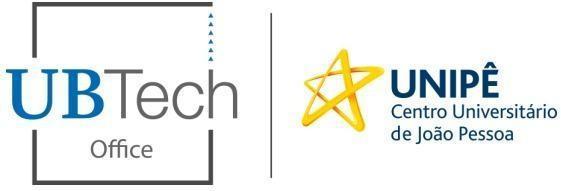
**Centro Universitário de João Pessoa - UNIPÊ Fábrica de Software do UBTech Office/UNIPÊ**

Documento de Visão de Sistema

# **OBJETIVO DESTE DOCUMENTO**

O objetivo deste documento é apresentar uma visão geral da plataforma **MusicStream, um aplicativo de streaming musical** que será desenvolvido. Serão abordados tópicos como escopo do produto, não escopo do produto, descrição dos envolvidos, visão geral do produto e restrições.

# **HISTÓRICO DE REVISÃO**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Histórico de Revisão*** | | | |
| **Data** | **Autor** | **Descrição** | **Versão** |
| 20/08 | Wellysson | Criação do documento | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Obs.: O redimensionamento das colunas das tabelas poderá ser alterado caso haja necessidade.

1. **ESCOPO DO PRODUTO**

A plataforma **MusicStream** terá como objetivo principal facilitar **os usuários ouvir músicas sob demanda, criar playlists, seguir artistas e compartilhar músicas com seus amigos**. O **aplicativo** será compatível com plataformas IOS, Android e Web.

O sistema deverá ser capaz de fornecer os seguintes serviços para atender as necessidades dos interessados:

**Perspectiva do Usuário (Ouvintes):**

* O aplicativo deve permitir o acessos a um catálogo extenso de músicas de diferentes gêneros.
* O aplicativo deve pesquisar por artistas, álbuns, músicas e playlists.
* O aplicativo deve ser capaz de criar, editar, e excluir playlists personalizadas.
* O aplicativo deve ser capaz de recomendar músicas para adicionar às playlists de forma automática.
* O aplicativo deve ser capaz de sugerir músicas baseadas no histórico de escuta do usuário.
* O aplicativo deve ser capaz de recomendar playlists e rádios personalizadas com base no gosto musical do usuário.
* O aplicativo deve ser capaz de seguir outros usuários e artistas.
* O aplicativo deve ser capaz de compartilhar músicas e playlists em redes sociais e com outros usuários do aplicativo.
* O sistema deve ser capaz de criar e gerenciar de perfis de usuário.
* O aplicativo deve ser capaz de buscar o histórico de músicas ouvidas e playlists criadas.
* O aplicativo deve ter funções de controle de reprodução (play, pause, avançar, retroceder).
* O aplicativo deve ser capaz de exibir as letras das músicas (shuffle, repetição)
* O aplicativo deve ser capaz de fazer download de músicas e playlits para ouvir offine.

# **NÃO ESCOPO DO PRODUTO**

O sistema **NÃO** deverá fornecer os serviços para atender as necessidades dos interessados:

* O aplicativo não deve suportar a reprodução de videoclipes ou vídeos relacionados a músicas.
* O aplicativo não deve incluir ferramentas de criação de música, como mixers ou editores de áudio.
* No aplicativo não deve haver suporte para streaming ou hospedagem de podcats.
* O aplicativo não deve permitir a compra ou download definitivo de músicas.
* No aplicativo não deve haver funcionalidades de controle por voz, como integração com assistentes virtuais, não serão implementadas.

# **DESCRIÇÃO DOS ENVOLVIDOS**

Os principais envolvidos na plataforma serão DESCREVER OS PRINCIPAIS USUÁRIOS.

# **Resumo dos Usuários**

|  |  |
| --- | --- |
| **Nome** | **Responsabilidades** |
| Gerente de Projeto | Responsável pela coordenação do projeto, gerenciamento de cronograma e comunicação com stakeholders. |
| Desenvolvedores | Encargados da codificação do aplicativo, incluindo backend e frontend. |
| Designer UX/UI | Responsável pela criação da interface do usuário e experiência de usuário. |
| Analista de QA | Realiza testes e garante a qualidade do aplicativo. |
| Especialista em Dados | Gerencia e analisa dados relacionados a preferências musicais e comportamento do usuário. |

# **VISÃO GERAL DO PRODUTO**

O aplicativo de streaming de música MusicStream permite aos usuários ouvir uma vasta biblioteca de músicas on-demand, criar playlists personalizadas, e descobrir novas faixas e artistas com base em suas preferências. O aplicativo estará disponível em plataformas móveis e web, oferecendo uma experiência integrada e intuitiva para amantes da música.

* 1. **Requisitos Funcionais**

|  |  |  |
| --- | --- | --- |
| **Nº** | **Nome** | **Descrição** |
| **RF001** | Biblioteca de Música | Permitir aos usuários acessar uma vasta gama de músicas e álbuns. |
| **RF002** | Playlists Personalizadas | Criar, editar e gerenciar playlists pessoais. |
| **RF003** | Reprodução de Música | Controles básicos de reprodução, como play, pause, skip, e repeat. |
| **RF004** | Descoberta e Recomendações | Sugestões de músicas e artistas com base no histórico e preferências do usuário. |
| **RF005** | Integração Social | Compartilhamento de músicas e playlists em redes sociais. |
| **RF006** | Multiplataforma | Suporte para acesso em dispositivos móveis e web. |
| **RF007** | Pesquisa de Música | Permitir aos usuários pesquisar músicas, álbuns, e artistas. |
| **RF008** | Histórico de Reprodução | Manter um registro das músicas reproduzidas recentemente. |
| **RF009** | Modo Offline | Opção para ouvir músicas armazenadas localmente (se aplicável). |
| **RF010** | Controle de Volume | Ajustar o volume de reprodução. |
| **RF011** | Equalizador de Áudio | Configurações de equalização personalizáveis (opcional). |

* 1. **Requisitos Não Funcionais**

|  |  |  |
| --- | --- | --- |
| **Nº** | **Nome** | **Descrição** |
| **RNF001** | Tempo de Resposta | O aplicativo deve responder a comandos do usuário em menos de 2 segundos. |
| **RNF002** | Escalabilidade | O sistema deve suportar um grande número de usuários simultâneos sem degradação significativa do desempenho. |
| **RNF003** | Proteção de Dados | Garantir que os dados pessoais e de pagamento dos usuários sejam protegidos e criptografados. |
| **RNF004** | Autenticação | Implementar métodos seguros de autenticação e autorização para acesso ao aplicativo. |
| **RNF005** | Interface Intuitiva | A interface deve ser fácil de usar e navegar para todos os tipos de usuários. |
| **RNF006** | Acessibilidade | O aplicativo deve ser acessível para usuários com deficiência, com suporte a leitores de tela e navegação por teclado. |
| **RNF007** | Plataformas | O aplicativo deve ser compatível com principais sistemas operacionais móveis (iOS e Android) e navegadores web modernos. |
| **RNF008** | Dispositivos | Deve ser adaptável a diferentes tamanhos de tela e resoluções. |
| **RNF009** | Atualizações | O aplicativo deve ser facilmente atualizável para correção de bugs e adição de novas funcionalidades. |
| **RNF010** | Documentação | Fornecer documentação técnica e de usuário abrangente. |

*Exemplo: Desempenho, Confiabilidade, Segurança, Portabilidade, Escalabilidade, Manutenção, Interoperabilidade, Adaptabilidade, Acessibilidade*

**6.1 Diagrama de Casos de Uso**

Nesta seção, apresentaremos o diagrama de casos de uso do sistema, que visa representar visualmente as interações entre os atores e o sistema. O diagrama de casos de uso fornece uma visão geral das funcionalidades principais e dos fluxos de trabalho mais relevantes do sistema.

INSERIR DIAGRAMA AQUI

* 1. **Diagrama Entidade-Relacionamento**

O diagrama ER (Entidade-Relacionamento) é uma representação gráfica utilizada na modelagem de sistemas e bancos de dados. Ele descreve a estrutura lógica do sistema, mostrando as entidades (objetos ou conceitos) envolvidos e os relacionamentos entre eles. Esse diagrama é fundamental para compreender como os dados estão organizados e interagem no sistema.

As associações são representações dos relacionamentos entre as entidades no diagrama ER. Elas indicam como uma entidade se relaciona com outra e podem ser do tipo um-para-um, um-para-muitos ou muitos-para-muitos. As cardinalidades são usadas para expressar quantos elementos de uma entidade estão relacionados com a outra entidade. Elas ajudam a definir as restrições e a natureza dos relacionamentos.

INSERIR DIAGRAMA AQUI

obs.: Incluir o diagrama ER no Documento de Visão é importante para garantir uma compreensão clara da estrutura lógica do sistema por parte da equipe e das partes interessadas. Ele facilita a comunicação e o desenvolvimento eficiente, fornecendo uma visão geral dos dados e das interações entre eles. O diagrama ER e as informações sobre associações e cardinalidades auxiliam na definição das necessidades do projeto e na criação de soluções adequadas para o sistema.

* 1. **Diagrama de Atividades ou de Sequência**

Diagrama de Sequência: O diagrama de sequência é uma representação visual que ilustra as interações e a ordem de mensagens trocadas entre objetos em um cenário específico de um sistema orientado a objetos.

INSERIR DIAGRAMA AQUI, SE HOUVER

Diagrama de Atividades: O diagrama de atividades é uma representação visual que descreve a sequência de atividades e decisões em um processo ou fluxo de trabalho.

INSERIR DIAGRAMA AQUI, SE HOUVER

# **7. RESTRIÇÕES**

Descreve as restrições que são impostas ao sistema ou ao processo de desenvolvimento. Para a plataforma NOME DO PROJETO, listo as restrições abaixo:

*Exemplo:*

* *Necessidade de novas licenças de software;*
* *Custos não cobertos pela métrica;*
* *Imposição de novas tecnologias;*
* *Ambiente físico ou plataformas;*
* *Utilização de diferentes sistemas operacionais;*
* *Deve ser funcional no ambiente web e mobile;*
* *Deve ser responsivo no ambiente web e mobile;*
* *Deve ser compatível com os browsers mais populares (chrome e firefox);*

# **8. POLÍTICA DE VERSIONAMENTO**

Descreve como será realizada a política de versionamento do projeto.

* *Gitlab;*
* *Branch Master;*
* *Cada alteração é acompanhada pela equipe antes de cada commit;*
* É mantido um histórico de versão por meio do gitlab e a cada *commit* são executados testes automatizados, por meio do componente de integração contínua do gitlab.

# **9. REFERÊNCIAS**

São referências para este documento de visão:

LISTAR REFERÊNCIAS

Não se aplica