

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO  
INSTITUTO DE MATEMÁTICA  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

DEOVAL LUIS DOS SANTOS JUNIOR  
JOSUÉ PEREIRA DA CONCEIÇÃO  
NATHÁLIA PEDRAL DE CASTRO

SMARTBOX - SISTEMA DE RECOMENDAÇÃO  
MUSICAL PARA AMBIENTES  
COMPARTILHADOS

RIO DE JANEIRO

2019

DEOVAL LUIS DOS SANTOS JUNIOR  
JOSUÉ PEREIRA DA CONCEIÇÃO  
NATHÁLIA PEDRAL DE CASTRO

SMARTBOX - SISTEMA DE RECOMENDAÇÃO  
MUSICAL PARA AMBIENTES  
COMPARTILHADOS

Trabalho de conclusão de curso de graduação apresentado ao Departamento de Ciência da Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Adriana Santarosa Vivacqua, D.Sc.

RIO DE JANEIRO

2019

S237s

Santos Junior, Deoval Luis dos

Smartbox: sistema de recomendação musical para ambientes compartilhados / Deoval Luis dos Santos Junior, Josué Pereira da Conceição, Nathália Pedral de Castro. – Rio de Janeiro, 2019.

[61] f.

Orientadora: Adriana Santarosa Vivacqua.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal do Rio de Janeiro, Instituto de Matemática, Bacharel em Ciência da Computação, Rio de Janeiro, 2019.

1. Smartbox. 2. Sistemas de recomendação. 3. Sistema de recomendação musical. 4. Colaboração. I. Conceição, Josué Pereira da. II. Castro, Nathália Pedral de. III. Vivacqua, Adriana Santarosa (Orient.). IV. Universidade Federal do Rio de Janeiro, Instituto de Matemática. V. Título.

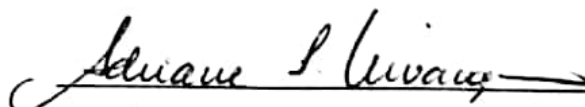
DEOVAL LUIS DOS SANTOS JUNIOR  
JOSUÉ PEREIRA DA CONCEIÇÃO  
NATHÁLIA PEDRAL DE CASTRO

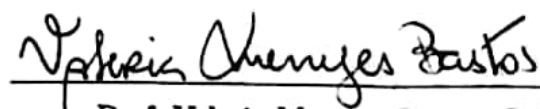
SMARTBOX - SISTEMA DE RECOMENDAÇÃO  
MUSICAL PARA AMBIENTES  
COMPARTILHADOS

Trabalho de conclusão de curso de graduação apresentado ao Departamento de Ciência da Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 17 de JANEIRO de 2019.

BANCA EXAMINADORA:

  
Prof. Adriana Santarosa Vivacqua, D.Sc.

  
Prof. Valeria Menezes Bastos, D.Sc.

  
Prof. Mônica Ferreira da Silva, D.Sc.

Dedicamos este trabalho às nossas famílias: nossos pais que sempre nos apoiaram em tudo o que fizemos desde que nascemos, e se esforçaram ao máximo para nos dar todo o suporte necessário para estarmos aqui hoje.

## AGRADECIMENTOS

Deoval Luis dos Santos Junior

Agradeço a Deus, que foi minha maior força nos momentos de angústia e desespero. Sem ele, nada disso seria possível. Obrigado, senhor, por colocar esperança, amor e fé no meu coração.

Dedico esta, bem como todas as minhas demais conquistas, aos meus familiares. Especialmente, meus pais, Deoval e Graça, que juntos enfrentaram tantas dificuldades para que eu pudesse estudar. Ao meu irmão, Gabriel, obrigado pelo apoio e torcida.

Josué Pereira da Conceição

Primeiramente sou grato a Deus, por ter me ajudado tanto nessa caminhada, me dando forças e motivação para lutar e jamais desistir.

Agradeço também aos meu pais, parentes, namorada e amigos que sempre estiveram comigo nos momentos bons e ruins, para me apoiar e comemorar as conquistas comigo.

Gostaria de agradecer também a todos os professores que nessa trajetória tiveram uma fundamental participação na minha formação, seja com conhecimento ou seja com maturidade e autodidatismo.

Muitos fracassos e muitas notas ruins tentaram nos abater, mas superando todas as derrotas, muitas vitórias foram conquistadas. E a maior delas, maior até que todas as aprovações, foi a resiliência adquirida nos altos e baixos, para que estivesse preparado para o que der e vier, graças a todos estes que estiveram ao meu lado.

Nathália Pedral de Castro

Agradeço imensamente a Deus, por ter me concedido saúde, força e disposição para finalizar a faculdade e o trabalho de final de curso. Sem Ele, nada disso seria possível. Também sou grata ao Senhor por ter dado saúde aos meus familiares e tranquilizado o meu espírito nos momentos mais difíceis da minha trajetória acadêmica até então.

Um agradecimento especial aos meus pais, Norma e Renato, que tanto lutaram pela minha educação e nunca me deixaram perder a fé. Agradeço pelo apoio, força e amor incondicional. Sem vocês a realização desse sonho não seria possível.

Agradeço às minhas amigas Amanda, Esther, Fernanda e Gabrielle. Obrigada pelos inúmeros conselhos, frases de motivação e puxões de orelha. As risadas, que vocês compartilharam comigo durante toda a minha vida acadêmica, também fizeram toda a diferença.

Agradeço aos professores que contribuíram com a minha trajetória acadêmica, especialmente à professora Adriana Vivacqua, responsável pela orientação do nosso projeto.

Por fim, agradeço à UFRJ, por oferecer os recursos necessários que de alguma forma contribuíram para a finalização do curso e realização desse trabalho.



"Quando tudo parecer dar errado em sua vida, lembre-se que o avião decola contra o vento, e não a favor dele."

**Henry Ford**

## RESUMO

Plataforma com o objetivo de reproduzir músicas em um ambiente compartilhado com vários indivíduos. Para isso o sistema deverá ser capaz de analisar quem está no ambiente em um determinado momento e criar uma lista de músicas que irão agradar a esses indivíduos. Antes da geração da playlist, o sistema deverá ser capaz de recomendar com base no perfil dos usuários quais músicas deverão ser adicionadas, analisando sempre os gostos de cada indivíduo e tentando traçar a maior intercessão possível dentre as características presentes.

***Palavras-chave:*** Smartbox. Jukebox. Sistema de Recomendação. Sistema de Recomendação Musical. Colaboração.

## ABSTRACT

*Platform whose main goal is to reproduce musics in a shared environment with multiple individuals. For that, the system must be capable of analyze who is in the environment at a certain moment and create a list of songs that will please those individuals. Before generating the playlist, the system must be capable of recommend, based on the user's profile, which musics will be added, always analyzing each individual's taste and trying to trace the biggest intercession possible between the present characteristics.*

**Keywords:** Smartbox. Jukebox. Recommendation System. Musical Recommendation System. Collaboration.

## LISTA DE FIGURAS

Figura 1:	Arquitetura do Sistema . . . . .	19
Figura 2:	Visão geral da modelagem do banco . . . . .	22
Figura 3:	Página Inicial do Sistema . . . . .	35
Figura 4:	Apresentação do Sistema . . . . .	36
Figura 5:	Lista de Permissões Necessárias . . . . .	37
Figura 6:	A nossa Equipe . . . . .	37
Figura 7:	Acesso ao Sistema precedido pelo Login do Spotify . . . . .	38
Figura 8:	Dashboard do usuário logado . . . . .	39
Figura 9:	A Smartbox do usuário logado . . . . .	40
Figura 10:	Área de acesso às outras Smartbox . . . . .	41
Figura 11:	Diagrama de interseção de gêneros grupo 1 . . . . .	43
Figura 12:	Gráfico de avaliação da playlist gerada para o grupo 1 (notas de 1 a 5, do menor para o maior) . . . . .	43
Figura 13:	Diagrama de interseção de gêneros grupo 2 . . . . .	44
Figura 14:	Gráfico de avaliação da playlist gerada para o grupo 2 . . . . .	45

## LISTA DE TABELAS

Tabela 1: Serviços disponibilizados pela nossa API . . . . .	29
--	----

## LISTA DE ABREVIATURAS E SIGLAS

SR	Sistema de Recomendação
JSON	JavaScript Object Notation
PaaS	Platform as a Service
API	Application Programming Interface

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	MOTIVAÇÃO	15
1.2	A IDEIA	16
1.3	SOBRE O SPOTIFY	16
1.4	PÚBLICO ALVO	16
1.5	SISTEMAS DE RECOMENDAÇÃO	17
1.6	TRABALHO CORRELATOS	17
<b>2</b>	<b>DESENVOLVIMENTO</b>	<b>19</b>
2.1	ARQUITETURA	19
2.2	INFRAESTRUTURA	20
2.2.1	Smartbox Web App	20
2.2.2	Smartbox Server	20
2.2.3	Banco de Dados na Nuvem	20
2.2.4	Deploy e Hospedagem	21
2.3	LICENÇA, VERSIONAMENTO E DISTRIBUIÇÃO	21
<b>3</b>	<b>O SERVIDOR</b>	<b>22</b>
3.1	MODELOS DO BANCO	22
3.1.1	Usuários	23
3.1.2	Músicas	25
3.1.3	Artistas	27
3.1.4	Álbuns	28
3.2	NOSSOS SERVIÇOS	28
3.2.1	Login	29
3.2.2	Callback	30
3.2.3	Coletar e Processar Informações do Usuário	30
3.2.4	Obter Usuário	31
3.2.5	Definir Status de Abertura da Smartbox	32
3.2.6	Obter Participantes da Minha Smartbox	32

3.2.7	Entrar na Smartbox de um Usuário . . . . .	32
3.2.8	Remover todos os Usuários da Smartbox . . . . .	33
3.2.9	Gerar Lista de Músicas a partir de uma Distribuição . . . . .	33
4	A APLICAÇÃO WEB . . . . .	35
4.1	FUNCIONALIDADES . . . . .	35
4.1.1	Página Inicial . . . . .	35
4.1.2	Login . . . . .	37
4.1.3	Dashboard . . . . .	38
4.1.4	Minha Smartbox . . . . .	39
4.1.5	Entrar em uma Smartbox . . . . .	40
5	RESULTADOS OBTIDOS . . . . .	42
6	CONCLUSÃO . . . . .	46
6.1	DIFICULDADES ENCONTRADAS E APRENDIZADO . . . . .	46
6.2	TRABALHOS FUTUROS . . . . .	46
6.2.1	Tratamento de Dados . . . . .	47
6.2.2	Melhorias de Funcionalidades . . . . .	47
	REFERÊNCIAS . . . . .	49
	ANEXO A . . . . .	51
	ANEXO B . . . . .	55



## 1 INTRODUÇÃO

Desde os princípios da humanidade, a música sempre esteve presente como um instrumento de percepção dos sons já presentes na natureza. Com a criação da sociedade moderna, a música se transformou em uma ferramenta para expressão artística que tinha como objetivo entreter e principalmente criar vínculos dos seres humanos entre si durante eventos sociais. Com o passar do tempo, o cenário musical se adaptou e evoluiu sempre acompanhando os desenvolvimentos da sociedade e suas necessidades.

Avançando para o cenário moderno, a tecnologia se faz cada vez mais presente no cotidiano das pessoas, as deixando por muitas vezes desconectadas da sociedade no mundo real. A música atual não diverge dessa realidade. Os principais sistemas voltados para esse mercado podem exemplificar esse fenômeno individualista vivido pela sociedade moderna. Empresas tais como Spotify, Apple Music e Soundcloud tem como objetivo criar a melhor experiência musical para cada usuário de uma forma particular, não necessariamente focando em ambientes compartilhados.

Tendo isso em vista, o projeto foca em grupos de pessoas que possam ter algum tipo de atividade em comum, desde ambientes de trabalhos até encontros sociais, com o objetivo de trazer novamente o carácter social da música para os dias atuais, utilizando algumas das mais recentes tecnologias disponíveis.

### 1.1 MOTIVAÇÃO

A principal motivação para o projeto partiu do desejo de se ter uma plataforma que permitisse a integração de preferências musicais de diversos indivíduos que compartilhem ambientes em comum, tais como: locais de trabalho, academias, encontros sociais, festas, entre outros. Após a realização de algumas pesquisas, foi possível constatar que esse tipo de sistema não era comum, visto que as principais empresas no mercado como Spotify, Pandora, Apple Music e SoundCloud focam na recomendação individual para seus usuários, e não em uma recomendação colaborativa para

um grupo deles.

## 1.2 A IDEIA

A ideia principal se baseia em criar um ambiente colaborativo onde as preferências de todos os usuários envolvidos possam ser levadas em consideração. Cada indivíduo pode criar sua própria 'sala' privada, podendo convidar para a mesma quantos usuários desejar. O sistema é visado para ser utilizado em locais onde grupos prevaleçam, já que a recomendação final tem como resultado a criação de uma playlist que será o produto do balanceamento entre os mais diversos gêneros musicais. Após criada, a lista de músicas fica disponível na conta do spotify do usuário dono da 'sala', podendo ser usufruída a qualquer momento.

## 1.3 SOBRE O SPOTIFY

O Spotify é um serviço de streaming de músicas, podcasts e vídeos lançado em 7 de outubro de 2008[1]. Trata-se de uma das plataformas musicais mais importantes da atualidade, contabilizando um acervo vasto de artistas, álbuns e gêneros para todos os gostos pessoais. A ferramenta permite buscas personalizadas, além de contar com um sistema de recomendação para o usuário baseado em suas playlists e músicas salvas. Em 2014[1], o Spotify lançou uma API Web, que permite o acesso à alguns conjuntos de dados por desenvolvedores externos, possibilitando a criação de diversos tipos de projetos inovadores.

## 1.4 PÚBLICO ALVO

Inicialmente, o principal alvo do sistema são usuários que já possuem uma conta ativa no Spotify, já que todo o desenvolvimento se baseia no uso de suas APIs. Além disso, tendo em vista o objetivo inicial do projeto, o propósito da ideia é focar em grupos de diversas vertentes sociais, desde pequenos encontros a atividades em conjunto. Contudo, o sistema também pode ser utilizado individualmente pelo

usuário como uma forma de recomendação para novas músicas, baseando-se no seu gosto pessoal.

## 1.5 SISTEMAS DE RECOMENDAÇÃO

De uma forma geral, um sistema de recomendação é um conjunto de algoritmos que utilizam técnicas computacionais na descoberta de novos produtos ou conteúdos para um indivíduo ou um grupo de indivíduos de forma personalizada. A longo prazo, o sistema tende a se aperfeiçoar e aprender com as escolhas do usuário ou grupo, deixando a recomendação mais exata. Atualmente existem duas metodologias principais de filtragem para gerar as recomendações: Baseada em conteúdo (*Content-based recommendation*) e Colaborativa (*Collaborative filtering*)[2].

Na metodologia baseada em conteúdo, o sistema utiliza produtos ou conteúdos que o usuário previamente já demonstrou interesse e recomenda novos produtos ou conteúdos que apresentem características similares ou em comum. Por exemplo, se um usuário assistir uma série de terror em uma plataforma que implementa um sistema de recomendação, a tendência é que em seguida seja recomendado à esse usuário séries com o mesmo gênero ou com os mesmos atores.

Na metodologia colaborativa, o sistema leva em consideração todos os usuários da plataforma. A recomendação é feita baseada nos fatores em comum entre os próprios usuários, e não leva em consideração o produto ou conteúdo que será recomendado de fato. O sistema verifica pontos de interseção entre os usuários, como por exemplo idade, gênero, localidade, entre outros, e então recomenda produtos ou conteúdos que um indivíduo já mostrou interesse previamente para um segundo indivíduo, que possui essas interseções com o primeiro.

## 1.6 TRABALHO CORRELATOS

A seguir iremos listar alguns trabalhos semelhantes, fazendo uma comparação deles com o nosso trabalho:

- **Adaptive Radio**[23]: Esse sistema seleciona diversas músicas para um ambiente compartilhado, mas ao invés de focar em recomendar o que os usuários querem ouvir, ele foca no que os usuários não querem ouvir, fazendo análises de similaridade para ver o que mais os usuários podem não gostar, baseando-se nos votos negativos dados nas músicas, caso contrário, será assumido que o usuário gosta de todas as músicas sendo tocadas.

A grande vantagem do nosso sistema comparado a esse, é que não há a necessidade do usuário realizar qualquer interação de voto, ou algo mais. Apenas serão coletadas as músicas que mais se escutam e feita uma análise de gosto em cima delas.

- **MusicFX**[24]: Esse é um sistema voltado para o ambiente de academia, onde o objetivo é recolher os votos de cada usuário, dado para determinados gêneros musicais mediante o preenchimento de um formulário. A partir disso, o algoritmo arbitra as músicas baseando-se nas preferências dos usuários presentes na academia.

A ideia desse sistema é bem semelhante a nossa, diferenciando-se apenas na parte da votação, sendo a nossa automatizada baseando-se nas músicas escutadas do indivíduo.

## 2 DESENVOLVIMENTO

### 2.1 ARQUITETURA

A arquitetura foi elaborada para utilizar o que há de mais moderno na nossa época. Implementando uma API em Node.js[3] com Express.js[4], onde estão os nossos serviços que conversam com a API do Spotify[5], e persistindo os dados em um banco de dados NoSQL na nuvem, o Firestore[6] da Google. Essa API é consumida por um Web App que faz a interface com o usuário.

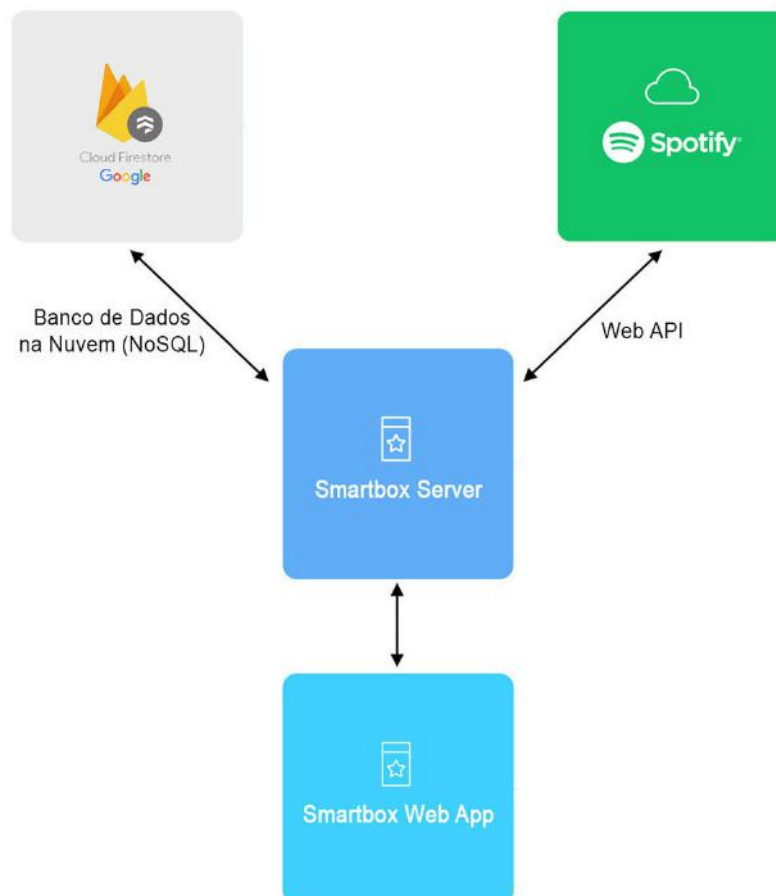


Figura 1: Arquitetura do Sistema

## 2.2 INFRAESTRUTURA

### 2.2.1 Smartbox Web App

Para montar a aplicação web, foram utilizadas as tecnologias básicas para desenvolvimento web, sendo elas o HTML[7] para marcação de hipertexto, o CSS[8] para definir os estilos em cascata, e o Javascript[9] para a manipulação dos elementos em tela, executar determinadas funcionalidades e comunicar com o nosso servidor para consumir os serviços criados por nós.

Para auxiliar a construção das funcionalidades e manipulação dos elementos em tela, utilizamos o jQuery[10], que é uma biblioteca com inúmeros recursos de manipulação e animação dos elementos, tratativas de eventos e comunicação com APIs, o que é essencial para o nosso desenvolvimento.

### 2.2.2 Smartbox Server

No lado do servidor, foi utilizado Node.js[3], que é um interpretador de código Javascript[9] para servidores, que foi baseado no motor V8[11] do Chrome[12]. Sua grande vantagem é a facilidade de criar aplicações de alta escalabilidade, que inclusive permitem muito mais acessos simultâneos, se comparado com seus concorrentes mais antigos que executam, por exemplo, PHP.

Utilizamos o Express.js[4] para auxiliar o desenvolvimento da nossa API, com uma fácil construção e gerenciamento dos acessos aos serviços que disponibilizamos. Ele é um framework web de alto desempenho que fornece um conjunto robusto de recursos para acelerar o desenvolvimento sem onerar o custo de tempo e processamento.

### 2.2.3 Banco de Dados na Nuvem

O banco de dados utilizado é o disponibilizado pela Google, o Cloud Firestore[6]. Ele é um banco de dados não-relacional (NoSQL), que possui uma versão gratuita

que limita a quantidade de consultas e escritas no banco, mas também poderá ser fechado um pacote de "pay on-demand", onde se paga pelo que for usado.

Como o objetivo do projeto era montar uma arquitetura orientada a microsserviços, optamos pela utilização de um banco de dados na nuvem, sendo o NoSQL a única opção encontrada nos produtos da Google.

#### **2.2.4 Deploy e Hospedagem**

Todo o sistema (Serviços e Web App) estão hospedados no Heroku[13]. O Heroku é um PaaS na nuvem, onde ele é o responsável por verificar alterações na branch master do repositório no Github e fazer o deploy, automaticamente. Atualmente, ele oferece suporte a várias linguagens de programação, e possui uma versão gratuita que atende à nossa demanda. Caso haja necessidade, os recursos de sistema necessários poderão ser facilmente escaláveis, pagando-se por isso.

### **2.3 LICENÇA, VERSIONAMENTO E DISTRIBUIÇÃO**

O código possui uma licença GNU General Public License v3.0, o qual permite o uso desse código, desde que o projeto derivado use também essa licença.

O código foi versionado usando o Git, utilizando um repositório remoto no Github, o qual poderá ser encontrado em <https://github.com/deoval/Smartbox>.

O sistema poderá ser acessado gratuitamente, e utilizado à vontade e quando desejar, no seguinte link: <https://smartbox-ufrj.herokuapp.com>.

### 3 O SERVIDOR

#### 3.1 MODELOS DO BANCO

A seguir, serão apresentados os modelos que são usados no banco de dados na nuvem (Firestore). Como se trata de um banco de dados NoSQL, os modelos (chamados de documentos) serão detalhados em formato JSON, e serão apresentados de forma geral com uma diagramação semelhante ao ER, pois não há um padrão de diagramação definido.

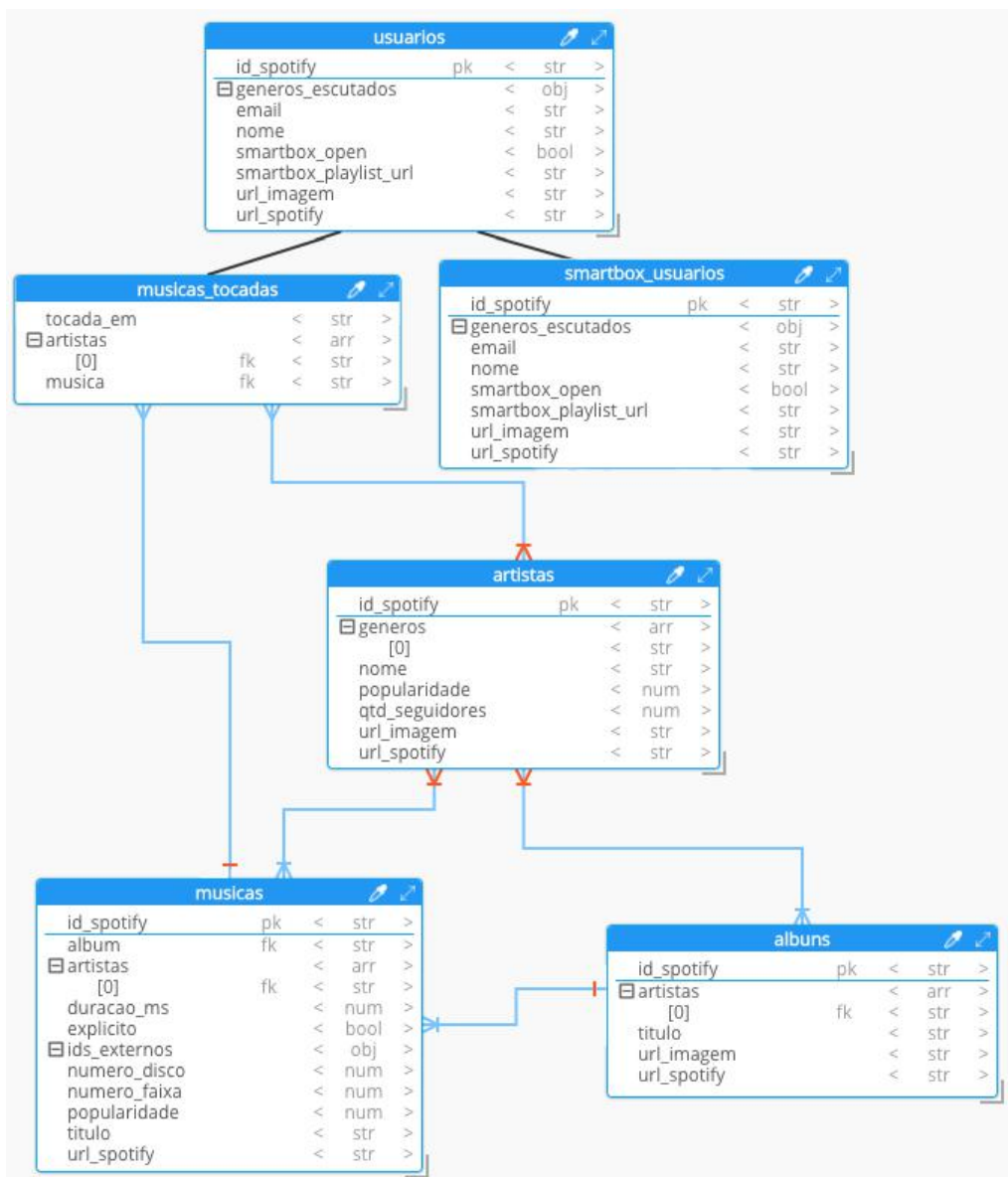


Figura 2: Visão geral da modelagem do banco



### 3.1.1 Usuários

Este é o modelo que representa a coleção *usuarios* do banco de dados, que é responsável por armazenar as informações pertinentes aos usuários que acessam o sistema:

```
{
  _id: "12165811***",
  email: "jos***@***.com",
  generos_escutados: {
    pop: 27,
    rock: 18,
    hip-hop: 11,
    // ...
  },
  id_spotify: "12165811***",
  nome: "Josue Pereira",
  smartbox_open: true,
  smartbox_playlist_url: "https://open.spot...",
  url_imagem: "https://scontent.xx.fbcd...",
  url_spotify: "https://open.spotify.co..."
}
```

- **\_id:** String - É a chave primária, que recebe o mesmo valor do campo `id_spotify`;
- **email, id\_spotify, nome, url\_imagem e url\_spotify:** String - São dados gerais do usuário, recuperados na API do Spotify;
- **generos\_escutados:** Object - Um objeto contendo os gêneros como atributos e respectivamente a quantidade de vezes escutado;
- **smartbox\_open:** Boolean - Campo que define o status de abertura da Smartbox para a entrada de participantes;

- **smartbox\_playlist\_url**: String - Link da última playlist gerada no Spotify.

### 3.1.1.1 Músicas Tocadas

Dentro da coleção *usuarios* existe uma coleção chamada *musicas\_tocadas*, que é responsável por armazenar as músicas escutadas pelo usuário, com data e referência para música e artistas:

```
{
  _id: "07nH4ifBxUB4lZcsf44Brn2018-12-06T01:10:47.172Z",
  artistas: [
    </artistas/7CajNmpbOovFoOoasH2HaY>,
    </artistas/34v5MVKeQnIo0CWYMbbrPf>,
    // ...
  ],
  musica: </musicas/07nH4ifBxUB4lZcsf44Brn>,
  tocada_em: "2018-12-06T01:10:47.172Z"
}
```

- **\_id**: String - É a chave primária, que recebe como valor a junção do id da música e da data de quando foi tocada;
- **artistas**: Array<Reference> - Um array com referências de documentos de artistas;
- **musica**: Reference - Um campo com a referência de um documento de música;
- **tocada\_em**: String - Um campo com a data e hora em ISO 8601, de quando a música foi escutada.

### 3.1.1.2 Smartbox Usuários

Este é um modelo da coleção *smartbox\_usuarios*, que fica dentro da coleção *usuarios*, responsável por armazenar os usuários que estão participando da Smartbox

de outro usuário:

```
{
  _id: "10164125***",
  usuario: {
    email: "nathal***@***.com",
    generos_escutados: {
      pop: 22,
      dance: 19,
      house: 13,
      // ...
    },
    id_spotify: "10164125***",
    nome: "Nathalia Pedral",
    smartbox_open: true,
    smartbox_playlist_url: "https://open.spotify...",
    url_imagem: "https://scontent.xx.fbcd...",
    url_spotify: "https://open.spotify.co..."
  }
}
```

- **\_id**: String - É a chave primária, que recebe o mesmo valor do campo id do usuário;
- **usuario**: Object - Lugar onde fica uma cópia do usuário participante, feita do momento em que acessou a Smartbox do usuário dono.

### 3.1.2 Músicas

Este é o modelo da coleção *musicas*, responsável por armazenar as informações das músicas recuperadas do Spotify:

```

{
  _id: "003eoIwxETJujVWmNFMoZy",
  album: </albuns/3mzHuQmDLVuecodQNKUd0T>,
  artistas: [
    </artistas/1BajNmpbOovFoOoasH3HhY>,
    </artistas/7Fv6MVKeQnIo0CWYMbbrBc>,
    // ...
  ],
  duracao_ms: 193680,
  explicito: false,
  id_spotify: "003eoIwxETJujVWmNFMoZy",
  ids_externos: {
    isrc: "USUM71808193"
  }
  numero_disco: 1,
  numero_faixa: 1,
  popularidade: 68,
  titulo: "Growing Pains",
  url_spotify: "https://open.spotify..."
}

```

- **\_id**: String - É a chave primária, que recebe o mesmo valor do campo `id_spotify`;
- **duracao\_ms**, **explicito**, **id\_spotify**, **titulo**, **url\_spotify**: String - São dados gerais da música, recuperados na API do Spotify;
- **numero\_disco**, **numero\_faixa**, **popularidade**: Number - São dados gerais da música, recuperados na API do Spotify;
- **artistas**: Array<Reference> - Um array com referências de documento de artistas;

- **ids\_externos:** Object - Um objeto contendo ids de algumas referências externas.

### 3.1.3 Artistas

Este é o modelo da coleção *artistas*, responsável por armazenar as informações dos artistas recuperadas do Spotify:

```
{
  _id: "06HL4z0CvFAxyc27GXpf02",
  generos: [
    "dance pop",
    "pop",
    "post-teen pop",
    // ...
  ]
  id_spotify: "06HL4z0CvFAxyc27GXpf02",
  nome: "Taylor Swift",
  popularidade: 89,
  qtd_seguidores: 16550218,
  url_imagem: "https://i.scdn.co/ima...",
  url_spotify: "https://open.spotify..."
}
```

- **\_id:** String - É a chave primária, que recebe o mesmo valor do campo `id_spotify`;
- **id\_spotify, nome, url\_imagem e url\_spotify:** String - São dados gerais do usuário, recuperados na API do Spotify;
- **popularidade, qtd\_seguidores:** Number - São dados gerais do usuário, recuperados na API do Spotify;

- **generos:** Array<String> - Um array contendo os gêneros musicais relacionados ao artista.

### 3.1.4 Álbuns

Este é o modelo da coleção *albuns*, responsável por armazenar as informações dos álbuns recuperadas do Spotify:

```
{
  _id: "00XeMtw9lHuOzG2stN4ny4",
  artistas: [
    </artistas/5JG9GISYjRLQUJMH2C6iJ5>,
    // ...
  ],
  id_spotify: "00XeMtw9lHuOzG2stN4ny4",
  titulo: "Misturadin (Ao Vivo)",
  url_imagem: "https://i.scdn.co/im...",
  url_spotify: "https://open.spotify..."
}
```

- **\_id:** String - É a chave primária, que recebe o mesmo valor do campo `id_spotify`;
- **id\_spotify, titulo, url\_imagem e url\_spotify:** String - São dados gerais do usuário, recuperados na API do Spotify;
- **artistas:** Array<Reference> - Um array com referências de documento de artistas.

## 3.2 NOSSOS SERVIÇOS

Para lidar com a parte pesada do processamento, segurança e armazenamento dos dados, foi criado um backend com serviços disponíveis para serem consumidos

por qualquer aplicação.

URI	Método	Request	Response
/login	GET	-	-
/callback	GET	code, state	access_token
/collectAndProcessUserInfo	GET	access_token	booleano
/getUser/{userId?}	GET	access_token	Usuario
/setSmartboxOpenStatus	GET	access_token, status	booleano
/getMySmartboxUsers	GET	access_token	Usuarios
/enterInSomeoneSmartbox	GET	access_token, userID	mensagem
/removeAllSmartboxUsers	GET	access_token	booleano
/generatePlaylist	POST	access_token, distribuicao, limit	link_playlist

Tabela 1: Serviços disponibilizados pela nossa API

Mais detalhes sobre cada serviço serão descritos a seguir.

### 3.2.1 Login

URI	Método	Request	Response
/login	GET	-	-

Redireciona o usuário para a autenticação OAuth 2.0[14] do Spotify, solicitando as credenciais de acesso, e em seguida retorna a resposta, com um código e estado, para o nosso serviço Callback (na Seção 3.2.2).

Nesse serviço são definidas as permissões de acesso à conta, que são requisitadas para o usuário utilizar o sistema. São elas:

- *user-read-private*: Acessar os detalhes da inscrição do usuário (tipo de conta de usuário);
- *user-read-email*: Obter o endereço de e-mail do usuário;

- *user-library-read*: Acessar o que o usuário escutou recentemente;
- *user-read-recently-played*: Acessar as faixas, álbuns e programas salvos;
- *playlist-modify-public*: Acesso às listas de reprodução públicas do usuário (para gerar as playlists).

### 3.2.2 Callback

URI	Método	Request	Response
/callback	GET	code, state	access_token

Serviço responsável, exclusivamente, por captar a resposta do login do usuário com um código e estado gerados, que serão usados para recuperar o token de acesso[14] do usuário logado (`access_token`).

Em seguida, após o sucesso da recuperação do token do usuário, é feita a coleta e processamento de informações do usuário (na Seção 3.2.3).

### 3.2.3 Coletar e Processar Informações do Usuário

URI	Método	Request	Response
/collectAndProcessUserInfo	GET	access_token	booleano

Recebe apenas o `access_token` para recuperar o usuário, e efetuar a coleta e o processamento de dados das últimas músicas escutadas.

Após recuperar o usuário no Spotify, os dados são salvos ou atualizados no banco de dados do sistema. Em seguida, verifica se existem músicas escutadas salvas no usuário, recuperando elas. Após isso, é feito um acesso à API do Spotify para pegar as últimas 50 músicas escutadas[15] (essencial para solucionar o problema do *Cold Start*), e adicionar em uma lista, retirando as músicas escutadas que sejam de uma playlist Smartbox própria, para impedir de poluir a recomendação do próprio usuário com músicas que foram escolhidas para um grupo.



Com essa lista de músicas escutadas filtrada é feito mais um filtro para retirar as músicas escutadas já salvas no usuário, ou seja, só irá pegar as novas músicas escutadas. Iterando sobre essa lista, insere ela na coleção de `musicas_tocadas` do usuário, e salva no banco de dados a música e o álbum, para popular o banco com dados para serem usados futuramente.

Nessa lista de músicas duplamente filtrada, a partir de cada música, é extraído e recuperado na API do Spotify todos os artistas que tem nela, em vários lotes de 50 ids de artistas[16]. Após recuperar todos os artistas, eles serão salvos no banco de dados para serem usados futuramente. Em seguida, será feita a contagem de gêneros escutados, a partir, dos gêneros de cada artista, juntamente com as musicas escutadas já salvas no usuário.

### 3.2.4 Obter Usuário

URI	Método	Request	Response
/getUser/{userId?}	GET	access_token	Usuario

Passando-se o token de acesso pela query e, opcionalmente, o id do usuário no URI, obtém um usuário salvo no banco de dados.

Ao passar um id, o usuário será buscado no banco e retornado. Ao omitir o id, será feita uma consulta à API do Spotify, usando o `access_token`, para buscar o usuário atualmente logado[17], e em seguida usar o seu id para buscar o usuário no banco de dados.

Ao receber os dados do usuário do banco de dados, é feito um filtro dos gêneros escutados salvos, para ser usado apenas o que for aceito pela busca do Spotify[18]. Em seguida, é feito um recálculo percentual de cada gênero filtrado escutado.

### 3.2.5 Definir Status de Abertura da Smartbox

URI	Método	Request	Response
/setSmartboxOpenStatus	GET	access_token, status	booleano

Envia o access\_token e o status, em booleano, para definir se a Smartbox do usuário logado ficará aberta para os outros usuários acessarem, ou não. Obtém como resposta se foi possível, ou não, efetuar essa mudança.

### 3.2.6 Obter Participantes da Minha Smartbox

URI	Método	Request	Response
/getMySmartboxUsers	GET	access_token	Usuarios

Monta uma lista com os usuários cadastrados na coleção de "smartbox\_usuarios" do usuário logado, e adiciona o próprio nessa lista.

Nessa lista de usuários é feito o filtro de gêneros (como mencionado na Seção 3.2.4), com o cálculo da distribuição percentual dos gêneros escutados de cada usuário.

### 3.2.7 Entrar na Smartbox de um Usuário

URI	Método	Request	Response
/enterInSomeoneSmartbox	GET	access_token, userID	mensagem

Recebe o access\_token e o id do usuário alvo ao qual deseja participar da Smartbox.

Existem 3 etapas de verificação antes de efetuar a entrada na Smartbox do usuário desejado:

- Verifica se o usuário alvo está cadastrado no Sistema;

- Verifica se o usuário alvo não é o próprio usuário logado;
- Verifica se o usuário alvo definiu o status de abertura da Smartbox para "Aberto".

Passando por essas etapas, é feito o cadastro na coleção de "smartbox\_usuarios" do usuário alvo, com a cópia do usuário logado que fez o pedido de entrada. Em seguida, emite mensagem através do Socket.IO[19] para indicar que a tela do usuário alvo precisar atualizar para o novo participante, e retorna uma mensagem de sucesso.

### 3.2.8 Remover todos os Usuários da Smartbox

URI	Método	Request	Response
/removeAllSmartboxUsers	GET	access_token	booleano

Recupera a coleção de "smartbox\_usuarios" do usuário logado e deleta todos os documentos dela.

### 3.2.9 Gerar Lista de Músicas a partir de uma Distribuição

URI	Método	Request	Response
/generatePlaylist	POST	access_token, distribuicao, limit	link_playlist

Recebe o access\_token, o limite de músicas que se deseja ter na playlist e uma lista com a distribuição percentual dos gêneros escutados no qual deseja se embasar para criar uma nova playlist, por exemplo:

```
{
  "pop": 34,
  "rock": 23,
  "hip-hop": 12,
  // ...
}
```

Varre-se essa lista de gêneros escutados pegando, por ordem decrescente, o percentual e buscando[20] na API do Spotify a quantidade de músicas desse percentual sobre o limite passado, filtrando a busca por músicas de no mínimo 50 de popularidade (sendo o máximo 100, devido a uma limitação do Spotify).

Após obter todas as músicas de cada gênero, é criada[21] uma playlist pública na conta do usuário logado, com o padrão de nome sendo: Smartbox <data> <hora>. Em seguida são adicionadas[22] todas essas músicas buscadas na playlist criada. E, por fim, retornado o link para a playlist gerada, estando disponível para ser reproduzida.

## 4 A APLICAÇÃO WEB

A aplicação web foi feita utilizando HTML5 e CSS3, juntamente com o jQuery para manipular os elementos do HTML e consumir os serviços disponibilizados pela API.

### 4.1 FUNCIONALIDADES

#### 4.1.1 Página Inicial

A página inicial consta de uma curta apresentação inicial a respeito do projeto e seus desenvolvedores, além de listar todas as permissões que são requisitadas do usuário para que se possa acessar os dados através da API do Spotify. Além disso, há um botão que redireciona o usuário para a tela de login do Spotify.

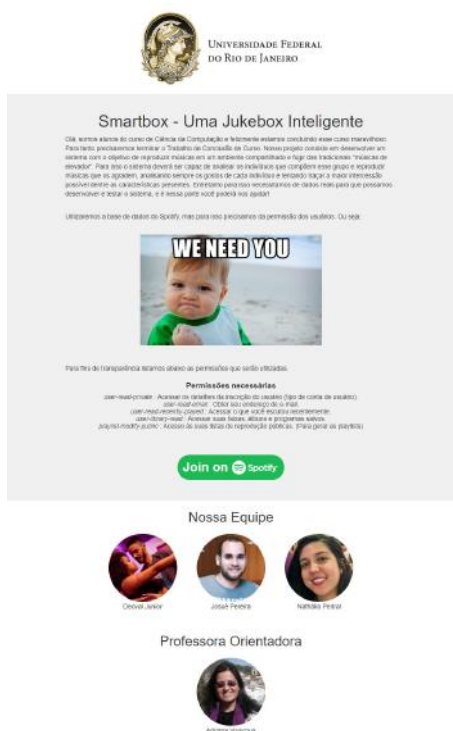


Figura 3: Página Inicial do Sistema

#### 4.1.1.1 Apresentação do Sistema

Um breve texto que apresenta o sistema, indicando rapidamente os principais objetivos do projeto, descrevendo como será seu funcionamento e a necessidade de dados de usuários reais para garantir a validação do sistema.



UNIVERSIDADE FEDERAL  
DO RIO DE JANEIRO

**Smartbox - Uma Jukebox Inteligente**

Olá, somos alunos do curso de Ciência da Computação e felizmente estamos concluindo esse curso maravilhoso. Para tanto precisaremos terminar o Trabalho de Conclusão de Curso. Nosso projeto consiste em desenvolver um sistema com o objetivo de reproduzir músicas em um ambiente compartilhado e fugir das tradicionais "músicas de elevador". Para isso o sistema deverá ser capaz de analisar os indivíduos que compõem esse grupo e reproduzir músicas que os agradem, analisando sempre os gostos de cada indivíduo e tentando traçar a maior intercessão possível dentre as características presentes. Entretanto para isso necessitamos de dados reais para que possamos desenvolver e testar o sistema, e é nessa parte você poderá nos ajudar!

Utilizaremos a base de dados do Spotify, mas para isso precisamos da permissão dos usuários. Ou seja:

A meme featuring a baby with a serious expression, wearing a green and white shirt, with the text "WE NEED YOU" overlaid in large, bold, white letters with a black outline.

Figura 4: Apresentação do Sistema

#### 4.1.1.2 Descrição de Permissões

Detalhamento das permissões que precisam ser concedidas por cada usuário para a utilização do sistema. As permissões garantem o acesso a certas informações da conta do Spotify do usuário em questão, sendo a principal o acesso às últimas músicas escutadas, que serve de base para a população do banco de dados.

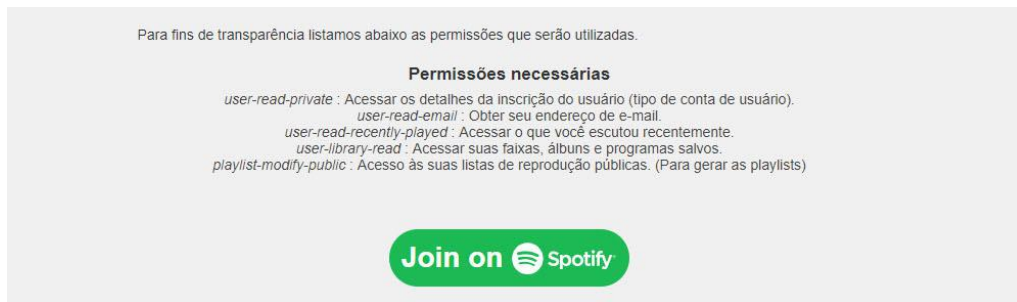


Figura 5: Lista de Permissões Necessárias

#### 4.1.1.3 Equipe

Uma pequena exibição e descrição dos membros da equipe, sendo os alunos responsáveis pelo desenvolvimento do TCC e a professora orientadora.

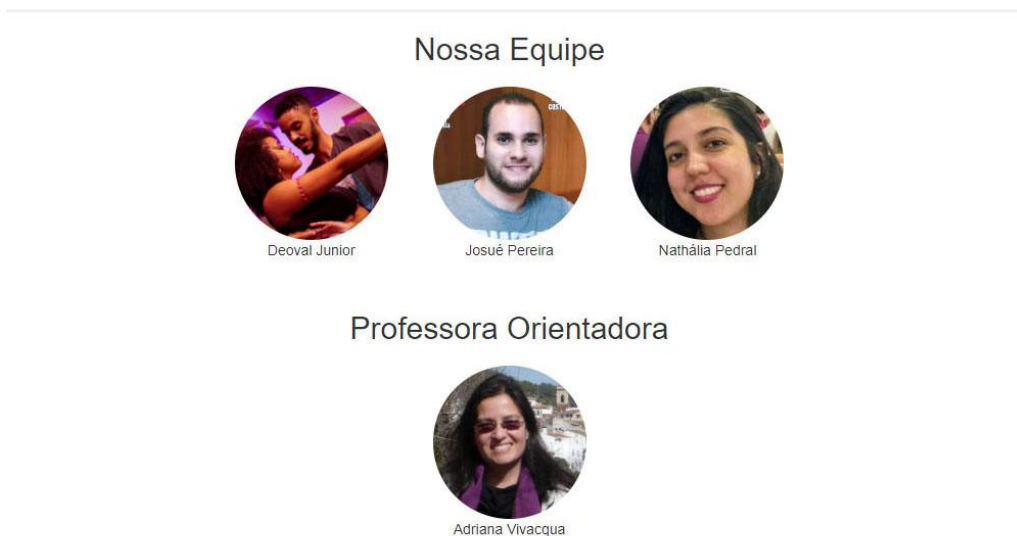


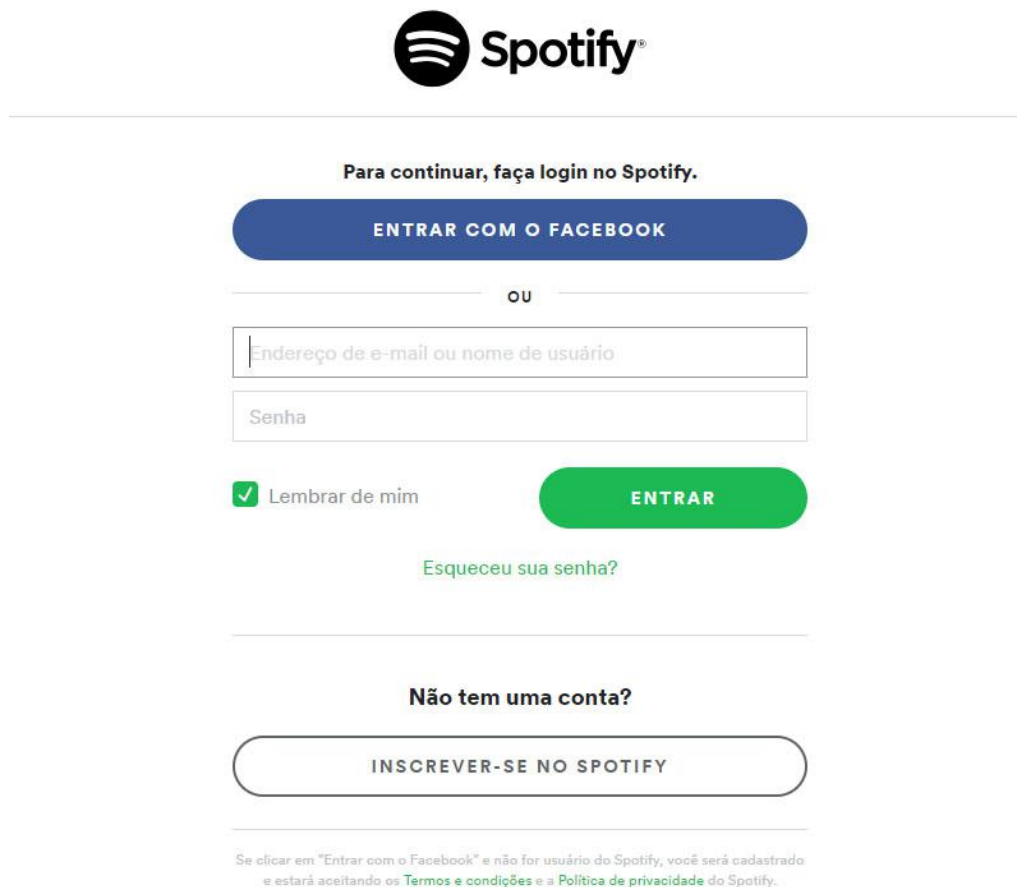
Figura 6: A nossa Equipe

#### 4.1.2 Login

Área de autenticação do usuário ao Spotify, sendo redirecionado em seguida para o nosso sistema com o token de acesso.

Nessa área, após entrar com email e senha, será solicitada as permissões de acesso do sistema Smartbox à conta do usuário, exibindo novamente os requisitos

mencionados na página inicial do sistema.



The image shows the Spotify login interface. At the top is the Spotify logo. Below it, the text reads "Para continuar, faça login no Spotify." There are two main options: "ENTRAR COM O FACEBOOK" (a blue button) and "OU" (or). Below "OU" are two input fields: "Endereço de e-mail ou nome de usuário" and "Senha". There is a checkbox labeled "Lembrar de mim" which is checked. To the right of the checkbox is a green "ENTRAR" button. Below the "ENTRAR" button is a link "Esqueceu sua senha?". At the bottom, there is a section titled "Não tem uma conta?" with a button "INSCREVER-SE NO SPOTIFY". At the very bottom, there is a small disclaimer: "Se clicar em 'Entrar com o Facebook' e não for usuário do Spotify, você será cadastrado e estará aceitando os Termos e condições e a Política de privacidade do Spotify."

Figura 7: Acesso ao Sistema precedido pelo Login do Spotify

### 4.1.3 Dashboard

Apresenta a página principal do sistema que contém informações básicas do usuário, além da análise de gêneros do usuário e os botões para acessar uma sala já existente de outro usuário (Entrar em uma Smartbox) ou criar uma nova sala, pertencente ao usuário logado (Abrir minha Smartbox).





Figura 8: Dashboard do usuário logado

#### 4.1.4 Minha Smartbox

Essa é a área mais importante do sistema, pois é nela que a proposta principal do sistema está sendo trabalhada. A "Minha Smartbox" tem o funcionamento semelhante ao de uma sala de chat, onde as pessoas entram e interagem, mas nesse caso sem a conversa. O objetivo é a pessoa entrar nessa sala utilizando a funcionalidade de "Entrar em uma Smartbox" (descrito mais a frente) e, após entrar, o sistema recalculará as recomendações do grupo com essa nova pessoa.

Apenas o dono da Smartbox poderá ver quem está nela e executar comandos, tais como:

- Remover Participantes, que serve para esvaziar a sala de todos os participantes, deixando apenas o dono nela;
- Gerar Nova Playlist, que serve para usar as informações coletadas de gêneros recomendados para o grupo, e buscar na API do Spotify músicas com esses gêneros (descrito no serviço de gerar playlist);

- Acessar Playlist Gerada, para redirecionar para o player do Spotify com a última lista gerada.

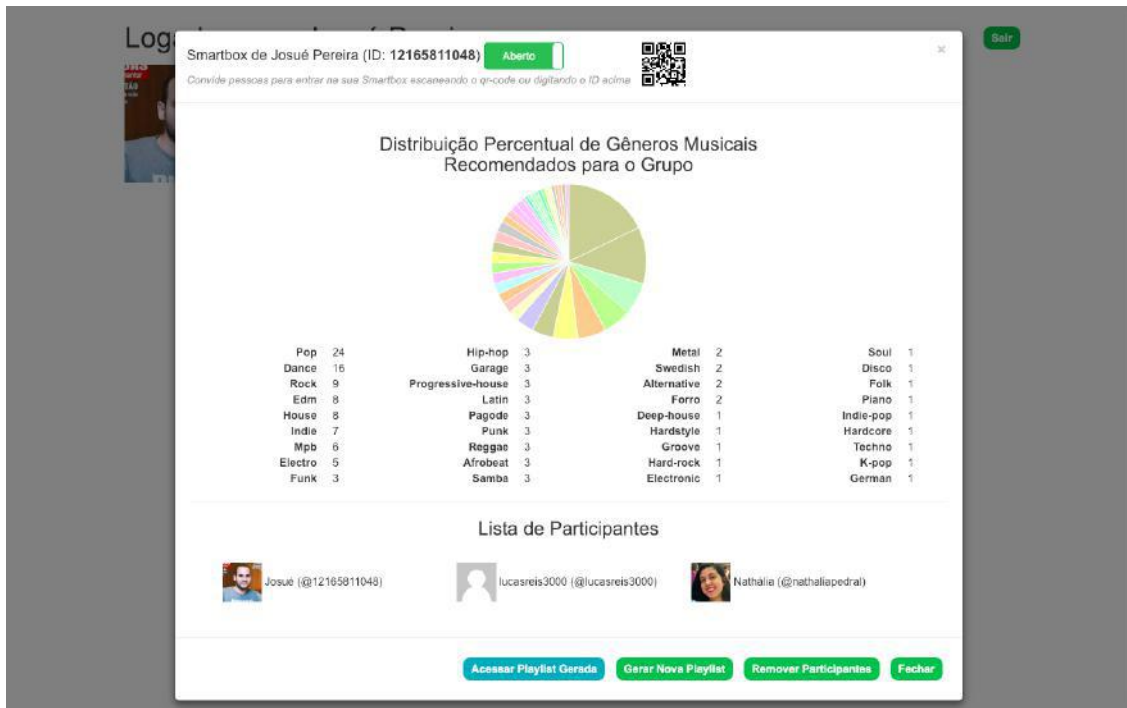


Figura 9: A Smartbox do usuário logado

#### 4.1.5 Entrar em uma Smartbox

Nessa área, o usuário poderá acessar a Smartbox de outro usuário de duas formas, sendo possível apenas se o dono da Smartbox tiver habilitado a entrada de novos participantes.

O usuário pode acessar outra Smartbox digitando o ID do usuário dono, ou direcionando a câmera (requer permissão) para o qr-code gerado na tela da Smartbox do usuário dono.

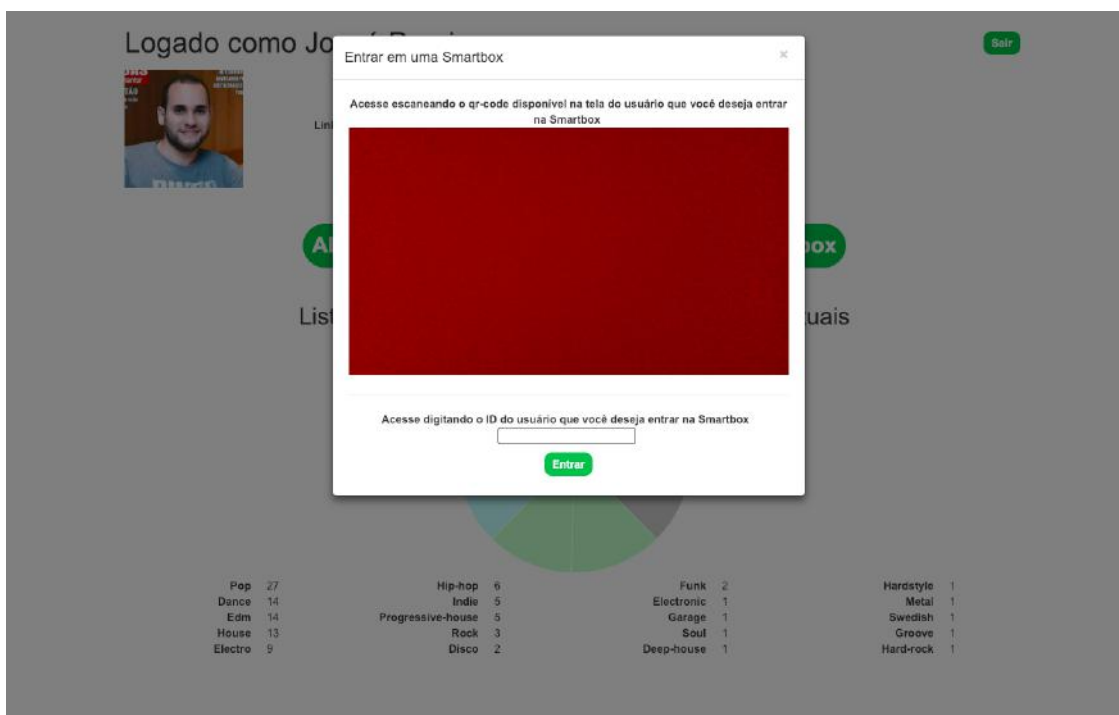


Figura 10: Área de acesso às outras Smartbox

## 5 RESULTADOS OBTIDOS

Para a validação e análise do sistema, foram realizados alguns testes com usuários reais. Os testes baseiam-se na criação de uma nova sala com os novos usuários. Uma vez que todos estejam logados, o sistema é capaz de gerar uma playlist recomendada a esse grupo específico. O dono da sala então compartilha a playlist, que possui entre 10 e 15 músicas, com os indivíduos envolvidos. Todos os participantes então devem escutar a mesma por completo, sendo escutado pelo menos um minuto e 30 segundos de cada música, para garantir uma análise mais completa sobre cada música. Então, os usuários respondem um rápido questionário (Anexo A) a respeito das músicas escutadas, seus gostos pessoais e a experiência com o sistema em si.

Para caráter de controle, cada grupo recebeu um número de identificação, para que pudesse ser feita uma análise mais completa dos resultados focando em cada grupo social atingido pelo teste. Além disso, os desenvolvedores do sistema agiram como donos das smartboxes utilizadas durante os testes, com o objetivo de guiar os participantes de forma mais apropriada durante todo o processo. Visto que os desenvolvedores fazem realmente parte dos grupos na vida real, essa decisão não acarretou em alterações nas recomendações finais nem influenciou as análises que serão detalhadas a seguir (o formulário não foi respondido pelos desenvolvedores).

Abaixo segue um breve relatório de cada resultado por grupo.

- Grupo 1: Social entre amigos (4 usuários)

O grupo 1 é definido por um grupo de amigos contendo quatro participantes na faixa etária de 25 a 30 anos, quase todo formado por mulheres. Para esse grupo em questão, o sistema seria usado principalmente durante os encontros sociais do mesmo.

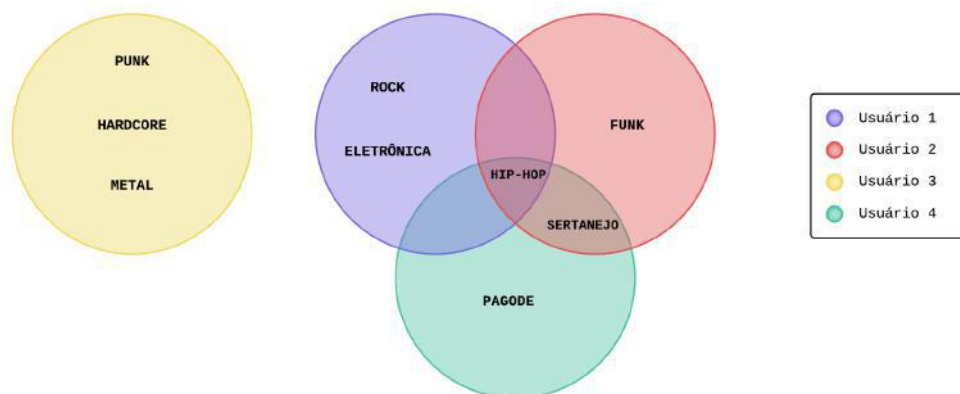


Figura 11: Diagrama de interseção de gêneros grupo 1

A figura 10 mostra a interseção entre os gêneros mais escutados pelos usuários do grupo 1 baseada nas respostas do questionário, com notas de 1 a 5, do menor para o maior. É possível observar que existem alguns gêneros em comum, o que leva-se a acreditar que haverá uma maior aceitação das músicas recomendadas pelo sistema, ainda que um dos usuários não participe da interseção com nenhum outro.

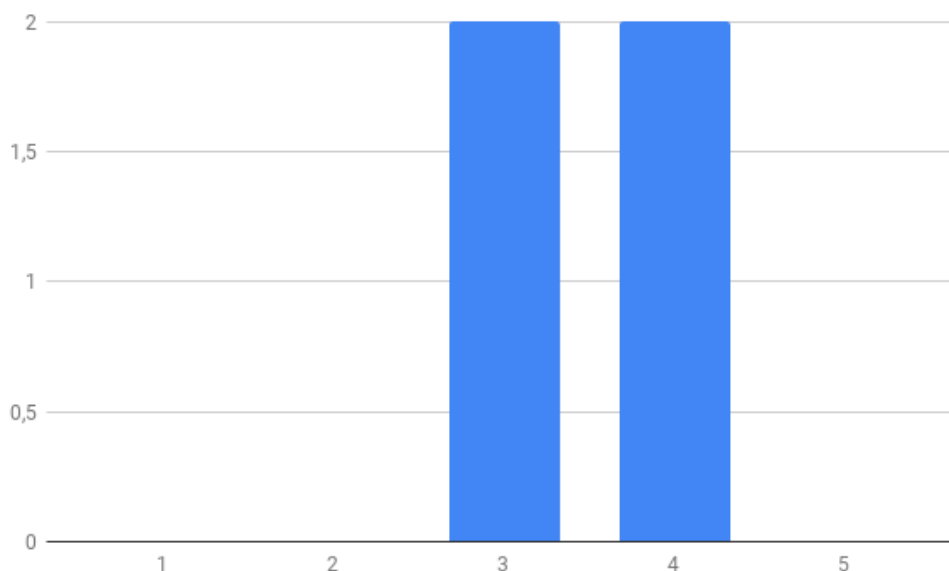


Figura 12: Gráfico de avaliação da playlist gerada para o grupo 1 (notas de 1 a 5, do menor para o maior)

O gráfico que representa a avaliação da playlist gerada, visto na figura 11, foi

retirado diretamente das respostas do grupo 1 para a pergunta número 5 do formulário (Anexo A). De uma forma geral, o resultado pode ser considerado positivo, validando o raciocínio anterior.

- Grupo 2: Grupo de dança (5 usuários)

O grupo 2 é definido por um grupo de dança de salão contendo cinco participantes na faixa etária de 25 a 35 anos.

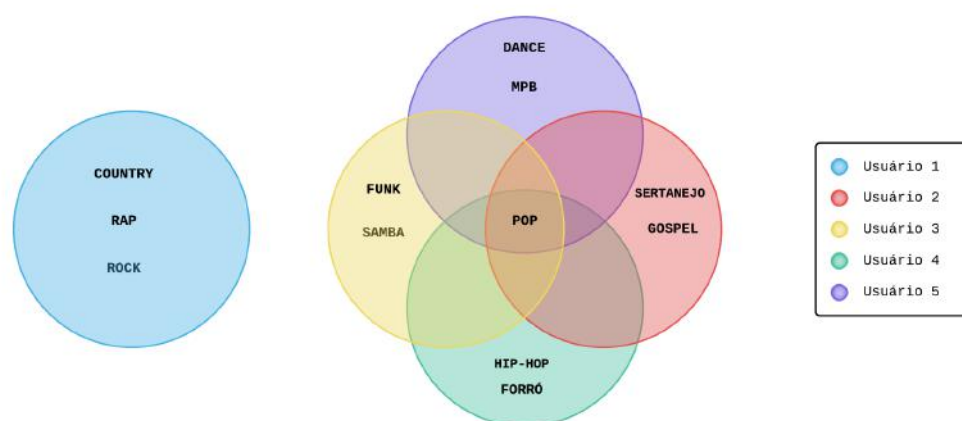


Figura 13: Diagrama de interseção de gêneros grupo 2

A figura 12 mostra a interseção de gêneros do grupo 2, com notas de 1 a 5, do menor para o maior. É possível observar que apenas um gênero aparece em comum com a maioria dos usuários da sala, sendo que um usuário não possui interseção com nenhum dos outros usuários.

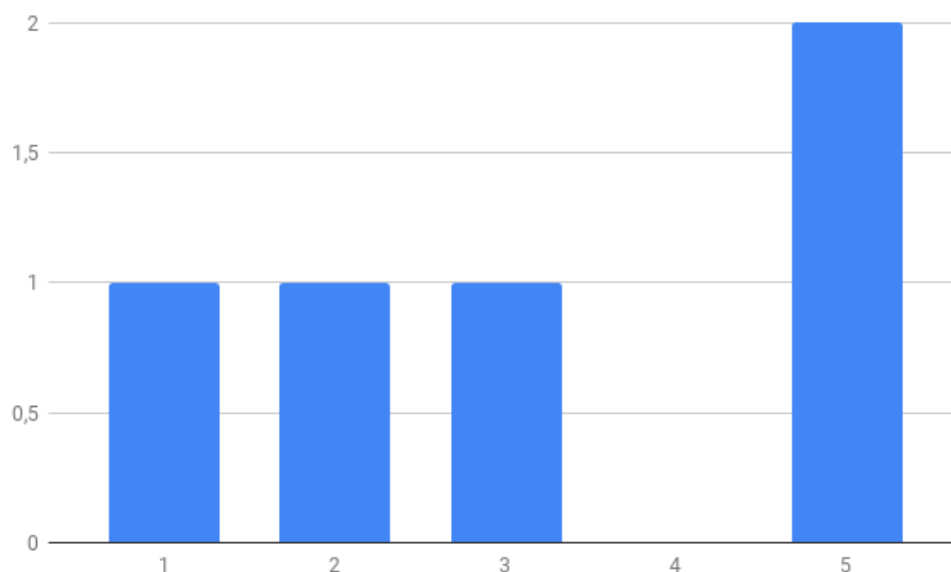


Figura 14: Gráfico de avaliação da playlist gerada para o grupo 2

Como pode ser visto na figura 13, a avaliação geral da playlist não foi boa, dado que o número de interseções foi mínimo, acarretando em uma lista de músicas genericamente mais diversas.

Analisando todos os resultados de forma conjunta, pôde-se observar que a base de dados escolhida (50 últimas músicas escutadas) não necessariamente representa os gêneros favoritos dos usuários, o que ocasionou em algumas discrepâncias nos gráficos individuais e do grupo. Além disso, a variedade de músicas desconhecidas parece ter sido um fator que influenciou bastante nas avaliações dos usuários, tanto positivamente quanto negativamente. Outra situação analisada foi o relato de gêneros nunca escutados presentes nos gráficos individuais. Após uma verificação nos dados buscando uma explicação plausível para tal evento, percebeu-se que existem gêneros diferentes que possuem o mesmo nome, o que pode gerar erros na recomendação final, e também artistas que trabalham com vários gêneros musicais.

## 6 CONCLUSÃO

### 6.1 DIFICULDADES ENCONTRADAS E APRENDIZADO

Foi possível contornar o problema do *Cold Start* e partir dele para o funcionamento padrão de coleta de dados do sistema. A solução foi usar o recurso do Spotify de coletar as últimas 50 músicas escutadas pelo usuário, dando assim uma boa margem para iniciar as recomendações, e sempre que for feito o login, serão coletadas novamente as últimas 50 músicas escutadas, sem repetição. Mas como o sistema ainda é novo, os resultados obtidos com os usuários terão um impacto diferente a longo prazo.

Durante o desenvolvimento do trabalho, houve diversas adversidades que complicaram a evolução do mesmo. Uma delas foi a limitação da API do Spotify em relação à quantidade de dados que podem ser coletadas por vez. Essa restrição dificultou no recolhimento de uma quantidade de dados mais volumosa para garantir uma recomendação ainda mais apurada.

Outra situação notada foi a falta de informações completas dos dados coletados. Determinadas músicas não apresentam os detalhes sobre o gênero, característica principal utilizada para a recomendação, sendo então necessário buscar pelo gênero do artista. Os artistas possuem gêneros demais que não podem ser utilizados na busca de músicas. E a busca de músicas do Spotify acabou sendo pouco precisa, ainda mais com gêneros diferentes e de mesmo nome, por exemplo: Funk. Para garantir uma confiança melhor dos dados, seria necessário realizar um refinamento nos mesmos, fato que será citado na próxima seção.

### 6.2 TRABALHOS FUTUROS

Como todo trabalho, projeto, produto ou serviço, nada é 100% perfeito, e sempre carecendo de melhorias e ajustes para aprimorar a qualidade do que é oferecido.

Graças ao aprendizado, aos resultados obtidos e às pesquisas de avaliação fei-



tas, pôde-se definir algumas questões que foram identificadas e julgadas necessárias aprimorar para se obter uma recomendação em grupo mais eficiente e precisa.

### 6.2.1 Tratamento de Dados

Abaixo é listada uma série de aprimoramentos identificados para o núcleo do sistema de recomendação, destinados a tratar e analisar os dados com mais precisão:

- Analisar melhor os gêneros das músicas e fazer uma interpretação desses dados para traçar um perfil mais correto de cada usuário. De preferência implementando algum algoritmo de análise de ondas sonoras para avaliação das músicas, com precisão, de forma a sugerir a qual gênero pertence. Tal estratégia se faz necessária pois o Spotify não classifica o gênero de cada música;
- Utilizar os metadados das músicas (acousticness, danceability, energy, instrumentalness, liveness, loudness, speechiness, valence e etc.), disponibilizado pelo Spotify[20], e ponderar eles também, de alguma forma a ser estudada, na recomendação. Isso seria bom para resolver o problema da diferenciação de gosto por músicas calmas e rápidas de um mesmo gênero, músicas instrumentais e etc;
- Fazer uma análise de desgosto musical de cada pessoa, para ponderar negativamente sobre o ranqueamento dos gêneros musicais na recomendação individual e em grupo.

### 6.2.2 Melhorias de Funcionalidades

Segue abaixo alguns itens de melhorias mapeadas para as funcionalidades do sistema:

- Permitir excluir um participante por vez;
- Permitir a criação de mais de uma sala por vez;

- Colocar as cores exibidas nos gráficos do lado dos gêneros na legenda;
- Gerar o gráfico com cores suaves e únicas, ou seja, sem repetir cores semelhantes;
- Enxugar os gêneros com pouco valor nos cálculos, talvez pegar os dez primeiros gêneros que são exibidos nos gráficos;
- O "dono da sala" poder bloquear algum gênero musical.

## REFERÊNCIAS

- [1] Sobre o Spotify. <https://pt.wikipedia.org/wiki/Spotify>.
- [2] Sistemas de Recomendação. <https://towardsdatascience.com/learning-how-recommendation-system-recommends-45ad8a941a5a>.
- [3] Node.js. <https://nodejs.org/en/>.
- [4] Express.js. <https://expressjs.com/pt-br/>.
- [5] Documentação da API do Spotify. <https://developer.spotify.com/documentation/web-api/>.
- [6] Google Cloud Firestore. <https://firebase.google.com/docs/firestore/?hl=pt-br>.
- [7] HTML. <https://www.w3schools.com/html/>.
- [8] CSS. <https://www.w3schools.com/css/>.
- [9] Javascript. <https://www.javascript.com/>.
- [10] jQuery. <https://jquery.com/>.
- [11] Motor V8. <https://v8.dev/>.
- [12] Google Chrome. <https://www.google.com/chrome/>.
- [13] Heroku. <https://www.heroku.com/>.
- [14] Spotify Authorization Guide. <https://developer.spotify.com/documentation/general/guides/authorization-guide/>.
- [15] Spotify Get Current User's Recently Played Tracks. <https://developer.spotify.com/documentation/web-api/reference/player/get-recently-played/>.
- [16] Spotify Get Several Artists. <https://developer.spotify.com/documentation/web-api/reference/artists/get-several-artists/>.

- [17] Spotify Get Current User's Profile. <https://developer.spotify.com/documentation/web-api/reference/users-profile/get-current-users-profile/>.
- [18] Spotify Get Available Genre Seeds. <https://developer.spotify.com/console/get-available-genre-seeds/>.
- [19] Socket.IO. <https://socket.io/>.
- [20] Spotify Get Recommendations Based on Seeds. <https://developer.spotify.com/documentation/web-api/reference/browse/get-recommendations/>.
- [21] Spotify Create a Playlist. <https://developer.spotify.com/documentation/web-api/reference/playlists/create-playlist/>.
- [22] Spotify Add Tracks to a Playlist. <https://developer.spotify.com/documentation/web-api/reference/playlists/add-tracks-to-playlist/>.
- [23] DENNIS L. CHAO, J. B., E FORREST, S. Adaptive radio: achieving consensus using negative preferences. In Proceedings of the 2005 international ACM.
- [24] MCCARTHY, J. F., E 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 - CSCW '98, pages 363–372, New York, New York, USA, November 1998. ACM Press.

## ANEXOS

### ANEXO A – Formulário de Pesquisa de Avaliação do Sistema

O anexo a seguir refere-se ao formulário de pesquisa e avaliação do sistema, distribuída para os usuários de grupos selecionados.

# Smartbox - Trabalho de Conclusão de Curso UFRJ

O formulário abaixo tem como objetivo validar o sistema colaborativo de recomendação de músicas. Responda o formulário somente se você já utilizou o sistema em questão. Obrigado!

\* Required

1. Informe de qual grupo você fez parte.  
(somente o número) \*

Caso não saiba de qual grupo faz parte, peça ajuda aos desenvolvedores.

---

2. Indique os três principais gêneros de música que você costuma escutar no Spotify. \*

---

## Recomendação de músicas

A seção abaixo deve ser respondida após escutar a playlist recomendada pelo sistema para seu grupo.

3. Se fosse recomendada somente para você, como seria a sua avaliação da playlist? \*

Mark only one oval.

1      2      3      4      5

---

---

4. Indique os principais fatores que te levaram a essa nota. \*

---

---

---

---

---

5. Sendo uma recomendação voltada para o ambiente do seu grupo em questão (trabalho, academia, festa, social, etc), como seria a sua avaliação da playlist? \*

Mark only one oval.

1      2      3      4      5

---

---

6. Indique os principais fatores que te levaram a essa nota. \*

---

---

---

---

---

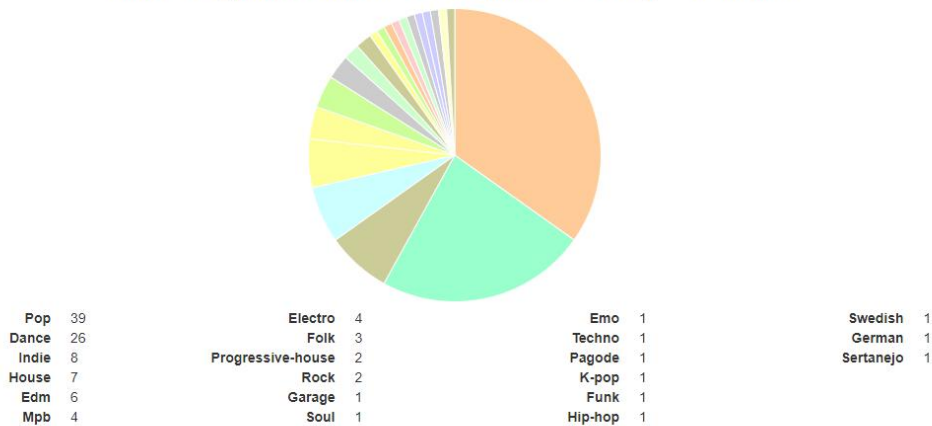
7. O gráfico abaixo é um exemplo da página inicial do sistema que apresenta uma avaliação por gênero baseada nas suas músicas escutadas no Spotify. O seu gráfico condiz com os seus gostos musicais de forma correta? (1- muito incorreta, 5 - muito correta) \*

Para responder essa pergunta é necessário analisar o gráfico que é exibido para você logo após o login no sistema (deve ser algo parecido com o da imagem).

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Lista de gêneros mais escutados e seus percentuais



8. Indique os principais fatores que te levaram a essa nota. \*

---

---

---

---

---

## Interface do sistema

Avaliação da usabilidade do sistema como todo, desde o momento de login até a geração da playlist.

9. Como você avaliaria a interface do sistema? (1 - muito ruim, 5 - muito bom) \*

Mark only one oval.

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

10. Indique algumas das dificuldades que possam ter ocorrido na utilização do sistema. \*

---

---

---

---

---



**ANEXO B** – Respostas do Formulário de Pesquisa de Avaliação do Sistema

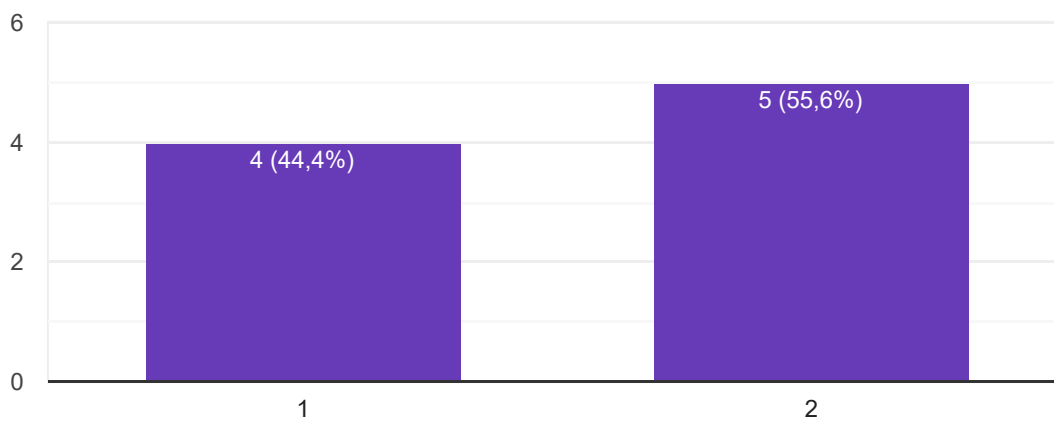
O anexo a seguir refere-se às respostas do formulário de pesquisa e avaliação do sistema indicado no anexo anterior.

# Smartbox - Trabalho de Conclusão de Curso UFRJ

9 respostas

Informe de qual grupo você fez parte. (somente o número)

9 respostas



Indique os três principais gêneros de música que você costuma escutar no Spotify.

9 respostas

Rock, hip hop e eletrônica

Sertanejo, funk e hip hop

Punk, Hardcore, Metal

Sertanejo, pagode e hip hop

Country, RAP, Rock

Sertanejo, pop e gospel

Pop, samba, funk

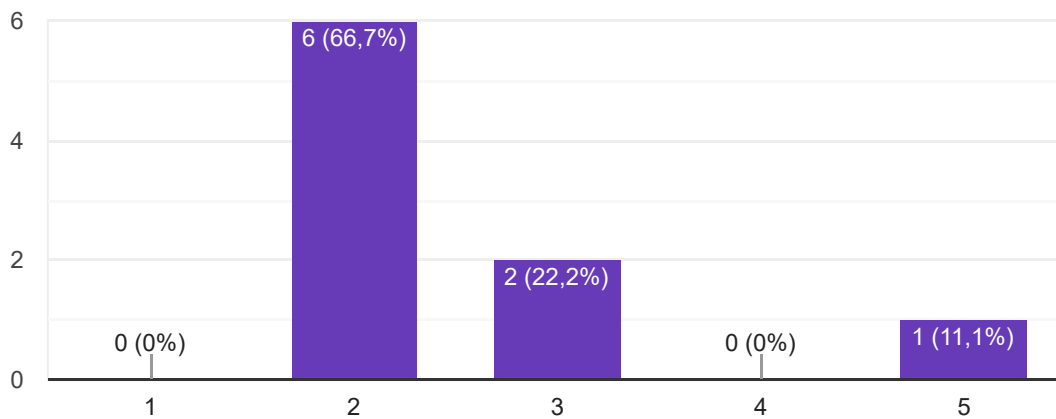
Forró, Hip Hop, Pop

Dance, pop e mpb

## Recomendação de músicas

Se fosse recomendada somente para você, como seria a sua avaliação da playlist?

9 respostas



Indique os principais fatores que te levaram a essa nota.

9 respostas

Gêneros de músicas que n costumo ouvir

A maioria das músicas e cantores eu não gosto

Playlist com poucas músicas limitando a variedade, menos de 10% das músicas seriam agradáveis ao meu gosto.

Não me agradou

Só conhecia os Novos Baianos da playlist. E das novidades, só uma ou duas me agradaram.

Existem algumas músicas desconhecidas e eu acabaria não escutando e pulando. Tenho costume de escutar apenas músicas conhecidas no spotify, quase não escuto playlists de recomendação.

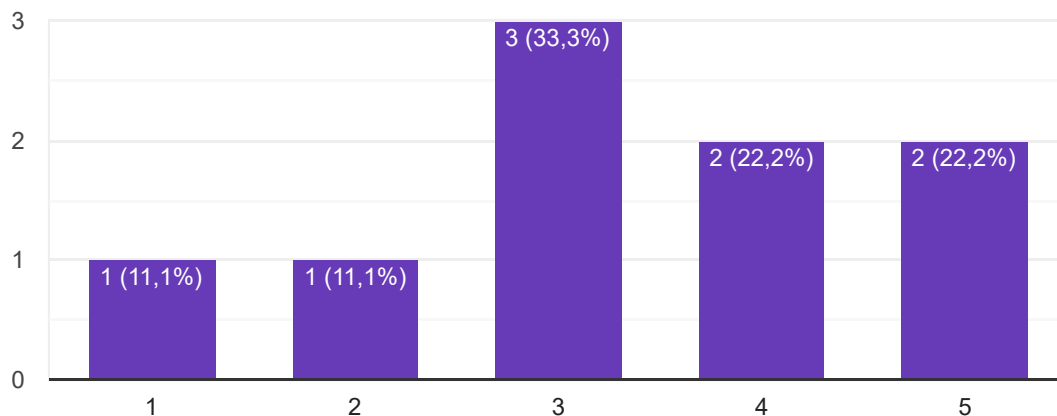
Gênero de várias das músicas não agradou, duração, repetitividade das letras.

Ele é aleatória demais.

Ótima seleção de músicas

Sendo uma recomendação voltada para o ambiente do seu grupo em questão (trabalho, academia, festa, social, etc), como seria a sua avaliação da playlist?

9 respostas



Indique os principais fatores que te levaram a essa nota.

9 respostas

Músicas muito calmas

O meu grupo gosta das músicas da playlist

Acredito que na visão geral do grupo seria de regular a boa visto o gosto musical compartilhado, no entanto outros se sentiriam menos representados na lista.

Meus amigos iriam gostar

As pessoas têm gostos bem variados. Acho que como a lista é bem variada, pode agradar a todos. Mas pelo mesmo motivo, agradaria por apenas uma ou duas músicas.

A playlist é bem diversa, provável que atenda aos gostos diferentes possíveis.

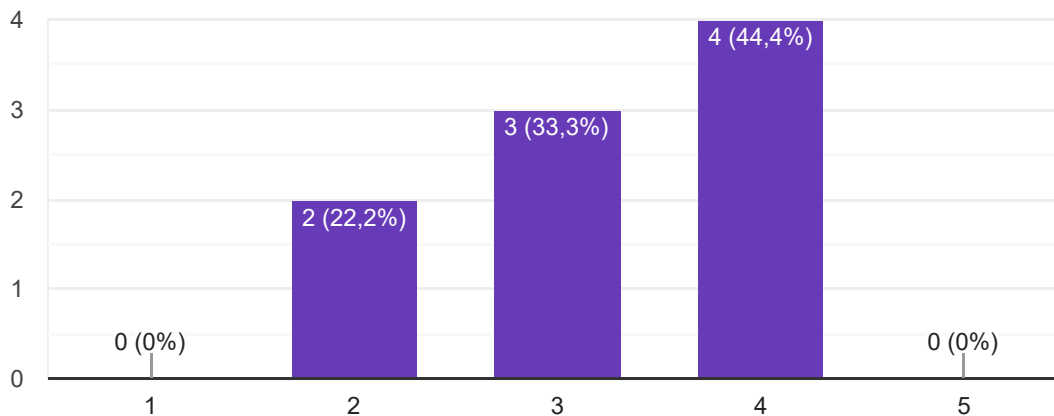
O gênero das músicas é bem diferente do que nós, em geral, costumamos ouvir

Creio que boa parte das músicas não agradariam os membros do meu grupo.

Músicas agradáveis para um grande grupo

O gráfico abaixo é um exemplo da página inicial do sistema que apresenta uma avaliação por gênero baseada nas suas músicas escutadas no Spotify. O seu gráfico condiz com os seus gostos musicais de forma correta? (1- muito incorreta, 5 - muito correta)

9 respostas



Indique os principais fatores que te levaram a essa nota.

9 respostas

Discrepância entre os dois generos favoritos: rock e pop e ausência do gênero hip hop

Tem alguns gêneros que eu não costumo ouvir

Apesar da maioria estar correta, poderia ser mais eficaz adicionando "subgêneros" como por exemplo do punk - "Celtic punk", "anarcopunk" e etc.

Sim, gosto de músicas muito variadas

Acertou em alguns temas, mas errou categoricamente em outros. Funk, por exemplo, está com 17% e eu nunca nem ouvi.

Sertanejo representaria a maior parte, assim como pagode e outros gêneros que nem estão representados (gospel).

O grande percentual relacionado à música pop condiz com a realidade, mas ouço muito alguns outros gêneros que estão com percentuais muito baixos.

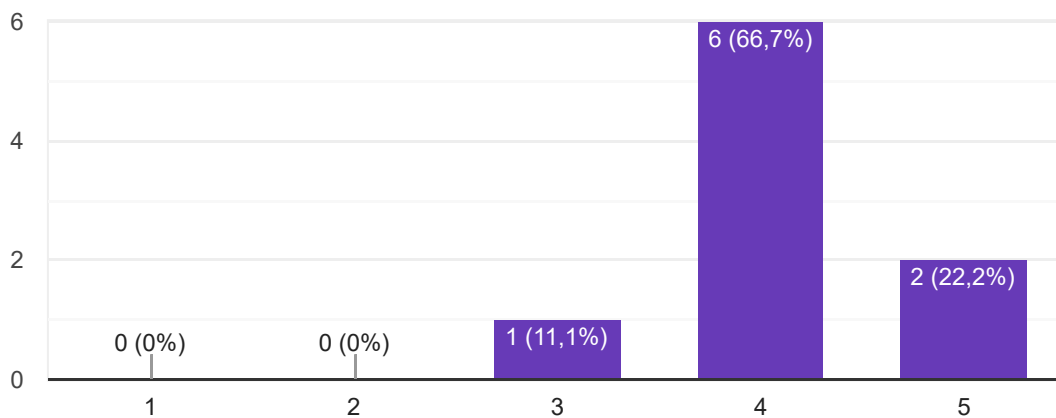
Eu usualmente uso o Spotify pra escutar recomendações, então não necessariamente as músicas que escutei condizem com o meu gosto.

Eu colocaria o mpb em terceiro lugar

## Interface do sistema

Como você avaliaria a interface do sistema? (1 - muito ruim, 5 - muito bom)

9 respostas



Indique algumas das dificuldades que possam ter ocorrido na utilização do sistema.

9 respostas

Única dificuldade foi o erro no login

Não houve dificuldades

Sem dificuldade

Nada foi notado

Eu só não soube onde olha, a princípio.

O sistema foi inicialmente capaz de classificar muito bem o meu padrão de uso, porém a playlist gerada não reflete tanto assim os meus gostos.

Nenhuma

Não tenho nenhuma dificuldade em mente.

Não tive nenhuma dificuldade, muito pratico de utilizar! Ótimo!

Este conteúdo não foi criado nem aprovado pelo Google. Denunciar abuso - Termos de Serviço

Google Formulários