

**Universidade São Judas Tadeu**  
**Projeto UC Modelagem de Software**

**Isadora Tavares Riegert**

**Jorge Leandro Piva**

**Mateus Gomes Dias**

**Rebeca Evedove Csato Sózio**

**Rodrigo Bergamin Guimarães da Silva**

**Trabalho para Unidade Curricular de modelagem de  
Software apresentado para aprovação na Unidade  
Curricular de Modelagem de Software**

**São Paulo**

**2020**

**Isadora Tavares Riegert**

**Jorge Leandro Piva**

**Mateus Gomes Dias**

**Rebeca E. C. Sózio**

**Rodrigo Bergamin Guimarães da Silva**

**Trabalho para Unidade Curricular de modelagem de Software  
apresentado para aprovação na Unidade Curricular de Modelagem  
de Software**

**Orientador(a)**

**Andreia Cristina Grisolio Machion**

**São Paulo**

**2020**

## AGRADECIMENTOS

Aos nossos familiares que tanto nos apoiam em nossa jornada tão esperada rumo ao sucesso, aos colegas que há tão pouco tempo juntos já se tornaram uma família, sabendo entender o momento de cada um e respeitando as dificuldades de cada um neste momento tão difícil que estamos passando e aos nossos professores, Hamilton por suas intermináveis listas de exercícios que ajudaram na fixação, Duduchi, pela forma irreverente e simpática de nos ensinar, ao Bossini, por sua extrema competência em ensinar e executar, de forma mágica tornando algo complexo, simples de entender e a Andreia que mesmo não olhando nos nossos olhos nos transmite o brilho do seu olhar ao falar de tecnologia e sistemas, sempre com muita alegria em ensinar, bom humor e competência.

So Houston, now we expect in the next pages we don't have problems, but bad things can happen.

## PREFÁCIO

Iniciar um curso superior não importa em qual área é um desafio, os cursos são longos, muitas coisas podem mudar durante esse longo período, nosso cotidiano está repleto de distrações vindos das próprias tecnologias que nós queremos tanto estudar, é uma tarefa árdua se manter focado e neste momento em que a humanidade passa por um grande desafio com a pandemia do Covid – 19 se torna mais difícil ainda manter o foco, muitas preocupações pairam as cabeças dos jovens, ou não tão jovens estudantes cheios de sonhos e ambições para a vida em TI, até o contato humano a reunião olho no olho se tornou impossível nos dias atuais então se tornou mais necessário ainda o uso da tecnologia para elaboração de nossos projetos.

Segundo um artigo na computerworld (69% dos universitários desistem dos cursos de tecnologia, afirma Brasscom, 2019) embasado em um levantamento feito pela Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação (Brasscom) o próprio título já diz 69% dos estudantes de tecnologia não terminam o curso, qual será essa evasão agora neste momento de pandemia em que muitos perdem seus empregos, entram em crises existenciais e estão sufocados pela mídia funerária que tem apavorado milhões e brasileiros nos últimos meses?

Este trabalho foi desenvolvido por um grupo de estudantes totalmente aleatórios que não se conheciam e tiveram altos e baixos durante todo o percurso, situações de stress, frustrações, dificuldade de conciliar agenda, trabalho fora da área pretendida gerando frustrações e deixando o tempo mais escasso para o desenvolvimento na área, dificuldade de se adaptar ao modelo de estudo Digital, não pela estrutura ou os professores, mas, novamente por questões de foco.

Mesmo com todas essas situações, muitos de nós estão remando na contra mão buscando nos especializar e estudar, nos esforçando e fazendo reuniões a noite e finais de semana, cursos paralelos na tentativa de acelerar a carreira e o aprendizado, passando horas programando, debatendo casos de uso, layout de telas, e diversas outras atividades acadêmicas, visando um futuro em uma profissão que tem crescido tanto e trazendo tantos benefícios para o ser humano, desta forma, submetemos esse documento entendendo termos planejado um software proposto pelo corpo docente da Universidade São Judas Tadeu, da melhor forma possível

dentro das limitações de um time de calouros e semi-calouros, mas que tem muito esforço, dedicação e vontade de fazer o correto.

Esperamos que todos deste grupo estejam logo formados e com bons empregos trazendo criatividade, inovação e conseguindo transformar o mundo em que vivem impactando milhares e milhares de pessoas.

## LISTA DE FIGURAS

Figura 1 Diagrama de Casos de Uso UML.....	12
Figura 2 Diagrama de Classes UML .....	13
Figura 3 Diagrama de Atividades UML.....	14
Figura 4 Diagrama de Sequência UML .....	15
Figura 5 Diagrama Entidade Relacionamento Conceitual).....	16
Figura 6 Diagrama Entidade Relacionamento Lógico .....	17

## SUMÁRIO

1	INTRODUÇÃO .....	8
2	LEVANTAMENTO DE REQUISITOS.....	9
2.1	Levantamento de <i>Stakeholders</i> .....	9
2.2	Técnica de Levantamento de Requisitos.....	9
2.3	Requisitos de Usuário .....	9
2.4	Requisitos de Sistema.....	9
2.5	Regra de negócio .....	10
2.6	Requisitos funcionais.....	10
2.7	Requisitos não funcionais.....	11
3	DIAGRAMAS UML.....	12
3.1	Diagrama de Casos de Uso .....	12
3.2	Diagrama de Classes .....	12
3.3	Diagrama de Atividades .....	14
3.4	Diagrama de Sequência.....	15
4	DIAGRAMAS DE BANCO DE DADOS .....	16
4.1	Modelo Conceitual ER do DB.....	16
4.2	Modelo Lógico Relacional do DB .....	16
5	CONCLUSÃO .....	18
6	BIBLIOGRAFIA .....	20

# 1 INTRODUÇÃO

O sistema Rock-me de recomendação de músicas é um sistema onde é possível selecionar gêneros preferidos, avaliar músicas cadastradas e caso seja solicitado o sistema apresenta uma recomendação de música através de um posto que é a avaliação que outros usuários deram para a mesma música.

Este sistema é focado nos gêneros musicais do gênero pai Rock, mas pode ser facilmente customizado para outros públicos através da estilização das telas, onde é possível mudar cores ou o logotipo, a manutenção do mesmo se dá de fácil modo uma vez que o usuário administrador pode cadastrar novas músicas e gêneros sempre que quiser.

O sistema foi confeccionado com a linguagem de modelagem UML para auxiliar nas etapas do desenvolvimento do projeto de forma segura, profissional e organizada.

Nas partes iniciais do documento temos os levantamentos de requisitos para ter uma explanação melhor das necessidades do sistema, afim de não deixar de instalar nenhuma funcionalidade chave do sistema, com todas as validações necessárias, nesta seção falaremos sobre os principais *stakeholders*, técnicas de levantamento de requisitos, requisitos de usuários, requisitos de sistema, regras de negócios, requisitos funcionais e requisitos não funcionais.

Após esta seção iremos apresentar os principais diagramas UML para modelagem de dados que são o diagrama de casos de uso, diagrama de classes, diagrama de atividades e diagrama de sequência.

Finalizada a etapa dos diagramas UML, iniciamos o planejamento do banco de dados, e apresentamos nesta seção os diagramas de Entidade relacionamento nos modelos conceitual e lógico.

Todas essas ferramentas são vitais para uma boa construção de um software assim como dito por (Sommerville, Ian, Engenharia de Software - 9ª edição) pág. 19 “a engenharia de software tem por objetivo apoiar o desenvolvimento profissional de software, mais do que a programação individual” e o que buscamos com este projeto é o desenvolvimento de uma aplicação por calouros universitários com qualidade de uma ferramenta profissional.



## 2 LEVANTAMENTO DE REQUISITOS

### 2.1 Levantamento de *Stakeholders*

Equipe, Usuários, Professores, participantes da Expo São Judas, alunos desenvolvedores do projeto.

### 2.2 Técnica de Levantamento de Requisitos

Entrevista - Perguntas e observações realizadas pelo grupo durante as aulas.

### 2.3 Requisitos de Usuário

- O usuário deverá cadastrar um login para acessar o sistema;
- O usuário precisará cadastrar uma senha da qual possa lembrar;
- O usuário poderá solicitar um e-mail de recuperação de senha;
- O usuário deverá escolher ao menos um gênero preferido para poder avaliar músicas
- O usuário poderá obter recomendações.
- Para avaliar suas músicas, o usuário precisará selecionar pelo menos um gênero preferido,
- O usuário deverá escolher a música desejada e avaliar a mesma digitando uma nota de 1 a 5.

### 2.4 Requisitos de Sistema

O sistema não aceita login sem prévio cadastro;

Sistema deverá receber um login com e-mail e senha e gravá-lo no banco de dados;

O sistema validará o login não permitindo duplicidade de login;

O sistema validará o e-mail não permitindo duplicidade de e-mail;

O sistema apresenta opções de gêneros pré cadastrados;

O sistema permite cadastro de novos gêneros pelo usuário administrador;

O sistema subentende que o usuário não deseja recomendações de determinado gênero se o usuário não o escolher como favorito;

O sistema não cadastrará músicas que não possuam ao menos um gênero

O sistema recebe avaliações de notas com o intervalo de 1 a 5 sendo 1 a nota mais baixa e 5 a nota mais alta.

O sistema irá apresentar para o usuário todas as músicas não avaliadas para os gêneros selecionados como preferidos.

O sistema trará uma lista com todas as músicas não avaliadas pelo usuário com o posto da música classificado em ordem decrescente recomendando as músicas com avaliação mais alta dentro outros usuários que avaliaram a mesma música.

Obs: O posto da música é a média das avaliações concedidas por todos os usuários que avaliaram a mesma música.

## **2.5 Regra de negócio**

- Ao se cadastrar o usuário não pode ter um login de outro usuário
- Ao se cadastrar o usuário não poderá ter o mesmo e-mail de outro usuário.
- Ao se cadastrar o usuário deverá digitar a sua senha duas vezes de forma igual.
- Antes de obter recomendações o usuário deverá selecionar pelo menos um gênero preferido.
- As avaliações deverão receber notas entre 1 e 5 sendo 1 a nota mais baixa e 5 a nota mais alta.
- Gêneros cadastrados poderão ser excluídos somente se não estiverem vinculados à uma música ou pessoa.
- Músicas Cadastradas poderão ser excluídas somente se não vinculadas à uma avaliação.
- Ao excluir um gênero não serão perdidas as músicas cadastradas nele.
- Antes de excluir uma avaliação o sistema deverá perguntar ao usuário se ele tem certeza da operação.

## **2.6 Requisitos funcionais**

- O usuário poderá efetuar login no sistema mediante a validação de suas credenciais
- O usuário poderá adicionar gêneros a uma lista de gêneros preferidos
- O usuário poderá avaliar as músicas disponíveis no catálogo
- O usuário receberá recomendações musicais com base nas avaliações feitas em músicas por outros usuários anteriormente
- O usuário poderá remover um gênero da sua lista de gêneros preferidos

## **2.7 Requisitos não funcionais**

- O sistema deverá ser desenvolvido na linguagem Java
- O sistema deverá utilizar o SGBD MySQL
- O sistema deverá utilizar um banco de dados estruturado
- O sistema irá apenas conter músicas cujos direitos e responsabilidades estejam dentro das normas legais.
- O sistema manterá as informações de login, usuário, senha, nome e e-mail do usuário.

### 3 DIAGRAMAS UML

#### 3.1 Diagrama de Casos de Uso

O diagrama de casos de uso UML é o diagrama de mais alto nível da modelagem UML, ou seja, é uma forma de visualizar o sistema mais próximo da linguagem do usuário, nele é possível prever todas as situações que o sistema precisa resolver, "Diagramas de casos de uso, que mostram as interações entre um sistema e seu ambiente." (Sommerville, Ian, Engenharia de Software - 9ª edição) pág. 99.

Na figura 1 podemos ver o diagrama de casos de uso para o sistema de recomendação de músicas.

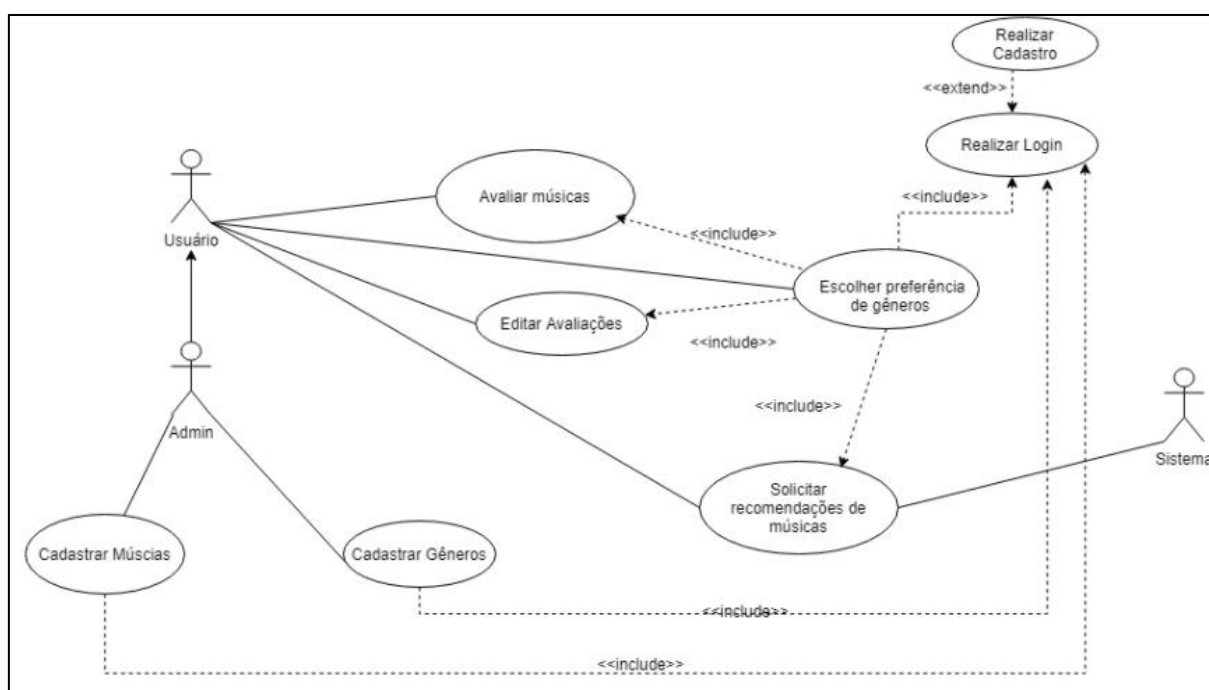


Figura 1 Diagrama de Casos de Uso UML

Um ponto chave desse diagrama é que todas as atividades principais do sistema dependem inicialmente da escolha do gênero preferido, desta forma deverá haver um conjunto forte de métodos de validação para garantir que o usuário não poderá acessar as músicas sem que tenha escolhido um gênero.

#### 3.2 Diagrama de Classes

"Diagramas de classe, que mostram as classes de objeto no sistema e as associações entre elas." (Sommerville, Ian, pág 99, Engenharia de Software - 9ª edição), com este diagrama é possível iniciar as primeiras etapas de programação a

criação dos primeiros objetos, nele já estão definidos como os objetos serão criados, seus atributos e métodos e seu nível de encapsulamento conforme a figura 2.

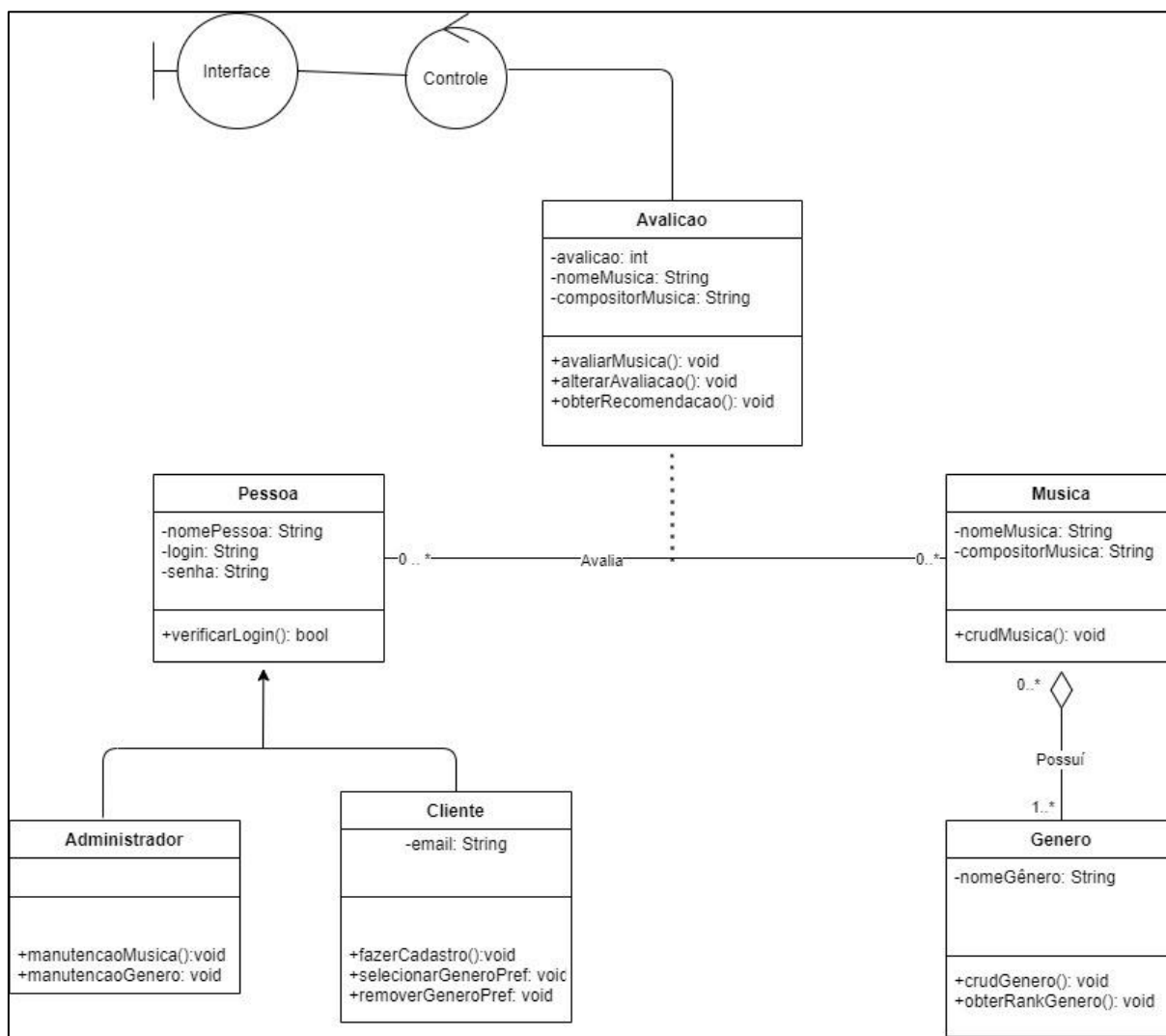


Figura 2 Diagrama de Classes UML

Em seus relacionamentos também é possível verificar a cardinalidade, se há herança ou uma relação de dependência entre as classes.

### 3.3 Diagrama de Atividades

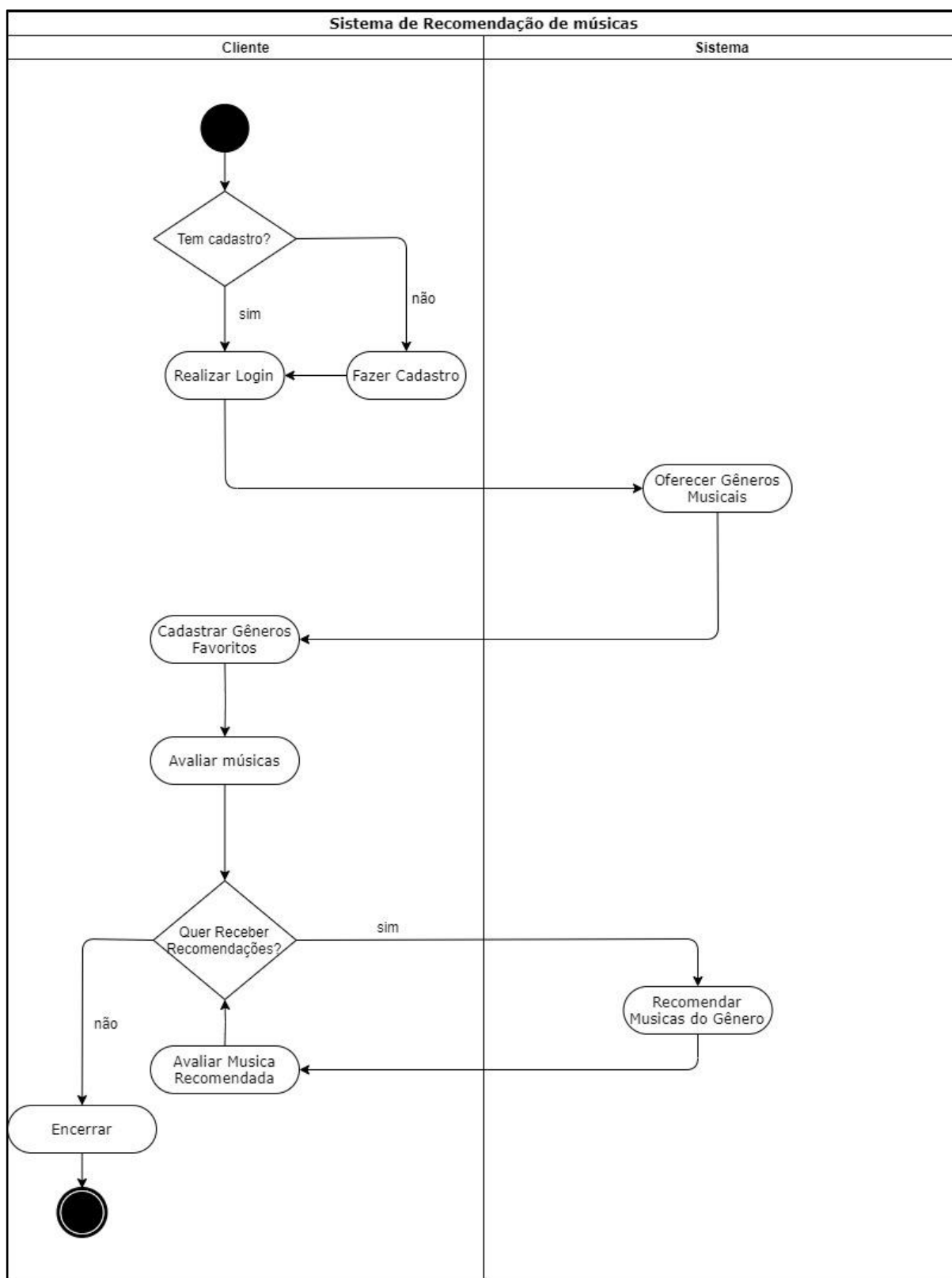


Figura 3 Diagrama de Atividades UML

O diagrama de atividades conforme a figura 3 se assemelha muito ao fluxo de um algoritmo, "A Object Management Group (OMG) estabelece o diagrama de

atividades como parte da Universal Modeling Language (UML) e que pode ser usado também para modelagem de processos de negócio" (Vazquez & Simões, 2016) pág. 222 é o que é percebido dentro do diagrama de atividades um processo que deve ser seguido e suas possíveis ramificações.

### 3.4 Diagrama de Sequência

"Os detalhes das interações envolvidas em um caso de uso de alto nível podem ser documentados em um diagrama de sequência" (Sommerville, Ian, Engenharia de Software - 9ª edição) pag.102, neste caso podemos ver o detalhamento da operação para avaliação de música e a clara relação entre o caso de uso e o diagrama de sequência conforme também observado por(Sommerville, Ian, , Engenharia de Software - 9ª edição) pág.102 "Os modelos de caso de uso e diagramas de sequência apresentam interações em diferentes níveis de detalhamento e, assim, podem ser usados juntos"

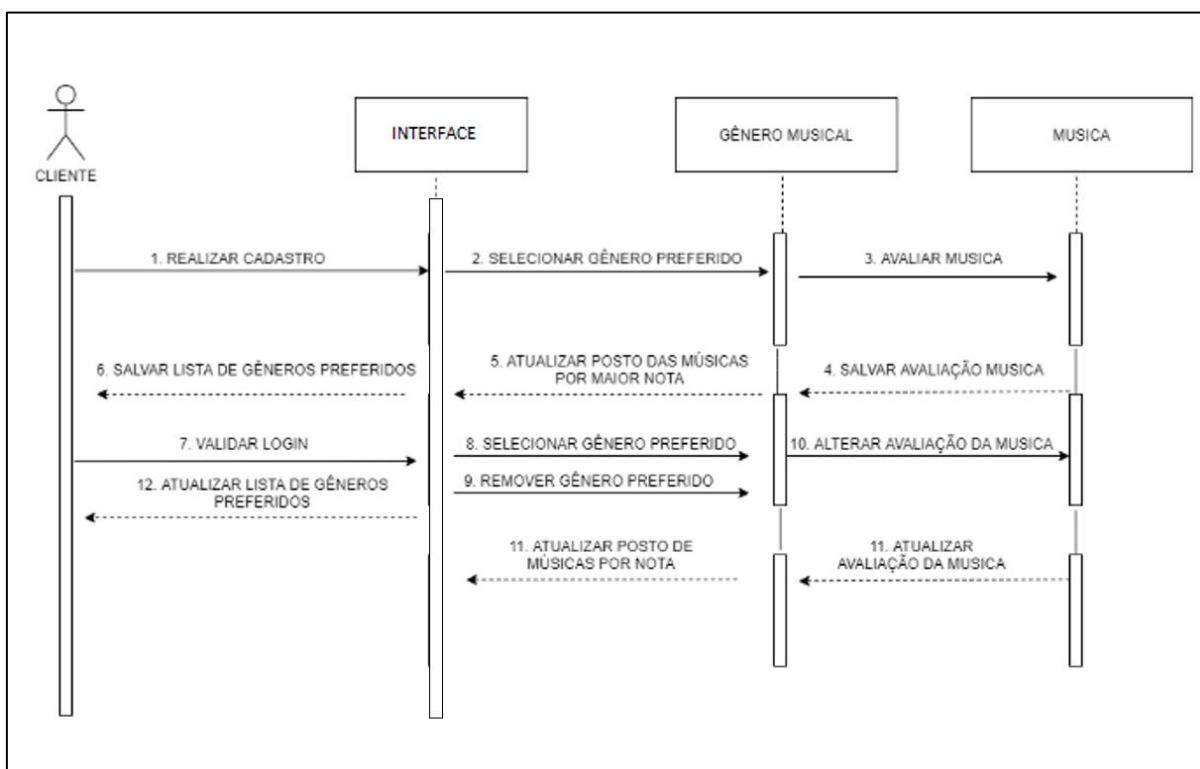


Figura 4 Diagrama de Sequência UML

Desta forma podemos ver o alto nível de abstração na figura 1 com o caso de uso de avaliar músicas e o detalhamento da atividade ainda em um alto nível já que não há o que se falar em programação ainda nesta etapa, mas com uma riqueza de detalhes muito maior.

## 4 DIAGRAMAS DE BANCO DE DADOS

### 4.1 Modelo Conceitual ER do DB

O modelo conceitual do diagrama Entidade Relacionamento (DER) traz de forma visual um esquema para o banco de dados, neste diagrama já é possível ver uma tabela de banco de dados para cada entidade e para cada relacionamento conforme mostrado na figura 5.

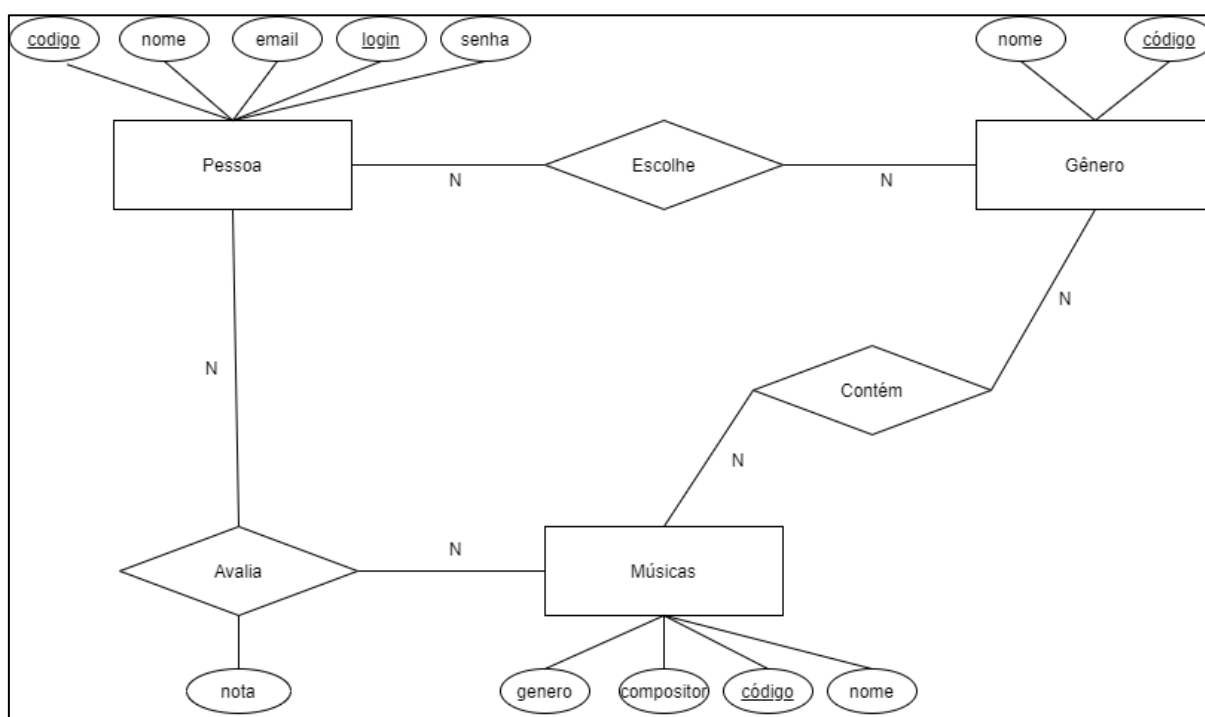


Figura 5 Diagrama Entidade Relacionamento Conceitual)

Neste nível ainda não temos definidas as chaves primárias e estrangeiras dos relacionamentos, mas já existe uma ideia estruturada de como será modelado o banco de dados.

### 4.2 Modelo Lógico Relacional do DB

O modelo Lógico Relacional do diagrama de entidade relacionamento baixa um pouco o nível e já temos algo muito próximo de uma estrutura de banco de dados, podemos ver as chaves primárias e estrangeiras, os relacionamentos entre as tabelas do sistema, através destes relacionamentos se garantida a integridade e unicidade do banco será possível criar suas tabelas e executar instruções CRUD (create read update e delete) no banco de dados para a execução de métodos com as especificações do CRUD o diagrama está exemplificado na figura 6.



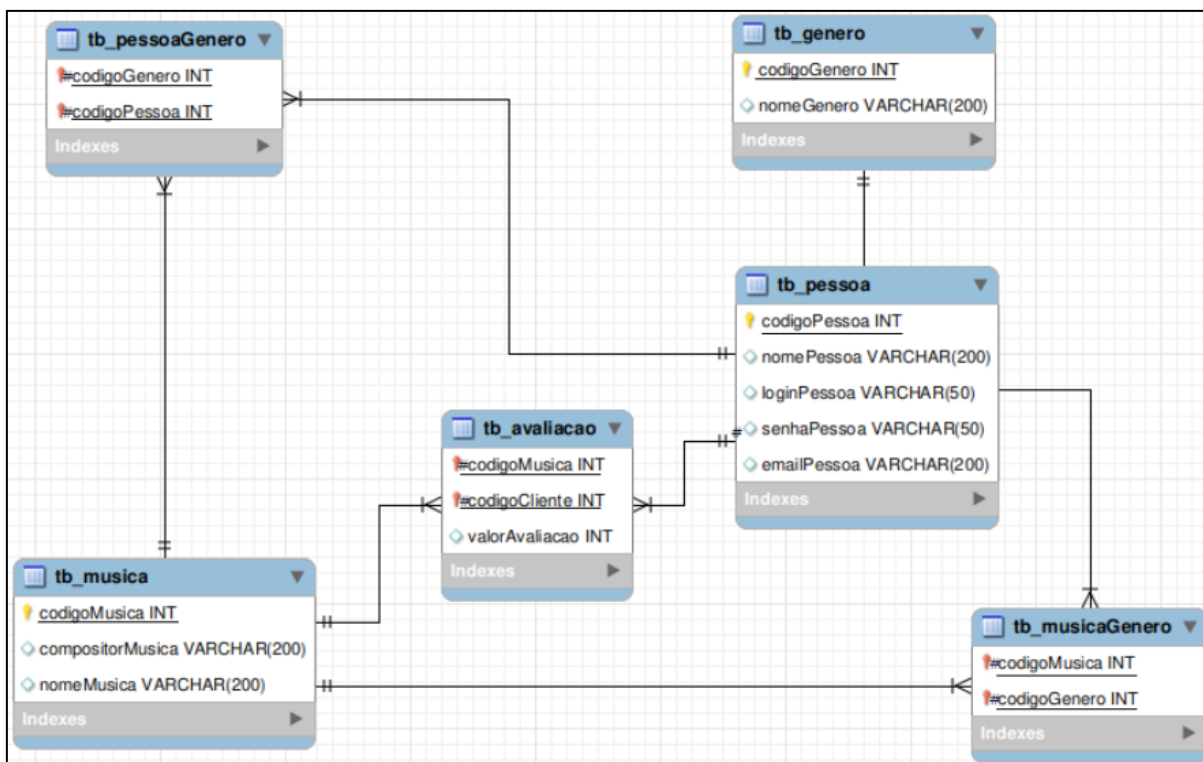


Figura 6 Diagrama Entidade Relacionamento Lógico

As chaves amarelas representam as primary key(chaves primárias das tabelas) também representadas pelo sublinhado, as chaves vermelhas são as foreign keys(chaves estrangeiras do banco de dados) que são informações que vem de uma chave primária de outra tabela para realizar consultas e relacionar dados, também estão representadas pelo símbolo # os aqui também temos o relacionamentos representados de forma gráfica.

## 5 CONCLUSÃO

Se parece difícil para alunos com nenhuma experiência na linguagem Java, pouco ou nenhum contato com programação, lógica, algoritmos, banco de dados e outras ferramentas utilizadas neste projeto seria difícil desenvolver um software que tivesse coerência e atendesse todas as especificações solicitadas em um prazo de apenas pouco mais de um mês, sem o mínimo de organização e padronização seria impossível ou praticamente um fiasco.

Com a modelagem de software baseada na linguagem UML foi possível ter essa organização e padronização, ficou bem claro para os integrantes do grupo como se dá a empatia pelo cliente usuário, ao mesmo tempo que desenvolvemos o projeto, conseguimos ter a visão do usuário através do diagrama de casos de uso feito através de um prévio levantamento de requisitos.

Com o diagrama de classes foi possível dar o pontapé inicial na programação, mesmo ele sofrendo readequações ao longo de todo o projeto durante a fase de implementação, o diagrama de classes tornou possível a equipe sair da estaca zero e iniciar a programação da aplicação, programação é uma atividade prática, não se faz um curso de programação assistindo um vídeo e refletindo os assuntos, se aprende programação implementando e o diagrama de classes é um convite à implementação.

O diagrama de atividades nos ajuda a raciocinar de forma lógica montando praticamente um algoritmo de atividades chave, estimula o raciocínio e nos faz prever erros na lógica que poderiam ser corrigidos com algumas horas de implementação.

Quando falamos do diagrama de sequência é possível perceber como os objetos e telas interagem durante a execução do sistema facilitando a implementação e comunicação entre objeto e interface gráfica além do usuário é claro.

Já nos diagramas ER do banco de dados foi possível planejar toda a estrutura do banco de dados e seus relacionamentos que caso não fossem executados da forma correta traria grandes problemas de execução nas instruções SQL o que comprometeria todos os métodos de inserção, remoção, atualização e consulta do banco, além de comprometer a confiabilidade e unicidade das tuplas de dados.

É possível para um desenvolvedor experiente programar sem uma linguagem de modelagem como a UML? Estes jovens calouros não possuem essa resposta, mas

talvez seja possível, caso ele nunca tenha usado tais ferramentas, mas será impossível para jovens ou não tão jovens alunos pensar em fazer futuramente algum tipo de aplicação sem a utilização deste conjunto de ferramentas que facilitam tanto o trabalho dos profissionais de tecnologia na sua árdua tarefa de impactar a vida de milhares de pessoas através da tecnologia.

Desta forma fica claro que um projeto por menor que seja ao maior que exista deve passar intensamente por diversos rabiscos e bonecos palito para chegar a ser algo grandioso, seja com linguagem UML ou com outra que permita um pré concepção do resultado que se espera atingir futuramente.

## 6 BIBLIOGRAFIA

*69% dos universitários desistem dos cursos de tecnologia, afirma Brasscom.* (09 de Setembro de 2019). Fonte: computerworld: <https://computerworld.com.br/carreira/69-dos-universitarios-desistem-dos-cursos-de-tecnologia-afirma-brasscom/>

Sommerville, I. (2011). *Engenharia de Software*. São Paulo: Prentice Hall.

Vazquez, C. E., & Simões, G. S. (2016). *Engenharia de Requisitos*. Rio de Janeiro: Brasport.