

# Documento de Requisitos

## Cadastro de músicas

O aplicativo tem como objetivo cadastrar várias músicas simulando uma playlist. Deve ser possível cadastrar, excluir, listar e alterar músicas. Sobre as músicas, deve-se guardar o nome, artista, duração e álbum.

## Requisitos funcionais

- **[RF001] Cadastrar Música**

O usuário deve conseguir cadastrar novas músicas

**Prioridade:**    ☒ Essencial            ☐ Importante            ☐ Desejável

**Entradas:** nome, artista, duração e álbum.

**Saídas e pós-condições:** Uma nova música é persistida no banco de dados.

- **[RF001] Alterar Música**

O usuário deve conseguir alterar músicas já cadastradas

**Prioridade:**    ☒ Essencial            ☐ Importante            ☐ Desejável

**Entradas:** nome, artista, duração e álbum.

**Saídas e pós-condições:** Os dados da música são alterados no banco de dados.

- **[RF001] Remover Música**

O usuário deve conseguir remover músicas.

**Prioridade:**    ☒ Essencial            ☐ Importante            ☐ Desejável

**Entradas:** idMusica

**Saídas e pós-condições:** A música é excluída do banco de dados.

- **[RF001] Listar Músicas**

O usuário deve conseguir listar as músicas

**Prioridade:**    ☒ Essencial            ☐ Importante            ☐ Desejável

**Saídas e pós-condições:** As músicas são recuperadas do banco de dados.

## Requisitos não funcionais

### Padrões

Esta seção descreve os requisitos não funcionais associados a padrões ou normas que devem ser seguidos pelo sistema ou pelo seu processo de desenvolvimento.

- **[NF001] Dividir o sistema em módulos**

O sistema deve priorizar a melhor separação entre as responsabilidades. O sistema deve ser feito utilizando camadas, cada uma delas possui apenas algoritmos relacionados com a sua responsabilidade.

**Prioridade:**    ☒ Essencial            ☐ Importante            ☐ Desejável

### Hardware e software

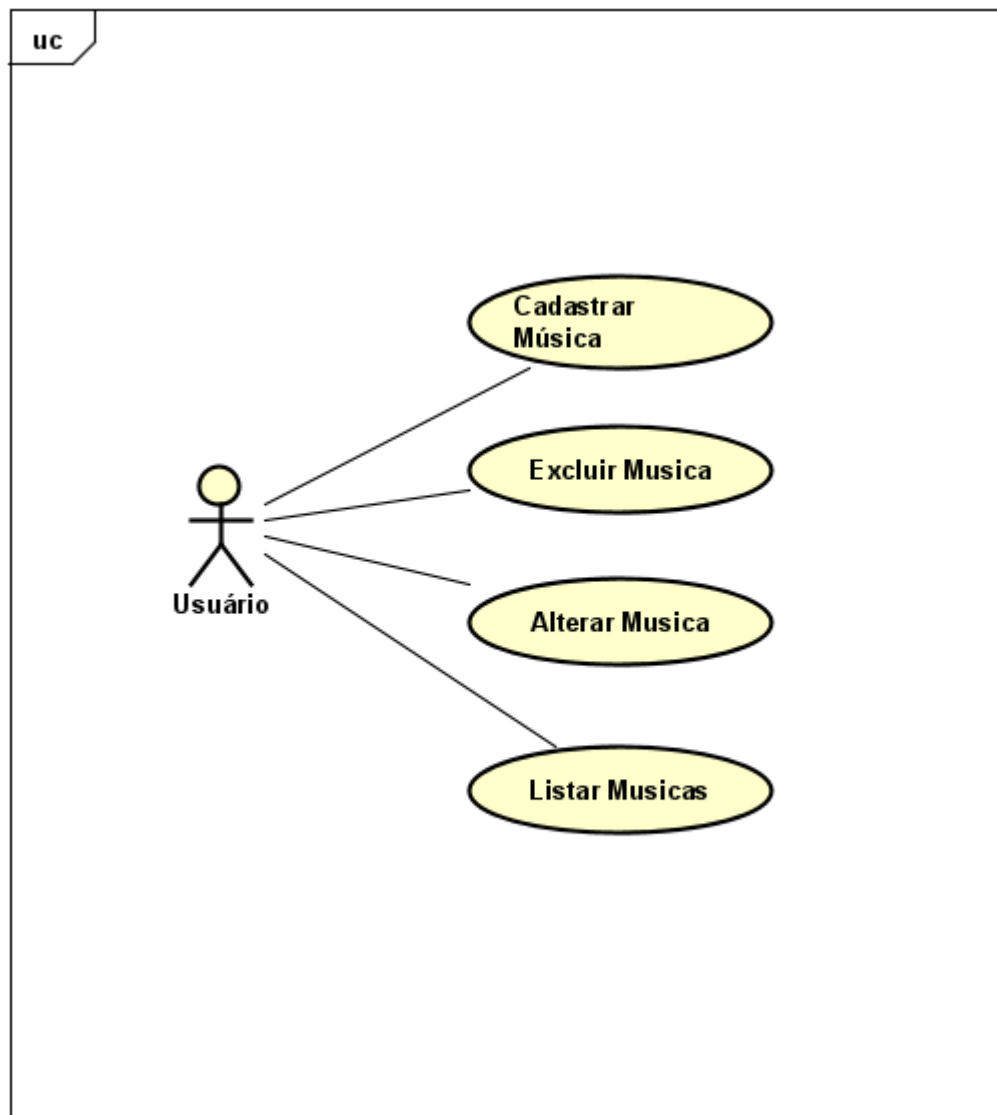
Esta seção descreve os requisitos não funcionais associados ao hardware e software usados para desenvolver ou para executar o sistema.

- **[NF001] Spring Boot**

O sistema deve utilizar Spring Boot.

**Prioridade:**    ☒ Essencial            ☐ Importante            ☐ Desejável

### Diagrama de caso de uso



- **CD001 - Cadastrar Música**

O usuário deve ser capaz de cadastrar uma nova música

**Ator:** Usuário

**Requisitos Funcionais relacionados:** RF001

**Pós-Condições:** Os dados da música estão armazenados no sistema

#### **Fluxo Básico**

1. O usuário informa os dados da música

2. O sistema valida os dados informados
3. O sistema gera o identificador e associa a música cadastrada
4. As alterações são persistidas no banco de dados

### **Fluxo Alternativo 1: Cancelar operação**

- 1 - O usuário cancela a operação
  - 1.1 A inserção é cancelada

## ● **CD002 - Editar Música**

O usuário deve ser capaz de editar uma música.

**Ator:** Usuário

**Requisitos Funcionais relacionados:** RF002

**Pós-Condições:** Os dados da música estão atualizados no sistema

### **Fluxo Básico**

1. Usuário escolhe a música que será editada
2. O usuário informa os dados a serem alterados
3. O sistema valida os dados
4. O sistema atualiza os dados da música

### **Fluxo Alternativo 1: Cancelar operação**

- 1 - O usuário cancela a operação
  - 1.1 As alterações são canceladas

### **Fluxo Alternativo 2: O usuário forneceu dados inválidos**

- 3 - O sistema invalidou os dados informados
  - 3.1 A alteração é cancelada
  - 3.2 O sistema solicita a entrada de dados válidos

## ● **CD003 - Excluir Música**

O usuário deve ser capaz de excluir uma música.

**Ator:** Usuário

**Requisitos Funcionais relacionados:** RF003

**Pós-Condições:** Os dados da música são removidos do banco de dados.

### **Fluxo Básico**

1. O usuário escolhe a música que será excluída
2. O sistema exclui a música selecionado

- **CD004 - Listar Músicas**

O usuário deve ser capaz de listar músicas.

**Ator:** Usuário

**Requisitos Funcionais relacionados:** RF004

**Pós-Condições:** Os dados da música são recuperados do banco de dados.

**Fluxo Básico**

1. O usuário solicita as músicas cadastradas
2. O sistema entrega a lista de músicas cadastradas

## Diagrama de classe

Música
- id : int - nome : String - artista : String - duracao : int - álbum : String
+ getId() : int + getNome() : String + getArtista() : String + getDuracao() : int + getAlbum() : String + setNome() : void + setArtista() : void + setDuracao() : void + setAlbum() : void

