

Documentação

OpFlix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Alameda Barão de Limeira, 539 – São Paulo/SP |  |  |  | sp.senai.br |

Sumário

1. [Resumo 2](#_Toc25560494)

[Objetivos](#_Toc25560495)

1. [Descrição do projeto 2](#_Toc25560496)

[Resumo do projeto](#_Toc25560497)

1. [Modelagem de Software 3](#_Toc25560498)

[Modelo Lógico](#_Toc25560499)

[Modelo Físico](#_Toc25560500)

[Modelo Conceitual](#_Toc25560501)

[Cronograma](#_Toc25560502)

1. [BACKEND 6](#_Toc25560503)

[Como acessar o Swagger](#_Toc25560504)

[Fazer deploy do projeto](#_Toc25560505)

[Pacotes do Nuget que foram utilizados](#_Toc25560506)

[Arquitetura do Projeto](#_Toc25560507)

[Postman](#_Toc25560508)

[Inicialização do programa](#_Toc25560509)

[Ferramentas utilizadas](#_Toc25560510)

1. [Funcionalidades 9](#_Toc25560511)

[Web](#_Toc25560512)

[Mobile](#_Toc25560513)

1. [Protótipos 10](#_Toc25560514)

[Web](#_Toc25560515)

[Mobile](#_Toc25560516)

1. [Referências 12](#_Toc25560517)

[Links](#_Toc25560518)

[Livros](#_Toc25560519)

Resumo

Objetivos

Este documento tem como objetivo mostrar todas as funcionalidades e detalhes de uma plataforma chamada OpFlix, na qual é um serviço de catalogar lançamentos de filmes e séries.

Descrição do projeto

A plataforma OpFlix deseja mostrar aos seus usuários quais serão os próximos lançamentos de mídia e consequentemente avisar sobre a plataforma e os dados de seu lançamento.

Resumo do projeto

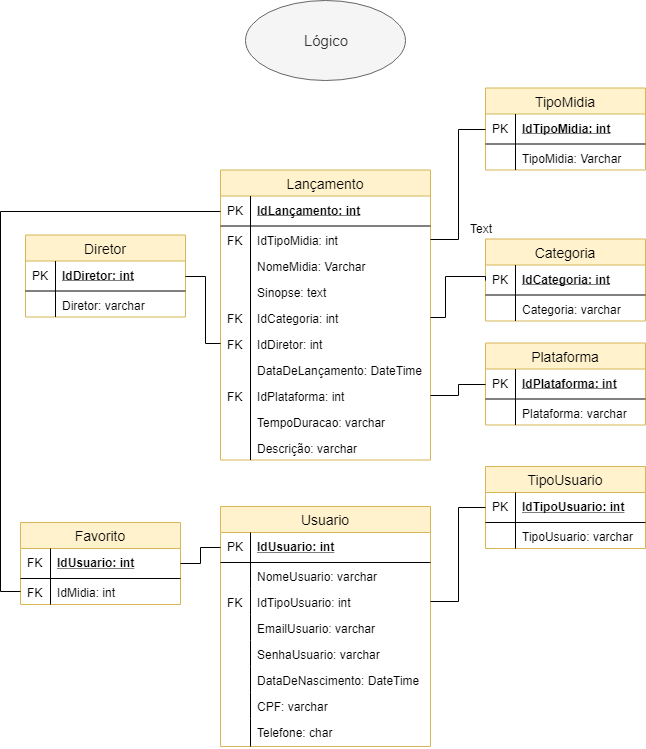
A plataforma OpFlix disponibiliza para seus usuários lançamento de novos filmes e séries, na qual contém os dados dos filmes e os dados de seus lançamentos.

A plataforma OpFlix permite que o usuário possa acessar diversos lançamentos de diversos gêneros. O cliente decidiu que um lançamento poderá ter apenas uma categoria, diretor, plataforma de estreia e tipo (se é filme, série, etc). Um usuário cadastrado poderá cadastrar os seus lançamentos favoritos para ficarem salvos. Os Usuários poderão ser um cliente normal, ou um administrador.

Modelagem de Software

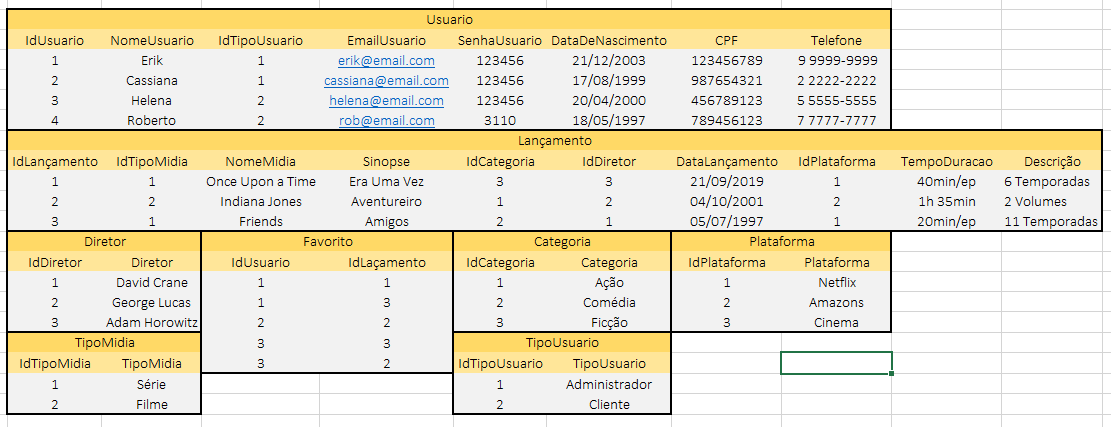
Esta sessão tende a demonstrar o que será desenvolvido no banco de dados e quais são as suas representações.

Modelo Lógico

O modelo lógico leva em conta algumas limitações e implementa recursos como adequação de padrão e nomenclatura, define as [chaves primárias e estrangeiras](https://www.luis.blog.br/chave%20primaria-chave-estrangeira-e-candidata.aspx), normalização, integridade referencial, entre outras.

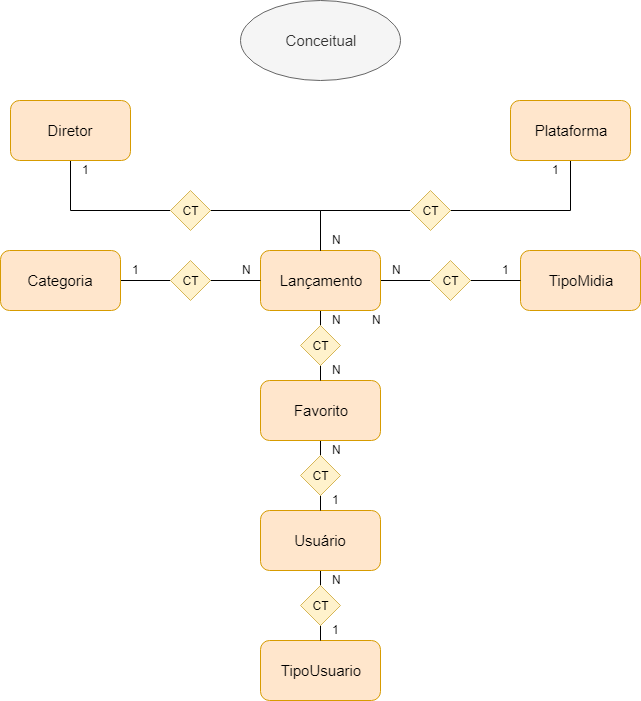
Modelo Físico

A modelagem física lida com o design do banco de dados real com base nos requisitos reunidos durante a modelagem lógica do banco de dados.



Modelo Conceitual

A modelagem conceitual baseia-se no mais alto nível e deve ser usada para envolver o cliente, pois o foco aqui é discutir os aspectos do negócio do cliente e não da tecnologia.



Para rodar o programa, execute na ordem respectivamente: “M\_01\_ArthurFoschiani\_DDL.sql”, “M\_02\_ ArthurFoschiani \_DML.sql” e “M\_03\_ ArthurFoschiani \_DQL.sql”.

Cronograma

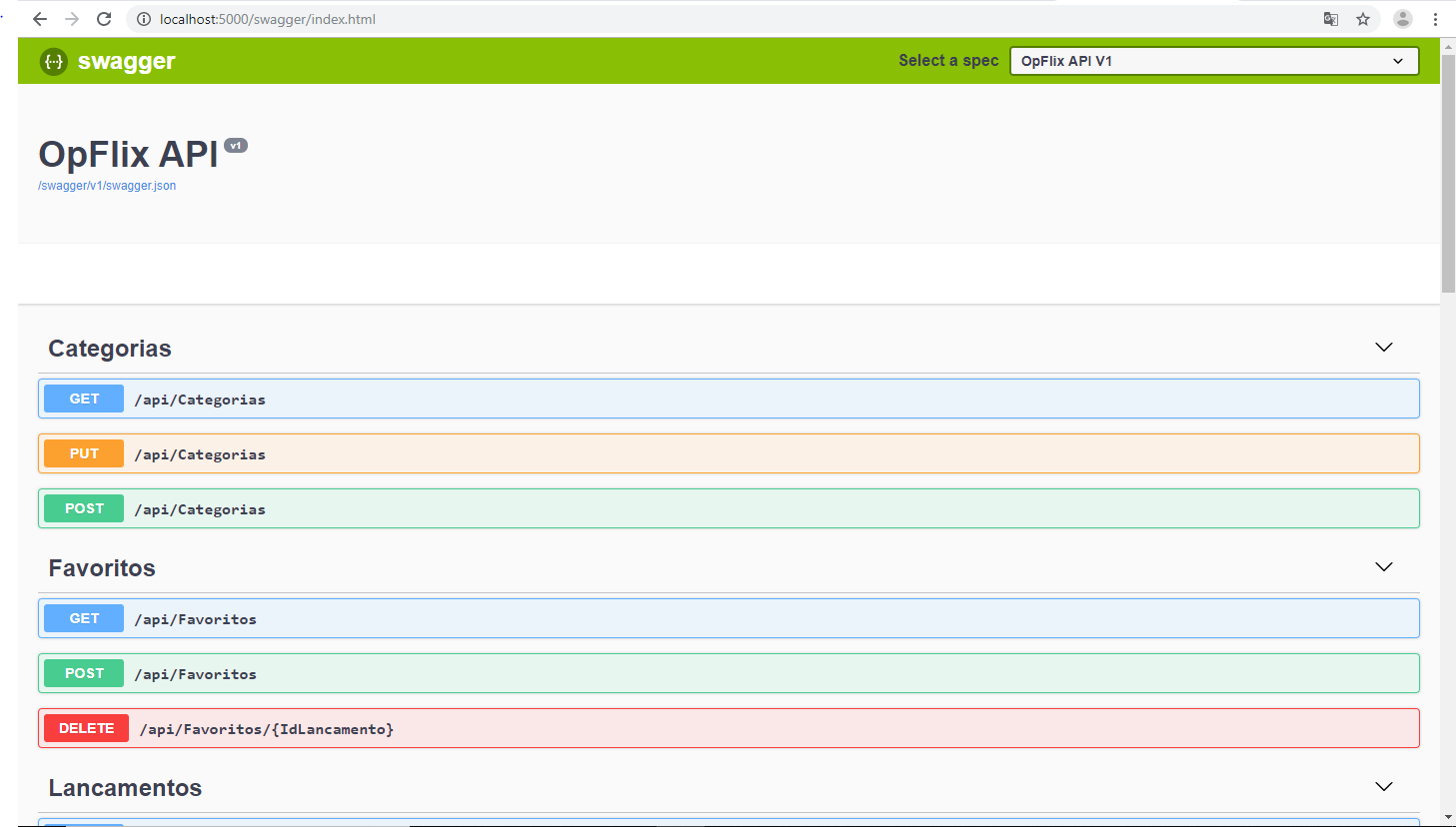
|  | Dia 1 | Dia 2 | Dia 3 | Dia 4 | Dia 5 | Dia 6 | Dia 7 | Dia 8 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Modelo Lógico** |  |  |  |  |  |  |  |  |
| **Modelo Físico** |  |  |  |  |  |  |  |  |
| Modelo Conceitual |  |  |  |  |  |  |  |  |

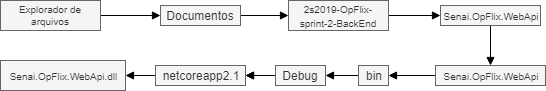
BACKEND

Como acessar o Swagger

1. Acesse o projeto “Senai.OpFlix.WebApi” no Visual Studio;
2. Execute o projeto clicando no botão:

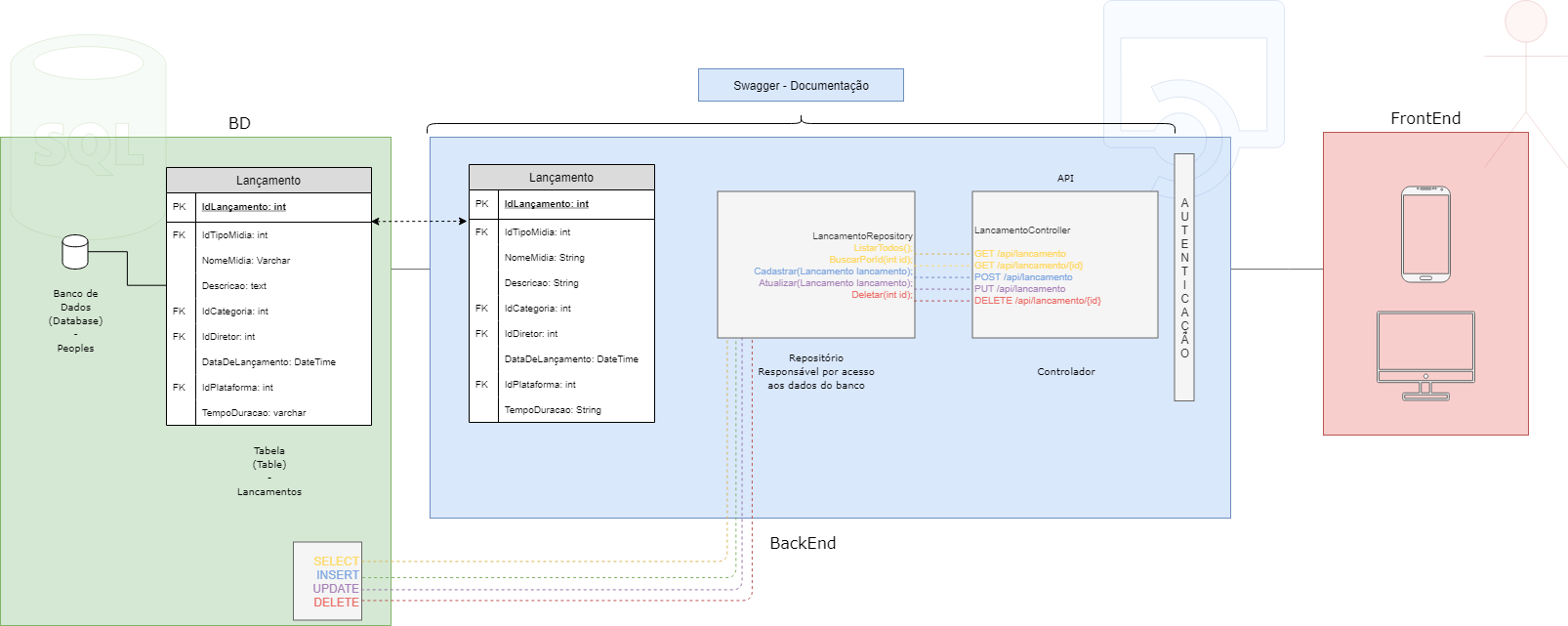


1. Após a compilação, abra seu navegador e insira o seguinte endereço na barra de pesquisa: <http://localhost:5000/swagger/index.html>.
2. Após abrir a página do Swagger, você conseguira visualizar todos os métodos do programa, suas funcionalidades e como testá-los.

Fazer deploy do projeto

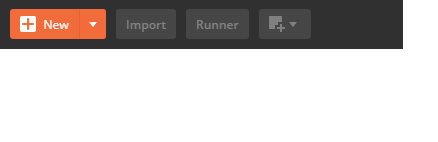
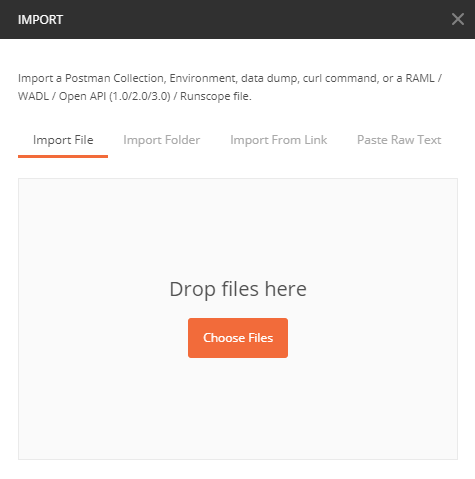
Pacotes do Nuget que foram utilizados

* Swashbuckle.AspNetCore 4.0.1
* System.Data.SqlClient 4.6.1
* Microsoft.EntityFrameworkCore.SqlServer 2.1.11
* Microsoft.EntityFrameworkCore.SqlServer.Design 1.1.6
* Microsoft.EntityFrameworkCore.Tools 2.1.11
* Microsoft.AspNetCore.Authentication.JwtBearer 2.1.1
* System.IdentityModel.Tokens.Jwt 5.5.0

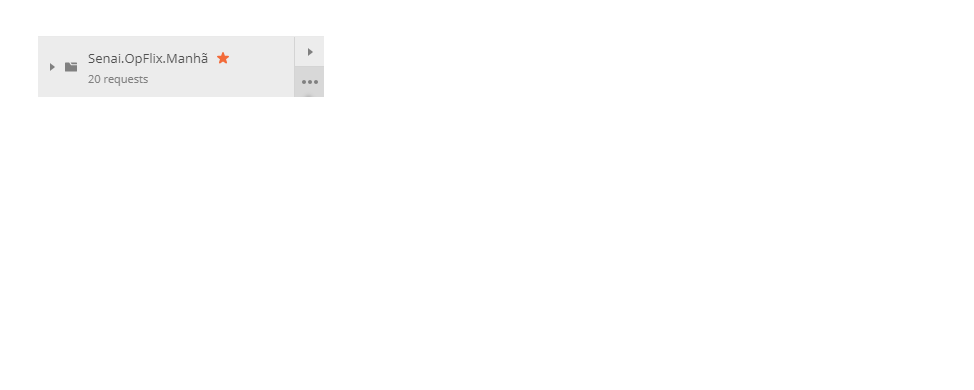
Arquitetura do Projeto

Postman

Importar

1. Abra o aplicativo “Postman” em seu desktop;
2. Clique no botão “Import” presente no menu:
3. Selecione a opção “Import File” e clique no botão “Choose Files”:
4. Selecione o arquivo “OpFlix\_Postman”, presente na pasta “2s2019-OpFlix-sprint-2-BackEnd”;
5. Todos os arquivos estarão presentes na pasta “Senai.OpFlix.Manhã”;

Exportar:

1. Abra o aplicativo “Postman” em seu desktop;
2. Na pasta “Senai.OpFlix.Manhã”, clique no botão com os três pontos:
3. Selecionar a opção “Export” na coleção “Collection v2.1”;
4. Colocar o nome e o local desejado para salvar;
5. Após isso, o projeto já está exportado para seus arquivos.

Inicialização do programa

Banco de Dados

Primeiramente acesse esse link do GitHub <https://github.com/arthurfoschiani/2s2019-OpFlix-sprint-2-BackEnd> e clone esse repositório na sua máquina. O presente arquivo será enviado junto a três fotos que dizem respeito aos modelos apresentados mais à frente do documento e a três arquivos SQL. Os arquivos SQL são uma espécie de roteiro de criação que deve ser reproduzido no programa para computador Microsoft SQL Server Management Studio 18. Os arquivos devem ser executados na seguinte ordem:

1° M\_01\_ArthurFoschiani\_DDL;

2° M\_02\_ ArthurFoschiani \_DML;

3° M\_03\_ ArthurFoschiani \_DQL.

Para executar, abra o programa descrito acima e arraste o primeiro arquivo para o programa e pressione o botão do teclado “F5”, depois faça o mesmo com o segundo e com o terceiro. O banco de dados já está pronto.

Aplicação API

É necessário, primeiramente, que o programa Visual Studio 2017 esteja instalado na máquina. Para executar a aplicação abra a pasta 2s2019-OpFlix-sprint-2-BackEnd/Senai.OpFlix.WebApi e clique duas vezes no arquivo “Senai.OpFlix.WebApi”. O arquivo será iniciado no programa descrito acima. Para iniciar as funcionalidades basta pressionar o botão do teclado “F5” ou clicar no botão que aparecerá na aplicação:



Ao executar, uma aba do Google Chrome abrirá instantaneamente com a seguinte URL “http://localhost:5000”. Desse modo, a aplicação já estará funcionando.

Ferramentas utilizadas

* Microsoft Visual Studio 2017
* Microsoft SQL Server Management Studio 18
* Postman
* Navegador de Web
* Swagger
* Jwt.io
* Trello
* GitHub

Funcionalidades

Web

As funcionalidades da aplicação pra web são as seguintes:

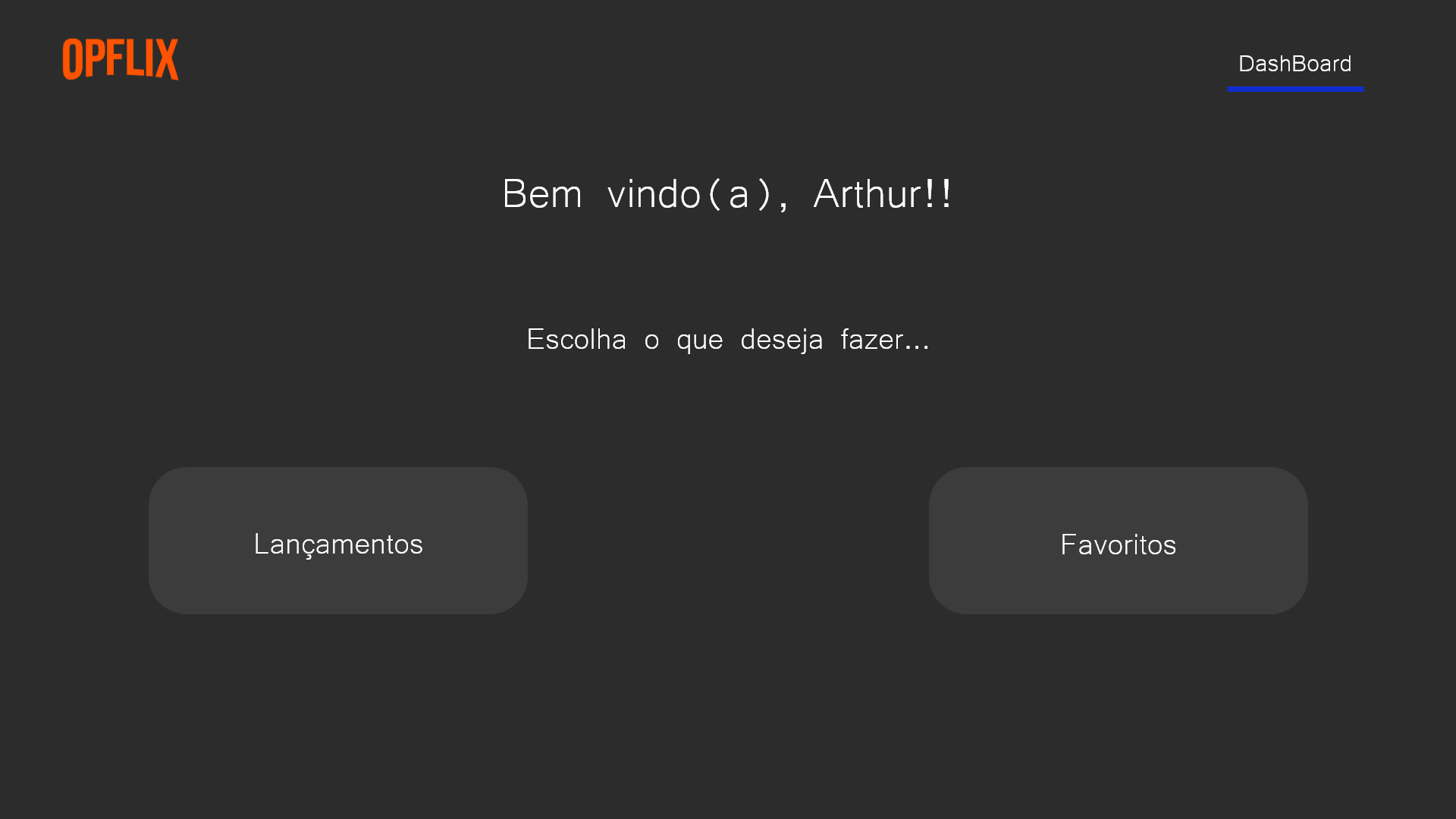
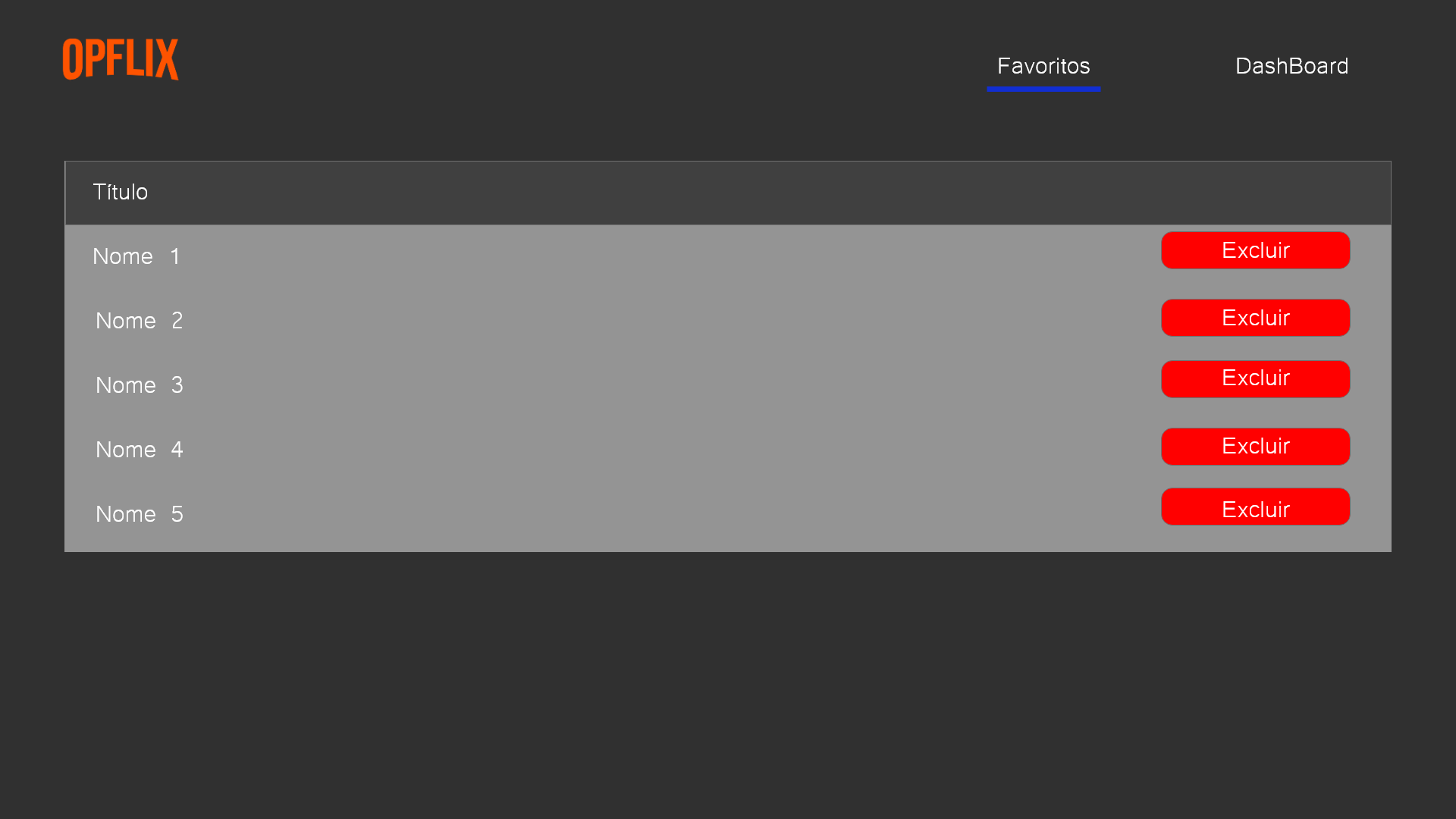
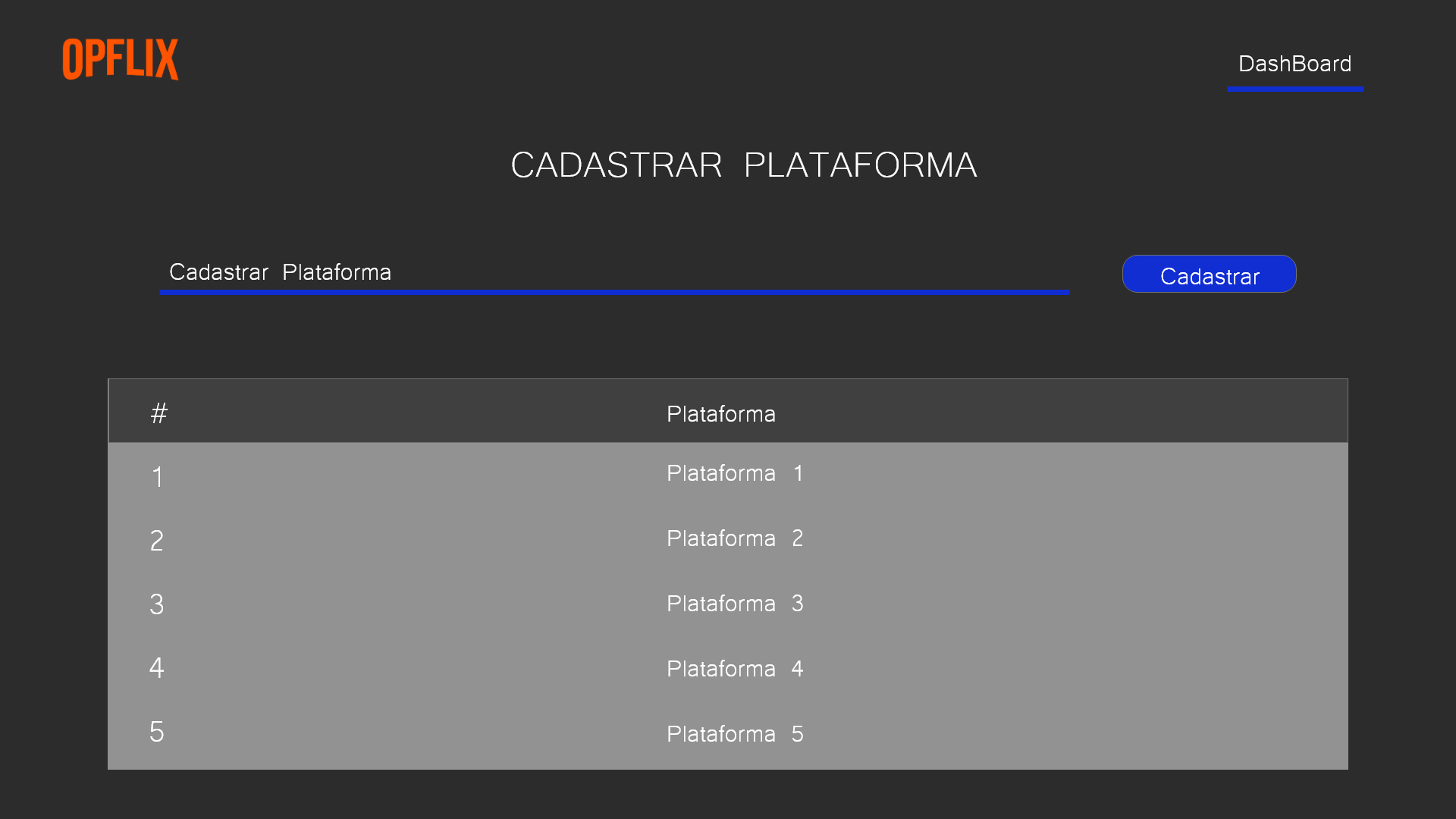
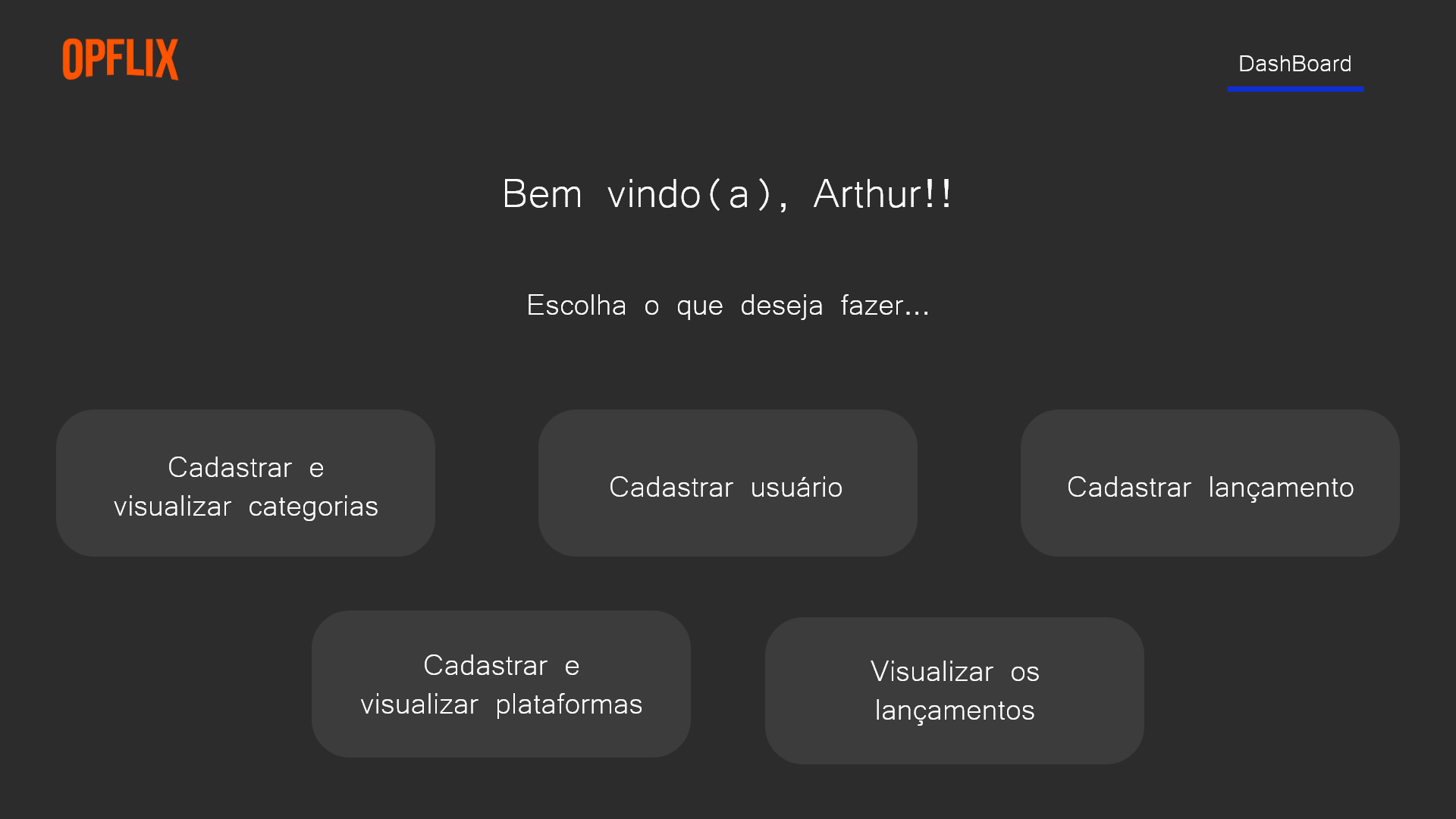
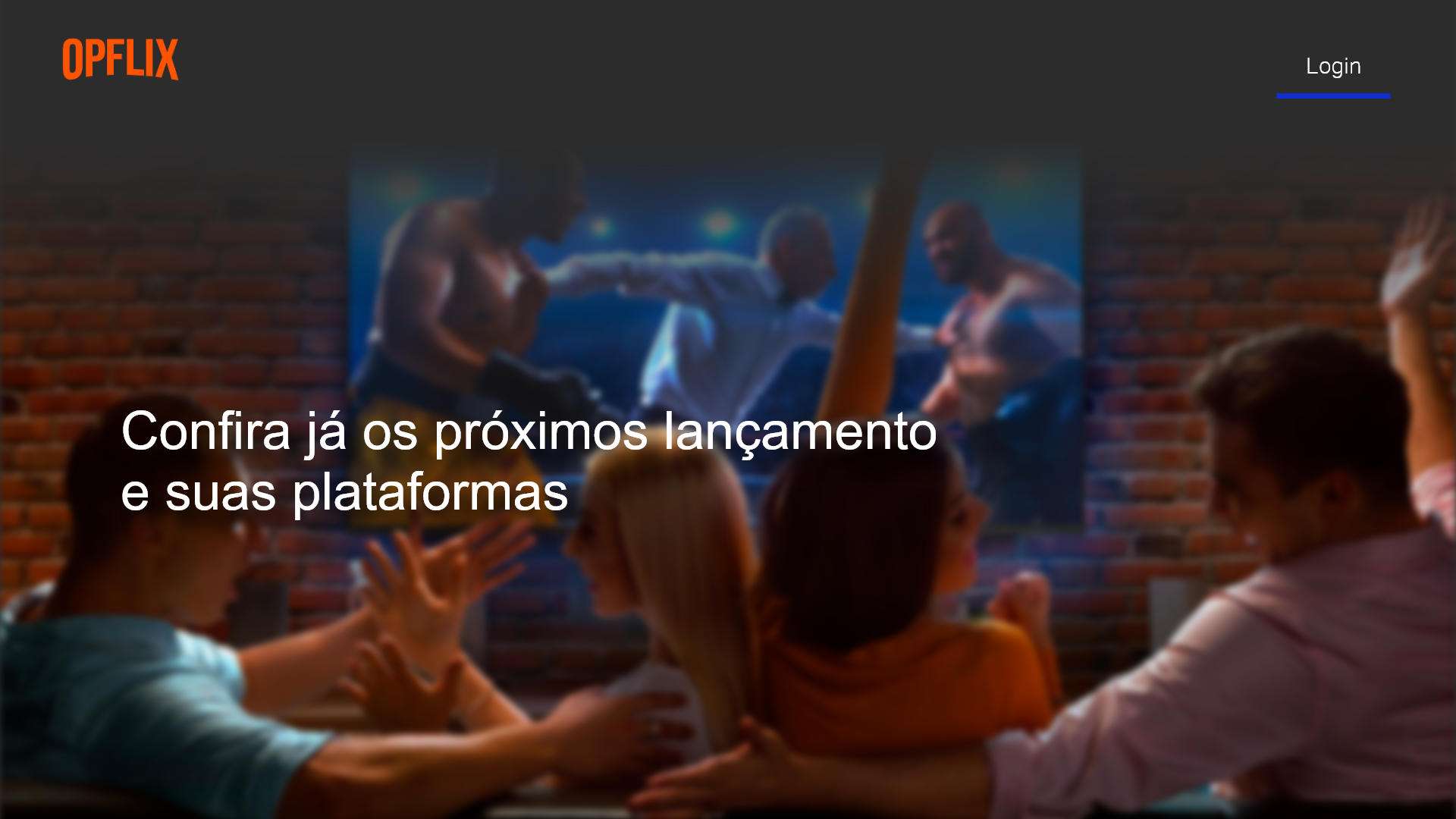
1. O administrador poderá cadastrar qualquer tipo de usuário (administrador ou cliente), também poderá listar os usuários cadastrados;
2. O administrador poderá cadastrar qualquer tipo de usuário (administrador ou cliente), também poderá listar os usuários cadastrados;
3. O administrador poderá cadastrar plataformas (contendo nome e mídia), poderá também listar e atualizar;
4. O administrador poderá cadastrar categoria (contendo nome), poderá também listar e atualizar;
5. O administrador poderá cadastrar o lançamento de um filme/série (contendo título, uma pequena sinopse, a categoria vinculada, tempo de duração, se é filme ou série e a data do primeiro lançamento daquele item), também poderá listar, atualizar, deletar, filtrar por data de lançamento e filtrar por nome da plataforma;
6. O cliente poderá visualizar todos os lançamentos publicados;

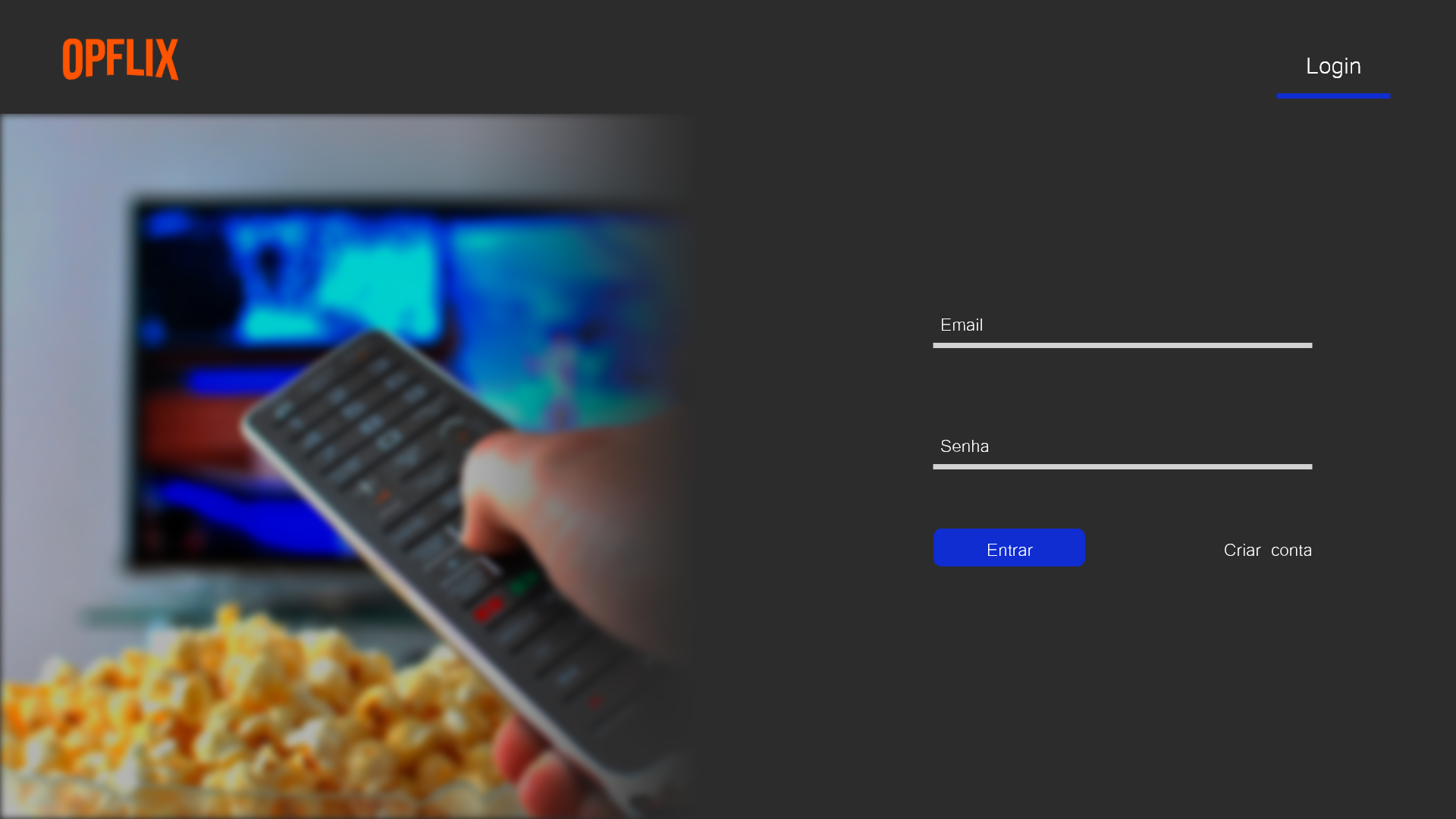
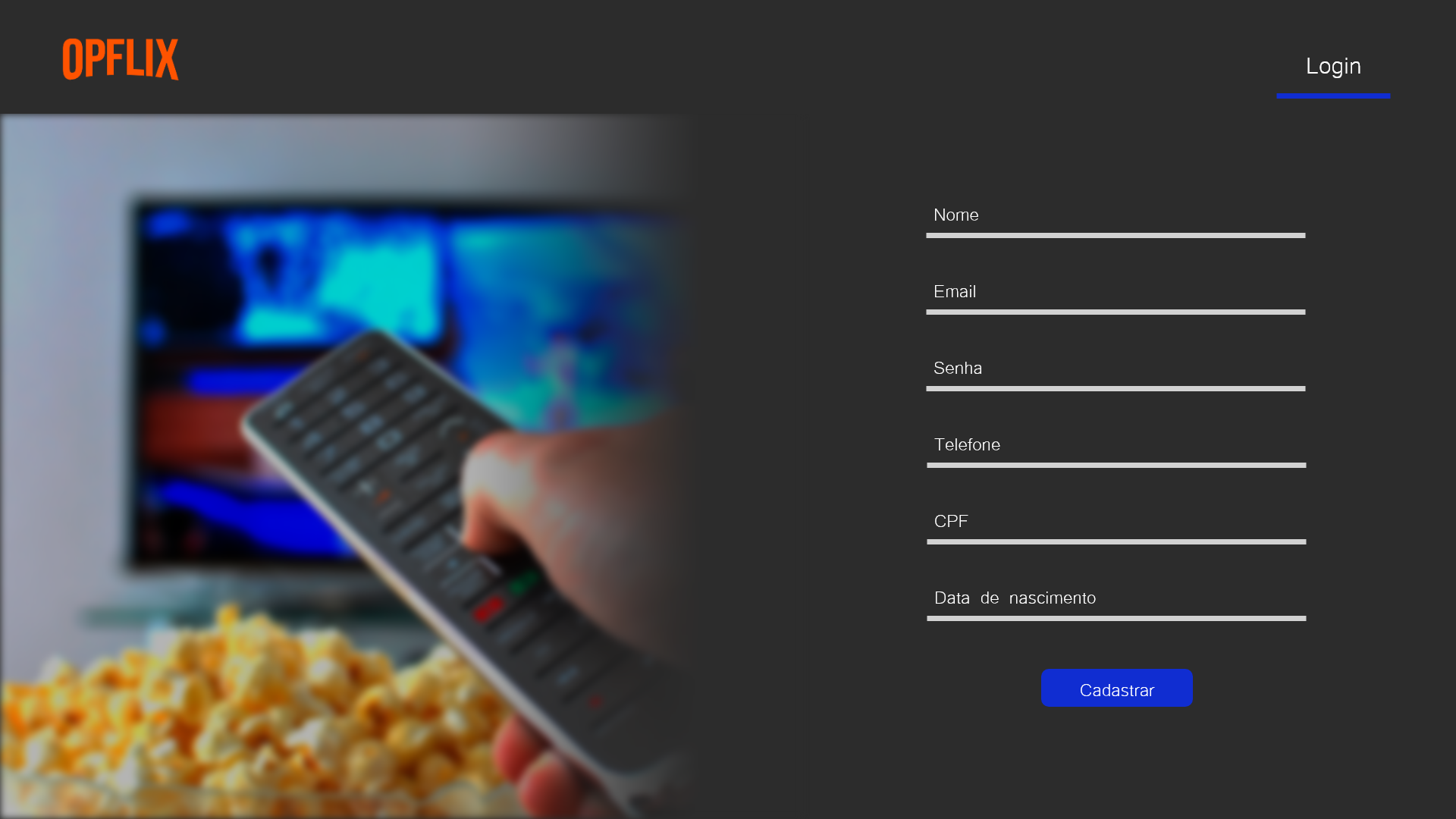
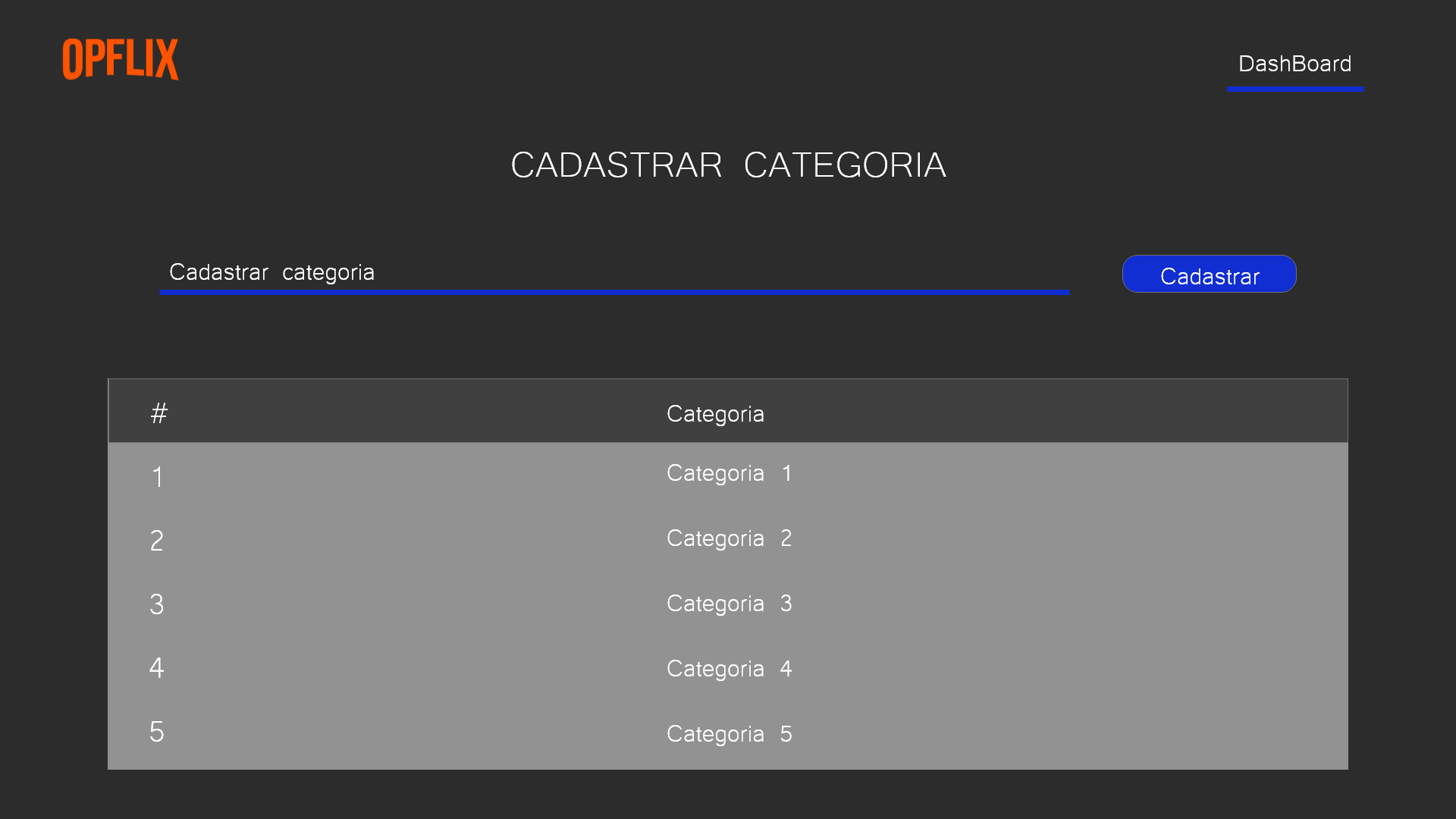
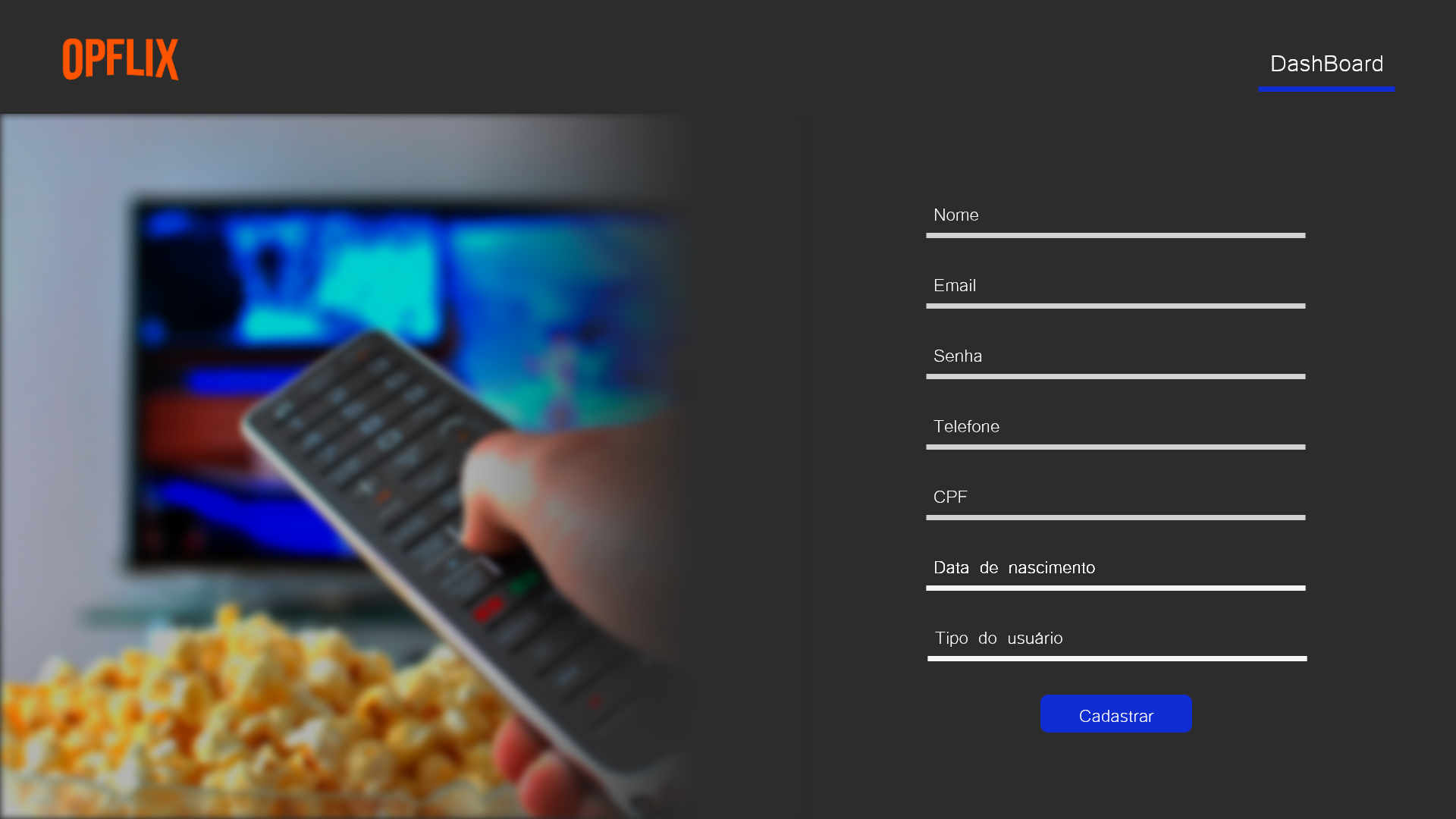
Mobile

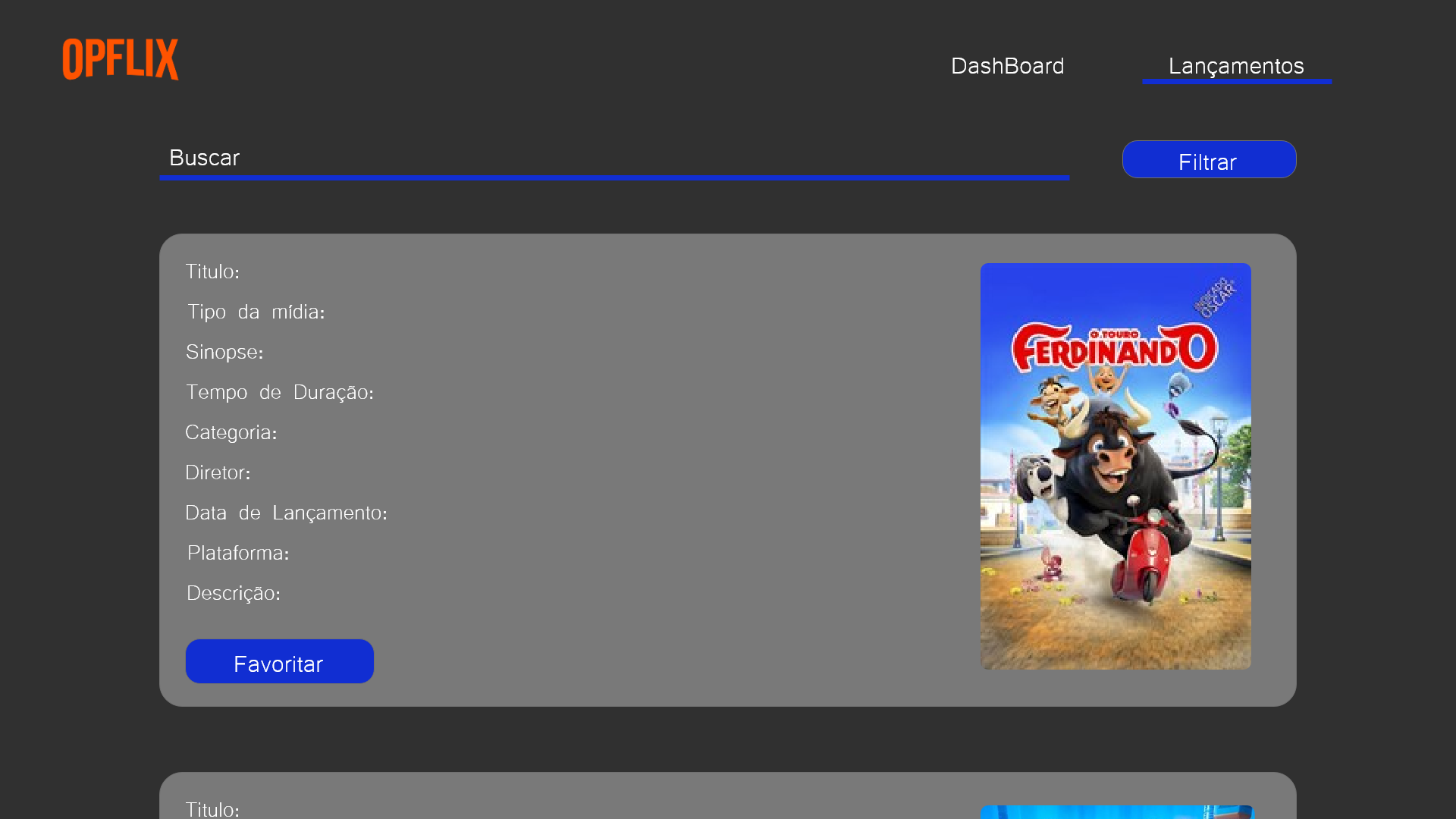
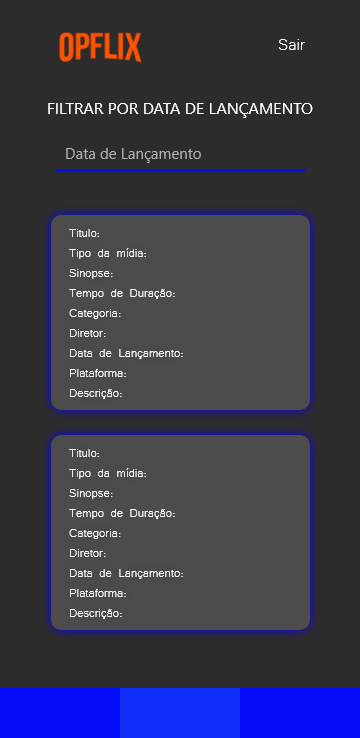
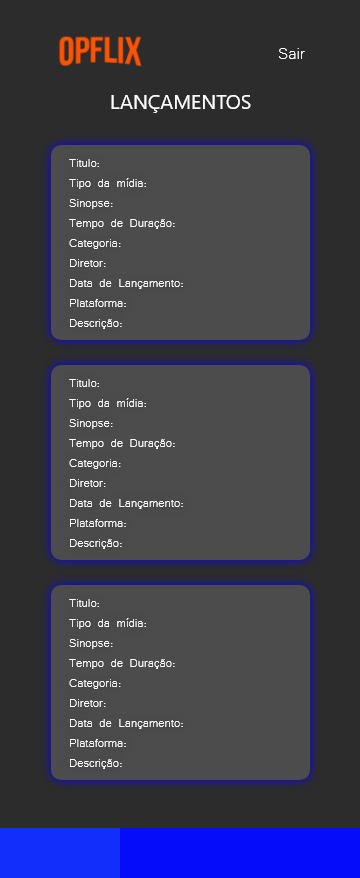
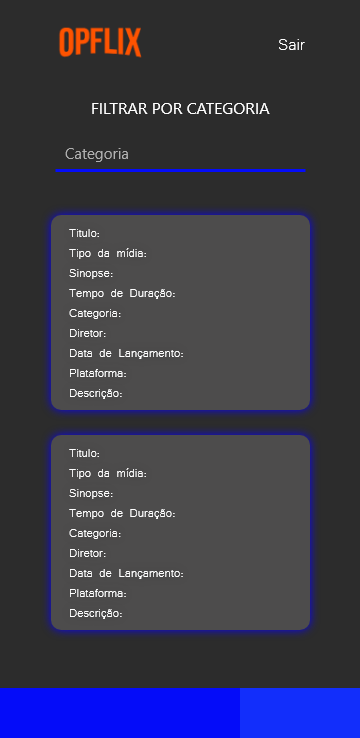
As funcionalidades da aplicação para mobile são:

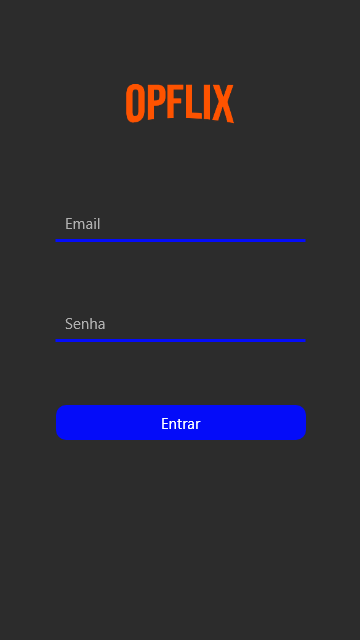
1. O cliente poderá visualizar todos os lançamentos publicados;
2. O cliente poderá selecionar por mês de lançamento, ou por nome da categoria;

Protótipos

Web



Mobile



Referências

Links

Livros