

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLAS DE CIÊNCIAS EXATAS E DA COMPUTAÇÃO
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



APLICAÇÃO PARA *REVIEW* DE FILMES E SÉRIES NA *NETFLIX*

VICTOR HUGO DE LISBOA

GOIÂNIA
2021

VICTOR HUGO DE LISBOA

APLICAÇÃO PARA *REVIEW* DE FILMES E SÉRIES NA *NETFLIX*

Trabalho de Conclusão de Curso apresentado à Escola de Ciências Exatas e da Computação, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Olegário Correa da Silva Neto

GOIÂNIA

2021

VICTOR HUGO DE LISBOA

APLICAÇÃO PARA *REVIEW* DE FILMES E SÉRIES NA *NETFLIX*

Este Trabalho de Conclusão de Curso julgado adequado para obtenção do título de Bacharel em Ciência da Computação , da Pontifícia Universidade Católica de Goiás, em ____/____/____.

Prof. Ma. Nome do coordenador(a) de TCC
Coordenador(a) de Trabalho de Conclusão de
Curso

Banca examinadora:

Orientador: Olegário Correa da Silva Neto

Prof. Me. Nome do membro 1

Prof. Dr. Nome do membro 2

GOIÂNIA
2021

DEDICATÓRIA

Este trabalho é dedicado à aqueles que embora não apareçam diretamente citados neste trabalho serviram de base e apoio para que este fosse realizado. Dedico este trabalho ao meu pai Sr. Wilmar Antônio de Lisboa, e a minha mãe mãe Rosany dos santos costa que entenderam minha ausência e são a razão por não desistir e a força que impulsiona. nossa motivação. Todo meu carinho, respeito e

admiração. Vocês são o motivo pelo qual não existe a palavra desistir em meu dicionário.

AGRADECIMENTOS

Sou grato primeiramente a Deus, que me fortaleceu nos momentos de dificuldades e que permitiu a finalização de mais um objetivo em minha vida.

Sou grato pela honra que é aprender ao lado de colegas que agora se tornaram amigos, que no lugar de competir por conhecimento compartilharam este fazendo com que ao invés de dividir este se multiplicasse.

Sou grato pela sorte de podermos contar com uma equipe de professores que se empenhou em tornar suas aulas fonte de aquisição de conhecimentos e habilidades que não nos permitiram sair da mesma forma que iniciamos este curso.

Sou grato a essa inspiradora fonte de ensino que é a Pontifícia Universidade Católica de Goiás, por ser inovadora e propiciar aos seus discentes um ambiente de ensino capaz de gerar conhecimentos que serão o diferencial no mercado de trabalho.

E meu agradecimento especial ao meu orientador o professor Olegário Correa da Silva Neto que não apenas me endicou o caminho a seguir mas caminhou junto comigo.

Muito Obrigado!

“Um cliente satisfeito é a melhor estratégia de negócios.”
Michael LeBoeuf

RESUMO

Este estudo tem como objetivo desenvolver o aplicativo para dispositivo móvel, denominado Netflix Review Module (NRM), cujo objetivo é permitir o registro e a consulta de reviews sobre os títulos disponibilizados na Netflix. Para atingir tal objetivo foi adotada a metodologia de desenvolvimento incremental que buscou criar novos requisitos, amadurecer e/ou mudar os já existentes. Os resultados deste estudo apresentaram indícios sobre a importância que os softwares de avaliação apresentam para a experiência dos usuários e seu potencial para o aumento de vendas e de engajamento. Usuários que tenham acesso a mais informações sobre determinado produto ou serviço ou mesmo comentários de outros usuários poderão tornar-se mais propensas a compras. O serviço de streaming da Netflix permite aprimorar a experiência do usuário ao procurar por um título para assistir. O Netflix Review Module (NRM) possibilita meios para classificação e comentários além de disponibilizar tais informações para outros usuários do sistema. Trata-se de um recurso adicional que complementa outro sistema já existente.

Palavras-chave: Netflix Review Module. Reviews. Streaming.

ABSTRACT

This study aims to develop the application for mobile device, called Netflix Review Module (NRM), whose objective is to allow the registration and consultation of reviews on titles made available on Netflix. To achieve this goal, the incremental development methodology was adopted, which sought to create new requirements, mature and/or change existing ones. The results of this study provided evidence about the importance that evaluation software has for the user experience and its potential for increasing sales and engagement. Users who have access to more information about a particular product or service or even comments from other users may become more prone to purchases. Netflix's streaming service allows you to improve the user experience when searching for a title to watch. The Netflix Review Module (NRM) provides means for rating and commenting in addition to making such information available to other users of the system. It is an additional feature that complements another existing system.

Keywords: Netflix Review Module. Reviews. Streaming.

LISTA DE ILUSTRAÇÕES

Figura 1 - Exemplo de uma página de um título da <i>Netflix</i>	13
Figura 2 - Tela para escolha de aplicativo que irá receber a intenção de compartilhamento	14
Figura 3 - Diagrama de Contexto	14
Figura 4 - Caso de uso “ <i>Netflix Review Module</i> ”	15
Figura 5 - Caso de uso “Manter URL”	16
Figura 6 - Caso de “Manter <i>Reviews</i> ”	17
Figura 7 - Caso de uso “Manter Usuários”	18
Figura 8 - Modelo Conceitual	22
Figura 9 - Lista de funções documentadas no Swagger	24
Figura 10 - Rota da API para registro de um usuário documentada no Swagger.	25
Figura 11 - Possíveis respostas da rota da API para registro de um usuário documentada no Swagger.	25
Figura 12 - Rota da API para autenticação de usuários no sistema documentada no Swagger	26
Figura 13 - Possíveis respostas da rota da API para autenticação de usuários no sistema documentada no Swagger.	27
Figura 14 - Rota da API para a função de criar ou buscar por um título documentada no Swagger	28
Figura 15 - Possíveis respostas da rota da API para a função de criar ou buscar por um título documentada no Swagger.	28
Figura 16 - Corpo da rota da API para busca da lista de <i>reviews</i> vinculados a um determinado título documentada no Swagger	29
Figura 17 - Possíveis respostas da API para busca da lista de <i>reviews</i> vinculados a um determinado título documentada no Swagger	30
Figura 18 - Rota da API para criar um <i>review</i> em um determinado título documentada no Swagger	31
Figura 19 - Possíveis respostas da rota da API para criar um <i>review</i> em um determinado título documentada no Swagger	31
Figura 20 - Diagrama de atividade sobre o acesso à um título da <i>Netflix</i>	32

Figura 21 - Diagrama de atividades sobre a criação de um <i>review</i> em um título	34
Figura 22 - Sistema de avaliação por estrelas da Google Play	35

LISTA DE SIGLAS

API	<i>Application Programming Interface</i>
id	Identificador
iOS	<i>iPhone operating system</i>
JS	<i>Javascript</i>
JWT	<i>JSON Web Token</i>
NRM	<i>Netflix Review Module</i>
REST	<i>Representational Transfer of Stat</i>
SQL	<i>Secure Sockets Layer</i>
URL	Uniform Resource Locator

SUMÁRIO

1 INTRODUÇÃO	11
1.1 Objetivo Geral	11
1.2 Objetivos Específicos	11
1.3 Metodologia	12
1.4 Escopo	12
2 DESENVOLVIMENTO	13
2.1 Descrição Geral do Produto	13
2.2 Diagrama de Contexto	14
2.3 Características do Usuário	15
2.4 Diagramas de Caso de Uso e suas Descrições	15
2.5 Requisitos Funcionais	18
2.6 Requisitos não Funcionais	19
2.6.1 <i>Requisitos de Usabilidade</i>	19
2.6.2 <i>Requisitos de Plataforma de Software</i>	20
2.6.3 <i>Requisitos de Segurança</i>	20
2.7 Projeto de Software	21
2.8 Modelo Conceitual	21
2.9 Documentação da API REST	22
2.9.1 <i>Função de registrar um usuário</i>	24
2.9.2 <i>Função de autenticação</i>	26
2.9.3 <i>Função de criar ou buscar por um título</i>	27
2.9.4 <i>Função de buscar por reviews</i>	29
2.9.5 <i>Função de criar um review</i>	30
2.10 Diagramas de Atividade	32
2.10.1 <i>Acessar uma página de reviews de um determinado título da Netflix</i>	32
2.10.2 <i>Adicionar um review à um título</i>	33
Contribuições Futuras	35
3 CONCLUSÃO	36
REFERÊNCIAS	37

1 INTRODUÇÃO

Existem diversas plataformas de *streaming* no mercado que oferecem diversos tipos de serviços. A *Netflix* é um exemplo, dedicada ao *streaming* de vídeos com séries, filmes e documentários. Seu conteúdo é vasto e indexado pelos respectivos títulos, sinopse, tipo (série ou filme), ano e censura. Além destas informações os títulos também contém relevância, que refere-se a uma taxa em porcentagem relativa ao julgamento dos usuários que já assistiram àquele vídeo.

Nesta plataforma a busca de um título é realizada através de palavras chave mencionadas pelo usuário, e seu resultado é ordenado pelo nível de coincidência das respectivas palavras chave com o título e sinopse. Apesar de eficiente, esta técnica não é suficiente para a avaliação subjetiva realizada pelo usuário. Algumas plataformas permitem aos seus usuários, não somente realizar um julgamento em escalas (de zero a cinco estrelas), mas também emitir uma opinião que será compartilhada com os outros usuários da plataforma. Esta forma de julgamento é muito utilizada pelos usuários como um dos critérios de seleção de vídeos.

Segundo *Netflix* (2020), um usuário pode classificar uma série e um filme com sinais de aprovação e desaprovação, mas avaliar um conteúdo só contribui para o algoritmo de recomendações de outros conteúdos para o usuário. No serviço da *Netflix* não existe um espaço dedicado para discussões entre usuários.

Sendo assim, este trabalho propõe o desenvolvimento de um aplicativo para celular que permita aos usuários da *Netflix* realizar um julgamento em duas escalas (“gostei” e “não gostei”) e registrar sua respectiva opinião em forma de texto.

1.1 Objetivo Geral

Desenvolver o aplicativo para dispositivo móvel, denominado *Netflix Review Module* (NRM), cujo objetivo é permitir o registro e a consulta de *reviews* sobre os títulos disponibilizados na *Netflix*.

1.2 Objetivos Específicos

Criar uma aplicação que:

- Adicione títulos da *Netflix* compartilhados por usuários no banco de dados;
- Permita que usuários adicionem *reviews* sobre os filmes e séries disponibilizados na *Netflix* através de um sistema de classificação por “gostei” e “não gostei”, com a possibilidade de adicionar um comentário que complemente seu *review*; Busque por *reviews* vinculados ao título da *Netflix* compartilhado pelo usuário;
- Cadastre usuários;
- Autentifique usuários;

1.3 Metodologia

A implementação da aplicação proposta foi realizada utilizando um desenvolvimento incremental. Segundo Sommerville (2011) “O desenvolvimento incremental é baseado na ideia de desenvolver uma implementação inicial, expô-la aos comentários dos usuários e continuar por meio da criação de várias versões até que um sistema adequado seja desenvolvido”. Devido a natureza deste trabalho se tratar de uma ideia e uma aplicação pessoal, a implementação inicial não foi exposta a nenhum usuário em potencial, porém, essa implementação foi utilizada para criar novos requisitos e amadurecer ou mudar os já existentes.

1.4 Escopo

O *Netflix Review Module* (NRM) é uma aplicação mobile para o sistema operacional Android que permite adicionar meios de classificação e comentários sobre os conteúdos disponibilizados na *Netflix*, e possibilitar a visualização dos mesmos pelos usuários da plataforma. O nome dado a aplicação é um nome de desenvolvimento, e recebe a palavra “*module*” (módulo) porque se trata de uma aplicação que complementa a funcionalidade de outro sistema (WAZLAWICK, 2014).

2 DESENVOLVIMENTO

2.1 Descrição Geral do Produto

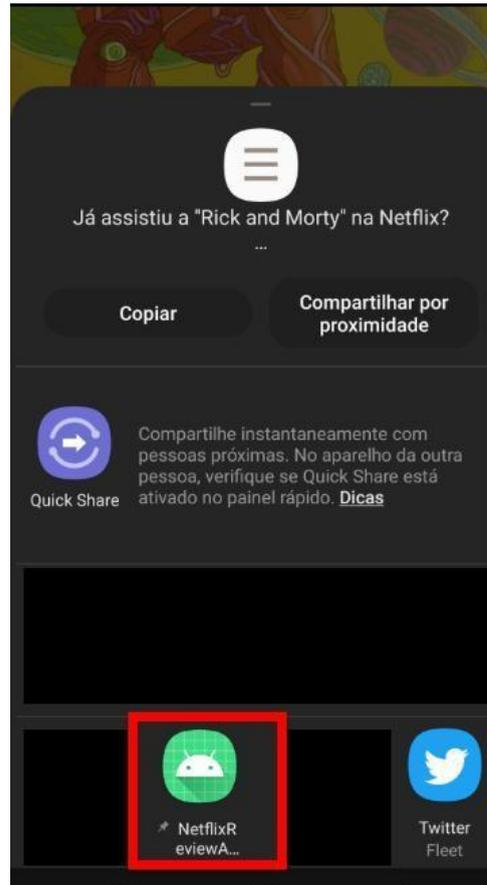
O sistema consiste em duas partes, uma *Application Programming Interface* (API) que persiste em um banco de dados, buscando informações como *reviews* vinculados a um título e inserindo informações como novos usuários, e uma aplicação mobile Android que será a interface que o usuário utilizará para acessar os recursos disponibilizados por essa API. Para usar o NRM primeiramente o usuário deve localizar o título desejado no aplicativo mobile da *Netflix* e, em seguida, localizar e clicar no botão “compartilhar” (figura 1) e escolher o NRM como opção de compartilhamento (figura 2), desta forma o usuário tem acesso à página de *reviews* daquele determinado título.

Figura 1 - Exemplo de uma página de um título da *Netflix*



Fonte: Autoria Própria

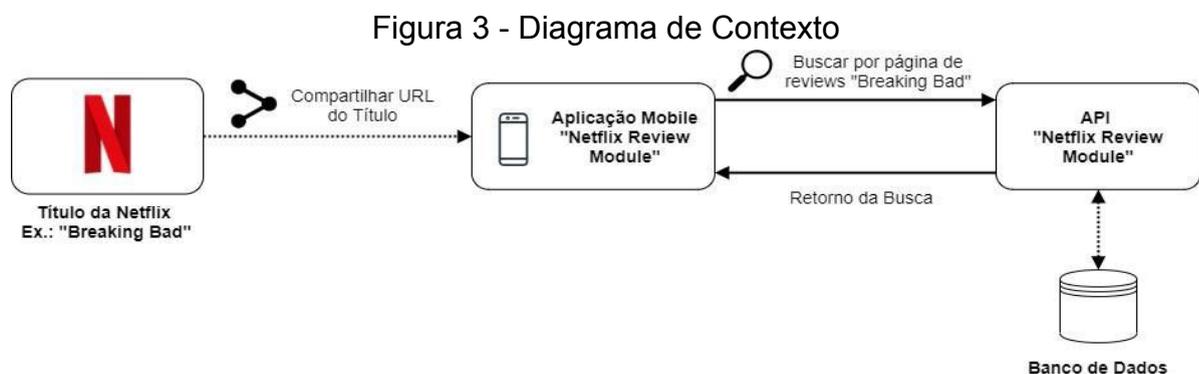
Figura 2 - Tela para escolha de aplicativo que irá receber a intenção de compartilhamento



Fonte: Autoria Própria

2.2 Diagrama de Contexto

O diagrama a seguir (figura 3) ilustra o funcionamento geral da aplicação, ao compartilhar um título da *Netflix* com o NRM, a aplicação faz uma busca no banco de dados para retornar informações sobre o título compartilhado e os *reviews* vinculados a ele.



Fonte: Autoria Própria

2.3 Características do Usuário

O sistema tem como público alvo usuários ativos do serviço de streaming da *Netflix* que fazem uso do aplicativo para celular da mesma, já que a proposta da aplicação é aprimorar a experiência do usuário ao procurar por um título para assistir.

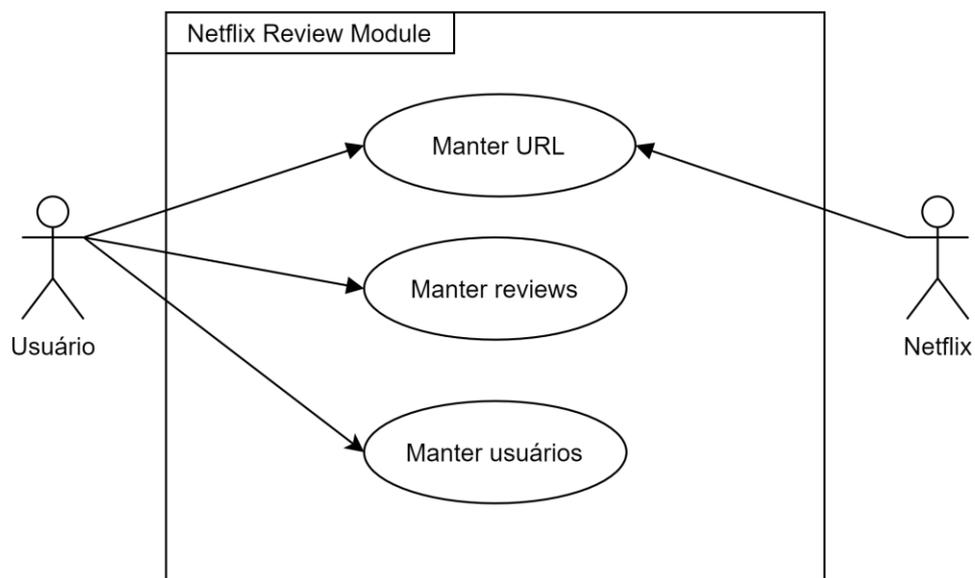
2.4 Diagramas de Caso de Uso e suas Descrições

Netflix Review Module

Atores: Usuário e *Netflix*

Descrição: Este caso de uso representa o sistema geral, que consiste em manter URL, manter *reviews* e manter usuários (figura 4).

Figura 4 - Caso de uso “*Netflix Review Module*”



Fonte: Autoria Própria

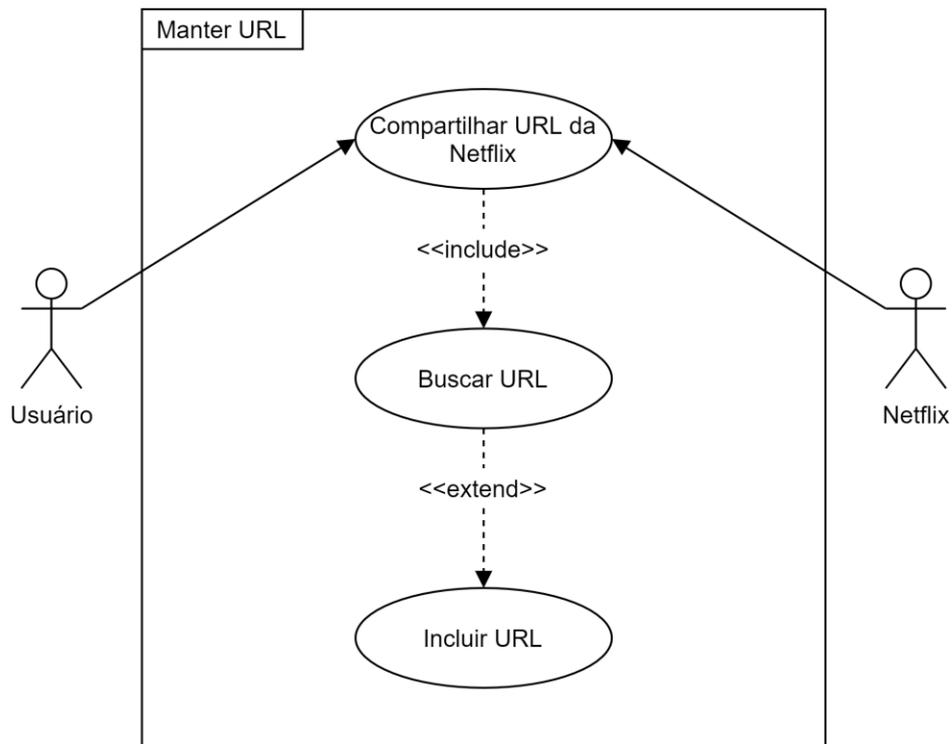
Manter URL

Atores: Usuário e *Netflix*

Descrição: Este caso de uso acontece quando um usuário compartilha uma URL de um título da *Netflix* com a aplicação NRM com o objetivo de acessar a página de *reviews* daquele título. O sistema irá buscar pela URL no banco de dados,

caso encontrado o sistema retorna a página do título em questão com todos os *reviews* atrelados a ela. Caso esta URL não seja encontrada, ela será adicionada ao banco de dados e a página do título recém criada é retornada ao usuário com 0 *reviews* atrelados a ela (figura 5).

Figura 5 - Caso de uso “Manter URL”



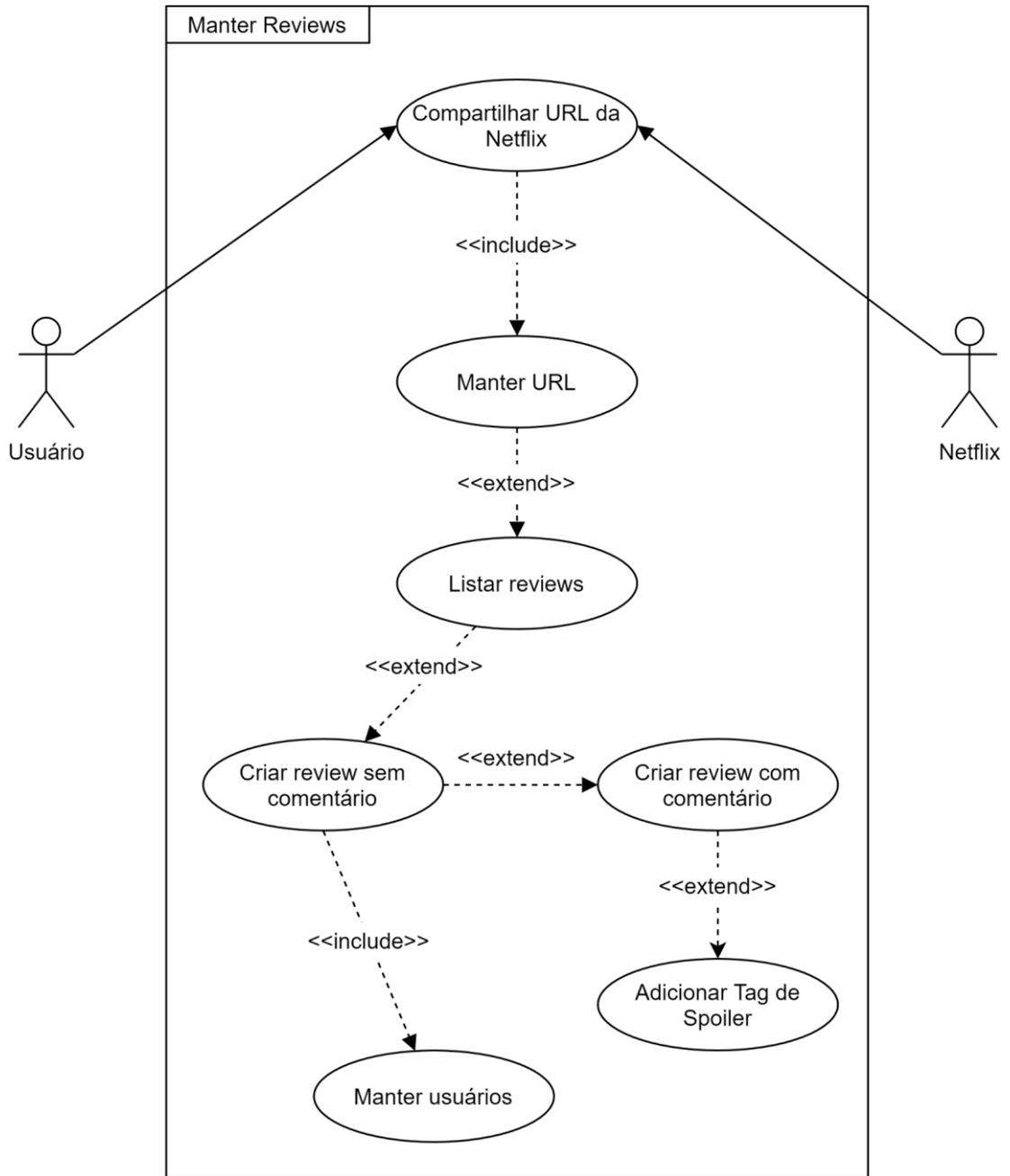
Fonte: Autoria Própria

Manter *Reviews*

Atores: Usuário e *Netflix*

Descrição: Este caso de uso acontece após um usuário compartilhar uma URL de um título e sistema busca ou adiciona a URL no banco de dados, dessa forma o usuário tem acesso a página de *reviews* daquele título e tem a opção de criar um *review*, porém, para criar um *review* o usuário obrigatoriamente deve estar logado. Ao criar um *review* o usuário escolhe entre “gostei” ou “não gostei” (“recomendo” ou “não recomendo”) e tem a opção de, se quiser, adicionar um comentário à avaliação e adicionar uma tag de *spoiler* ao comentário (figura 6).

Figura 6 - Caso de “Manter *Reviews*”



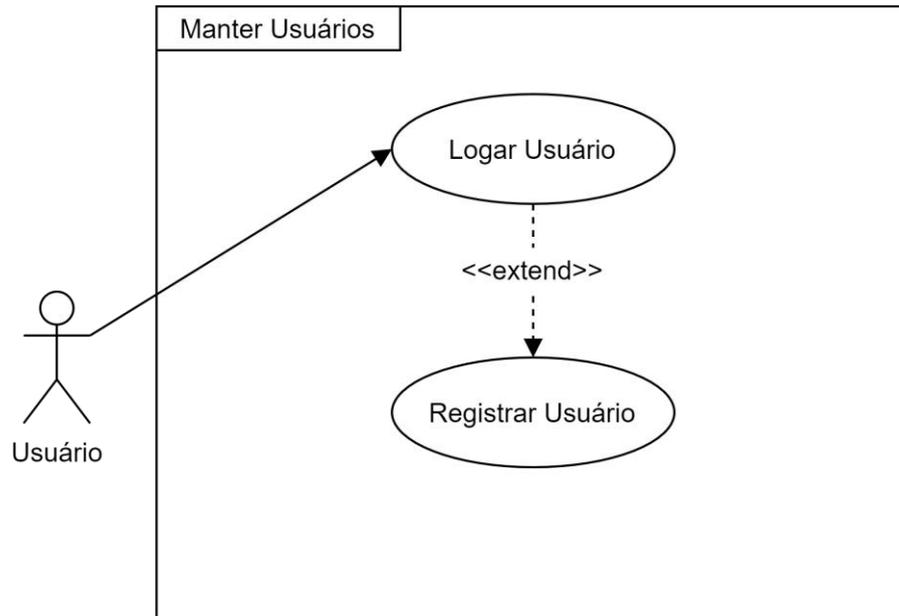
Fonte: Autoria Própria

Autenticar Usuário

Atores: Usuário

Descrição: Este caso de uso ocorre quando um usuário decide logar com email e senha para adicionar um *review* à uma página, caso o usuário não tenha uma conta ele pode criar uma com nome de usuário, email e senha (figura 7).

Figura 7 - Caso de uso “Manter Usuários”



Fonte: Autoria Própria

2.5 Requisitos Funcionais

Título: Autenticar usuário

Descrição: Um usuário deve poder efetuar login utilizando o email e senha.

Título: Registrar usuário

Descrição: Um usuário deve poder se registrar no sistema utilizando um nome de usuário, email e senha.

Título: Deslogar um usuário

Descrição: Um usuário deve poder se deslogar do sistema.

Título: Botão de atalho para a *Netflix* na tela de apresentação

Descrição: A aplicação deve conter um botão que redireciona o usuário para a aplicação da *Netflix* na tela de apresentação.

Título: Visualizar lista de *reviews* de um título da *Netflix*

Descrição: Ao compartilhar um título da *Netflix* com a aplicação, o usuário tem acesso a lista de *reviews* do título em questão.

Título: Criar um *review* (avaliação “gostei” ou “não gostei”)

Descrição: Um usuário, se logado, deve poder adicionar um *review* à um título da *Netflix*.

Título: Adicionar um comentário ao *review*

Descrição: Ao criar uma avaliação, o usuário tem a opção de adicionar um comentário para detalhar o porquê ele gostou ou não daquele título.

Título: Marcar um comentário com uma tag de *spoiler*

Descrição: Ao adicionar um comentário ao *review* um usuário pode marcar o comentário com uma tag de *spoiler*, alertando ao outros usuários que no seu comentário possui revelações do enredo de determinado título da *Netflix*.

Título: Atualizar a página de *reviews*

Descrição: O usuário deve poder atualizar a página de *reviews* de um título da *Netflix* para acessar possíveis novos *reviews*.

Título: Informações de uma página de um título da *Netflix*

Descrição: Uma página de um título da *Netflix* deve apresentar o número total de *reviews* vinculados à página, número de *reviews* positivos, número de *reviews* negativos e ilustrar a porcentagem de *reviews* positivos em relação ao total.

2.6 Requisitos não Funcionais

2.6.1 Requisitos de Usabilidade

Título: “*Shimmer Effect*”

A aplicação deve apresentar o chamado “*Shimmer Effect*” quando o usuário está buscando por uma página de *reviews*. “*Shimmer Effect*” é o nome dado ao efeito visual que indica um estado de carregamento de forma discreta.

Título: Notificações “*Toast*”

A aplicação deve passar mensagens de sucesso ou falha de uma ação na aplicação através de notificações “*toast*”. “Uma notificação “*toast*” fornece um *feedback* simples sobre uma operação em uma pequena janela *pop-up*. Ele só ocupa a quantidade de espaço necessária para a mensagem. Notificações *toast* desaparecem automaticamente após um tempo limite”. (DEVELOPER, 2021, *on-line*).

Título: Campo de inserção de senha

O campo de inserção de senha deve esconder os caracteres inseridos substituindo-os por pontos.

Título: Botões sensíveis aos campos de inserção

Os botões para realização de cadastro, autenticação e criação de *review* devem permanecer desativados até que todos os campos necessários para realização da ação sejam preenchidos.

2.6.2 Requisitos de Plataforma de Software

Ferramentas para o desenvolvimento da API:

JavaScript (Linguagem de programação)

Node JS (Ambiente de execução Javascript)

postgreSQL (Banco de dados relacional)

Ferramentas para o desenvolvimento do aplicativo *Android*:

Android Studio (Ambiente de desenvolvimento de aplicações *Android*)

Kotlin (Linguagem de programação)

2.6.3 Requisitos de Segurança

Título: JWT (JSON Web Token) como ferramenta para autenticar usuários.
 Descrição: O login do usuário deve ser realizado através do JWT. “O JWT é um método para realizar autenticação entre duas partes por meio de um token assinado que autentica uma requisição web.” A única funcionalidade do sistema que é protegida, ou seja, que exige a autenticação do usuário, é a de “Criar um *Review*”. Para criar um *review* o usuário precisa obter um JWT válido logando no sistema (DEV MEDIA, 2021).

Título: Salvar senhas no banco de dados

Descrição: Ao criar um usuário, a senha deve ser criptografada. Senhas são informações sensíveis, portanto é importante criptografá-las antes de salvá-las no banco de dados, para caso a confidencialidade do banco seja comprometida.

2.7 Projeto de Software

2.8 Modelo Conceitual

A aplicação possui três entidades com seus respectivos atributos e relacionamentos como ilustra a figura 8.

As entidades são:

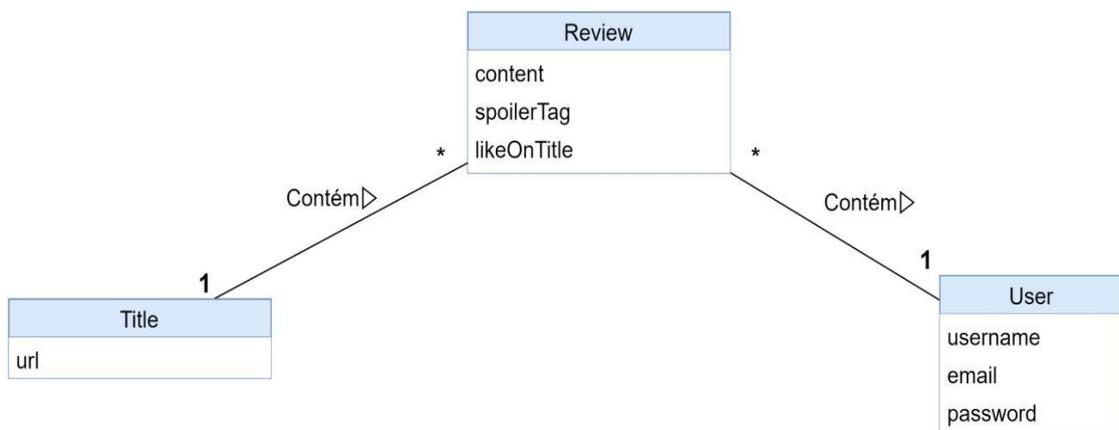
- “*Title*”: Representa um título da *Netflix*, composto pelo atributo “url” que guarda uma URL de um título da *Netflix*.
- “*Review*”: Representa um *review*, composto pelos atributos “*content*” representando o comentário textual do *review*, “*spoilerTag*” que indica se o

comentário possui revelações de enredo e *“likeOnTitle”* que indica se o *review* é positivo (“gostei”) ou negativo (“não gostei”).

- *“User”*: Representa um usuário do sistema, composto pelos atributos *“username”* que representa o nome de usuário, *“email”* que representa o email do usuário e *“password”* que representa a senha do usuário.

As três entidades se relacionam entre si, um título contém de zero a inúmeros *reviews*, mas um *review* está atrelado a somente um título. Um *review* está atrelado a um usuário, mas um usuário pode ter de zero a inúmeros *reviews*.

Figura 8 - Modelo Conceitual



Fonte: Autoria Própria

2.9 Documentação da API REST

Segundo Massé (2011) em uma definição geral, uma API expõe um conjunto de dados e funções para facilitar as interações entre programas de computador e permitir que eles troquem informações.

REST é um estilo arquitetural comumente aplicado no design de APIs. Uma API em conformidade com o estilo de arquitetura REST é uma API REST (MASSÉ, 2011).

APIs REST utilizam verbos para definir qual é a finalidade da requisição que está sendo enviada (HOSTGATOR, 2019).

Os dois verbos utilizados na API REST deste trabalho são:

POST: Tipo de requisição utilizada para criar um recurso em uma determinada API (HOSTGATOR, 2019).

GET: Indica que a requisição é um pedido de dados para a API (HOSTGATOR, 2019).

“APIs REST também utilizam códigos de status em suas respostas para indicar se uma requisição foi corretamente concluída” (MOZILLA, 2021).

Os códigos de status utilizados neste trabalho foram:

200 (“OK”): Indica que a requisição foi bem sucedida. (MOZILLA, 2021).

201 (“Created”): Indica que a A requisição foi bem sucedida e um novo recurso foi criado como resultado (MOZILLA, 2021).

400 (“Bad Request”): Indica que o servidor não entendeu a requisição pois está com uma sintaxe inválida (MOZILLA, 2021).

401 (“Unauthorized”): Indica que o cliente deve se autenticar para obter a resposta solicitada (MOZILLA, 2021).

A documentação da API foi feita utilizando o Swagger. “O Swagger é uma ferramenta para documentação de APIs REST. O documento do Swagger especifica a lista de recursos que estão disponíveis na API e as operações que podem ser chamadas nesses recursos.” (IBM, 2020).

A figura 9 ilustra todas as cinco funções que são disponibilizadas pela API, documentadas no Swagger.

Figura 9 - Lista de funções documentadas no Swagger

POST	<code>/user/login</code>	Autenticar um usuário	✓ ↩
POST	<code>/user/register</code>	Registrar um novo usuário	✓ ↩
POST	<code>/title</code>	Criar ou buscar por uma página de um título da Netflix	✓ ↩
GET	<code>/review/title/{titleId}</code>	Buscar a lista de reviews	✓ ↩
POST	<code>/review/title/{titleId}</code>	Criar um review em um determinado título	✓ ↩

Fonte: Autoria Própria

2.9.1 Função de registrar um usuário

A função de registrar um novo usuário utiliza os dados de email, nome de usuário e senha como o corpo da requisição como ilustrado na figura 10, tendo como possíveis respostas os códigos 200 (“OK”) para um cadastro realizado com sucesso e 400 (“*Bad Request*”) para a ocorrência de um erro na realização de um cadastro, ilustradas na figura 11.

Figura 10 - Rota da API para registro de um usuário documentada no Swagger.

POST /user/register Registrar um novo usuário

Registrar um novo usuário utilizando o email, username e senha.

Parameters Try it out

Name	Description
body * required object (body)	Corpo do usuário a ser criado

Example Value | Model

```
{
  "email": "string",
  "username": "string",
  "password": "string"
}
```

Parameter content type
application/json

Fonte: Autoria Própria

Figura 11 - Possíveis respostas da rota da API para registro de um usuário documentada no Swagger.

Responses Response content type application/json

Code	Description
200	OK
400	Bad Request

Fonte: Autoria Própria

2.9.2 Função de autenticação

A autenticação de um usuário é realizada fornecendo o e-mail e a senha como corpo da requisição ilustrado na figura 12. Caso as credenciais estejam corretas a API retorna o código 200 (“OK”) com um token de acesso no qual o usuário poderá utilizar para acessar recursos protegidos. Caso as credenciais estejam incorretas a API retorna um erro 400 (“*Bad Request*”) indicando que o corpo da requisição é inválido. Os dois possíveis retornos estão ilustrados na figura 13.

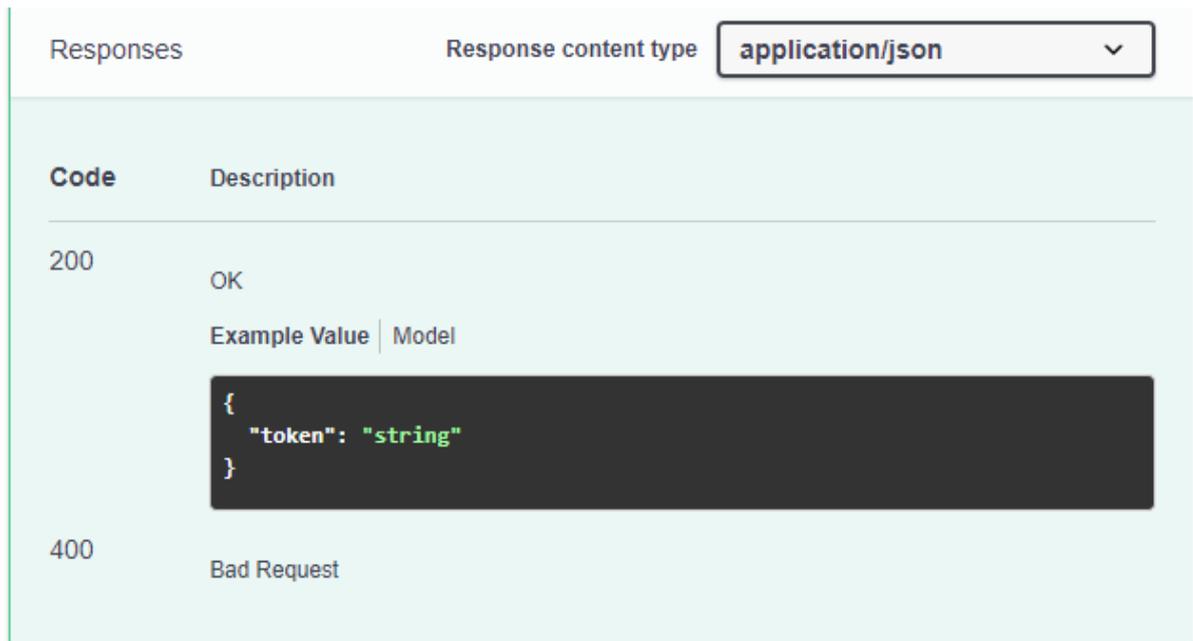
Figura 12 - Rota da API para autenticação de usuários no sistema documentada no Swagger

The image shows a Swagger API documentation interface for a POST endpoint. At the top, it says 'POST /user/login Autenticar um usuário'. Below this, there is a 'Parameters' section with a 'Try it out' button. The main part of the interface is a table with 'Name' and 'Description' columns. The first entry is 'body * required', which is described as 'Corpo do usuário a ser autenticado com email e senha'. Below this, there is an 'Example Value' field containing a JSON object: { "email": "string", "password": "string" }. At the bottom, there is a 'Parameter content type' dropdown menu set to 'application/json'.

Name	Description
body * required object (body)	Corpo do usuário a ser autenticado com email e senha Example Value Model <pre>{ "email": "string", "password": "string" }</pre> Parameter content type application/json

Fonte: Autoria Própria

Figura 13 - Possíveis respostas da rota da API para autenticação de usuários no sistema documentada no Swagger.



The image shows a Swagger API documentation interface. At the top, there is a 'Responses' section with a 'Response content type' dropdown menu set to 'application/json'. Below this, a table lists the response codes and their descriptions. The first entry is for status code 200, described as 'OK', with an example JSON response: `{ "token": "string" }`. The second entry is for status code 400, described as 'Bad Request'.

Code	Description
200	OK Example Value Model <pre>{ "token": "string" }</pre>
400	Bad Request

Fonte: Autoria Própria

2.9.3 Função de criar ou buscar por um título

Na função de criar ou buscar por um título da *Netflix* no banco de dados, o corpo da requisição é a URL de um título da *Netflix* ilustrado na figura 14, a API faz uma busca por essa URL no banco de dados e, caso a busca seja bem sucedida, a API retorna o código 200 (“OK”) com o corpo do título encontrado, que é composto por id, url, número de *reviews*, número de likes e número de dislikes. Caso a busca seja mal sucedida a API adiciona a URL ao banco de dados e retorna o código 201 (“Created”) com o corpo do novo título adicionado que consiste em id, url, número de *reviews*, likes e dislikes. Quando a API adiciona um novo título ao banco o número de *reviews*, likes e dislikes será por padrão zero, porque como o título acabou de ser criado não existem *reviews* vinculados a ele. Os dois possíveis retornos estão ilustrados na figura 15.

Figura 14 - Rota da API para a função de criar ou buscar por um título documentada no Swagger

POST /title Criar ou buscar por uma página de um título da Netflix

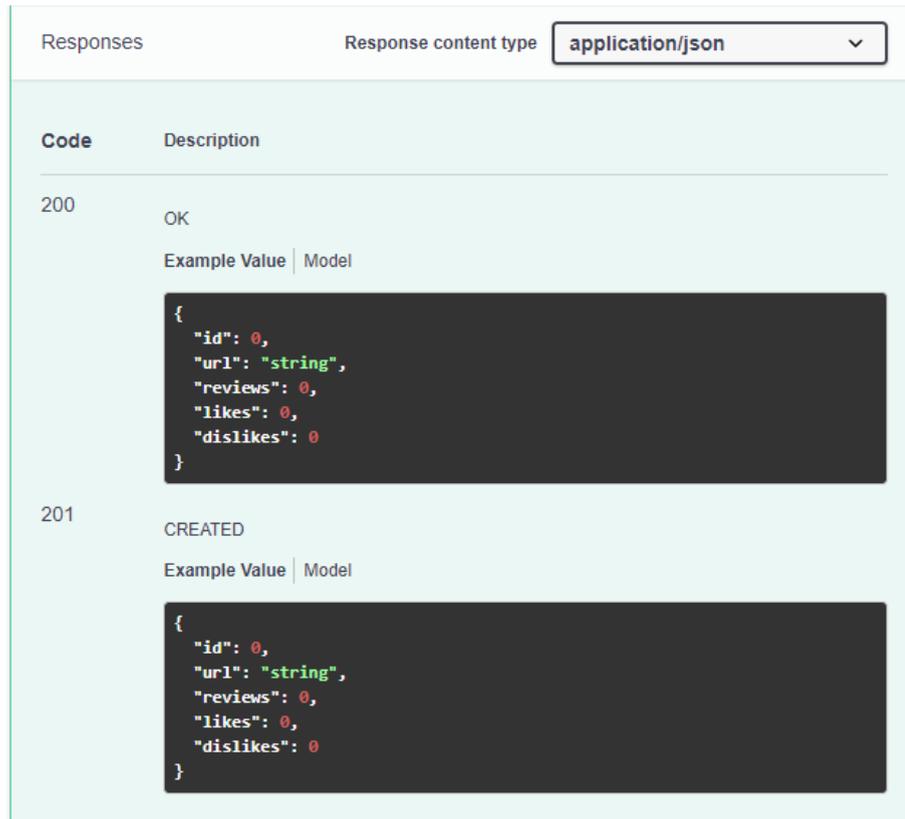
Criar ou buscar por uma página de um título da Netflix no banco de dados a partir de uma URL.

Parameters Try it out

Name	Description
body * required	Corpo da requisição
object (body)	Example Value Model
	<pre>{ "url": "string" }</pre>
Parameter content type	application/json

Fonte: Autoria Própria

Figura 15 - Possíveis respostas da rota da API para a função de criar ou buscar por um título documentada no Swagger.



Fonte: Autoria Própria

2.9.4 Função de buscar por reviews

A função de buscar por *reviews* retorna todos os *reviews* atrelados ao título em questão. Esta busca é realizada fornecendo o “id” (identificador) do título como ilustra a figura 16. O retorno da requisição é o código 200 (“OK”) com uma lista de *reviews* ilustrada na figura 17. Cada *review* retornado contém os atributos “id” (identificador), “username” (nome do usuário), “content” (conteúdo textual do *review*), “likeOnTitle” (indica se o *review* é positivo ou negativo) e “spoilerTag” (indica se o comentário possui revelações de enredo)

Figura 16 - Corpo da rota da API para busca da lista de *reviews* vinculados a um determinado título documentada no Swagger

GET /review/title/{titleId} Buscar a lista de reviews

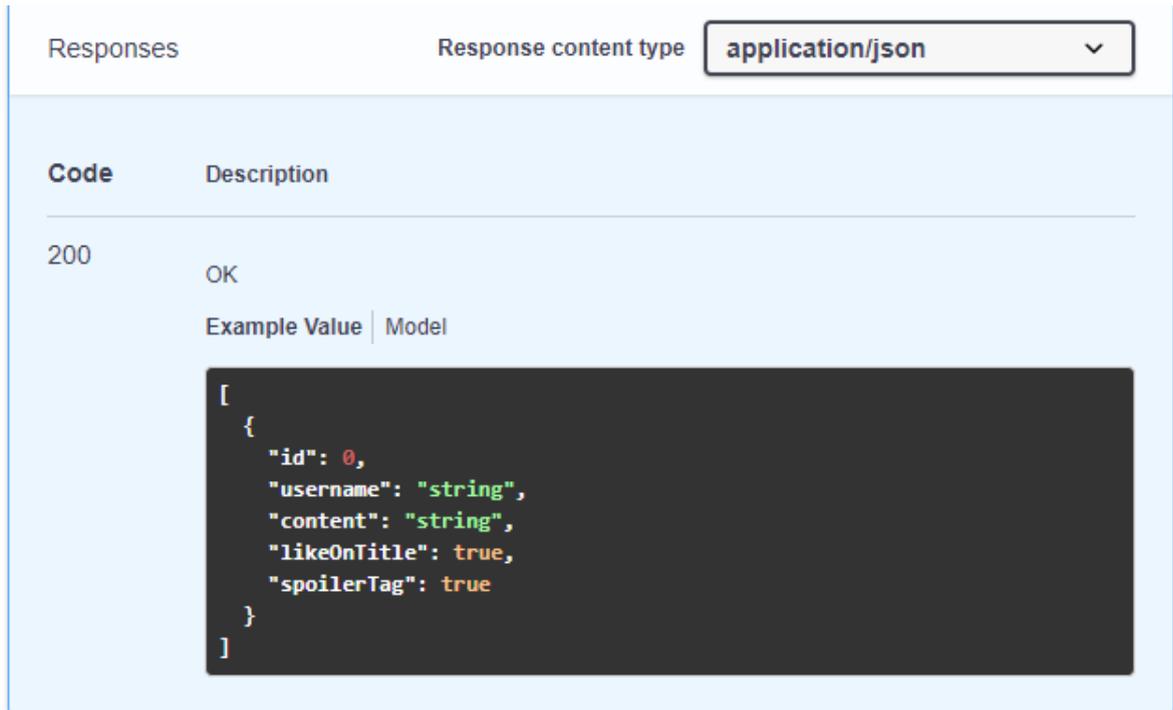
Buscar a lista de reviews de um determinado título da Netflix

Parameters Try it out

Name	Description
titleId * required	"titleId" é o identificador de um título da Netflix no banco de dados
string (path)	<input type="text" value="titleId"/>

Fonte: Autoria Própria

Figura 17 - Possíveis respostas da API para busca da lista de *reviews* vinculados a um determinado título documentada no Swagger



Responses Response content type application/json

Code	Description
200	OK

Example Value | Model

```
[
  {
    "id": 0,
    "username": "string",
    "content": "string",
    "likeOnTitle": true,
    "spoilerTag": true
  }
]
```

Fonte: Autoria Própria

2.9.5 Função de criar um review

Na função de criar um *review* em um determinado título, o usuário deve obrigatoriamente estar autenticado na aplicação e enviar seu token de acesso e o identificador do título em questão junto à requisição além do corpo como ilustra a figura 18.

O corpo da requisição é composto por “*likeOnPage*” (“gostei” ou “não gostei”), “*content*” (comentário do *review*) e “*spoilerTag*” (tag que alerta se o comentário contém revelações sobre o enredo). Ao criar um *review*, a única informação obrigatória que o usuário deve fornecer é a “*likeOnPage*”, um *review* não necessariamente deve possuir um comentário. As possíveis respostas da requisição ilustradas na figura 19 são 200 (“OK”) para um *review* criado com sucesso e 401 (“*Unauthorized*”) como um erro para usuários que acessam o recurso sem estarem devidamente autenticados.

Figura 18 - Rota da API para criar um *review* em um determinado título documentada no Swagger

POST /review/title/{titleId} Criar um review em um determinado título

Criar um review em um determinado título, isso só pode ser feito com um usuário logado

Parameters Try it out

Name	Description
body * required object (body)	Corpo do review a ser criado Example Value Model <pre>{ "content": "string", "likeOnTitle": true, "spoilerTag": true }</pre> Parameter content type application/json
titleId * required string (path)	"titleId" é o identificador de um título da Netflix no banco de dados titleId

Fonte: Autoria Própria

Figura 19 - Possíveis respostas da rota da API para criar um *review* em um determinado título documentada no Swagger

Responses Response content type application/json

Code	Description
200	OK
401	Unauthorized

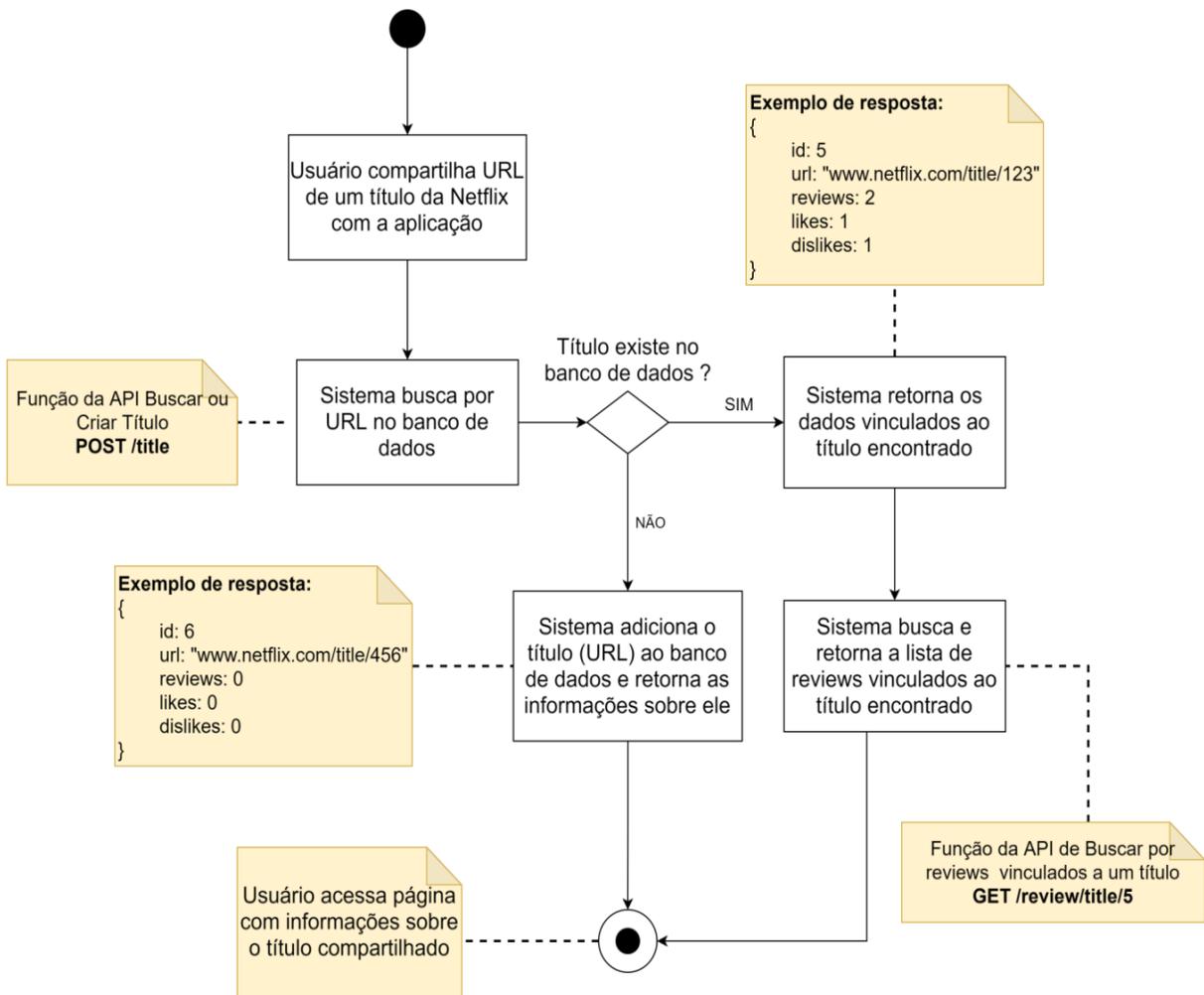
Fonte: Autoria Própria

2.10 Diagramas de Atividade

2.10.1 Acessar uma página de reviews de um determinado título da Netflix

A figura a seguir (20) ilustra o funcionamento da aplicação no acesso a uma página de *reviews*. Para acessar a página de um título o usuário primeiramente deve compartilhar um título da *Netflix* com a aplicação NRM, após o compartilhamento a aplicação acessa a função de busca ou criação de um título, caso o título seja encontrado o sistema retorna os dados do mesmo e busca por *reviews* vinculados a ele. Caso o título não seja encontrado ele é adicionado ao banco e seus dados são retornados. Quando um título é recém criado, o sistema não passa pela função de buscar *reviews* atrelados a um título.

Figura 20 - Diagrama de atividade sobre o acesso à um título da *Netflix*

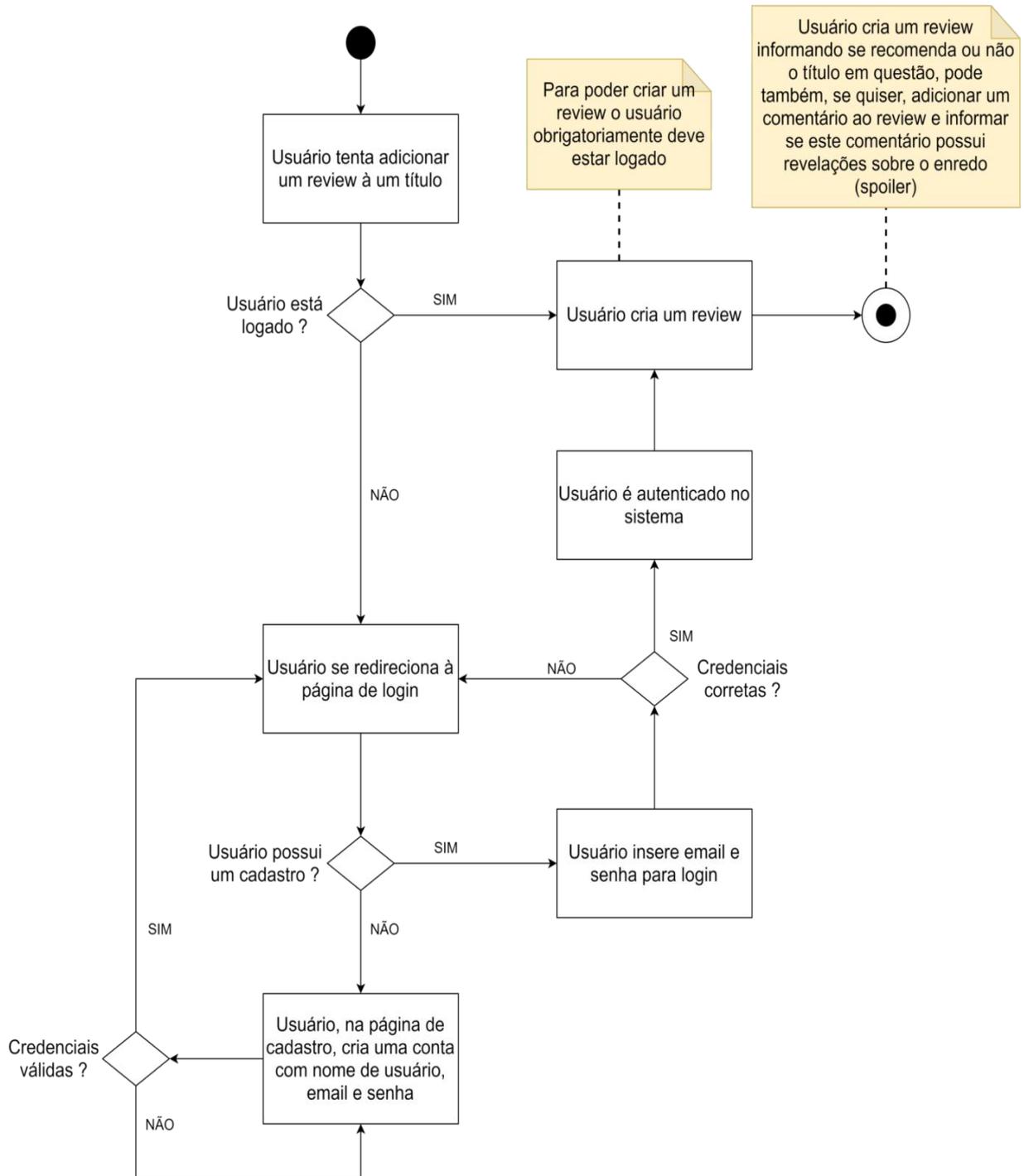


Fonte: Autoria Própria

2.10.2 Adicionar um review à um título

A figura 21 ilustra o funcionamento da aplicação na criação de um novo *review*. Para que um usuário possa adicionar um *review* à um título ele deve estar autenticado, caso ele não esteja a aplicação o informará na tentativa de criar um *review*. Ao se direcionar à página de login, o usuário tem a opção de se direcionar à página de registro para caso ele não possua um cadastro. Para se cadastrar o usuário deve informar um email, nome de usuário e senha válidos. Assim que um usuário estiver devidamente cadastrado no sistema e autenticado, ele pode criar um *review*. Caso o usuário insira credenciais inválidas o sistema retorna o erro 400 (“*Bad Request*”) e o aplicativo notifica o usuário através do “*toast*”.

Figura 21 - Diagrama de atividades sobre a criação de um *review* em um título



Fonte: Autoria Própria

Contribuições Futuras

Porte para iOS: Para desenvolver aplicações para o sistema operacional iOS é necessário um computador com o sistema operacional da Apple (MacOS), como foi utilizado um computador Windows para a realização deste projeto não foi possível considerar o porte para aparelhos iOS.

Aprimoramento do sistema de *Review*: Atualmente um usuário pode avaliar um título da *Netflix* com um sinal de “gostei” ou “não gostei” porém um mecanismo de estrelas como o sistema de avaliação de aplicativos da Google Play (figura 22) é o ideal. Um sistema de estrelas apresenta uma escala maior de opções com 1 estrela sendo péssimo, 2 estrelas sendo ruim, 3 estrelas sendo mediano, 4 estrelas sendo bom e 5 estrelas sendo ótimo.

Figura 22 - Sistema de avaliação por estrelas da Google Play



Fonte: Print screen da aplicação da Google Play

3 CONCLUSÃO

O desenvolvimento da aplicação deste projeto possibilitou uma análise de como uma adição de um função em um software já existente pode complementar a experiência de um usuário.

No comércio eletrônico, a função de *reviews* é uma parte crítica por aumentar vendas e engajamento.

Funções de *review* são importantes porque podem levar ao aumento das vendas, fornecendo aos clientes as informações sobre um produto ou serviço que eles estão pensando em comprar. As pessoas são mais propensas a comprar um produto ou serviço com base nas avaliações e comentários de outras pessoas (OGINO, 2021).

Se considerarmos os títulos presentes na plataforma da *Netflix* como produtos a serem consumidos, a mesma lógica se aplica. Usuários são mais propensos a assistir um filme ou uma série com base nos *reviews* de usuários que já assistiram e compartilharam suas experiências.

REFERÊNCIAS

DEVELOPER. **Visão geral das notificações toast.** Disponível em: <<https://developer.android.com/guide/topics/ui/notifiers/toasts?hl=pt-br>> Acesso em: 17 de ago. 2021.

DEVMEDIA. **Crie uma loja virtual completa com Visual Studio e ASP.NET.** Disponível em: <<https://www.devmedia.com.br/curso/crie-uma-loja-virtual-completa-com-visual-studio-e-asp-net/62?aula-assine>> Acesso em: 17 de ago.2021.

HOSTGATOR. **API Restful:** conceito, princípios e como criar. 7 novembro, 2019. Disponível em: <<https://www.hostgator.com.br/blog/api-restful/>> Acesso em: 17 de ago. 2021

IBM - Integration Bus. **Swagger**. versão 10.0.0.18, 2020. Disponível em: <<https://www.ibm.com/docs/pt-br/integration-bus/10.0?topic=ssmkhh-10-0-0-com-ibm-ertools-mft-doc-bi12018--htm>>. Acesso em: 17 de ago. 2021.

MASSÉ, M. **Rest API Design Rulebook** 1 ed. California: O'Reilly Media, 2011.

MOZILLA. **Códigos de status de respostas HTTP**, 2021. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status>>. Acesso em: 17 de ago. 2021.

NETFLIX. Como classificar séries e filmes. Disponível em: <<https://help.Netflix.com/pt/node/9898>>. Acesso em: 19 abr. 2021.

OGINO, Sean. **4 Reasons Online Reviews Are Important For Your Ecommerce Business**. Ratings and Reviews , 2021. Disponível em: <<https://www.annexcloud.com/blog/4-reasons-online-reviews-important-ecommerce-business/>> Acesso em: 17 de ago. 2021.

SOMMERVILLE, I. **Engenharia de software**. 9 ed. São Paulo: Pearson Prentice Hall, 2011.

WAZLAWICK, R. S. **Metodologia de pesquisa para computação**. 2 ed. Elsevier Editora Ltda. 2014.



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
GABINETE DO REITOR

Av. Universitária, 1069 • Setor Universitário
Cidade Postal 05 • CEP 74605-010
Goiânia • Goiás • Brasil
Fone: (62) 3946.1000
www.pucgoias.edu.br • reitoria@pucgoias.edu.br

RESOLUÇÃO n° 038/2020 – CEPE

ANEXO I

APÊNDICE ao TCC

Termo de autorização de publicação de produção acadêmica

O(A) estudante Victor Hugo de Lisboa
do Curso de Ciências da Computação, matrícula 20161002803558,
telefone: 62981190534 e-mail victorhugodelisboa@gmail.com, na qualidade de titular dos
direitos autorais, em consonância com a Lei n° 9.610/98 (Lei dos Direitos do autor),
autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o
Trabalho de Conclusão de Curso intitulado
Aplicação para review de filmes e séries da Netflix

, gratuitamente, sem ressarcimento dos direitos autorais, por 5
(cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial
de computadores, no formato especificado (Texto (PDF); Imagem (GIF ou JPEG); Som
(WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros, específicos da
área; para fins de leitura e/ou impressão pela internet, a título de divulgação da
produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 23 de Agosto de 2021.

Assinatura do(s) autor(es): Victor Hugo de Lisboa

Nome completo do autor: Victor Hugo de Lisboa

Assinatura do professor-orientador: Olegário Correa da Silva Neto

Nome completo do professor-orientador: Olegário Correa da Silva Neto, MSc.