

Documentação

OpFlix

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

Sumário

1. [Resumo 2](#_Toc533767843)

[Objetivos](#_Toc533767844)

1. [Descrição do projeto 2](#_Toc533767845)

[Resumo do projeto](#_Toc533767846)

1. [Modelagem de Software 3](#_Toc533767847)

[Modelo Lógico](#_Toc533767848)

[Modelo Físico](#_Toc533767849)

[Modelo Conceitual](#_Toc533767850)

[Cronograma](#_Toc533767851)

1. [Funcionalidades 5](#_Toc533767852)

[Web](#_Toc533767853)

[Mobile](#_Toc533767854)

1. [Protótipos 6](#_Toc533767855)

[Web](#_Toc533767856)

[Mobile](#_Toc533767857)

1. [Front-End 7](#_Toc533767858)
2. [Mobile 8](#_Toc533767859)
3. [Arquitetura do Projeto 9](#_Toc533767860)
4. [Referências 10](#_Toc533767861)

[Links](#_Toc533767862)

[Livros](#_Toc533767863)

Resumo

Objetivos

Exibir o desenvolvimento do projeto, com os modelos e descrição deles, o avanço conforme o cronograma, desenvolvimento do front-end, back-end e aplicação mobile.

Descrição do projeto

Este projeto foi dado para que se desenvolva um aplicativo web e mobile em que haja um banco de dados, front-end e API para integração do back-end do web e mobile, contando com um banco de dados relacional.

Resumo do projeto

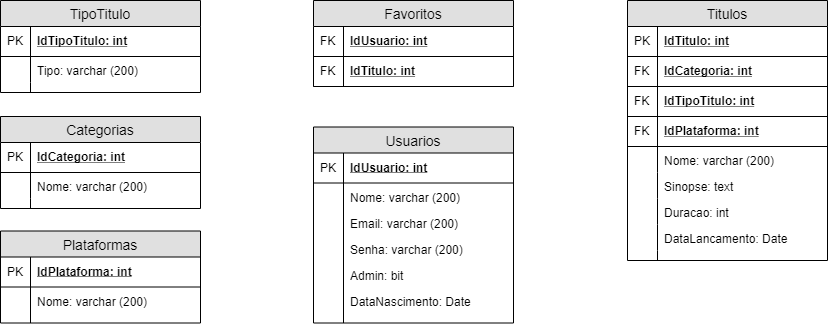
O projeto OpFlix é um aplicativo web e mobile em que tem como principal objetivo facilitar o acesso aos futuros lançamentos de filmes, séries e documentários. Com acesso aos dias restantes, categorias e plataformas em que estarão disponíveis.

Modelagem de Software

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

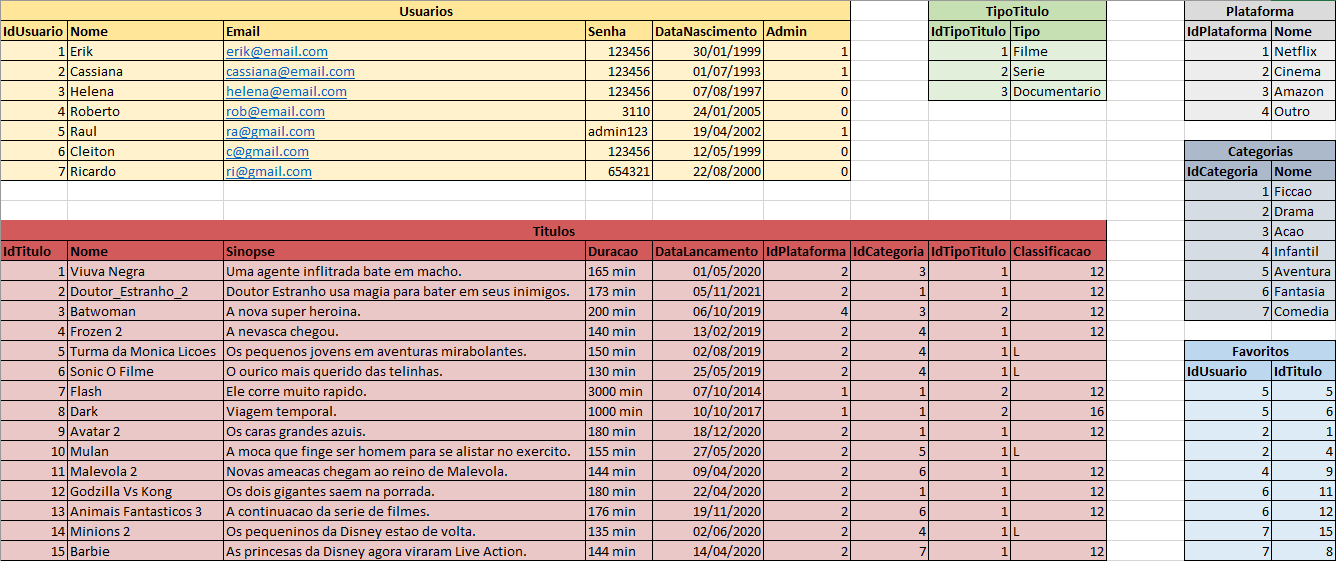
Modelo Lógico

O modelo Lógico é o modelo que mais se aproxima do banco de dados, já identificando o tipo de dado e as relações entre chave primária e secundária.



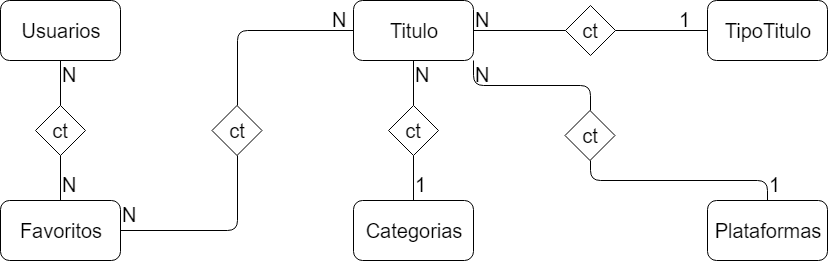
Modelo Físico

O modelo Físico é um teste para validar os valores que serão inseridos, é o modelo que o entendimento é mais fácil, já que não exige um grande entendimento de banco de dados para interpreta-lo.



Modelo Conceitual

O modelo Conceitual é uma visualização macro das entidades e dos relacionamentos, onde tais relações entre tabelas podem ser 1:1, 1:N e N:N.



Cronograma

|  | Dia 1 | Dia 2 | Dia 3 | Dia 4 | Dia 5 | | Dia 6 | | Dia 7 | | Dia 8 | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Modelo Lógico** | X |  |  |  |  | |  | |  | |  | |
| **Modelo Físico** | X | X |  |  |  | |  | |  | |  | |
| **MODELO CONCEITUAL** | X |  |  |  |  | |  | |  | |  | |
| **DDL** |  | X | X |  |  | |  | |  | |  | |
| **DML** |  | X | X |  |  | |  | |  | |  | |
| **DQL** |  | X | X |  |  | |  | |  | |  | |
| Documentação |  |  | X |  | |  | |  | |  | |  | |

Funcionalidades

Web

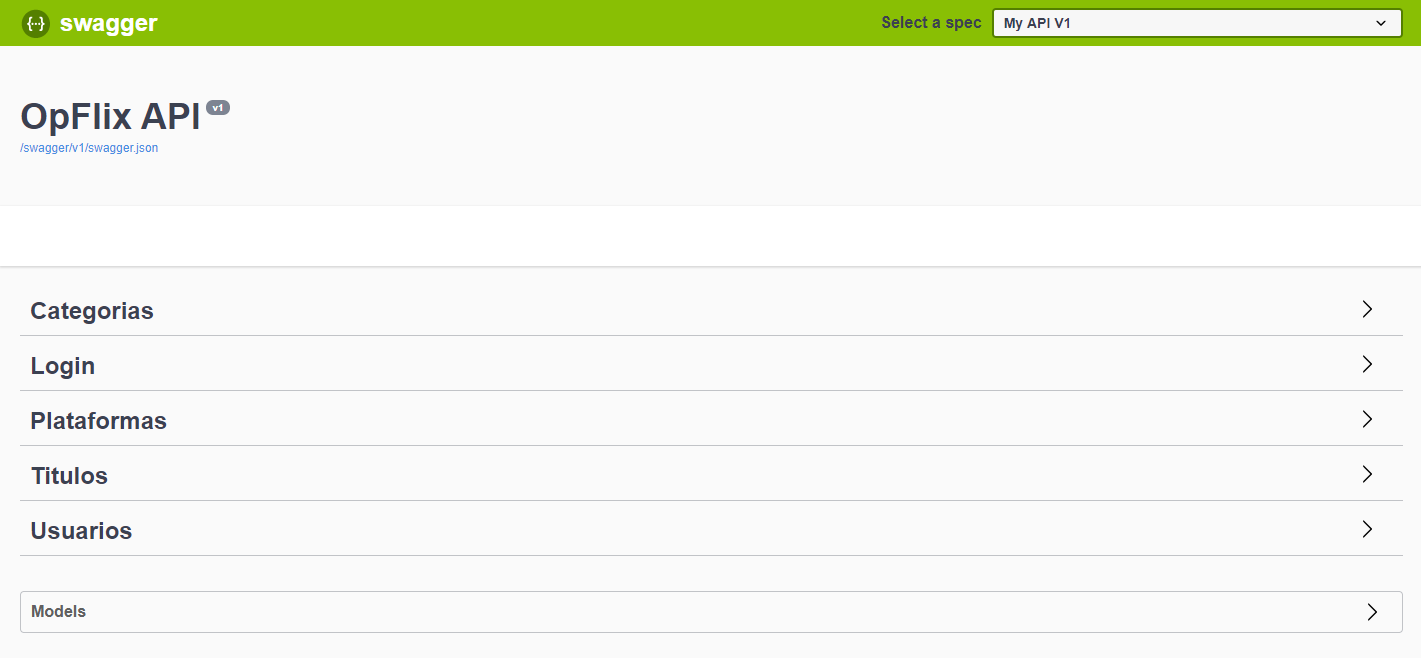
A aplicação tem como funcionalidades o Login, para diferenciação de permissão entre usuário e administrador, pessoas não logadas podem criar contas e após logadas podem consultar todos lançamentos registrados no sistema, filtrar por datas, além disto podem colocar lançamentos como favoritos e exibir quais lançamentos estão como favoritos. Administradores podem cadastrar/alterar/deletar novos lançamentos, categorias e plataformas.

Mobile

Por conta da aplicação ter sido desenvolvida em API as funcionalidades feitas no back-end são tanto para web como para mobile, sendo assim usuários mobile terão acesso as mesmas funcionalidades que os usuários web.

**Swagger**

Para acessar o Swagger com todos end-points do projeto basta acessar o link: <http://localhost:5000/swagger/index.html>



**Postman**

Para importar o postman basta clicar no botão de importar no canto superior esquerdo e selecionar o Arquivo “Senai.OpFlix.postman\_collection.json”.

**Pacotes Nuget usados**

-Swashbuckle.AspNetCore 4.0.1

-System.IdentityModel.Tokens.Jwt 5.5.0

-Microsoft.VisualStudio.Web.CodeGeneration.Design 2.1.9

-Microsoft.EntityFrameworkCore.SqlServer 2.1.11

-Microsoft.AspNetCore.Razor.Design 2.1.2

-Microsoft.AspNetCore.Authentication.JwtBearer 2.1.1

**Arquitetura do projeto**

WebApi.

**Criação do Banco de Dados**

Para criar o banco de dados você deve ter instalado o “Microsoft SQL Server Management Studio 18”, sendo assim basta abrir o arquivo “DDL-1”, na pasta “BD”, e executar os comandos, em seguida faça o mesmo com o “DML-2”.

**Executar back-end**

Para executar o back-end você deve ter instalado o “Visual Studio 2017”, sendo assim basta abrir o arquivo “Senai.OpFlix.WebApi.sln” na pasta “Senai.OpFlix.WebApi” e pressionar a tecla F5 ou o botão 

**Ferramentas usadas para desenvolvimento**

Microsoft SQL Server Management Studio 18;

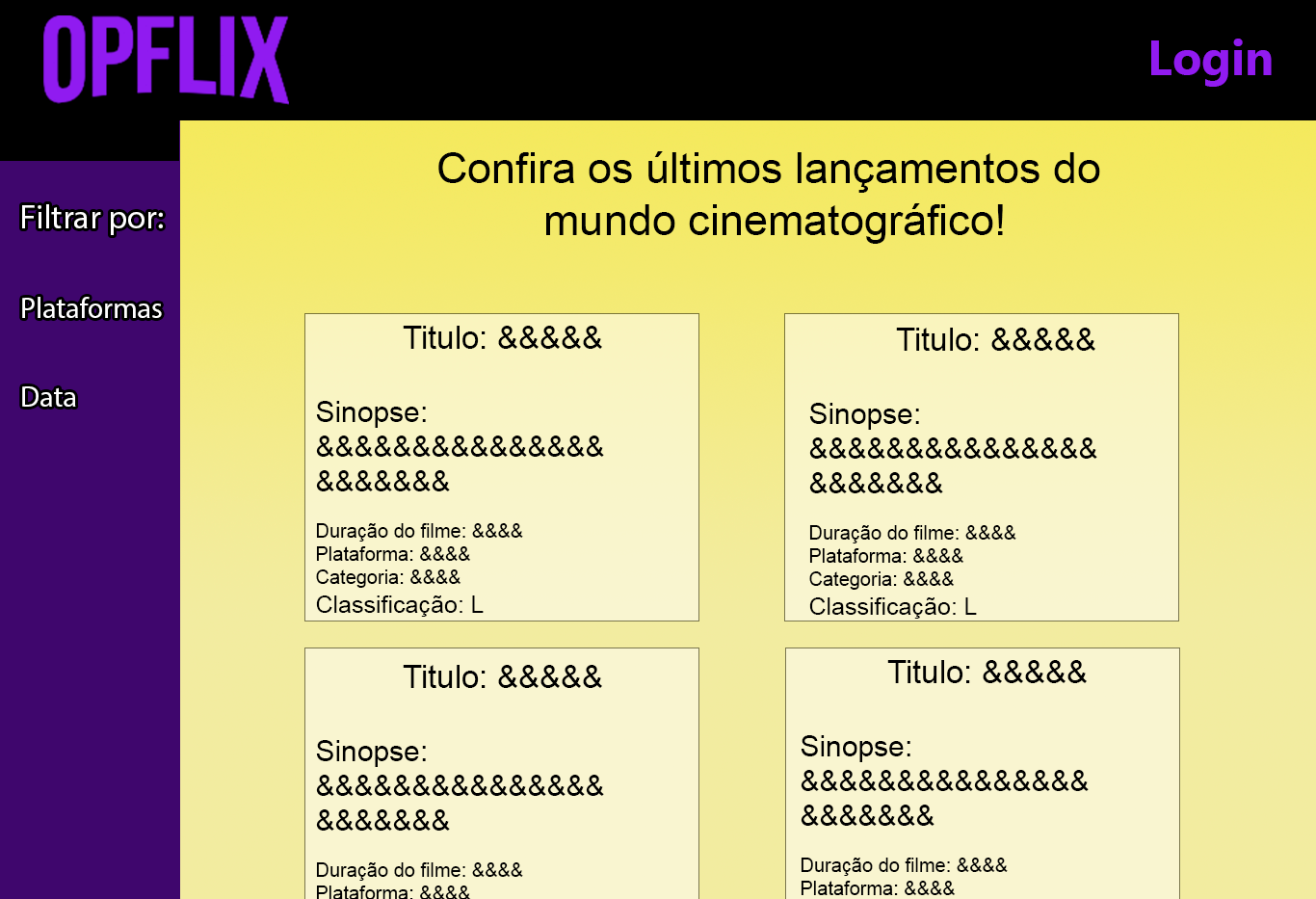
Visual Studio 2017;

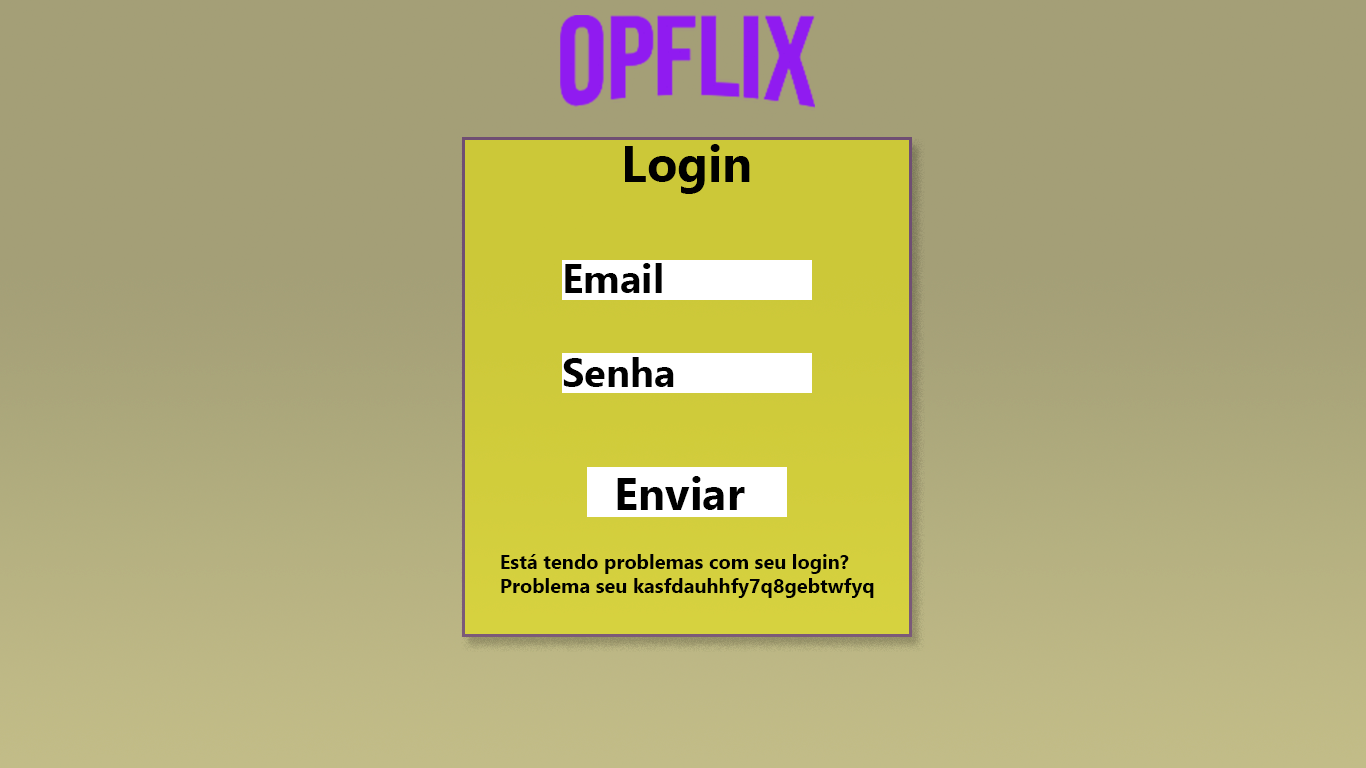
Postman.

Protótipos

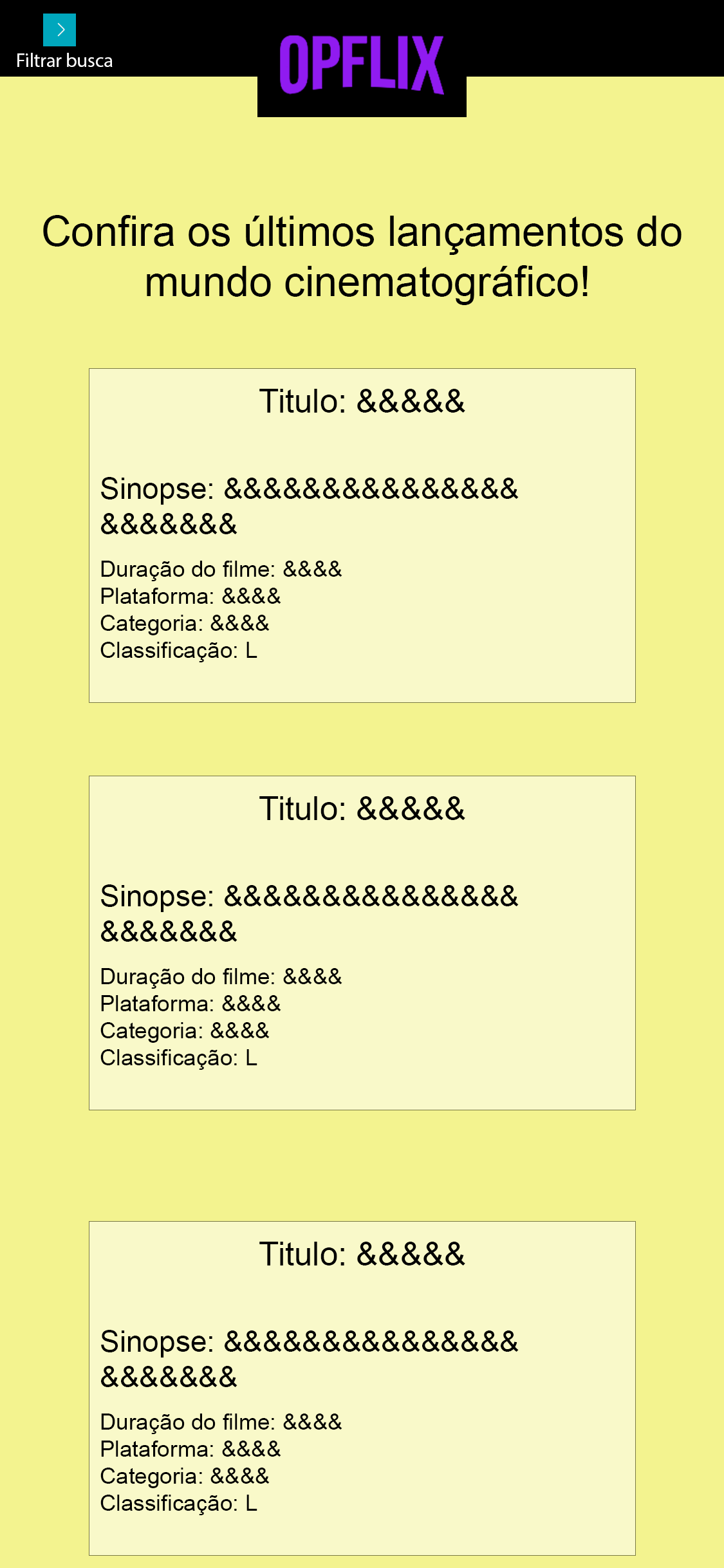
Web

Main:

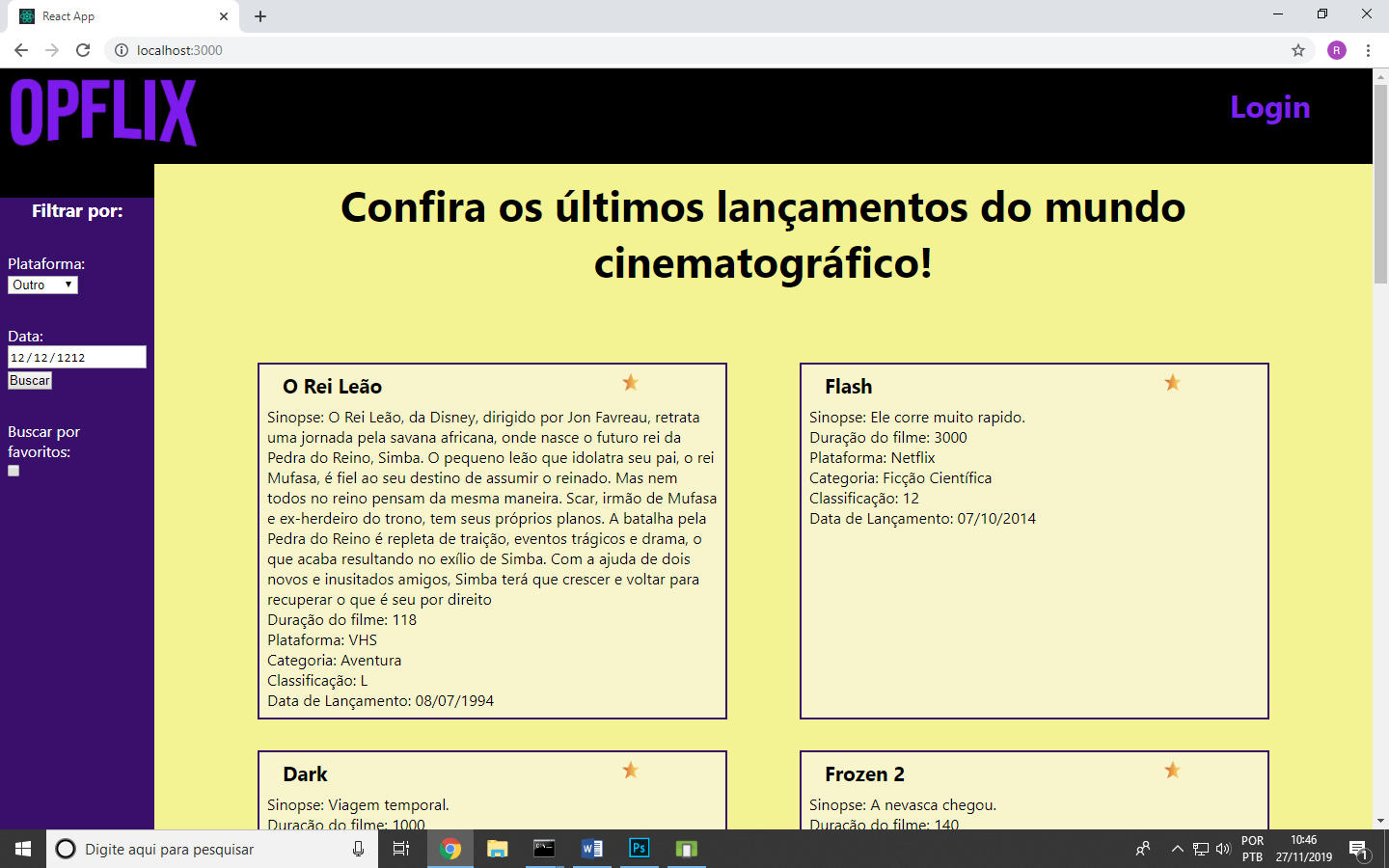


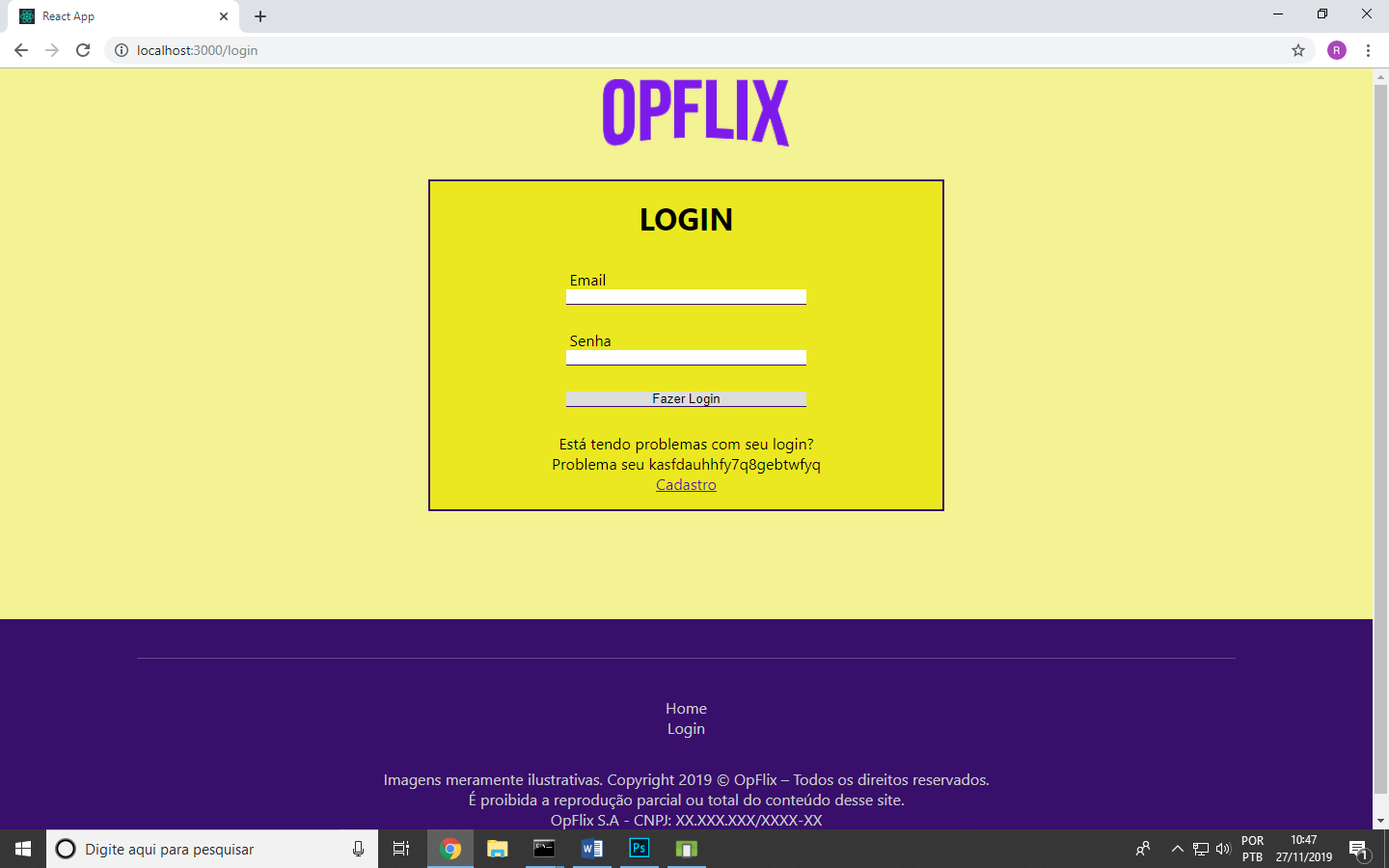
Login: 

Mobile



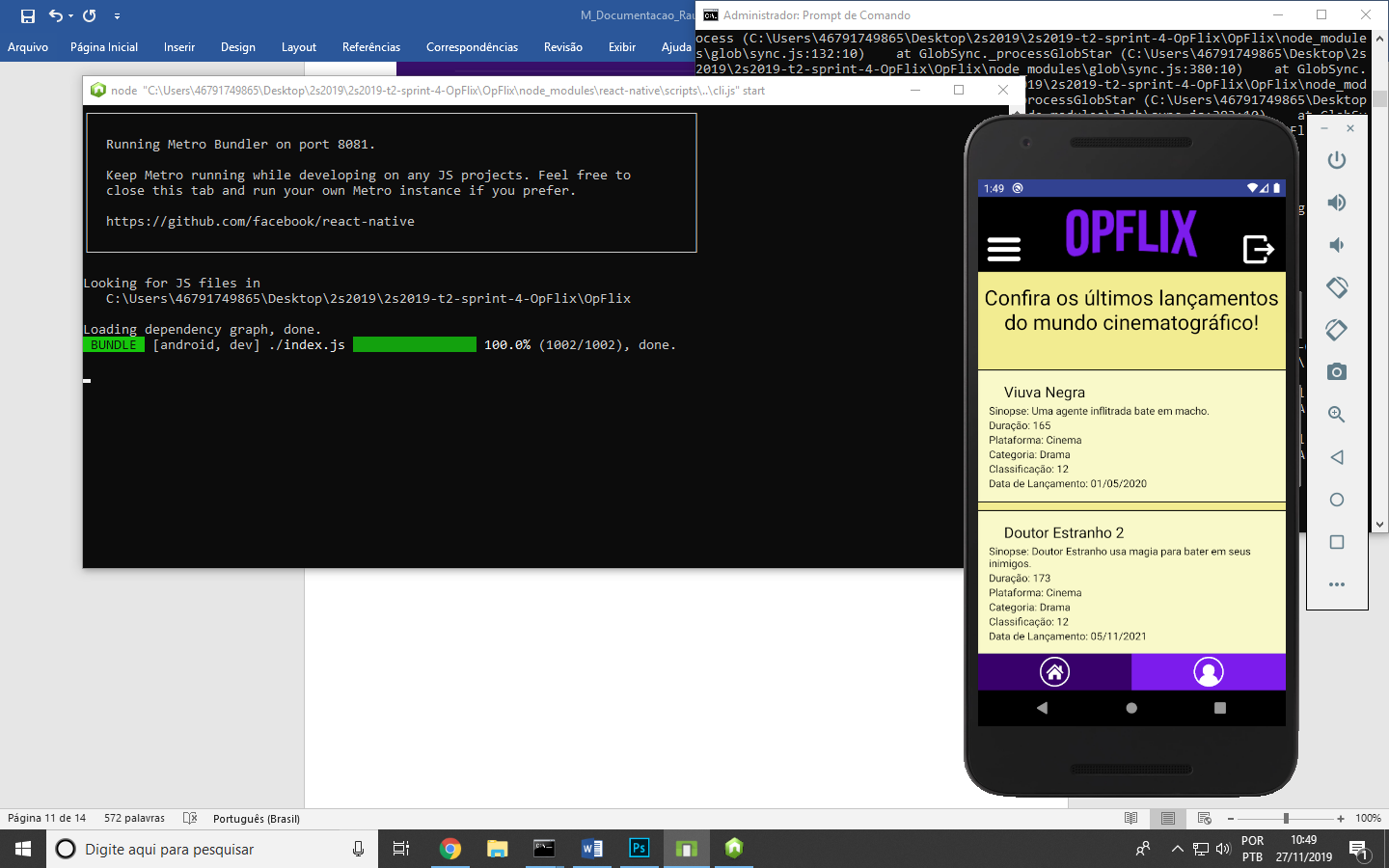
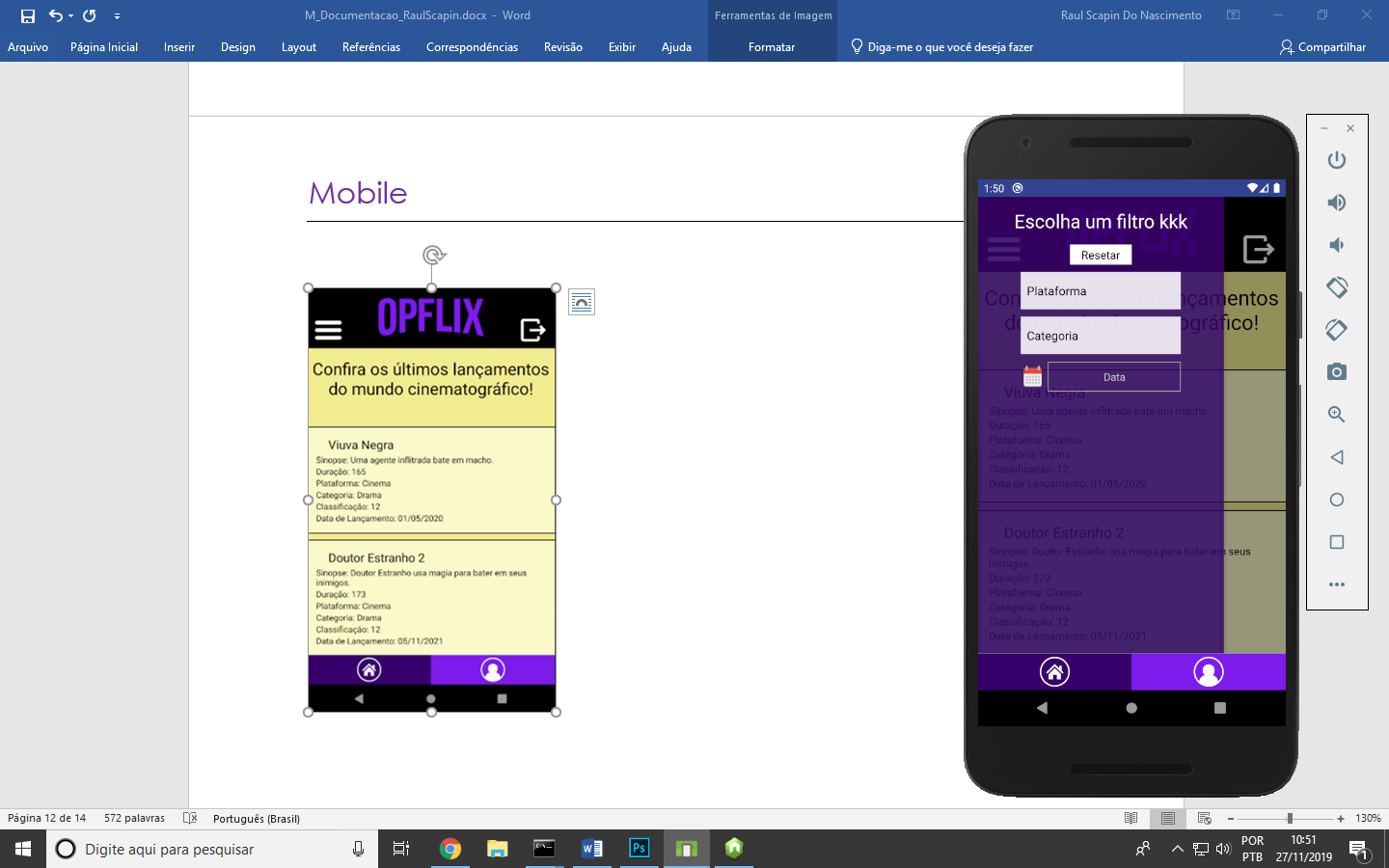
Front-End

Main:

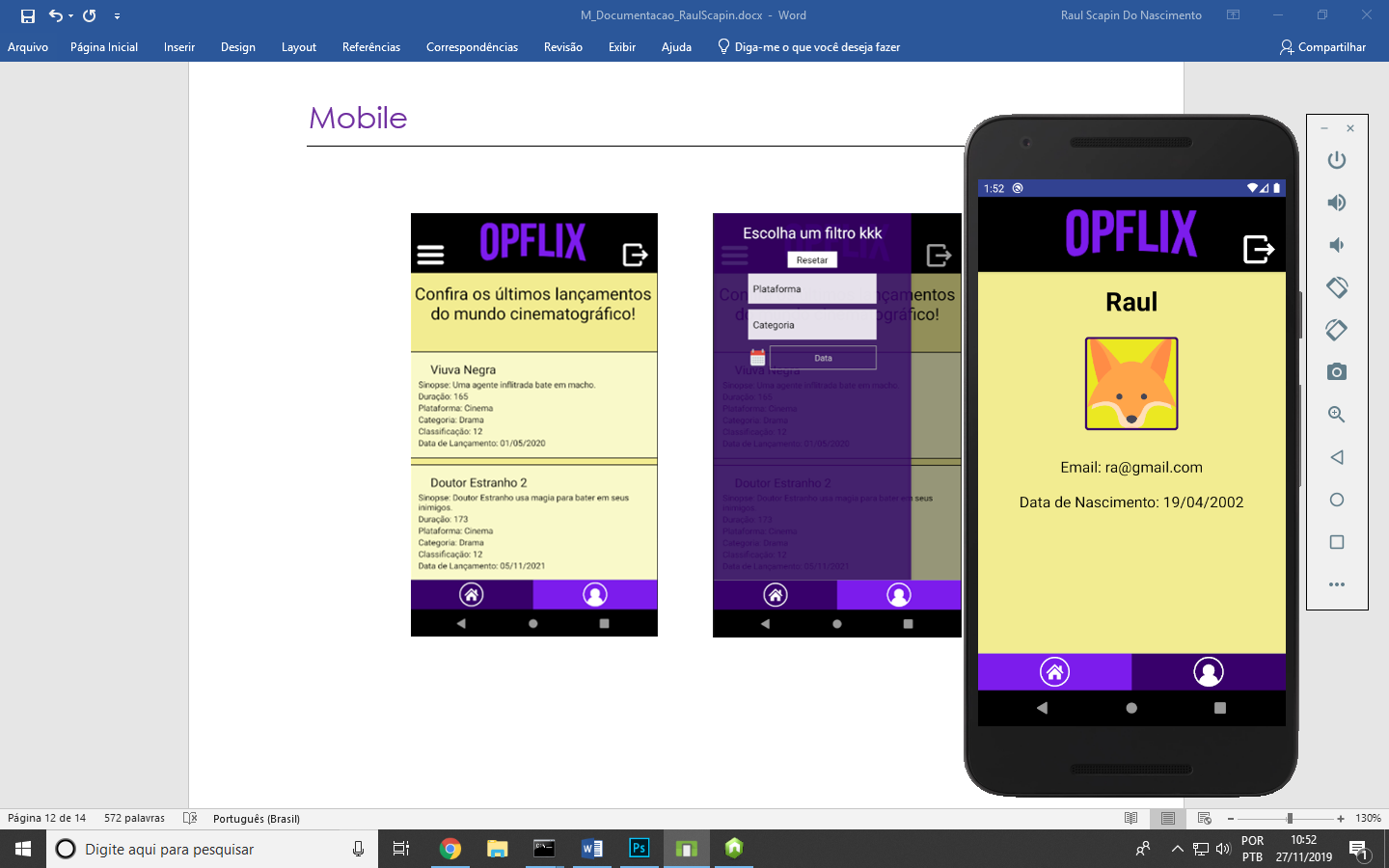
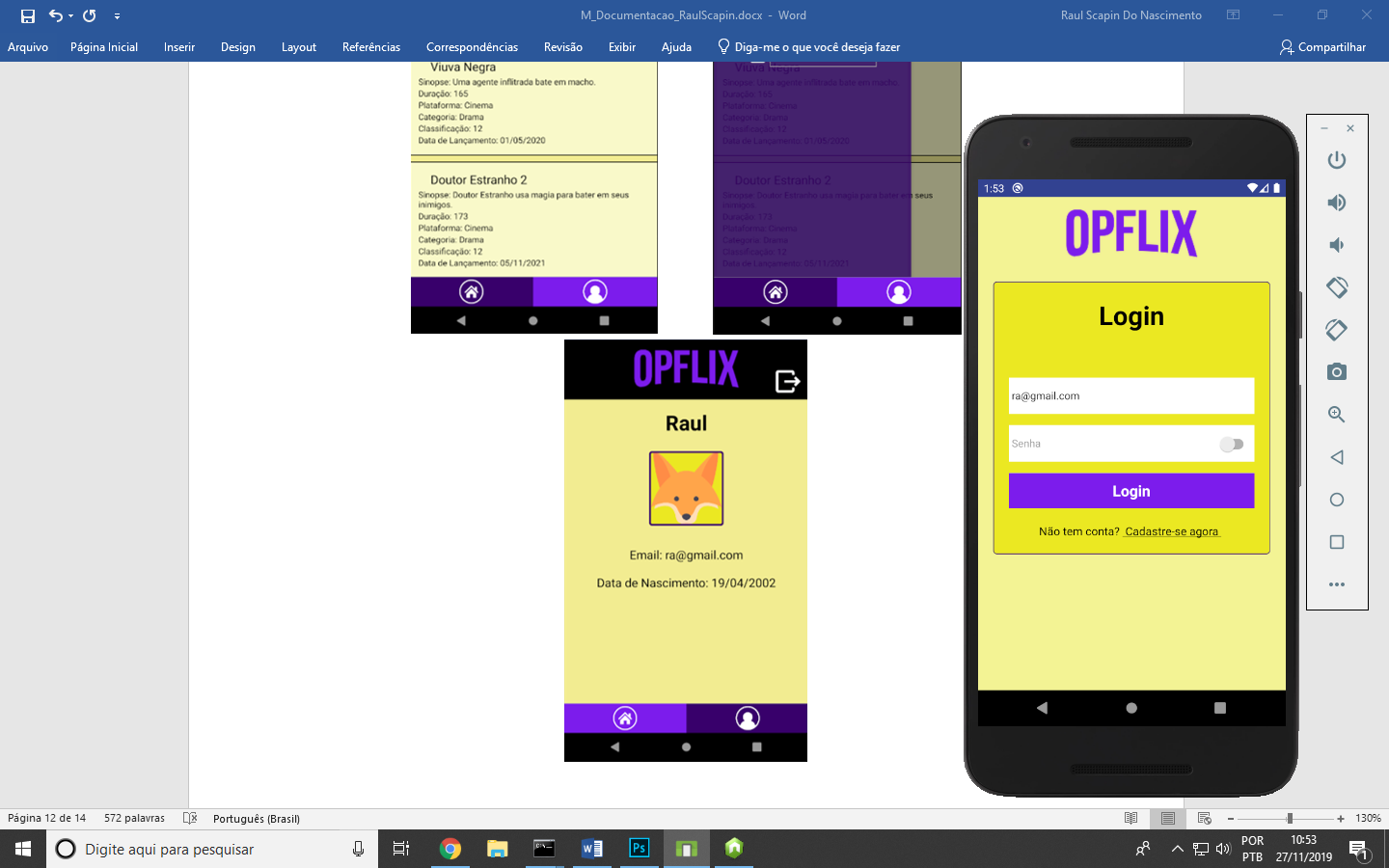
Login:

Mobile

Home Home com SideBar aberto

Página de Usuário Login

Arquitetura do Projeto

Referências

Links

Livros