**BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**UNIVERSIDADE DE PERNAMBUCO**

Sound Mood - Sistema de Recomendação de Música

Alunos:

Bianca Moreira

Leylane Ferreira

Iago Richard

Washington Filipe

Turma:

5º período

Professor:

Wylliams Barbosa Santos

Caruaru, 20 de abril de 2015.

Sumário

1. Descrição do Projeto 3
2. Requisitos 4
3. Casos de Uso 7
4. Diagrama de Classes 16
5. Anexos 16
6. Referências 19
7. Descrição do Projeto

O sistema sugerido será um sistema de recomendação. Estes sistemas se utilizam de mecanismos de busca personalizada para melhor atender o usuário. Com o advento do consumo e comércio eletrônico, os sistemas de recomendação tornaram-se extremamente ativos e necessários.

Com base neste tipo de sistema, pensamos em criar para a cadeira de Projeto de Orientação a Objeto um sistema de recomendação musical. Este sistema sugerirá músicas a partir de informações como: Humor do usuário, o que o usuário está fazendo, gênero escolhido pelo usuário no momento de execução do sistema.

Para melhor atender às necessidades do mercado, fizemos uma pesquisa dos principais aplicativos de música para computador. Encontramos o Shazam, Spotify e Deezer.

O Shazam reconhece músicas e faixas a partir do som gravado pelo dispositivo móvel. Também, descobre informações sobre a música reconhecida. Porém, não trabalha com sistema de recomendações, muito menos trabalha com autenticação.

O Spotify pesquisa artistas, músicas e playlists. O usuário monta novas playlists e playlists baseadas no humor. Porém, não trabalha com o conceito de recomendação a partir do gosto musical do usuário, humor e atividade realizada atualmente. Pode rodar músicas offline apenas para usuários *premium*.

Achamos também o Deezer, que trabalha com um conceito bem parecido com o sugerido pela nossa equipe neste projeto. Porém, encontramos diferenças significativas: as recomendações do Deezer se dão a partir do gênero musical e artistas que o usuário curtiu a priori no aplicativo. Diferentemente do nosso projeto, que trabalhará com o conceito de recomendação voltado ao humor e à atividade realizada pelo usuário.

Concluindo, estamos trabalhando em um sistema que se assemelha a outros existentes no mercado, porém com o diferencial a recomendação por “status”. Ou seja, o sistema recomendará músicas a partir do que o usuário estiver fazendo e como o usuário estiver se sentindo. Este conceito ajudará o usuário a conhecer novas músicas e bandas que previamente aquele não conhecia.

A princípio, em nosso o usuário precisará de autenticação. Após autenticar, o usuário navegará pelas telas principais do sistema que são: A tela onde o usuário especificará o seu estado de HUMOR, após isto, o que ESTÁ FAZENDO, e por último, o gênero musical que deseja escutar. A partir desta informação, o sistema calculará e montará uma playlist com base no banco de dados. Utilizaremos JAVA como linguagem de programação e NetBeans como IDE. Utilizaremos também, MySQL para o Banco de Dados e o padrão de projeto MVC.

1. Requisitos

Nessa fase deveremos esclarecer os requisitos do sistema. Definiremos os requisitos funcionais e não funcionais. Os requisitos serão agrupados de acordo com as suas características.

De acordo com Sommerville (2007), requisitos podem ser definidos como sendo o conjunto de funções, objetivos, propriedades e restrições que um sistema apresenta, seguindo a necessidade do usuário final e cliente. Os requisitos fazem parte de todo o ciclo de desenvolvimento de um software, não apenas do seu começo. Serve para descrever funcionalidades do sistema, como também para avaliar se o que o cliente deseja é compatível com a real capacidade da equipe de produção. Ou seja, se o sistema pode realmente ser desenvolvido ou não. Os requisitos são divididos em Requisitos Funcionais e Não Funcionais.

Segundo Pressman (2006), os requisitos são esclarecidos pelo engenheiro de software no processo de engenharia de requisitos. Esta fornece os mecanismos mais adequados para o entendimento do que o cliente deseja. Pressman afirma, também, que nesse processo, analisa-se as necessidades do cliente, avalia-se a viabilidade, negocia-se uma solução razoável, ao mesmo tempo que se especifica a solução sem ambiguidades, e, por fim, valida-se a especificação.

A etapa de esclarecimento de requisitos abrange sete tarefas distintas, entre elas: Concepção, levantamento, elaboração, negociação, especificação, validaç’ao e gestão. Podem ocorrer em simultâneo ou sozinhas, porém todas são adaptadas às necessidades apresentadas no projeto.

* 1. Requisitos Funcionais

Segundo Rezende (2005), os requisitos funcionais descrevem as necessidades do sistema com base no que o cliente precisa. A equipe do projeto especifica o sistema, suas funções, o desempenho, interfaces e restrições (REZENDE, 2005).

* RF001:
  + O sistema deverá efetuar cadastro do usuário: Permite que um usuário cadastre informações de acesso do usuário que deve ser utilizado na hora do login. Não deve haver outra maneira de começar a utilizar o sistema (entrada)
* RF002:
  + O sistema deverá permitir o log in do usuário: Permite que o usuário utilize o nome de usuário e a senha previamente cadastrados no sistema para ter acesso a este. Não há outra forma de entrada no sistema (entrada)
* RF003:
  + O sistema deverá permitir o logoff do usuário: Permite que o usuário termine todas as suas atividades no sistema e saia do mesmo (saída)
* RF004:
  + O sistema deverá permitir que o usuário escolha opções de “status”: O usuário do sistema escolherá entre uma das opções impressas na tela para dar seguimento ao uso do sistema, pois se trata de um sistema de recomendação e apenas após a escolha de opção, o sistema poderá calcular a *playlist* de músicas baseada na escolha do usuário (entrada)
* RF005:
  + O sistema deverá retornar uma playlist: Baseando-se no que o usuário escolheu nas opções de status e humor, o sistema calculará uma lista de músicas e retornará para o usuário no player de música (saída)
* RF006:
  + O sistema para a música atual: Se por algum evento do meio externo, o usuário decida parar a música, este requisito será necessário. O usuário deve clicar no botão “pause” para efetuar com sucesso (entrada e saída)
* RF007:
  + O sistema retorna para a música anterior: Volta para a música que estava tocando previamente. Esta função apenas funciona para usuários do tipo PREMIUM (entrada e saída)
* RF008:
  + O sistema passa para a próxima música: Passa para a música que vem em seguida a que está atualmente tocando. Esta função apenas funciona para usuários do tipo PREMIUM (entrada e saída)
* RF009:
  + Trocar *playlist:* Esta função dar ao usuário a opção de mudar de *playlist* se aquele não gostou desta. O usuário PREMIUM poderá realizar esta função quantas vezes desejar, porém, o usuário FREE apenas poderá realizar 5 (cinco) vezes.
  1. Requisitos Não-Funcionais

Pressman (2006) define Requisitos Não-Funcionais como os fatores de qualidade de software de forma indireta ou como características que ficam implícitas no sistemas, mas são esperadas de todo software profissionalmente desenvolvido.

Os Requisitos Não-Funcionais descrevem como o sistema fará o que foi especificado nos Requisitos Funcionais. Eles não estão ligados às funcionalidades do sistema diretamente, mas como estas se comportarão em termos de qualidade. Definem propriedades e restrições do sistema, como por exemplo, tempo, espaço, linguagem de programação utilizada, versão do compilador utilizada, SGBD, sistemas operacionais, método de desenvolvimento, entre outros. Eles abordam aspectos de qualidades. Em suma, se não forem levados em consideração como devem ser, o sistema poderá ser desenvolvido com baixa qualidade e poderá se tornar inconsistente.

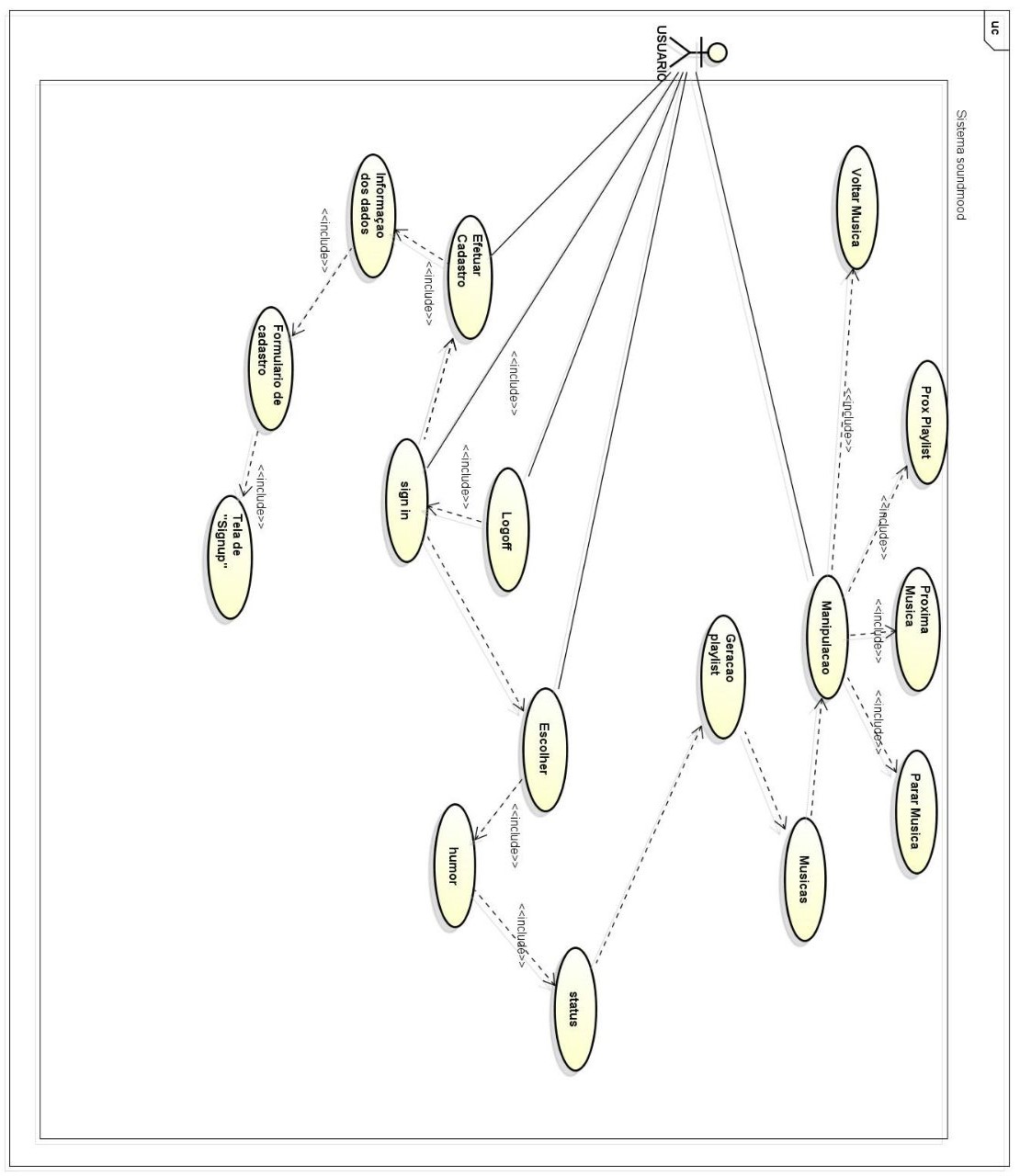
* RNF001:
  + Linguagem de programação: A linguagem utilizada para o desenvolvimento do software será Java.
* RNF002:
  + Plataformas Compatíveis: O sistema será multiplataforma.
* RNF003:
  + SGBD: O sistema usará como manipulação da base de dados o SGBD Oracle.
* RNF004:
  + Tempo de clique de um botão: O tempo de resposta de um botão quando acontecer o evento de clique deve ser de 4 segundos. Esse requisito está sujeito a alterações posteriores com o decorrer do desenvolvimento do software. \*Observação
* RNF005:
  + Espaço ocupado pelo software: O software deverá ocupar 10 MB no disco. Esse requisito está sujeito a alterações posteriores com o decorrer do desenvolvimento do software.
* RNF006:
  + Facilidade de uso: O software possuirá uma tela de ajuda para nortear seus usuários.
* RNF007:
  + Confiabilidade: O tempo médio para que o software venha a falhar será de . Esse requisito está sujeito a alterações posteriores com o decorrer do desenvolvimento do software.
* RNF008:
  + Versão do Compilador: O compilador utilizado será o JavaC encontrado na versão 1.48 do JDK.
* RNF009:
  + Robustez: O tempo de reinício após falha do sistema será de 4 segundos. Esse requisito está sujeito a alterações posteriores com o decorrer do desenvolvimento do software.

1. Casos de Uso

De acordo com Booch (2000), um caso de uso descreve cada representação de interação de itens externos ao sistema - os atores - com o sistema e suas principais abstrações, esta descrição se dá em sequência.

* 1. Diagrama de Casos de Uso

O diagrama de caso de uso é o principal diagrama utilizando em UML. Segundo Furlan (1998), ele fornece um modo de descrever a visão externa de um sistema de informação e suas interações com o mundo.



* 1. Detalhamento dos Casos de Uso

Detalhamento dos casos de uso do nosso sistema:

|  |  |
| --- | --- |
| **Número do Caso de Uso:** | **Nome do caso de Uso:** |
| UC0001 | Efetuar Cadastro |
| **Objetivos:** | **Descrição:** |
| Permitir que o usuário venha a usufruir das funcionalidades do sistema estando cadastrado. | Insere um novo usuário no sistema. |
| **Ator:** | **Prioridade:** |
| Um novo usuário do sistema. | [ x ] Alta [ ] Média [ ] Baixa |
| **Pré-condição:** | **Pós-condições:** |
| Não se aplica. | O ator terá um cadastro no sistema do tipo FREE ou PREMIUM, e poderá usar os dados der Username e Password inseridos para entrar no sistema. |
| **Cenários principais de sucesso:** | |
| 1. O ator seleciona a opção “SignUP” na tela principal do sistema;  2. O sistema exibe o formulário de cadastro de um novo usuário;  3. O ator informa os dados do usuário a ser cadastrado;  4. O ator clica no botão “Cadastrar”;  5. O sistema cadastra as informações no banco de dados e será exibida uma mensagem de retorno para o usuário. | |
| **Fluxos Alternativos:** | |
| * ator fornece um username já cadastrado no banco de dados; * A mensagem “Username já utilizado, utilize outro username!” é exibida. | |
| * ator fornece um Email já cadastrado no banco de dados; * A mensagem “Email já utilizado, utilize outro Email!” é exibida. | |
| * ator não preenche todos os campos do formulário de cadastro;   • A mensagem “Preencha todos os campos!” é exibida. | |

|  |  |
| --- | --- |
| **Número do Caso de Uso:** | **Nome do caso de Uso:** |
| UC0002 | Efetuar Login |
| **Objetivos:** | **Descrição:** |
| Permitir que o usuário venha a usufruir das funcionalidades do Sistema a partir do momento em que acessa o sistema. | Permite que o usuário utilize o nome de usuário e a senha previamente cadastrados no sistema para ter acesso a este. Não há outra forma de entrada no sistema. |
| **Ator:** | **Prioridade:** |
| Usuário do sistema. | [x ] Alta [ ] Média [ ] Baixa |
| Pré-condição: | Pós-condições: |
| Não se aplica. | O ator terá acesso às funcionalidades do sistema que lhe dizem respeito. |
| **Cenários principais de sucesso:** | |
| 1. Estando na tela inicial do sistema, o ator deve preencher os campos “login” e “senha”;  2. O ator então clica no botão “SignIN”;  3. O ator será logado no sistema. | |
| Fluxos Alternativos: | |
| * O ator fornece um login não cadastrado no sistema; * A mensagem “Usuário inexistente” é exibida. | |
| * ator fornece um login e uma senha não correspondentes; * A mensagem “Senha incorreta” é exibida. | |

|  |  |
| --- | --- |
| **Número do Caso de Uso:** | **Nome do caso de Uso:** |
| UC0003 | Efetuar logoff |
| **Objetivos:** | **Descrição:** |
| Permitir ao usuário sair do Sistema. | Permite que o usuário termine todas as suas atividades no sistema e saia do mesmo. |
| **Ator:** | **Prioridade:** |
| Usuário do sistema. | [ x ] Alta [ ] Média [ ] Baixa |
| **Pré-condição:** | **Pós-condições:** |
| O ator deve estar logado no sistema no momento da execução dessa operação. | O ator deixa de ter acesso às funcionalidades do sistema. O sistema retorna à tela inicial. |
| **Cenários principais de sucesso:** | |
| * O ator clica no botão “Sair”; * O sistema finaliza todas as operações que estão em execução devido às requisições feitas por esse ator. | |

|  |  |
| --- | --- |
| **Número do Caso de Uso:** | **Nome do caso de Uso:** |
| UC0004 | Escolher Opções de “Status” |
| **Objetivos:** | **Descrição:** |
| Permitir que o usuário escolha um Status. | Escolhe uma opção de status do usuário. |
| **Ator:** | **Prioridade:** |
| Usuário do sistema. | [ ] Alta [ x ] Média [ ] Baixa |
| **Pré-condição:** | **Pós-condições:** |
| O ator deve estar logado no sistema. | O status do usuário (o que ele está fazendo e seu humor) deverá será alterado e será gerada uma playlist de acordo com os dados coletados. |
| **Cenários principais de sucesso:** | |
| * Na tela de escolher status, o ator terá 5 opções de escolha para humor e o que está fazendo no momento da execução do programa; * O ator seleciona uma opção para humor * O ator seleciona uma opção para o que está fazendo * O clica no botão “OK”; * O status do ator no sistema será alterado e a tela onde serão executadas playlist será abertae a mesma começará a ser executada automaticamente. | |
| **Fluxos Alternativos:** | |
| * O ator não seleciona ou o status, ou o humor, ou os dois, e clica no botão “OK” * A mensagem “Selecione um status e um humor para poder gerar a playlist.” será exibida. | |

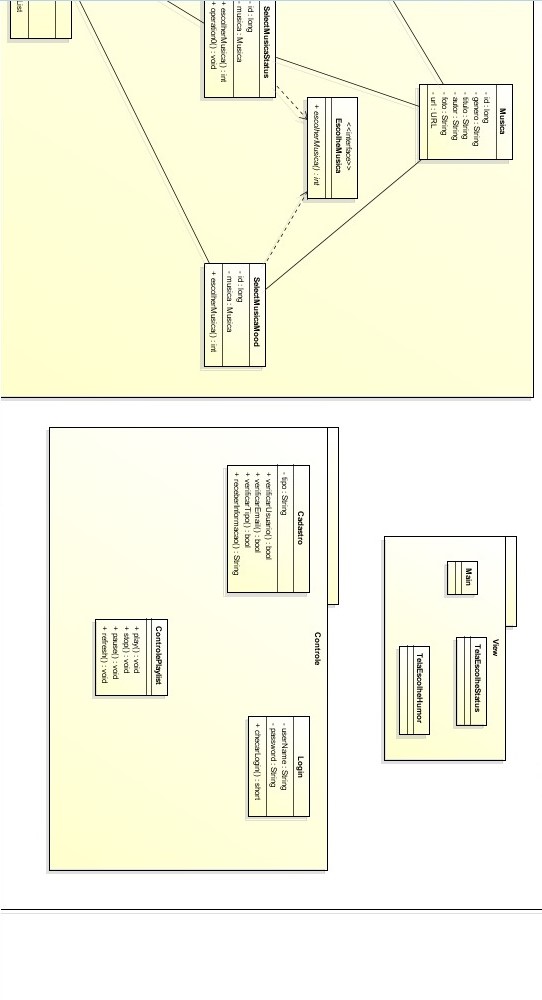
|  |  |
| --- | --- |
| **Número do Caso de Uso:** | **Nome do caso de Uso:** |
| UC0005 | Parar Música |
| **Objetivos:** | **Descrição:** |
| Permitir que o usuário pare a música de acordo com sua preferência. | Se por algum evento do meio externo, o usuário decida parar a música, este requisito será necessário. O usuário deve clicar no botão “pause” para efetuar com sucesso. |
| **Ator:** | **Prioridade:** |
| Usuário do sistema. | [ ] Alta [ x ] Média [ ] Baixa |
| **Pré-condição:** | **Pós-condições:** |
| A playlist deve estar em execução. | A execução de uma música será interrompida. |
| **Cenários principais de sucesso:** | |
| * Na tela de execução da playlist o ator clica no botão referente a parar música; * A execução da música atual será imediatamente interrompido. | |
| **Fluxos Alternativos:** | |
| * Na tela de execução da playlist o ator clica no botão referente a parar música sem ter nenhuma música em execução; * A mensagem “Para se pausar uma música é necessário que a mesma esteja em execução.” | |

|  |  |
| --- | --- |
| **Número do Caso de Uso:** | **Nome do caso de Uso:** |
| UC0006 | Voltar Música |
| **Objetivos:** | **Descrição:** |
| Permitir que o usuário volte a música de acordo com sua preferência. | Volta para a música que estava tocando previamente. Esta função apenas funciona para usuários do tipo PREMIUM. |
| **Ator:** | **Prioridade:** |
| Usuário do sistema. | [ ] Alta [x ] Média [ ] Baixa |
| **Pré-condição:** | **Pós-condições:** |
| A música que está sendo executada pela playlist não deve ser a primeira da mesma. | A música anterior da playlist será executada, se o ator for usuário PREMIUM. |
| **Cenários principais de sucesso:** | |
| * Na tela de execução da playlist o ator clica no botão referente a voltar música; * A música anterior da playlist será imediatamente executada. | |
| **Fluxos Alternativos:** | |
| * Na tela de execução da playlist o ator clica no botão referente a voltar música, sendo que a música em execução é a primeira música da playlist; * A mensagem “Você não pode voltar uma música, visto que a música em execução é a primeira da playlist”. | |
| * Na tela de execução da playlist o ator clica no botão referente a voltar música, sendo que o usuário é um usuário free; * A mensagem “Faça um upgrade para uma conta premium para poder usar esta funcionalidade!” será exibida. | |

|  |  |
| --- | --- |
| **Número do Caso de Uso:** | **Nome do caso de Uso:** |
| UC0007 | Passar Para Próxima Música |
| **Objetivos:** | **Descrição:** |
| Permitir que o usuário possa trocar a música de acordo com sua preferência. | Passa para a música que vem em seguida a que está atualmente tocando. Esta função apenas funciona para usuários do tipo PREMIUM. |
| **Ator:** | **Prioridade:** |
| Usuário do sistema. | [ ] Alta [ x] Média [ ] Baixa |
| **Pré-condição:** | **Pós-condições:** |
| A música que está sendo executada pela playlist não deve ser a última da mesma. | A próxima música da playlist será executada, se o ator for usuário PREMIUM. |
| **Cenários principais de sucesso:** | |
| * Na tela de execução da playlist o ator clica no botão referente a avançar para a próxima música; * A próxima da playlist será imediatamente executada. | |
| **Fluxos Alternativos:** | |
| * Na tela de execução da playlist o ator clica no botão referente a avançar para próxima música, sendo que a música em execução é a última música da playlist; * A mensagem “Você não pode avançar uma música, visto que a música em execução é a útima da playlist”. | |
| * Na tela de execução da playlist o ator clica no botão referente a avançar para próxima música, sendo que o usuário é um usuário free; * 4. A mensagem “Faça um upgrade para uma conta premium para poder usar esta funcionalidade!” será exibida. | |

|  |  |
| --- | --- |
| **Número do Caso de Uso:** | **Nome do caso de Uso:** |
| UC0008 | Passar Para Próxima Playlist |
| **Objetivos:** | **Descrição:** |
| Permitir que o usuário possa trocar a playlist de acordo com sua preferência. | Esta função dá ao usuário a opção de mudar de playlist se aquele não gostou desta. O usuário PREMIUM poderá realizar esta função quantas vezes desejar, porém, o usuário FREE apenas poderá realizar 5 (cinco) vezes. |
| **Ator:** | **Prioridade:** |
| Usuário do sistema. | [ ] Alta [x ] Média [ ] Baixa |
| **Pré-condição:** | **Pós-condições:** |
| A tela de execução da playlist deve estar em execução | A playlist deverá ser trocada, se o ator for usuário PREMIUM. |
| **Cenários principais de sucesso:** | |
| * Na tela de execução da playlist o ator clica no botão referente a trocar de playlist; * A próximaplaylist aleatória será imediatamente executada. | |
| **Fluxos Alternativos:** | |
| * Na tela de execução da playlist o ator clica no botão referente a trocar de playlist, sendo que o ator é um usuário FREE; * A mensagem “Faça um upgrade para uma conta premium para poder usar esta funcionalidade!” será exibida. | |

## Diagrama de Classes Um diagrama de classe UML descreve o objeto e as estruturas usadas pelo seu aplicativo internamente e comunicação com seus usuários de informações. Ele descreve as informações sem referência a qualquer implementação específica. Suas classes e relacionamentos podem ser implementados de diversas maneiras, como tabelas de banco de dados, nós XML ou composições de objetos de software. 5. Anexos



**Referências**

BOOCH, Grady; RAMBAUGH, James; JACOBSON, Ivar. UML: Guia do Usuário. Rio de Janeiro: Campus, 2000.

FURLAN, Furlan, José Davi. Modelagem de Objetos através da UML. São Paulo: MAKRON Books, 1998.

PRESSMAN, Roger. Engenharia de Software. 6a ed., São Paulo: Makron Books, 1995.

SOMMERVILLE, Ian. Engenharia de Software. 6a ed., São Paulo: Addison Wesley, 2003.