. 3, p. 261–273, 2015.
- JOHANSSON, J. *Create React App vs NextJS: A comparison of two ReactJS based web application frameworks*. 2021.
- JSX, R. *W3S Scholl*. 2022. <https://www.w3schools.com/react/react_jsx.asp>. [Online; accessed 11-July-2022].
- KADAM, N. *An introduction to group recommender systems*. 2020. <<https://towardsdatascience.com/an-introduction-to-group-recommender-systems-8f942a06db56>>. Accessed: 2022-05-27.
- KAŠŠÁK, O.; KOMPAN, M.; BIELIKOVÁ, M. Personalized hybrid recommendation for group of users: Top-n multimedia recommender. *Information Processing & Management*, Elsevier, v. 52, n. 3, p. 459–477, 2016.
- KIM, H.-N.; SADDIK, A. E. A stochastic approach to group recommendations in social media systems. *Information Systems*, Elsevier, v. 50, p. 76–93, 2015.
- KLUG, B. An overview of the system usability scale in library website and system usability testing. *Weave: Journal of Library User Experience*, Michigan Publishing, University of Michigan Library, v. 1, n. 6, 2017.
- KUMAR, A. *Global Online Music Streaming Growth Slowed Down in Q2 2020*. 2020. <<https://www.counterpointresearch.com/global-online-music-streaming-growth-slowed-down-in-q2-2020/>>. Accessed: 2021-06-06.
- LAKSHMI, S. S.; LAKSHMI, T. A. Recommendation systems: Issues and challenges. *International Journal of Computer Science and Information Technologies*, Citeseer, v. 5, n. 4, p. 5771–5772, 2014.
- LÜ, L.; MEDO, M.; YEUNG, C. H.; ZHANG, Y.-C.; ZHANG, Z.-K.; ZHOU, T. Recommender systems. *Physics reports*, Elsevier, v. 519, n. 1, p. 1–49, 2012.
- MAASØ, A.; HAGEN, A. N. Metrics and decision-making in music streaming. *Popular Communication*, Taylor & Francis, v. 18, n. 1, p. 18–31, 2020.
- MCCARTHY, J. F.; ANAGNOST, T. D. Musicfx: an arbiter of group preferences for computer supported collaborative workouts. In: *Proceedings of the 1998 ACM conference on Computer supported cooperative work*. [S.l.: s.n.], 1998. p. 363–372.
- MELVILLE, P.; SINDHWANI, V. Recommender systems. *Encyclopedia of machine learning*, v. 1, p. 829–838, 2010.
- MORRIS, J. W.; POWERS, D. Control, curation and musical experience in streaming music services. *Creative Industries Journal*, Taylor & Francis, v. 8, n. 2, p. 106–122, 2015.

- NASSAR, N.; JAFAR, A.; RAHHAL, Y. A novel deep multi-criteria collaborative filtering model for recommendation system. *Knowledge-Based Systems*, Elsevier, v. 187, p. 104811, 2020.
- NODEJS. *Every Noise At Once*. 2022. <<https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html>>. [Online; accessed 10-July-2022].
- NODEJS. *NodeJS*. 2022. <<https://nodejs.org/en/about/>>. [Online; accessed 20-July-2022].
- NORTH, A. C.; HARGREAVES, D. J.; HARGREAVES, J. J. Uses of music in everyday life. *Music perception*, University of California Press USA, v. 22, n. 1, p. 41–77, 2004.
- ORD, M. “every noise at once”: Online music discovery maps and cosmopolitan subjectivities. *The Bloomsbury Handbook of Popular Music and Social Class*, Bloomsbury Publishing USA, p. 117, 2020.
- ORTEGA, F.; BOBADILLA, J.; GUTIÉRREZ, A.; HURTADO, R.; LI, X. Artificial intelligence scientific documentation dataset for recommender systems. *IEEE Access*, IEEE, v. 6, p. 48543–48555, 2018.
- ORTEGA, F.; SANCHEZ, J.-L.; BOBADILLA, J.; GUTIERREZ, A. Improving collaborative filtering-based recommender systems results using pareto dominance. *Information Sciences*, Elsevier, v. 239, p. 50–61, 2013.
- OWASP. *A02:2021 – Cryptographic Failures*. 2022. <https://owasp.org/Top10/A02_2021-Cryptographic_Failures/>. [Online; accessed 20-July-2022].
- OWASP. *OWASP Top Ten*. 2022. <<https://owasp.org/www-project-top-ten/>>. [Online; accessed 20-July-2022].
- PATEL, B.; DESAI, P.; PANCHAL, U. Methods of recommender system: A review. In: *IEEE. 2017 international conference on innovations in information, embedded and communication systems (ICIIECS)*. [S.l.], 2017. p. 1–4.
- POSTGRESQL. *PostgreSQL*. 2022. <<https://www.postgresql.org/about/>>. [Online; accessed 20-July-2022].
- PRISMA. *Prisma ORM*. 2022. <<https://www.prisma.io/>>. [Online; accessed 01-July-2022].
- RESNICK, P.; VARIAN, H. R. Recommender systems. *Communications of the ACM*, ACM New York, NY, USA, v. 40, n. 3, p. 56–58, 1997.
- RICCI, F.; ROKACH, L.; SHAPIRA, B. Introduction to recommender systems handbook. In: *Recommender systems handbook*. [S.l.]: Springer, 2011. p. 1–35.
- SALEHI-ABARI, A.; BOUTILIER, C. Preference-oriented social networks: Group recommendation and inference. In: *Proceedings of the 9th ACM Conference on Recommender Systems*. [S.l.: s.n.], 2015. p. 35–42.
- SCHEDL, M.; KNEES, P.; MCFEE, B.; BOGDANOV, D.; KAMINSKAS, M. Music recommender systems. In: *Recommender systems handbook*. [S.l.]: Springer, 2015. p. 453–492.

SONG, Y.; DIXON, S.; PEARCE, M. A survey of music recommendation systems and future perspectives. In: *9th international symposium on computer music modeling and retrieval*. [S.l.: s.n.], 2012. v. 4, p. 395–410.

SPOTIFY. *Authorization Scopes*. 2021. <<https://developer.spotify.com/documentation/general/guides/authorization/scopes/>>. [Online; accessed 03-March-2021].

THORAT, P. B.; GOUDAR, R. M.; BARVE, S. Survey on collaborative filtering, content-based filtering and hybrid recommendation system. *International Journal of Computer Applications*, Foundation of Computer Science, v. 110, n. 4, p. 31–36, 2015.

TOULSON, R.; GRINT, B.; STAFF, R. Embedding isrc identifiers in broadcast wave audio files. *KES Transactions on Innovation in Music*, v. 1, n. 1, p. 213, 2013.

TSCHMUCK, P. The emergence of the phonographic industry within the music industry. *Creativity and innovation in the music industry*, Springer, p. 1–17, 2006.

VAUGHN, S.; SCHUMM, J. S.; SINAGUB, J. M. *Focus group interviews in education and psychology*. [S.l.]: Sage, 1996.

ZHANG, Q.; LU, J.; JIN, Y. Artificial intelligence in recommender systems. *Complex & Intelligent Systems*, Springer, v. 7, n. 1, p. 439–457, 2021.

APÊNDICE A – FORMULÁRIO DE PESQUISA

Questionário SUS - DeejaI

Questionário para avaliação do nível de usabilidade do deejai.

***Obrigatório**

1. Acho que gostaria de usar este sistema com frequência. *

Marcar *apenas uma oval.*

Discordo fortemente

1 ☐

2 ☐

3 ☐

4 ☐

5 ☐

Concordo fortemente

APÊNDICE B – FORMULÁRIO DE PESQUISA

2. Achei o sistema desnecessariamente complexo. *

Marcar *apenas uma oval.*

Discordo fortemente

1 ☐

2 ☐

3 ☐

4 ☐

5 ☐

Concordo fortemente

3. Achei o sistema fácil de usar. *

Marcar *apenas uma oval.*

Discordo fortemente

1 ☐

2 ☐

3 ☐

4 ☐

5 ☐

Concordo fortemente

APÊNDICE C – FORMULÁRIO DE PESQUISA

4. Acho que precisaria do apoio de um técnico para poder utilizar este sistema. *

Marcar *apenas uma oval.*

Discordo fortemente

1 ☐

2 ☐

3 ☐

4 ☐

5 ☐

Concordo fortemente

5. Achei que as várias funções neste sistema estavam bem integradas. *

Marcar *apenas uma oval.*

Discordo fortemente

1 ☐

2 ☐

3 ☐

4 ☐

5 ☐

Concordo fortemente

APÊNDICE D – FORMULÁRIO DE PESQUISA

6. Achei que havia muita inconsistência neste sistema. *

Marcar *apenas uma oval.*

Discordo fortemente

1 ☐

2 ☐

3 ☐

4 ☐

5 ☐

Concordo fortemente

APÊNDICE E – FORMULÁRIO DE PESQUISA

7. Eu imagino que a maioria das pessoas aprenderia a usar este sistema muito rapidamente. *

Marcar apenas uma oval.

Discordo fortemente

1 ☐

2 ☐

3 ☐

4 ☐

5 ☐

Concordo fortemente

APÊNDICE F – FORMULÁRIO DE PESQUISA

8. Achei o sistema muito complicado de usar. *

Marcar *apenas uma oval.*

Discordo fortemente

1 ☐

2 ☐

3 ☐

4 ☐

5 ☐

Concordo fortemente

9. Eu me senti muito confiante ao usar o sistema. *

Marcar *apenas uma oval.*

Discordo fortemente

1 ☐

2 ☐

3 ☐

4 ☐

5 ☐

Concordo fortemente

APÊNDICE G – FORMULÁRIO DE PESQUISA

10. Eu precisei aprender muitas coisas antes de poder usar este sistema. *

Marcar *apenas uma oval.*

Discordo fortemente

1 ☐

2 ☐

3 ☐

4 ☐

5 ☐

Concordo fortemente

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários

APÊNDICE H – RESPOSTAS SUS