

Colégio Técnico de Campinas  
Universidade Estadual de Campinas

## **Relatório de Pesquisa**

### **MELODIUM: um sistema de recomendação musical alternativo**

Luiza Viana Souza  
Pedro dos Santos Sakai  
Júlia Lopes de Campos  
Orientador Professora Andréia Cristina de Souza

Campinas, 2022



## SUMÁRIO

<b>1. Resumo</b>	<b>3</b>
<b>2. Introdução</b>	<b>3</b>
<b>3. Problema</b>	<b>4</b>
<b>4. Objetivos</b>	<b>4</b>
4.1. Objetivo Geral	4
4.2. Objetivos Específicos	4
<b>5. Justificativa</b>	<b>5</b>
<b>6. Hipótese</b>	<b>5</b>
<b>7. Metodologia e Materiais</b>	<b>5</b>
7.1. Metodologia	5
7.2. Materiais	6
<b>8. Desenvolvimento</b>	<b>8</b>
8.1. Etapas do Desenvolvimento	9
8.2. Pré-Desenvolvimento	9
8.2.1. Levantamento de Dados	9
8.2.2. Protótipo de Telas	9
8.3. Desenvolvimento	12
8.3.1. Sprints	12
8.4. Pós-Desenvolvimento	13
<b>9. Conclusão</b>	<b>13</b>
<b>10. Referências Bibliográficas</b>	<b>14</b>

## 1. Resumo

Antes da popularização do uso de plataformas digitais para ouvir e descobrir novas músicas, álbuns e artistas, a população apenas tinha contato com a arte sonora a partir de rádios ou TV. No início do século XXI, com a popularização do uso de plataformas online para ouvir e descobriu-se a possibilidade de o próprio usuário decidir aquilo que ele quer ouvir no momento, abrindo um leque de diferentes possibilidades para a inserção de novos artistas no mercado e usuários com mais interesse naquilo que eles estavam ouvindo.

## 2. Introdução

Antes da popularização da música e da democratização do acesso à esse tipo de arte, a população apenas ouvia e descobria novas canções através de rádios, televisão e outros meios de comunicação analógicos, que, visando um maior lucro, apenas mostravam artistas e bandas que já estavam no mercado e agradavam as grandes massas. Por isso, no caso de alguém desenvolver um interesse maior em ouvir algo mais personalizado, era necessário comprar discos, fitas cassetes e CDs, que não eram acessíveis a todos, às vezes incentivando até a pirataria e o ferimento dos direitos autorais. Entretanto, durante os anos 2000, com o surgimento de plataformas de *streaming*<sup>1</sup>, os ouvintes passaram a consumir música de uma maneira muito diferente. Foram descobertas novas maneiras de escutar novas músicas, álbuns e artistas e surgiu a possibilidade de o próprio usuário a criar *playlists*<sup>2</sup> e decidir aquilo que ele quer ouvir no momento, abrindo um leque de diferentes possibilidades para a inserção de novos artistas no mercado e consumidores com mais interesse naquilo que eles estavam escutando.

Com essa grande carga de conteúdo hoje em plataformas digitais como o Spotify, YouTube, Deezer, entre outros, há uma demanda muito grande para o que os usuários deveriam consumir em seguida. Contraditoriamente, surge o problema de que há tanto conteúdo na internet, que é difícil para as plataformas recomendarem algo que o usuário vai gostar de escutar em seguida. Para tapar esse buraco e fazer com que os usuários continuem ouvindo, a maioria das empresas passaram a recomendar músicas e artistas a partir do quesito social, ou seja, eles deduzem que, se grande parte dos ouvintes que escutam uma banda X, também escutam a cantora B, é provável que esse novo usuário que gosta bastante de um álbum da cantora B, goste da banda X.

Um dos primeiros algoritmos desenvolvidos para a recomendação musical, foi o “Ringo: Um sistema de recomendação musical personalizado”, desenvolvido por uma dupla de pesquisadores do MIT, em 1995. Onde o algoritmo desenvolvia um perfil musical para cada usuário da plataforma e testava sua compatibilidade com

---

<sup>1</sup>Um estilo de difusão de dados, geralmente transmitido em rede através de pacotes. Um fluxo de mídia ou transmissão contínua.

<sup>2</sup> Uma lista de músicas, canções e podcasts criada para ser ouvida continuamente.

outros usuários. Essa similaridade era testada por um sistema de notas que cada cliente dava para as músicas que ele escutava dentro da plataforma, ou seja, quanto mais conteúdo escutado, mais o perfil era adequado para recomendação. O algoritmo foi ensinado que usuários com perfis parecidos, tendem a ter o mesmo gosto musical. Então, para estes eram recomendadas as músicas com as maiores notas dadas por perfis parecidos. (SHARDANAD, U.; MAES, P., 1995)

Quase 30 anos depois, essa funcionalidade persiste em grande parte dos serviços de *streaming*. Entretanto, é um dos maiores motivos de insatisfação dos usuários, trazendo uma grande insuficiência para esse mercado.

É tentando suprir esse déficit, que a Melodium surgiu. Uma solução usando a tecnologia de Inteligência Artificial para recomendar novas músicas e artistas a partir dos aspectos musicais, e não sociais. O algoritmo da Melodium analisa uma música inserida pelo usuário e deduz características importantes para a experiência da música, como BPM (Batidas por Minuto), compasso, tom, campo harmônico e até os instrumentos musicais contidos. Através dos dados analisados, a aplicação *web* é capaz de recomendar novas músicas de uma maneira mais eficiente e trazendo exatamente o que o usuário quer ouvir em seguida.

### **3. Problema**

Como aumentar a experiência do usuário ao ouvir uma música recomendada por uma plataforma de *streaming* e realmente fazê-lo escutar algo conveniente no momento?

### **4. Objetivos**

#### **4.1. Objetivo Geral**

Desenvolver e tornar acessível uma plataforma de recomendação musical independente, que permite que usuários de aplicações digitais como o Spotify, Deezer, Apple Music ou Amazon Music, pesquisarem e selecionarem uma música que gostam ou que estejam ouvindo no momento e lhes é devolvida a recomendação de uma outra para ser escutada em seguida, com aspectos musicais parecidos que fazem sentido para o gosto do usuário.

#### **4.2. Objetivos Específicos**

- Entender o contexto da situação das plataformas de *streaming* e o sistema de recomendação musical;
- Estudar o desenvolvimento de software e ver como adequar-se da melhor maneira para que a aplicação seja acessível a todos os interessados;

- Inserir novas soluções para a recomendação de músicas no mercado de plataformas digitais que sejam benéficas aos usuários e também às empresas que criaram essas tecnologias;
- Aprimorar nossas habilidades como programadores e resolvedores de problemas tecnológicos;

## 5. Justificativa

Na indústria musical atual, focada na divulgação e consumo dentro de plataformas digitais, há uma grande lacuna a respeito dos algoritmos de recomendação das próximas músicas para o usuário. É preenchendo esse espaço tão necessário em um setor bilionário em crescimento acelerado que nasce nosso propósito: criar uma inteligência que seja capaz de desenvolver a experiência do usuário ao utilizar novas tecnologias para o consumo de música.

Além disso, o ramo tecnológico de softwares tem grande demanda por Inteligências Artificiais, cada vez mais estudadas e aprimoradas. Fazer um projeto que utiliza duas indústrias criativas, atuais e em crescimento demonstra a necessidade atual da Melodium e como as oportunidades à frente do projeto podem ser aproveitadas.

## 6. Hipótese

Espera-se que a Melodium seja capaz de suprir as necessidades atuais do mercado de plataformas digitais para o consumo de meios musicais e que traga uma experiência do usuário cada vez mais aprimorada.

## 7. Metodologia e Materiais

### 7.1. Metodologia

Durante o desenvolvimento do projeto, foram utilizadas tecnologias de gerenciamento e frameworks como o Scrum e outras metodologias ágeis.

O framework Scrum, se baseia no planejamento e na divisão de tarefas em *sprints*<sup>3</sup>, de 2 meses, que são planos para a conclusão do *Product Backlog*<sup>4</sup>. Após cada *sprint*, é realizada uma reunião comentando tudo o que foi concluído ou não, que era planejado para aquela data. Além disso, também é debatida a produtividade da equipe e é dado um *feedback* para a melhoria das funcionalidades implementadas.

---

<sup>3</sup> Uma sprint é a divisão das atividades pré-definidas pelo Product Backlog, geralmente a divisão é feita por um prazo estabelecido para a conclusão de tal atividade.

<sup>4</sup> O Product Backlog é uma lista ordenada de todas as atividades necessárias para o desenvolvimento de um certo projeto.

A pesquisa está sendo realizada em grande parte no Colégio Técnico de Campinas, mas também nas residências dos autores do projeto usando plataformas de comunicação e compartilhamento de arquivos online.

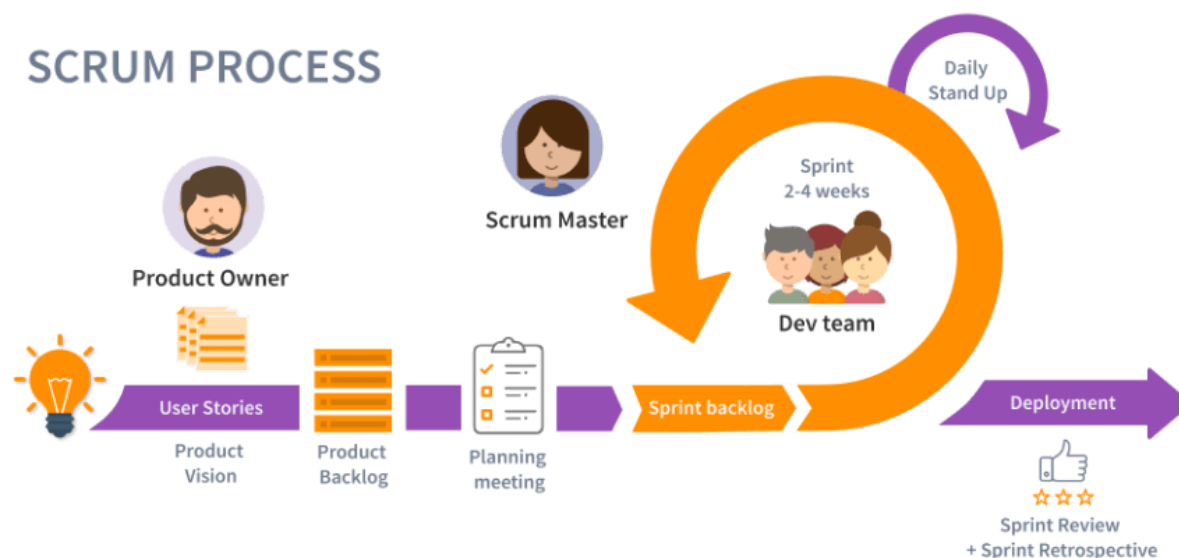


Imagem 1. Representação visual esquematizada do Processo Scrum.

Fonte: DEV Community. Disponível em: <<https://dev.to/stanley/entrega-agil-com-scrum-4bn9>>

Em nosso projeto, nossas atividades foram separadas nas seguintes sprints:

Numeração da sprint	Previsão do período de desenvolvimento
Sprint 1	De 03/03 à 30/04
Sprint 2	De 02/05 à 30/06
Sprint 3	De 03/08 à 01/10
Sprint 4	De 03/10 à 15/12

Tabela 1. Separação das sprints por período de desenvolvimento

## 7.2. Materiais

Para o desenvolvimento do projeto Melodium e suas features foram utilizadas diversas linguagens de programação e frameworks, como React.JS, API.NET, Node.JS, JavaScript, Python, API Web Scraper, Json, HTML, CSS, Next.JS.

Além de estudarmos e aprendermos através de vídeos, reportagens e artigos a respeito de aspectos musicais que deixam as estruturas musicais parecidas, descobrimos como acessar estes atributos usando bibliotecas como Puppeteer, Axios e Spotify for Developers, que mesmo com algumas incompatibilidades com outras funcionalidades foram recompensadoras durante o processo de aprendizado do grupo.

### 7.2.1. Materiais Pré-Desenvolvimento

Para o planejamento do design do projeto, que estará em contato direto com o usuário, era necessário o conhecimento básico de recursos como o UX/UI Design<sup>5</sup>, que foram estudados e seus conceitos aplicados através de plataformas online como Figma<sup>6</sup>, Paleta de Cores<sup>7</sup> e NicePage<sup>8</sup>.

As funcionalidades planejadas durante o primeiro semestre do projeto foram as seguintes:

- Um front-end React JS, utilizando JavaScript para a interação com o usuário;
- Um back-end desenvolvido em Python utilizando bibliotecas preparadas para a análise musical, estudando e aplicando os conceitos de Inteligência Artificial aprendidos;
- Uma API Restful<sup>9</sup> utilizando ASP.NET Core para a integração de ambas as peças.

### 7.2.1. Materiais para o Desenvolvimento

Durante o desenvolvimento do projeto, houveram alguns obstáculos de incompatibilidade entre as partes, e tiveram que ser feitas mudanças para a finalização do projeto.

Mesmo sem a presença das seguintes tecnologias na versão final do projeto, elas foram estudadas e aplicadas no projeto na tentativa de reestruturar o Melodium e superar esses obstáculos de incompatibilidade durante o caminho:

---

<sup>5</sup> UX/UI é um design de software utilizado para aplicativos altamente interativos. Esse tipo de design é focado na experiência do usuário, combinando design visual com design interativo, para fornecer layouts e cenários específicos, demonstrando como o aplicativo interage com o usuário.

<sup>6</sup> [www.figma.com](http://www.figma.com)

<sup>7</sup> [www.paletadecores.com](http://www.paletadecores.com)

<sup>8</sup> [www.nicepage.com](http://www.nicepage.com)

<sup>9</sup> Uma Interface de Programação de Aplicações que utiliza um estilo de arquitetura de software que define um conjunto de restrições a serem usadas para a criação de serviços web, com o objetivo de integrar diferentes partes de um programa.

- |               |                |              |
|---------------|----------------|--------------|
| • React JS    | • JSON         | • HTML       |
| • Node JS     | • HTTP         | • CSS        |
| • Next JS     | • Python       | • TSX        |
| • Spotify API | • Puppeteer    | • Angular JS |
| • JavaScript  | • API.NET Core | • GitHub     |
| • TypeScript  | • C#           | • TypiCode   |

Ao final da estruturação do projeto, as tecnologias que continuaram presentes em sua última versão foram:

- |               |              |       |
|---------------|--------------|-------|
| • React JS    | • JavaScript | • CSS |
| • Node JS     | • JSON       |       |
| • Spotify API | • HTML       |       |

## 8. Desenvolvimento

### 8.1. Etapas de desenvolvimento

Com intenção de organizar o desenvolvimento do projeto, ele foi separado nas seguintes fases, cada uma com suas subfases:

- Pré-desenvolvimento:
  - Levantamento de dados para a estruturação da pesquisa científica;
  - Desenvolvimento do protótipo de telas do projeto, utilizando o Figma<sup>10</sup>;
- Desenvolvimento:
  - Sprint 1:
    - Formalização da proposta do projeto;
    - Leitura dos guias Decola Beta
    - Criação do Diário de Bordo;
    - Curso Ápice;
    - Plano de Pesquisa;
  - Sprint 2:
    - Pitch;
    - Protótipo do Front-end;
    - Monografia preliminar;
    - Pesquisa bibliográfica preliminar;

---

<sup>10</sup> Figma é um editor gráfico de prototipagem, muito utilizado por desenvolvedores de software, para o desenvolvimento de telas.



- Sprint 3:
  - Desenvolvimento do servidor para conexão de texto;
- Sprint 4:
  - Desenvolvimento do servidor para a conexão de voz;

## 8.2. Pré-Desenvolvimento

### 8.2.1. Levantamento de Dados

Ao começo da elaboração do projeto, foi feito um levantamento pelos autores para coletar dados sobre a satisfação dos consumidores das plataformas de streaming em relação às recomendações oferecidas pelas plataformas de novas músicas. Durante a pesquisa, aproximadamente 72% dos alunos relataram achar insatisfatórias as recomendações oferecidas pelos serviços de streaming, destacando como principal motivo a falta de similaridades entre as músicas.

Durante a mesma pesquisa, foi perguntado se os alunos usavam frequentemente as funções de recomendação de músicas nos serviços, e enquanto 34% disse que já fez uso das ferramentas de recomendação mais de uma vez, 66% disseram que usaram apenas uma vez ou que não se lembram de ter usado.

Ao final dessa pesquisa, foi concluído que para o melhor proveito da ferramenta de recomendação é necessário que essa se torne mais assertiva e de fato faça o que se propõe a fazer.

### 8.2.2. Protótipo de Telas

Com todos os diagramas prontos, é possível desenhar os primeiros protótipos do projeto, pensando em como seria a paleta de cores selecionada, o logo do aplicativo e, principalmente, como será a introdução do usuário com a interface.

A paleta de cores selecionada foi escolhida a dedo, com cores que chamam a atenção, mas que não são muito fortes a ponto de o usuário ficar cansado delas. A seleção de cores para um aplicativo possui grande importância para chamar a atenção do usuário e mantê-lo focado na plataforma.



Figura 1. Paleta de cores do aplicativo. Fonte: Autoria própria.

Usando a paleta de cores escolhida, foi criado o logotipo do Melodium. A logo foi desenvolvida com a intenção de remeter a um pensamento ou ideia gerado a partir de uma música escolhida, já que o objetivo do site é encontrar novas músicas semelhantes ou parecidas a partir da escolhida.



Figura 2. Logotipo do aplicativo. Fonte: Autoria própria.

O protótipo de telas também foi realizado utilizando a logo e a paleta de cores anteriormente citadas. Esses dois processos anteriores foram extremamente úteis para que o desenvolvimento da interface do usuário fosse desenvolvido sem conflitos, deixando espaço para criatividade.

O esquema de telas foi desenvolvido utilizando designs UX/UI<sup>11</sup>, utilizando o Figma. Foram desenvolvidas as seguintes telas:

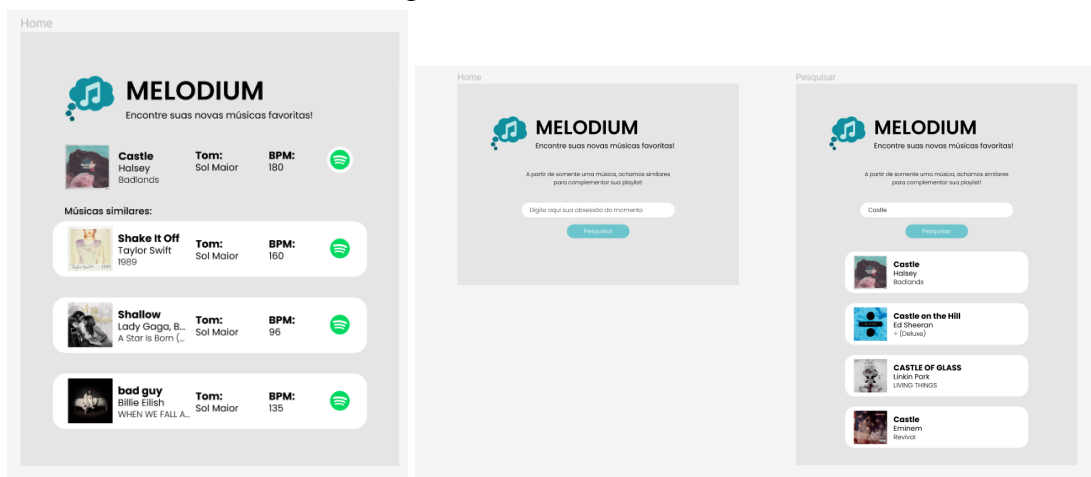


Figura 3. Protótipo de Telas. Fonte: Autoria própria

Entre as telas, podemos destacar:

- Página de pesquisa (inicial);
- Sugestões encontradas a partir da pesquisa;

<sup>11</sup> UX/UI é um design de software utilizado para aplicativos altamente interativos. Esse tipo de design é focado na experiência do usuário, combinando design visual com design interativo, para fornecer layouts e cenários específicos, demonstrando como o aplicativo interage com o usuário.

- Página com seleções de músicas encontradas a partir da música escolhida.

### 8.3. Desenvolvimento

Para o desenvolvimento do Melodium, separamos todas nossas atividades em um Product Backlog. Utilizando os princípios do framework Scrum, o aplicativo se encontra em desenvolvimento, sempre incrementando o software a partir de ciclos de feedback.

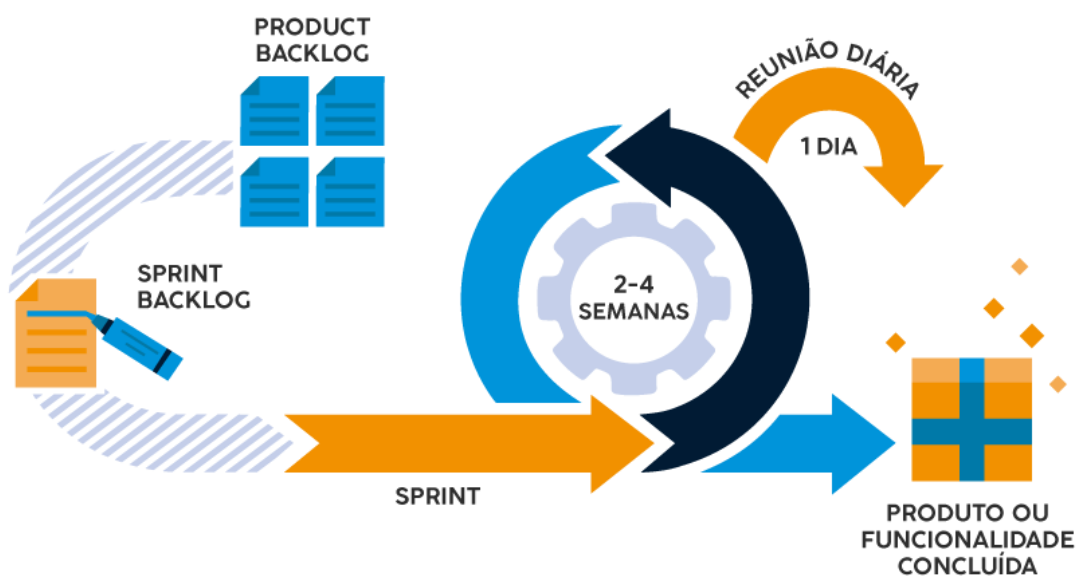


Figura 6. Representação visual dos principais artefatos do Scrum Framework.

Fonte: Tecnicon. Disponível em:  
<<https://www.tecnicon.com.br/upload/public/Blog/post-scrum.png>>

O tempo estimado para a conclusão de cada sprint tenha sido de 14 a 21 dias, as sprints do desenvolvimento Scrum são flexíveis, ou seja, elas podem durar mais ou menos, dependendo da necessidade de cada atividade.

### 8.3.1. Sprints

Como já mencionado, todas as atividades foram separadas em 4 sprints, períodos de desenvolvimento da aplicação, geralmente com duração de 30 a 60 dias, dependendo do número de tarefas e do tempo estimado para a execução das mesmas.

De acordo com as atividades anteriormente separadas, era evidente que algumas sprints demorariam mais tempo para serem concluídas que outras, logo foram dados prazos adequados para o fechamento como o esperado.

Como existem tarefas que deveriam ser executadas antes e que são necessárias para realizar as outras funcionalidades do programa, as quatro sprints foram enumeradas dependendo de quais atividades deveriam ser começadas primeiro.

Abaixo podemos observar a numeração das sprints e a previsão de seu período de desenvolvimento.

Tabela 1. Separação das sprints por período de desenvolvimento

<b>Numeração da sprint</b>	<b>Previsão do período de desenvolvimento</b>
Sprint 1	De 03/03 à 30/04
Sprint 2	De 02/05 à 30/06
Sprint 3	De 03/08 à 19/10
Sprint 4	De 20/10 à 22/11

OBS: Apenas estão sinalizadas as coisas que deram certo e estavam prontas no momento da reunião!

- Sprint 1:
  - Ideias e desenvolvimento inicial, apenas planejamento e documentação.
- Sprint 2:
  - Criação dos protótipos de tela.
- Sprint 3:

- Telas feitas em projeto react com css, server javascript com webscraping.
- Sprint 4:
  - Funções com a Spotify API, integração Front-Back, algoritmo de recomendação e projeto finalizado.

## 8.4. Pós-Desenvolvimento

Após o desenvolvimento completo do Melodium, planeja-se introduzir o site para o uso público a fim de testar a eficiência do projeto em si e ver se os resultados serão os previstos. Pretende-se implementar uma pesquisa de opinião com os alunos de nossa sala para que seja possível compreender se ainda há algo a ser implementado ou corrigido.

A intenção da pesquisa é determinar o nível de sucesso da solução do problema apresentado. Fazendo uma avaliação técnica e sempre recebendo feedbacks, planeja-se adaptar o projeto às necessidades reais dos usuários e fazê-lo uma ferramenta cada vez mais útil.

## 9. Conclusão

A partir do levantamento de dados e análise dos serviços de streaming em relação a recomendações feitas a partir de músicas já tocadas, concluímos que o desenvolvimento de um site que analisa uma música escolhida por seus quesitos musicais como bpm e tom pode ser uma alternativa eficiente para a reversão desse déficit.

A pesquisa ainda se encontra em fase de desenvolvimento e pretendemos ainda continuar implementando ela a partir dos resultados coletados. Entretanto, até a presente data, pode-se concluir que é necessária uma ferramenta assertiva, eficaz e eficiente que ajude as pessoas a encontrar com maior facilidade músicas que agradem seus gostos pessoais.

Como esta pesquisa se encontra em andamento, os resultados apresentados são parciais e, portanto, não podem concluir os benefícios que o Melodium possa vir a trazer. Este projeto está sendo desenvolvido há oito meses e é esperado que ele venha a ser implementado corriqueiramente durante o ano seguinte, tendo um período total de desenvolvimento de aproximadamente dois anos.

Nesse meio tempo, pretende-se finalizar as sprints, 2, 3 e 4. Quando o aplicativo receberá suas principais funcionalidades, assim como o chat de voz e de texto. Dessa forma, ao final de sua codificação, iniciaremos uma pesquisa de satisfação, para determinar o grau de sucesso da ferramenta. Dessa maneira, é .

## 10. Referências Bibliográficas

EDOπ : Uma biblioteca em Python para música microtonal. \*UNIVERSIDADE FEDERAL FLUMINENSE\*, 2021. Disponível em: <<https://app.uff.br/riuff/bitstream/handle/1/22789/2%20TCC%20FLÁVIA%20E%20LUA%20N.pdf?sequence=1&isAllowed=y>>. Acesso em: 22 de Mar. 2022.

SOCIAL information filtering: Algorithms for automating "word of mouth". **SHARDANAD, U.; MAES, P.** Disponível em: <<https://dl.acm.org/doi/fullHtml/10.1145/223904.223931>>. Acesso em: 24 de Mai. 2022.

VISUALIZAÇÃO de tags para explicar e filtrar recomendações de músicas. **SATO, Juliana.** \*UNIVERSIDADE DE SÃO PAULO\*, 2013. Disponível em: <<https://www.teses.usp.br/teses/disponiveis/45/45134/tde-06062013-142529/publico/dissertacaoJuliana.pdf>>.

PESQUISA Nacional por Amostra de Domicílios Contínua. **IBGE**, 2019. Disponível em: <[https://biblioteca.ibge.gov.br/visualizacao/livros/liv101794\\_informativo.pdf](https://biblioteca.ibge.gov.br/visualizacao/livros/liv101794_informativo.pdf)>. Acesso em 10 de Out. 2021.