

# DOCUMENTO DE REQUISITOS

## MusicLy



MusicLy

**Aluno:** Luana da Costa Cordeiro  
**Número:** 19

Russas-CE

# Sumário

<b>1. Introdução.....</b>	<b>3</b>
1.1. Requisitos.....	3
1.1.1. O que são requisitos funcionais?.....	3
1.1.2. O que são requisitos não funcionais?.....	4
1.2. Descrição do Usuário.....	4
1.2.1. Atores.....	4
<b>2. Requisitos Funcionais.....</b>	<b>5</b>
2.1. Sistema de Login (RF001).....	5
2.2. Gerenciamento de Músicas para Administradores (RF002).....	5
2.3. Gerenciamento de Álbuns para Administradores (RF003).....	6
2.4. Gerenciamento de Artistas para Administradores (RF004).....	6
2.5. Gerenciamento de Gêneros para Administradores (RF005).....	7
2.6. Gerenciamento de Playlists para Usuários (RF006).....	7
2.7. Gerenciamento de Álbuns para Artistas (RF007).....	7
<b>3. Requisitos Não-Funcionais.....</b>	<b>8</b>
3.1. Usabilidade (RNF001).....	8
3.1.1. Responsividade.....	8
3.1.1.1 Responsividade Bootstrap.....	8
3.1.2. Validação de Campos.....	9
3.1.2.1 Parsley.....	9
3.1.2.2 Bootstrap.....	9
3.1.3. Mensagens de Aviso.....	9
3.2. Banco de Dados MySQL (RNF002).....	10
3.3. PHP (RNF003).....	11
3.4. Frameworks (RNF004).....	11
3.4.1. Bootstrap.....	12
3.5. Requisitos Mínimos Para o Uso (RNF005).....	12
3.5.1. Bootstrap.....	12
3.5.1.1 Bootstrap 5.0.....	13
3.5.2. PHP.....	14
3.5.2.1. PHP 8.0.....	14
3.5.3. PDO.....	14
3.5.4 Parsley.....	15
<b>4. Referências Bibliográficas.....</b>	<b>15</b>

# 1. Introdução

O MusicLy é um projeto que funciona como um Catálogo de Músicas. O administrador pode criar, observar, atualizar ou deletar músicas, álbuns, artistas e gêneros. É de desejo que o projeto também tenha diferentes atividades para o usuário comum e o artista. O usuário pode observar as músicas, álbuns, artistas e gêneros presentes no catálogo e pode criar playlists, adicionando músicas. O artista pode criar álbuns e adicionar músicas, além de ver outros artistas e gêneros presentes também no catálogo. Esta aplicação pode ser utilizada por lojas de discos ou aplicativos que buscam armazenar as músicas, artistas e álbuns presentes em seu estabelecimento/software.

Este documento apresenta uma descrição dos requisitos do software MusicLy, detalhando os requisitos funcionais e não funcionais. Além disso, descreve os tipos de usuários e suas interações com o sistema, bem como as tecnologias e frameworks utilizados no desenvolvimento do projeto.

## 1.1. Requisitos

Os requisitos de software são as necessidades e expectativas dos usuários e outras partes em relação ao sistema que está sendo desenvolvido. Eles definem o que o sistema deve fazer, como ele deve se comportar e quais são os seus limites e restrições. Os requisitos são divididos em requisitos funcionais e não funcionais, definindo o que o software deve fazer e como ele deve operar em diferentes ambientes e situações. Esses requisitos moldam a experiência do usuário, a eficiência operacional e a adaptabilidade do software.

### 1.1.1. O que são requisitos funcionais?

Os requisitos funcionais descrevem as ações específicas que um sistema ou aplicativo deve ser capaz de executar. Eles são as capacidades concretas e as

operações que o software deve realizar para atender às necessidades e expectativas do usuário.

**Exemplos:**

- Processamento de transações
- Gerenciamento de usuários
- Busca e filtragem de dados
- Geração de relatórios

### 1.1.2. O que são requisitos não funcionais?

Os requisitos não funcionais especificam como o software deve fazer isso. Eles são cruciais para garantir a qualidade e a eficiência do software, abrangendo aspectos como desempenho, segurança, confiabilidade e usabilidade. Esses requisitos não estão diretamente ligados às funções específicas do software, mas sim à sua operação e ambiente.

**Exemplos:**

- Desempenho
- Segurança
- Confiabilidade
- Usabilidade

## 1.2. Descrição do Usuário

Um usuário, no contexto da tecnologia da informação, é qualquer pessoa que interage com um sistema computacional, seja ele um software, um aplicativo, um site ou um dispositivo eletrônico.

### 1.2.1. Atores

**Administrador/Funcionário:** O administrador ou o funcionário pode editar a tabela do artista, do álbum, das músicas e dos gêneros, podendo criar, deletar ou modificar o conteúdo dessas tabelas.

**Artista:** O artista pode adicionar álbuns e também introduzir músicas em seus respectivos álbuns. O artista não pode modificar a tabela de artistas.

**Usuário Comum:** O usuário comum pode criar playlists e adicionar músicas nas playlists criadas. O usuário não pode adicionar músicas (diretamente na tabela), álbuns, artistas ou gêneros.

## 2. Requisitos Funcionais

### 2.1. Sistema de Login (RF001)

O sistema deve permitir que o usuário crie sua conta e faça o login, caso o usuário e a senha sejam válidos.

**Entradas:** Nome de usuário e senha.

**Saídas:** Mensagens de sucesso ou erro.

**Pré-condições:** Nenhuma.

- O usuário deve ser redirecionado para a página inicial caso o login seja bem sucedido.
- O sistema deve exibir uma mensagem de erro caso o login ou a senha estejam incorretos.
- O sistema deve exibir uma mensagem ao criar uma conta.
- O sistema deve exibir uma mensagem caso os inputs estejam vazios.
- O sistema deve exibir uma mensagem de erro caso o usuário já exista ao criar uma conta.

### 2.2. Gerenciamento de Músicas para Administradores (RF002)

O sistema deve permitir que o administrador crie novas músicas no catálogo, atualize informações de músicas existentes, delete músicas do catálogo e visualize a lista completa de músicas.

**Pré-condições:** O usuário deve ter permissões de administrador.

**Entradas:** Informações das músicas (idMusica, nome, álbum, gênero, data\_lanç, etc).

**Saídas:** Mensagem de sucesso ou erro.

- O sistema deve exibir uma mensagem caso o processo seja executado corretamente.
- O sistema deve exibir uma mensagem caso ocorra um erro.
- O sistema deve permitir que o administrador escolha o artista da música.
- O sistema deve permitir que o administrador escolha o gênero da música.
- O sistema deve permitir que o administrador escolha o álbum da música.

## 2.3. Gerenciamento de Álbuns para Administradores (RF003)

O sistema deve permitir que o administrador crie novos álbuns, atualize informações de álbuns existentes, delete álbuns do catálogo e visualize a lista completa de álbuns.

**Pré-condições:** O usuário deve ter permissões de administrador.

**Entradas:** Informações dos álbuns (idalbum, nome, data\_lanç, etc).

**Saídas:** Mensagem de sucesso ou erro.

- O sistema deve exibir uma mensagem caso o processo seja executado corretamente.
- O sistema deve exibir uma mensagem caso ocorra um erro.
- O sistema deve permitir que o administrador escolha o artista do álbum

## 2.4. Gerenciamento de Artistas para Administradores (RF004)

O sistema deve permitir que o usuário crie novos artistas, atualize informações de artistas existentes, delete artistas do catálogo e visualize a lista completa de artistas.

**Pré-condições:** O usuário deve ter permissões de administrador.

**Entradas:** Informações dos artistas (idartista, nome, etc).

**Saídas:** Mensagem de sucesso ou erro.

- O sistema deve exibir uma mensagem caso o processo seja executado corretamente.

- O sistema deve exibir uma mensagem caso ocorra um erro.

## 2.5. Gerenciamento de Gêneros para Administradores (RF005)

O sistema deve permitir que o usuário adicione gêneros, atualize os gêneros existentes, delete gêneros do catálogo e visualize a lista completa de gêneros.

**Pré-condições:** O usuário deve ter permissões de administrador.

**Entradas:** Informações do gênero (idgenero, nome, etc).

**Saídas:** Mensagem de sucesso ou erro.

- O sistema deve exibir uma mensagem caso o processo seja executado corretamente.
- O sistema deve exibir uma mensagem caso ocorra um erro.

## 2.6. Gerenciamento de Playlists para Usuários (RF006)

O sistema deve permitir que o usuário crie novas playlists, adicione músicas às playlists, remova músicas das playlists e visualize suas playlists e as músicas contidas nelas.

**Pré-condições:** O usuário deve estar autenticado.

**Entradas:** Informações das playlists (nome, músicas, etc.).

**Saídas:** Mensagem de sucesso ou erro.

- O sistema deve exibir uma mensagem caso o processo seja executado corretamente.
- O sistema deve exibir uma mensagem caso ocorra um erro.

## 2.7. Gerenciamento de Álbuns para Artistas (RF007)

O sistema deve permitir que o artista crie novos álbuns, adicione músicas relacionando com seus respectivos álbuns, atualize informações de seus álbuns e delete seus álbuns do catálogo.

**Pré-condições:** O usuário deve ter permissões de artista.

**Entradas:** Informações dos álbuns (idalbum, nome, data\_lanç, etc).

**Saídas:** Mensagem de sucesso ou erro.

- O sistema deve exibir uma mensagem caso o processo seja executado corretamente.
- O sistema deve exibir uma mensagem caso ocorra um erro.

## 3. Requisitos Não-Funcionais

### 3.1. Usabilidade (RNF001)

A usabilidade em aplicativos é uma área crucial do design de experiência do usuário (UX), focada em garantir que os usuários possam interagir de forma eficiente e satisfatória com o aplicativo.

#### 3.1.1. Responsividade

A forma como o software se ajusta a diferentes dispositivos e tamanhos de tela, sem comprometer a funcionalidade e a estética do aplicativo. O Musicly utiliza responsividade em seus inputs, botões e textos, se adaptando a celulares como Moto G4.

##### 3.1.1.1 Responsividade Bootstrap

Bootstrap é uma das bibliotecas front-end mais populares para o desenvolvimento de sites e aplicações web responsivas. Ele oferece uma série de ferramentas e componentes que ajudam a criar layouts que se ajustam automaticamente a diferentes tamanhos de tela e dispositivos. A responsividade é um dos pontos fortes do Bootstrap, alcançada principalmente através de seu sistema de grid (grade) flexível e componentes adaptáveis.

- **Sistema de Grid Flexível:** Bootstrap utiliza um sistema de grid baseado em flexbox dividindo a página em 12 colunas.
- **Breakpoints:** Utiliza breakpoints (pontos de interrupção) onde o layout pode mudar para se adaptar melhor ao tamanho da tela. (xs, sm, md, lg, xl, xxl).
- **Classes de Utilidade:** Classes de utilidade que ajudam a aplicar rapidamente estilos responsivos.
- **Componentes Responsivos:** Componentes do Bootstrap, como navegação, modais e cards, são projetados para serem responsivos por padrão.



### 3.1.2. Validação de Campos

A validação dos campos de formulários veio com o objetivo de filtrar os dados para que essas informações sejam devidamente preenchidas. Desse modo, algo que poderia facilitar a validação e deixar mais intuitivo ao usuário seria a aplicação de máscaras nos campos de formulário. Para a validação de campos foi utilizado parsley, para mostrar o aviso de “Este campo é obrigatório” e ao redor foi utilizado a validação do bootstrap.

#### 3.1.2.1 Parsley

Parsley é uma biblioteca JavaScript que facilita a validação de formulários no lado do cliente. Ela permite validar os dados de entrada dos usuários. Com Parsley, você pode adicionar regras de validação diretamente nos elementos do formulário usando atributos HTML ou configurá-las via JavaScript. É feita a validação de acordo com os atributos de cada input e são apresentadas as mensagens de erro, caso existam.

#### 3.1.2.2 Bootstrap

A validação de campos do Bootstrap é um recurso que facilita a validação de formulários HTML com estilos visuais e mensagens de feedback. A partir da versão 5, o Bootstrap oferece classes específicas para adicionar validação visual e mensagens de feedback aos formulários. O Bootstrap utiliza classes CSS para mostrar visualmente o estado de validação dos campos de formulário.

- was-validated
- is-valid
- is-invalid

### 3.1.3. Mensagens de Aviso

Mensagens de aviso são mensagens que aparecem quando o usuário realiza alguma ação, informando se a ação ocorreu de forma correta e atualizando o usuário sempre que ele interage com a interface da aplicação. As mensagens de aviso foram feitas com o php e aparecem para o usuário por meio de alertas do Bootstrap. São elas:

- Tente novamente! Este usuário já existe.
- Tente novamente! As senhas não combinam.

- Sucesso! O usuário foi cadastrado.
- Tente novamente! Usuário ou senha incorretos.
- Preencha todos os campos!
- Sucesso! o(a) artista/gênero/música/álbum foi deletado
- Sucesso! o(a) artista/gênero/música/álbum foi atualizado
- Sucesso! o(a) artista/gênero/música/álbum foi adicionado
- Ocorreu algum erro! Tente novamente.

## 3.2. Banco de Dados MySQL (RNF002)

Um banco de dados é uma coleção de dados estruturados onde esses dados são armazenados e gerenciados. A palavra relacional significa que os dados armazenados estão organizados em tabelas e cada tabela está relacionada de alguma maneira. Caso o software não suporte o modelo relacional, então chamamos DBMS.

O MySQL é um sistema gerenciador de banco de dados relacional de código aberto usado na maioria das aplicações gratuitas para gerir suas bases de dados. O MySQL utiliza a linguagem SQL (Structure Query Language – Linguagem de Consulta Estruturada), que é a linguagem mais popular para inserir, acessar e gerenciar o conteúdo armazenado num banco de dados.

MySQL é um Banco de Dados relacional (RDBMS – *Relational Database Management Systems*) com um modelo de cliente-servidor. RDBMS é um software de código aberto ou serviço usado na criação e gerenciamento de bancos de dados baseados no modelo relacional.

### **Vantagens:**

Uma das principais vantagens do MySQL é sua natureza de código aberto. Isso significa que qualquer pessoa pode baixar, usar e modificar o MySQL de acordo com suas necessidades.

- Código aberto.
- Facilidade de uso.
- Compatibilidade.
- Suporte da comunidade.

- Segurança.

### 3.3. PHP (RNF003)

PHP é um acrônimo recursivo para Hypertext Preprocessor. É uma linguagem de script open source de uso geral e adequada para o desenvolvimento web, podendo ser embutida dentro do HTML. Os Scripts PHP são executados no servidor e é gratuito para baixar e usar. O PHP pode gerar conteúdo de página dinâmico, criar, abrir, ler, escrever, excluir e fechar arquivos no servidor, coletar dados de formulário, enviar e receber cookies, adicionar, excluir, modificar dados em seu banco de dados, pode ser usado para controlar o acesso do usuário e criptografar dados.

#### **Vantagens:**

A melhor coisa em usar o PHP é que ele é extremamente simples para um iniciante, mas oferece muitos recursos avançados para um programador profissional.

- É executado em várias plataformas (Windows, Linux, Unix, Mac OS X, etc.).
- É compatível com quase todos os servidores usados atualmente (Apache, IIS, etc.).
- Suporta uma ampla variedade de bancos de dado.
- É gratuito.
- É fácil de aprender e funciona com eficiência no lado do servidor.

### 3.4. Frameworks (RNF004)

Frameworks são estruturas compostas por um conjunto de códigos genéricos que permite o desenvolvimento de sistemas e aplicações. Um framework funciona como uma espécie de template ou modelo que, quando utilizado, oferece certos artifícios e elementos estruturais básicos para a criação de alguma aplicação ou software.

O framework utilizados no projeto foi:

- Bootstrap.

### 3.4.1. Bootstrap

Bootstrap é um kit de ferramentas de front-end poderoso e repleto de recursos. É um framework web com código-fonte aberto para desenvolvimento de componentes de interface e front-end para sites e aplicações web, usando HTML, CSS e JavaScript, baseado em modelos de design para a tipografia, melhorando a experiência do usuário em um site amigável e responsivo.

#### **Vantagens:**

- Velocidade de desenvolvimento.
- Compatibilidade.
- Responsividade.
- Open source.
- Fácil adoção.
- Personalização.
- Popularidade.
- Integração.

## 3.5. Requisitos Mínimos Para o Uso (RNF005)

São os requisitos de versão mínima do navegador para rodar o site, compatibilidade com os principais navegadores (Chrome, Firefox, Safari, Edge), requisitos de hardware e software para o servidor e os dispositivos clientes.

### 3.5.1. Bootstrap

- Navegadores alternativos que usam a versão mais recente do WebKit, Blink ou Gecko (tanto diretamente, quanto através da API web view da plataforma) não suportam explicitamente o Bootstrap.
- Suporte a navegadores modernos como Google Chrome, Mozilla Firefox, Safari, Microsoft Edge.

- Suporte básico ao Internet Explorer 11 (embora o Bootstrap 5 tenha removido o suporte oficial, projetos legados podem exigir).
- Organização do projeto que permita a inclusão dos arquivos principais do Bootstrap (CSS e JS) e suas dependências, como Popper.js e jQuery (caso necessário).
- Estrutura de pastas adequada para arquivos de estilos personalizados, scripts e assets.
- Minificação dos arquivos CSS e JS para melhorar o desempenho do carregamento da página.

#### 3.5.1.1 Bootstrap 5.0

- Bootstrap 5.0 é baseado em HTML5, portanto, seu ambiente de desenvolvimento precisa suportar as especificações HTML5.
- O bootstrap é um framework CSS, então você precisa ter suporte para CSS em seu ambiente de desenvolvimento.
- **JavaScript (opcional):** Se você estiver usando componentes interativos, como modals ou carousels, precisará do JavaScript.
- Baixar os arquivos do bootstrap 5.0 diretamente do site oficial ou incluí-los através de um CDN.
- Incluir os arquivos CSS e JavaScript do Bootstrap em seu projeto para que os estilos e funcionalidades sejam aplicados corretamente.
- O bootstrap 5.0 requer o Popper.js para alguns de seus componentes JavaScript, como tooltips e popovers.
- O bootstrap 5.0 é compatível com os navegadores modernos mais comuns, mas a compatibilidade exata pode depender das funcionalidades específicas do CSS e JavaScript que você está utilizando.

### 3.5.2. PHP

- Configuração adequada do servidor para processar arquivos PHP (geralmente via módulo ou FastCGI).
- Apache, Nginx, IIS ou qualquer outro servidor web compatível com PHP.
- PHP é compatível com diversos sistemas operacionais, incluindo Linux, Windows, macOS e outros sistemas baseados em Unix.

#### 3.5.2.1. PHP 8.0

- Servidor Web: Apache 2.4 ou Nginx 1.18.
- Sistema Operacional: Ubuntu 20.04, Windows 10, macOS Catalina.
- Extensões: ctype, fileinfo, mbstring, openssl, PDO, tokenizer, xml.
- Memória: 1GB RAM (mínimo para produção).
- Banco de Dados: MySQL 5.7+, MariaDB 10.2+, PostgreSQL 10+, SQLite 3.31+.
- Ferramentas: Composer, Git.

### 3.5.3. PDO

- **PHP 5.1.0 ou superior:** PDO foi introduzido no PHP 5.1.0, então você precisa pelo menos dessa versão. No entanto, é recomendado usar uma versão mais recente do PHP, como PHP 7.x ou 8.x, para garantir suporte e segurança.
- **Servidor de banco de dados compatível:** O servidor de banco de dados que você planeja usar deve estar instalado e configurado adequadamente. Isso inclui servidores como MySQL, PostgreSQL, SQLite, etc.
- **Servidor Web:** Um servidor web como Apache, Nginx, ou IIS, configurado para processar arquivos PHP.
- **Sistema Operacional:** Qualquer sistema operacional compatível com PHP, como Linux, Windows, ou macOS.

### 3.5.4 Parsley

- **HTML5:** Parsley utiliza atributos de dados para definir regras de validação, o que é uma característica do HTML5.
- **JavaScript:** Como Parsley é uma biblioteca JavaScript, você precisa de um ambiente onde o JavaScript esteja habilitado e funcional.
- **Biblioteca jQuery (opcional):** Embora Parsley possa funcionar de forma independente, a integração com jQuery facilita a manipulação do DOM e a inicialização.
- Incluir o Parsley diretamente de um CDN ou baixá-lo e hospedá-lo localmente em seu projeto.
- É compatível com todos os navegadores modernos, incluindo Chrome, Firefox, Safari, Edge e IE 10+.
- Se for utilizar o Parsley sem jQuery, inclua apenas o arquivo parsley.min.js. Se estiver utilizando com jQuery, inclua jquery.min.js antes de parsley.min.js.

## 4. Referências Bibliográficas

<https://pt.linkedin.com › pulse › requisitos-de-software-u...>  
<https://querobolsa.com.br/revista/requisitos-funcionais-e-nao-funcionais>  
<https://escolalbk.com.br>  
<https://rankmyapp.com>  
<https://coodesh.com>  
<https://www.hostinger.com.br/tutoriais/o-que-e-mysql>  
<https://www.techtudo.com.br/noticias/2012/04/o-que-e-e-como-usar-o-mysql.ghhtml>  
<https://awari.com.br>  
<https://www.hostgator.com.br/blog/mysql-e-suas-vantagens/>  
<https://www.w3schools.com/>  
[https://www.php.net/manual/pt\\_BR/intro-what-is.php](https://www.php.net/manual/pt_BR/intro-what-is.php)  
[https://pt.wikipedia.org/wiki/Bootstrap\\_\(framework\\_frontend\)](https://pt.wikipedia.org/wiki/Bootstrap_(framework_frontend))  
<https://tailwindcss.com/>  
<https://tailwindcss.com/>  
<https://www.hostmidia.com.br/blog/bootstrap-10-razoes-para-usar/>  
<https://www.dio.me>  
<https://www.geekalicious.pt>

Acessado em 01/05/2024