DOCUMENTAÇÃO

OpFlix

Alamedo Barão de Limeira,

539 – São Paulo/SP SENAI-SP

# Sumário

[Sumário 2](#_Toc25908121)

[Resumo: 3](#_Toc25908122)

[Descrição do projeto: 3](#_Toc25908123)

[Modelagem de Software: 4](#_Toc25908124)

[Modelo Lógico 4](#_Toc25908125)

[Modelo Físico 5](#_Toc25908126)

[Modelo Conceitual 5](#_Toc25908127)

[BackEnd: 6](#_Toc25908128)

[Resumo: 6](#_Toc25908129)

[Utilização: 6](#_Toc25908130)

[FrontEnd Desktop e Mobile: 8](#_Toc25908131)

[Resumo: 8](#_Toc25908132)

# Resumo:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Objetivos

O aplicativo solicitado pela(o) empresa/cliente OpFlix (Cliente denominado Tadeu), consiste em uma biblioteca de divulgações de filmes, desde sua data de lançamento até seus mínimos detalhes.

# Descrição do projeto:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Resumo do projeto

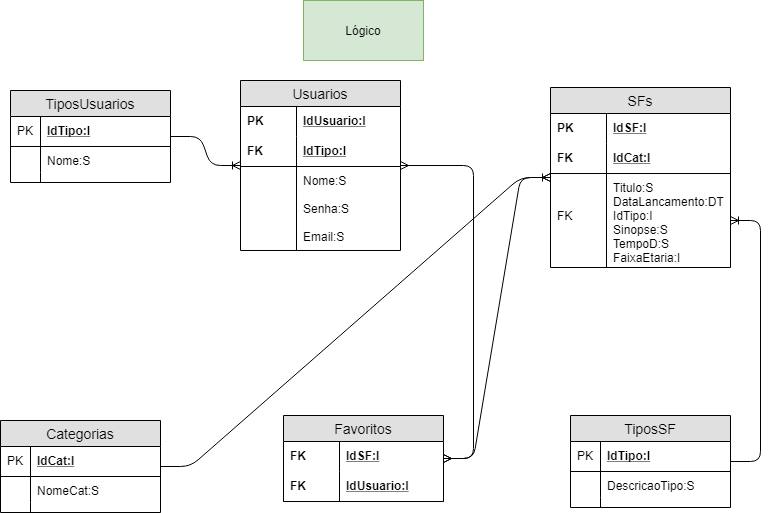
Assim como solicitado pelo cliente Tadeu, o novo programa da empresa OpFlix irá dar total acesso para o leitor(visualizador) receber informações sobre o lançamento de novos filmes no cinema, mostrando sua plataforma, seu título, tempo de duração, entre outros.

# Modelagem de Software:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Modelo Lógico

O modelo lógico é o que mais se assemelha a realidade de como será a produção final do projeto, assim mostrando todas as entidades mantendo relações umas com as outras.



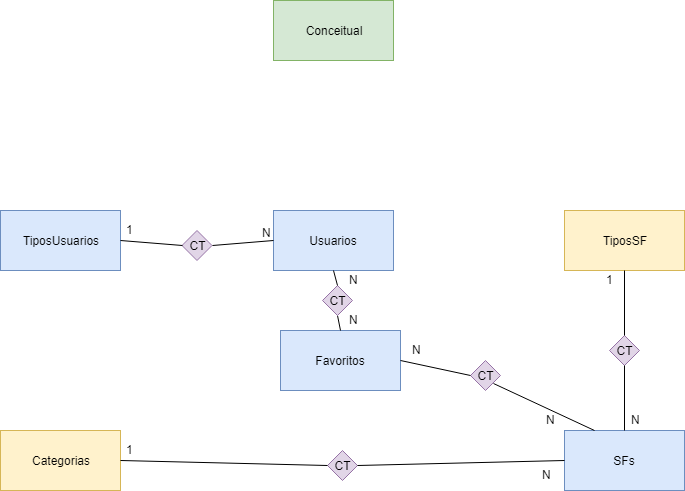
## Modelo Físico

O modelo físico tem como sua principalmente característica a função de por a prova o sistema que será formado, assim mostrando as entidades em um funcionamento beta, ou seja, ainda não finalizado. Em resumo podemos dizer que é a base de testes.



## Modelo Conceitual

O modelo conceitual é responsável por nos dar uma visão geral de como será o projeto e quais entidades serão necessárias para o funcionamento do programa, nessa fase não é aprofundado com detalhes sobre a criação final do projeto, mas sim uma visão geral das entidades principais e a relação delas.



# BackEnd:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Resumo:

O BackEnd em resumo se deu na elaboração de uma API da qual integra dentre várias tecnologias o Entity Framework (EF), assim como o Swagger, do qual realiza a facilitação de acesso a todos EndPoints.

Por ser elaborado em API, seu uso pode ser tanto Mobile, ou seja, dispositivos móveis, como Desktops (Computadores de escritório/doméstico), quanto sistemas IOS. O uso da API torna o programa mais versátil e assim economiza e minimiza futuros problemas em relação a disposição do mesmo para diversas plataformas, o que o torna mais integral no mercado, além da estrutura feita em estrutura MVC, dividindo a aplicação em inúmeras áreas diferentes de vários acessos.

Dentre inúmeros privilégios podemos citar também a maior segurança que o sistema possuí tanto pela linguagem utilizada para a programação, quanto no use de Interfaces, das quais tornam o acesso ao banco de dados (BD) mais restrito, evitando falhas de segurança.

Como dito anteriormente, o sistema se utiliza do Swagger, do qual disponibiliza todos os EndPoints do mesmo, assim permitindo a visualização de cada função ali integrada. Para o acesso ao Swagger, deve-se acessar a seguinte URL: (www.NomeDoSite/swagger/index.html), assim sendo visível de uma forma facilitada todas as funções do programa.

Além do Swagger, o Postman também é utilizado como um programa de terceiro. O mesmo tem a função de testar todos os EndPoints do sistema, evitando então futuros erros que ocasionalmente poderiam vir a acontecer, possibilitando uma maior facilidade de detecção dos mesmos e assim corrigindo-os.

## Utilização:

Para realizar o uso do sistema, o usuário/administrador deve inicialmente abrir o script do banco de dados anteriormente disponibilizado com o nome “M\_OpFlix\_2019” e em seguida realizar as seguintes instruções:

1º - Selecionar a seguinte linha com o mouse “CREATE DATABASE M\_OpFlix\_2019”;

2º - Apertar a tecla F5 do teclado e esperar a confirmação do sistema (Comandos concluídos com êxito);

3º - Selecionar a seguinte linha com o mouse “USE M\_OpFlix\_2019”;

4º - Apertar a tecla F5 do teclado e esperar a confirmação do sistema (Comandos concluídos com êxito);

5º - Selecionar a seguinte linha “CREATE TABLE TiposUsuarios” e todas suas linhas seguintes, até o primeiro ponto em virgula seguido de um parêntese encontrado.

**Obs: O “ ; “ representa o final do comando.**

6º - Apertar a tecla F5 do teclado e esperar a confirmação do sistema (Comandos concluídos com êxito);

7º - Selecionar a seguinte linha “CREATE TABLE Usuarios” e todas suas linhas seguintes, até o primeiro ponto em virgula seguido de um parêntese encontrado.

8º - Apertar a tecla F5 do teclado e esperar a confirmação do sistema (Comandos concluídos com êxito);

9º - Selecionar a seguinte linha “CREATE TABLE Categorias” e todas suas linhas seguintes, até o primeiro ponto em virgula seguido de um parêntese encontrado.

10º - Apertar a tecla F5 do teclado e esperar a confirmação do sistema (Comandos concluídos com êxito);

11º - Selecionar a seguinte linha “CREATE TABLE TiposSF” e todas suas linhas seguintes, até o primeiro ponto em virgula seguido de um parêntese encontrado.

12º - Apertar a tecla F5 do teclado e esperar a confirmação do sistema (Comandos concluídos com êxito);

13º - Selecionar a seguinte linha “CREATE TABLE SFs” e todas suas linhas seguintes, até o primeiro ponto em virgula seguido de um parêntese encontrado.

14º - Apertar a tecla F5 do teclado e esperar a confirmação do sistema (Comandos concluídos com êxito);

15º - Selecionar a seguinte linha “CREATE TABLE Plataformas” e todas suas linhas seguintes, até o primeiro ponto em virgula seguido de um parêntese encontrado.

16º - Apertar a tecla F5 do teclado e esperar a confirmação do sistema (Comandos concluídos com êxito);

17º - Selecionar a seguinte linha “CREATE TABLE Favoritos” e todas suas linhas seguintes, até o primeiro ponto em virgula seguido de um parêntese encontrado.

18º - Apertar a tecla F5 do teclado e esperar a confirmação do sistema (Comandos concluídos com êxito);

19º - Após a criação de todas tabelas seguindo os passos acima, deve-se selecionar tudo desde a seguinte linha “Insert TiposUsuarios (Nome)” até a seguinte linha “Values (‘Documentário’), (‘Drama’);”

20º - Apertar a tecla F5 do teclado e esperar a confirmação do sistema (Comandos concluídos com êxito);

**Obs: O cadastro das plataformas será feito diretamente na programação, portanto não há necessidade de inserção pelo banco de dados!**

21º - Toda inserção e criação do banco de dados foi realizada. Deve-se agora abrir a pasta “Projeto API BackEnd” fornecida junta com a documentação, em seguida abrir a pasta seguinte e então executar o arquivo “WebAPI.Senai.OpFlix\_Manha.sln”.

22º - Após o programa ter sido aberto, deve-se apertar a tecla F5 e então o código já estará funcionando.

# FrontEnd Desktop e Mobile:

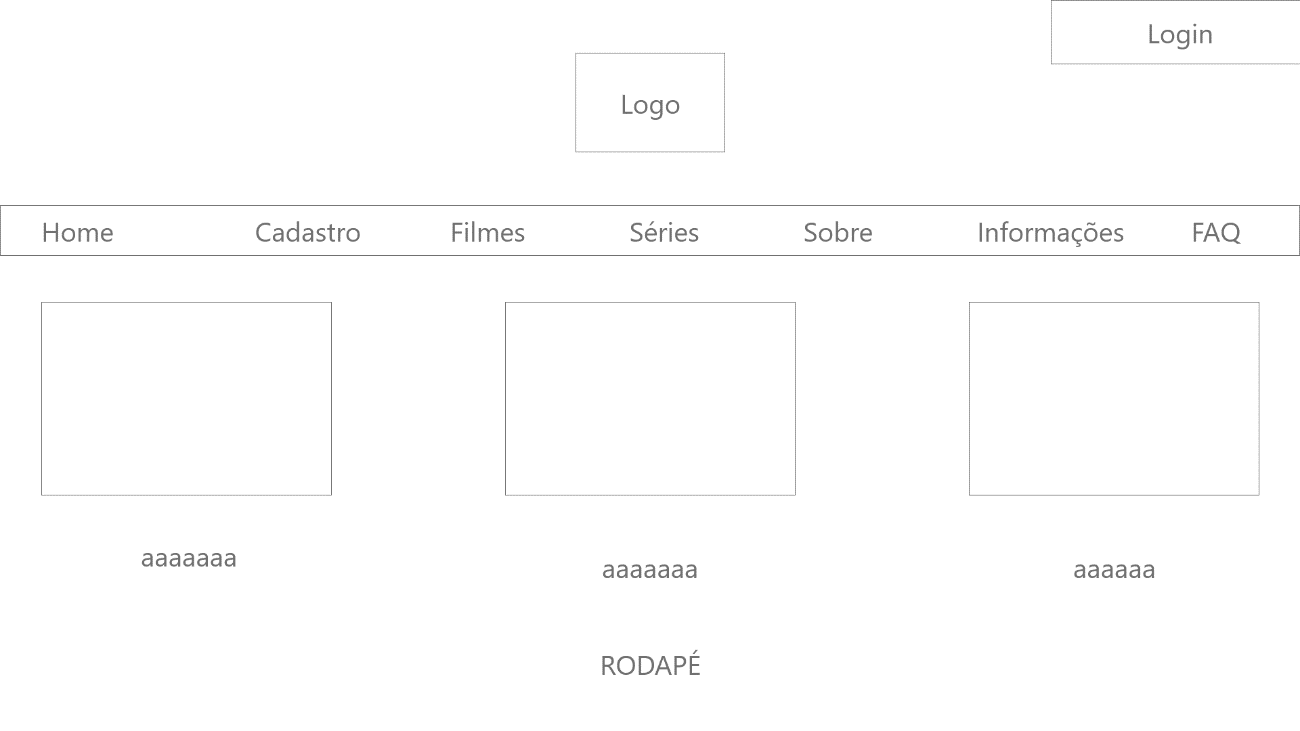
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

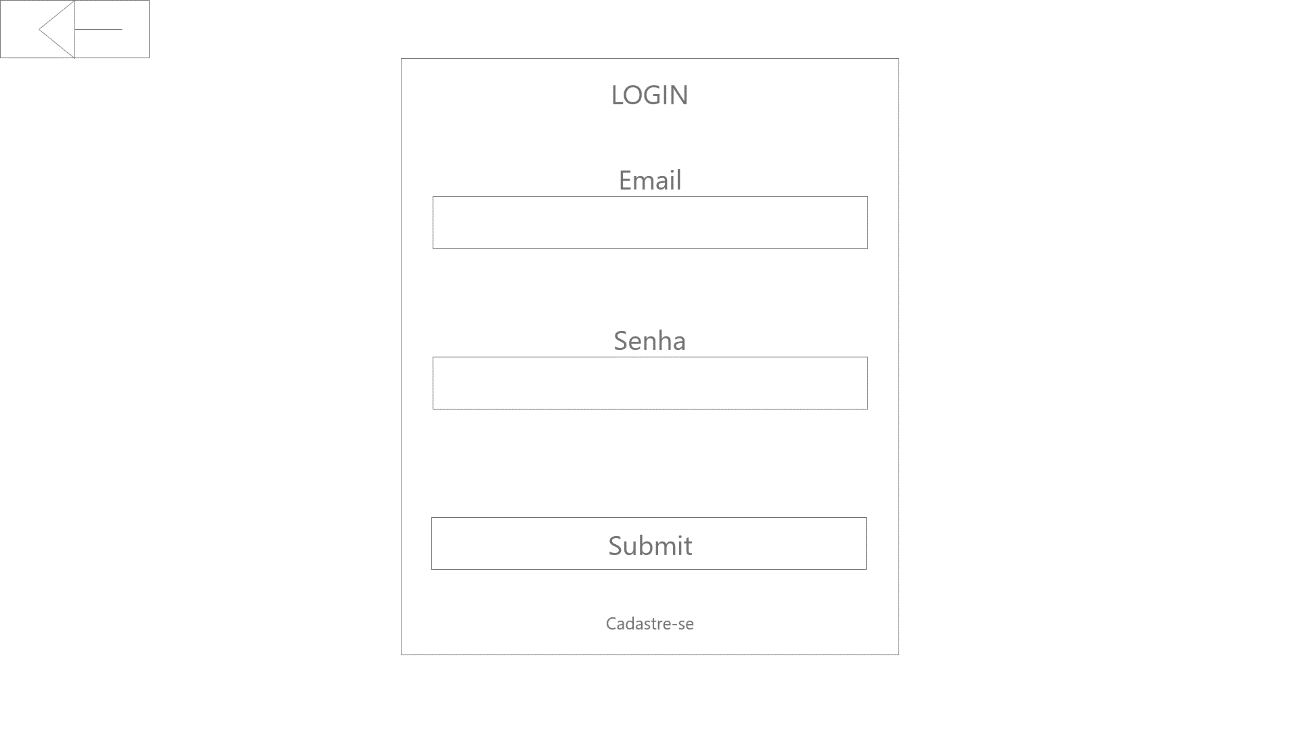
## Resumo:

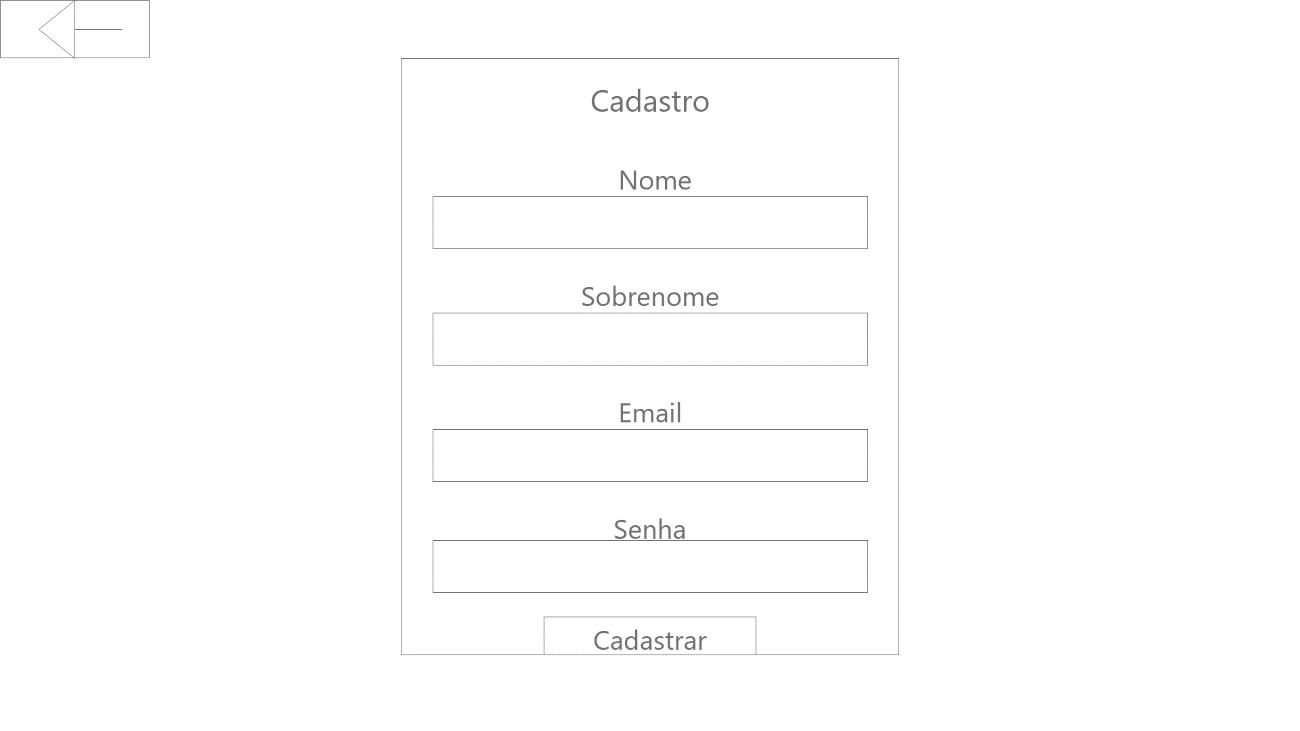
O FrontEnd é a parte visual do projeto, todo e qualquer aspecto visual se integra no FrontEnd, seja em um Desktop, quanto em um dispositivo Mobile, IOS e etc...

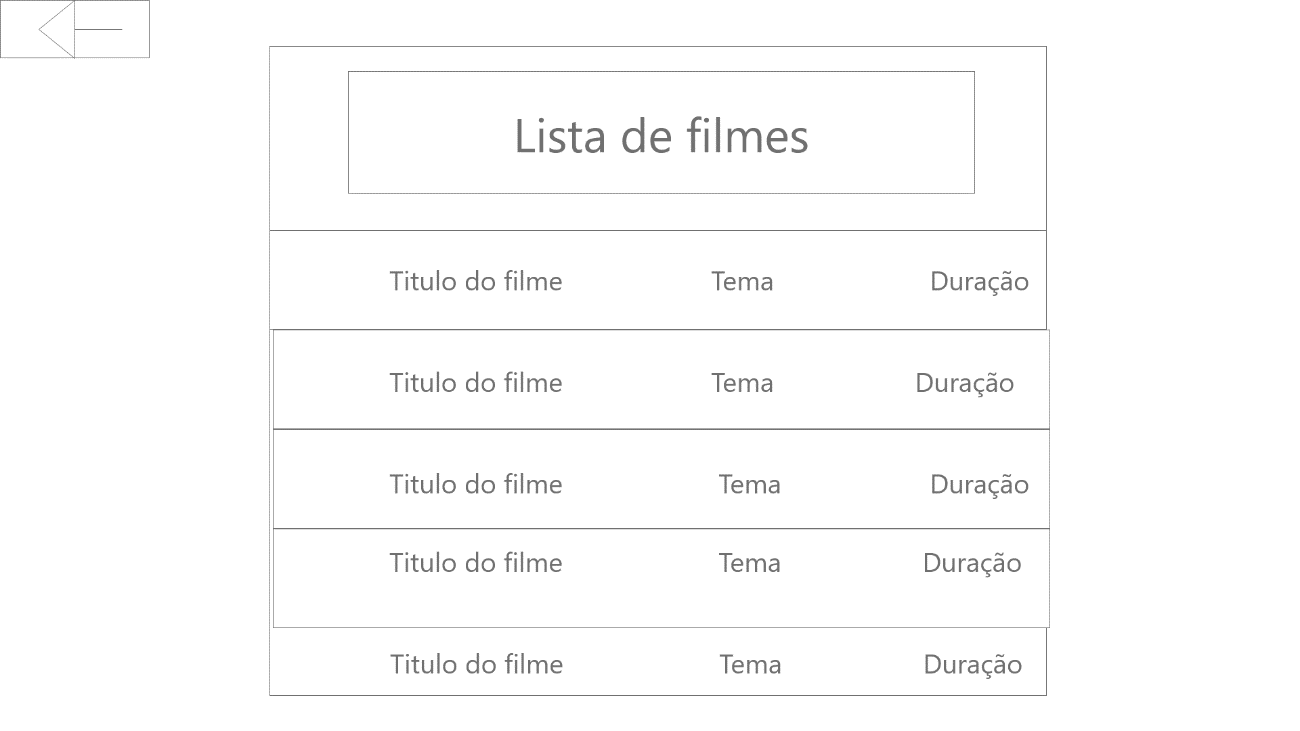
Toda a elaboração do FrontEnd na aplicação do OpFlix foi feita com React e React-Native para que sua integração com a API fosse facilitada e integrada de uma forma perfeita e funcional, para todos os dispositivos que conectassem com a aplicação. Para evitar problemas ao longo do desenvolvimento, foi-se elaborado um esboço básico do qual possui o nome de Layout de Baixa Fidelidade, ou Esboço Sujo.

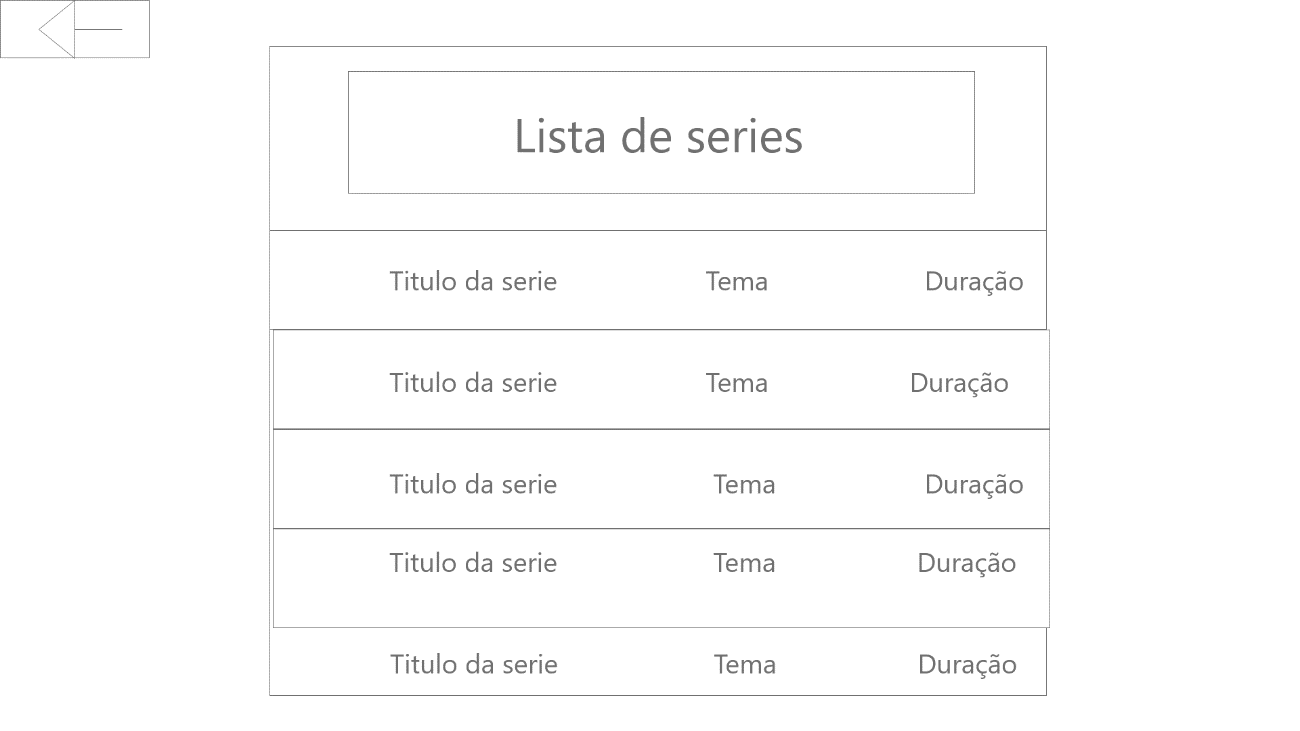
Imagens do Layout abaixo:











Podemos ver nas imagens que alguns aspectos foram alterados e outros foram aprimorados para uma melhor relação com o futuro usuário da aplicação, deixando a interface mais amigável para o olho humano, assim como a sensação aprimorada ao utilizar da aplicação.

Na elaboração do Mobile seguiu-se o mesmo padrão, foi utilizado o React-native, assim permitindo uma integração com a API de forma direta, e alterações mais fáceis no código da aplicação, facilitando a manutenção de algum possível erro ou mudança de dados.

O layout do Website em relação ao do Mobile se assemelha bastante, porém de um tamanho resumido para que caiba na palma de nossas mãos, mantendo o alinhamento e forma dos objetos.