

# Megadados - FetchFlix

## 1. Descrição

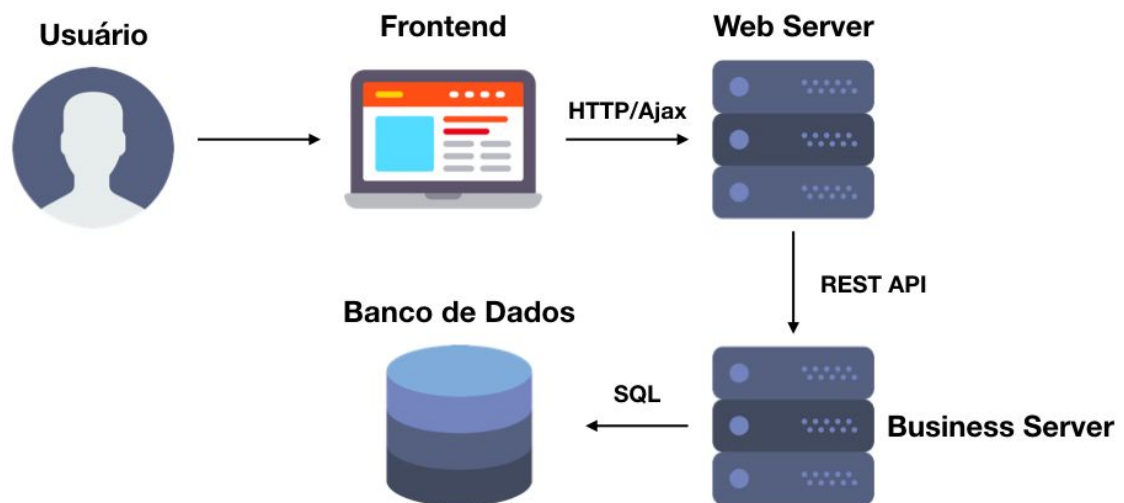
O projeto será um banco de dados com informações sobre séries. Ele será usada para obter informações, como por exemplo: onde assistir, nota média, filtrá-las de acordo com o gênero, ator, produtora, e as as preferências do usuário.

## 2. Funcionalidades

- Cadastrar usuários
- Filtrar séries de acordo com o gênero
- Filtrar séries de acordo com o ator
- Filtrar séries de acordo com a produtora
- Adicionar séries na lista de vistos
- Dar nota às séries vistas
- Listar séries de acordo com múltiplas condições (por exemplo: vistas com determinado ator, da produtora A e de romance...)

Github: <https://github.com/gabsmoreira/FirstBigdata>

### 3. Arquitetura do Sistema



O frontend foi feito em JavaScript, usando a biblioteca React js. Há um arquivo (*auth.js*) que é o web server (feito em JavaScript) e faz os *requests* (descritos na API REST, item 6 do documento) para o business server (*connect.py*). Esse último é feito em Python, com o uso de Flask. E é nesse arquivo que é feita a conexão com o banco de dados no MySQL, e comunicam-se por *queries*.

Obs.: O script *megadados.sql* cria o banco de dados, as tabelas, o trigger e o stored procedure, e o *insertData.sql* insere os dados iniciais para teste.

### 4. Dicionário de Dados

Tv\_Show

- id: número de identificação
- name: nome da série
- id\_producer: chave estrangeira identificando a produtora
- number\_of\_seasons: inteiro representando o número de temporadas

- avg\_score: nota média da série (há um trigger que atualiza esse valor sempre que uma série é adicionada como vista, ou seja, adicionada na tabela Rel\_Tv\_show\_Watcher)
- where\_to\_find: Nome do canal ou serviço de streaming onde pode-se assisti-la
- download\_link: link para download
- photo: Cartaz da série, em binário

#### Producer

- id: número de identificação
- name: nome do produtor

#### Genre

- id: número de identificação
- name: nome do gênero da série (ação, romance, suspense...)

#### Watcher

- id: número de identificação
- username: nome do usuário
- password: senha do usuário

#### Rel\_Tv\_show\_Watcher

- id\_watcher: número de identificação do usuário
- id\_tv\_show: número de identificação da série
- score: nota dada à série (por esse usuário) 1-5

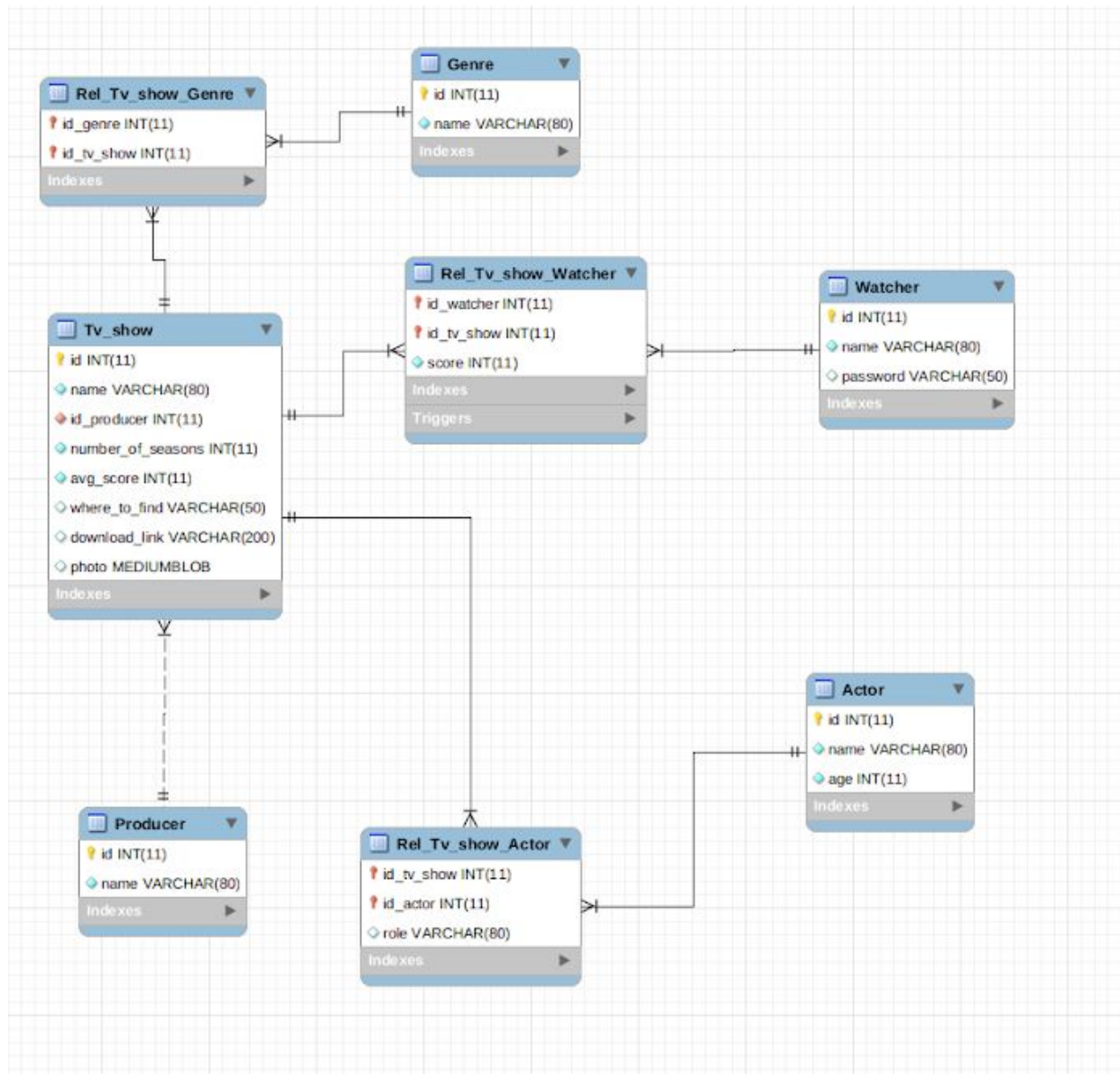
#### Rel\_Tv\_show\_Actor

- id\_tv\_show: número de identificação da série
- id\_actor: número de identificação do ator
- role: papel do ator na série

#### Rel\_Tv\_show\_Genre

- id\_tv\_show: número de identificação da série
- id\_genre: número de identificação do gênero

## 5. Diagrama E-R



## 6. REST Api

Resources/Methods	GET	POST	PUT	DELETE
/users	<p>Retorna a lista de todos os usuários</p> <p><b>RETURN:</b>  <b>Status - OK:</b>            ((id_user1, name...), (id_user2, name2)...)         </p> <p><b>Status - Erro:</b>  <b>Nada</b> </p>	<p>Cria um novo usuário</p> <p><b>Parâmetros:</b>            username;            senha         </p> <p><b>RETURN</b>  <b>Status - OK</b> </p> <p><b>Status - Erro:</b>  <b>403</b> </p>		<p>Deleta Usuário</p> <p><b>Parâmetros:</b>            username         </p> <p><b>RETURN</b>  <b>Status - OK</b> </p> <p><b>Status - Erro :</b>  <b>403</b> </p>
/login		<p>Retorna se o usuário existe ou não</p> <p><b>Parâmetros:</b>            username;            senha         </p> <p><b>RETURN</b>  <b>Status - OK</b>            Retorna 1 ou 0         </p> <p><b>Status - Erro:</b>  <b>403</b> </p>		
/tvshows	<p>Retorna a lista de todas as séries</p> <p><b>RETURN:</b>  <b>Status - OK:</b>            ((id_tv_show1, name...), (id_tv_showr2,         </p>			

	name2)...) Status - Erro: Nada			
--	-----------------------------------	--	--	--

Resources/Methods	GET	POST	PUT	DELETE
/search		Retorna a lista de séries que se encaixam nos parâmetros. <b>Parâmetros</b> <b>search:</b> palavra chave (que será procurada entre os atores e nomes de série); <b>genre:</b> gênero da série;  <b>RETURN:</b> Status - OK: ((id_tv_show1, name...), (id_tv_showr2, name2)...) Status - Erro: Nada		
/seen	Retorna a lista de filmes já vistos do usuário. <b>Parâmetros:</b> <b>user_id:</b> id do usuário;  <b>RETURN:</b> Status - OK: ((id_tv_show1, name...),	Retorna se uma determinada série foi vista ou não pelo usuário. <b>Parâmetros:</b> <b>user_id:</b> id do usuário; <b>id_film:</b> id da série;  <b>RETURN</b> Status - OK Retorna 1 ou 0	Usado para adicionar uma série na categoria de "visto" pelo usuário. <b>Parâmetros:</b> <b>user_id:</b> id do usuário; <b>id_film:</b> id da série;  <b>RETURN</b>	

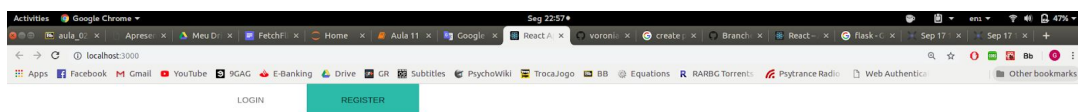
	(id_tv_showr2 , name2)...)	<b>Status - Erro:</b> <b>403</b>	<b>Status - OK</b>  <b>Status - Erro:</b> <b>403</b>	
	<b>Status - Erro:</b> <b>Nada</b>			

## 7. Features SQL

Foi usado, de SQL:

- Creates, inserts e selects
- Tabelas temporárias e join tables (com agrupamento) para as queries do sistema de buscas
- Stored procedure para diminuir a repetição na criação de tabelas temporárias para as queries citadas acima
- Trigger para a atualização da média de avaliações de cada série na criação de uma linha na tabela de relacionamento de séries com usuários

## 8. Manual de Uso



Username

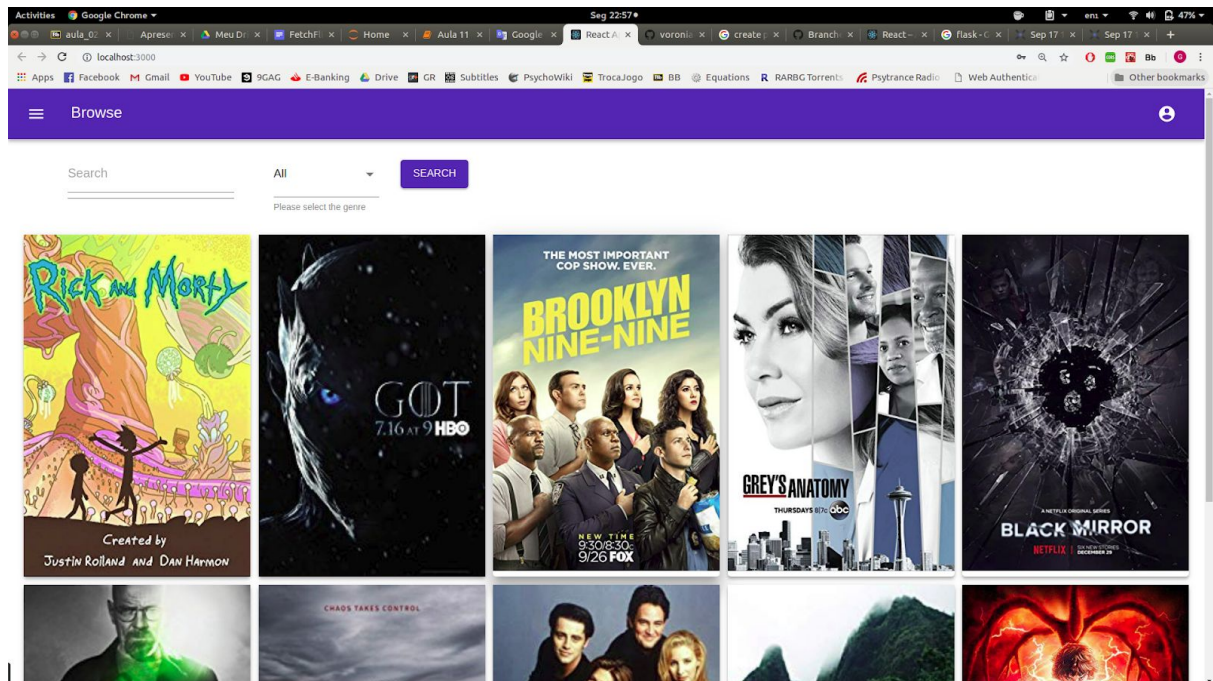
Password

Confirm password

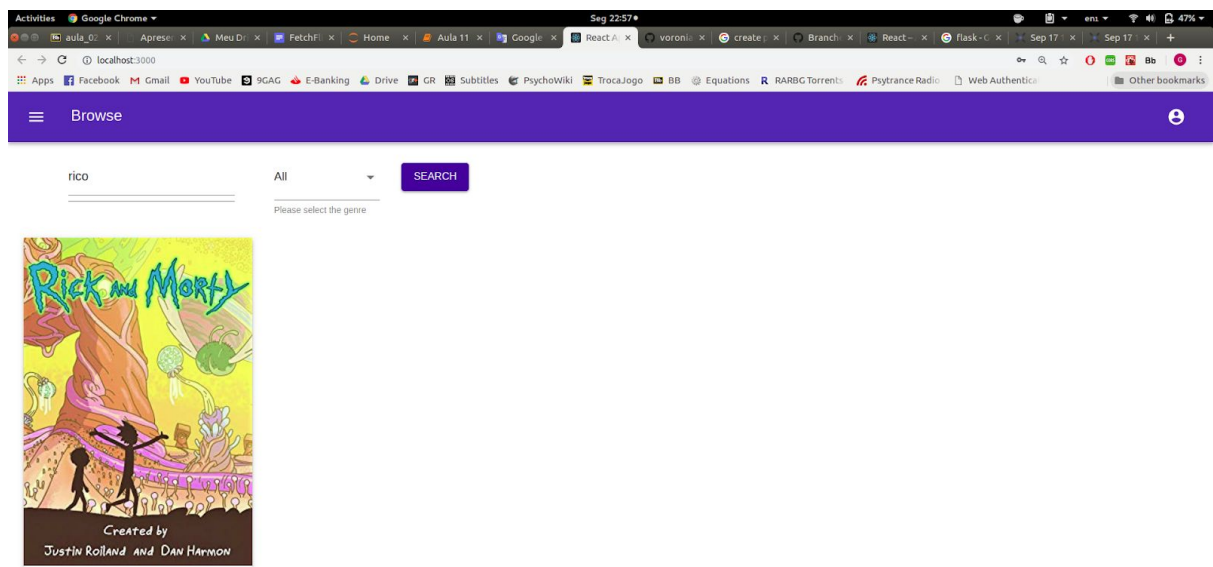
REGISTER

|

*Pagina de login e registro de novos usuários. Ao se registrar o usuário fica automaticamente logado*

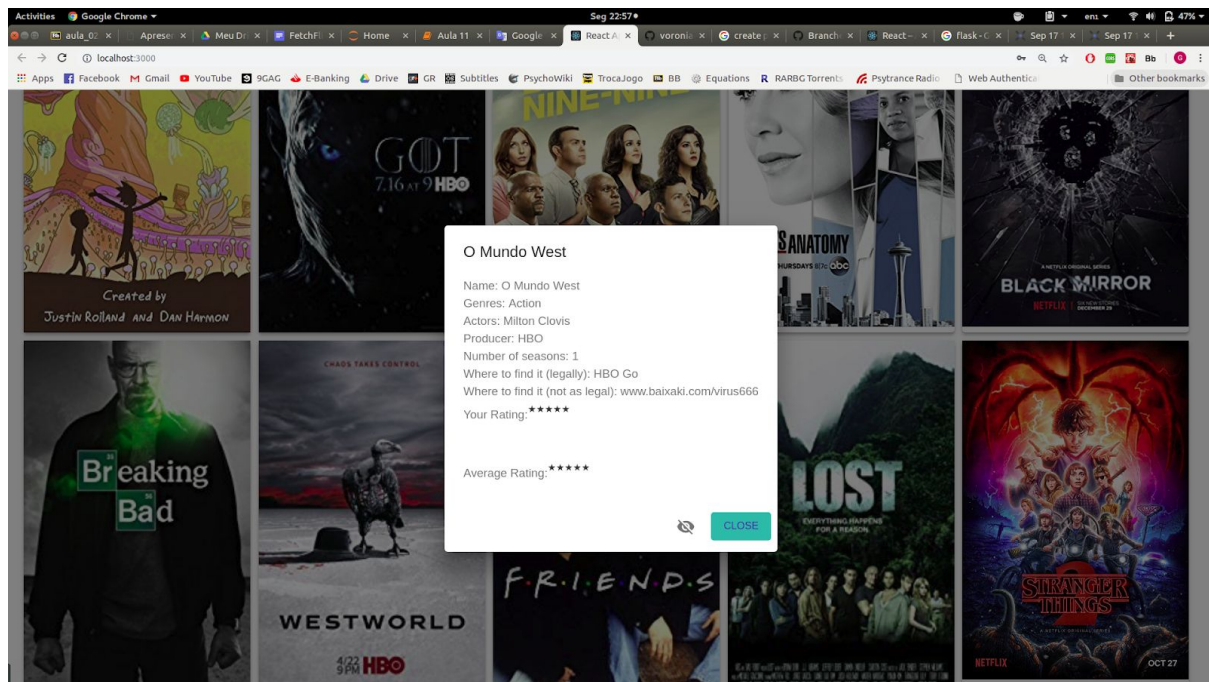


*Pagina principal, lista todas as séries. Para fazer buscas o usuário pode escrever parte do nome da série ou de atores no lugar de "Search" e/ou escolher um gênero no dropdown e clicar no botão Search.*

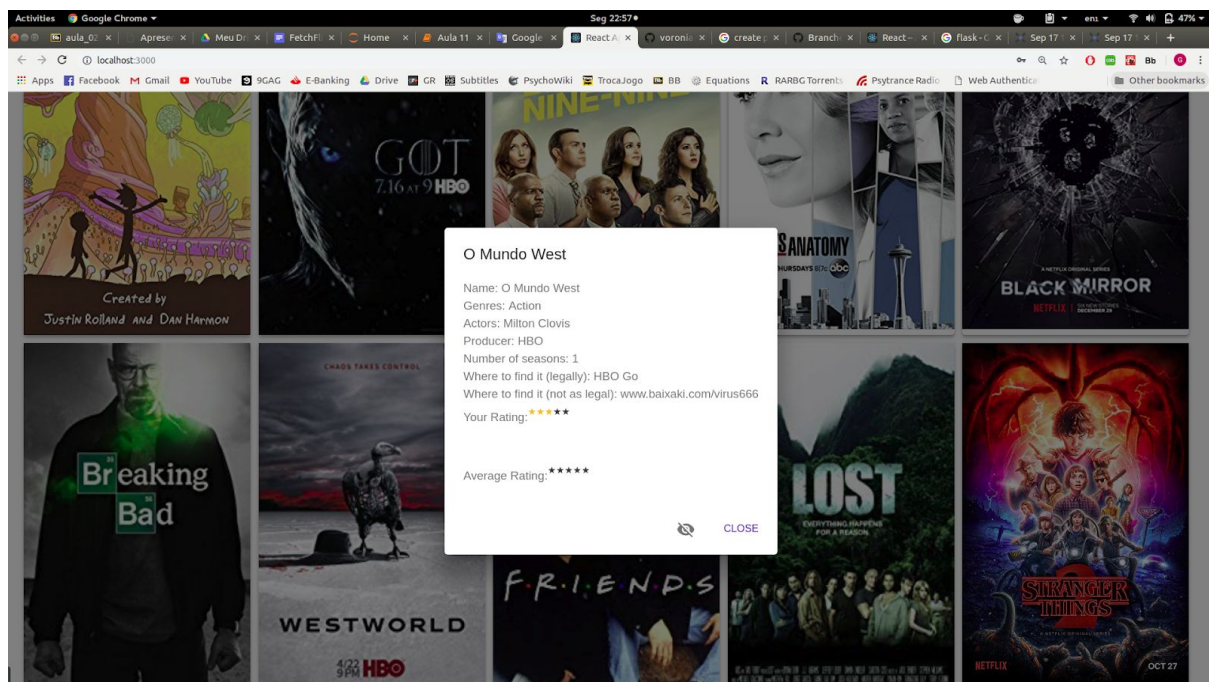


*Exemplo de busca. (o nome da série é Rico e Morte)*

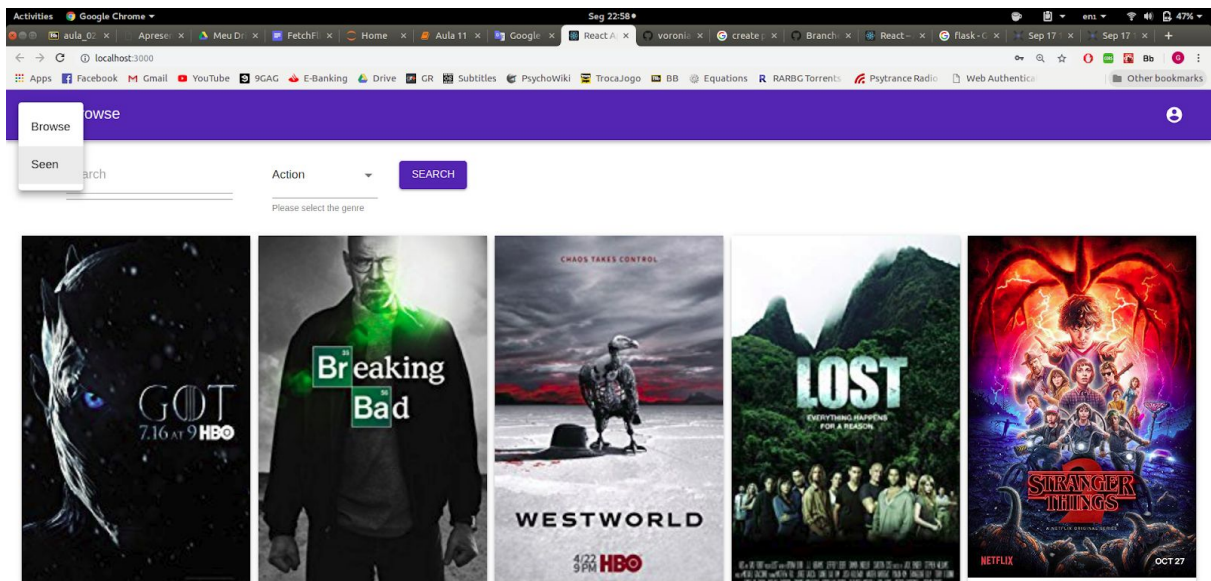




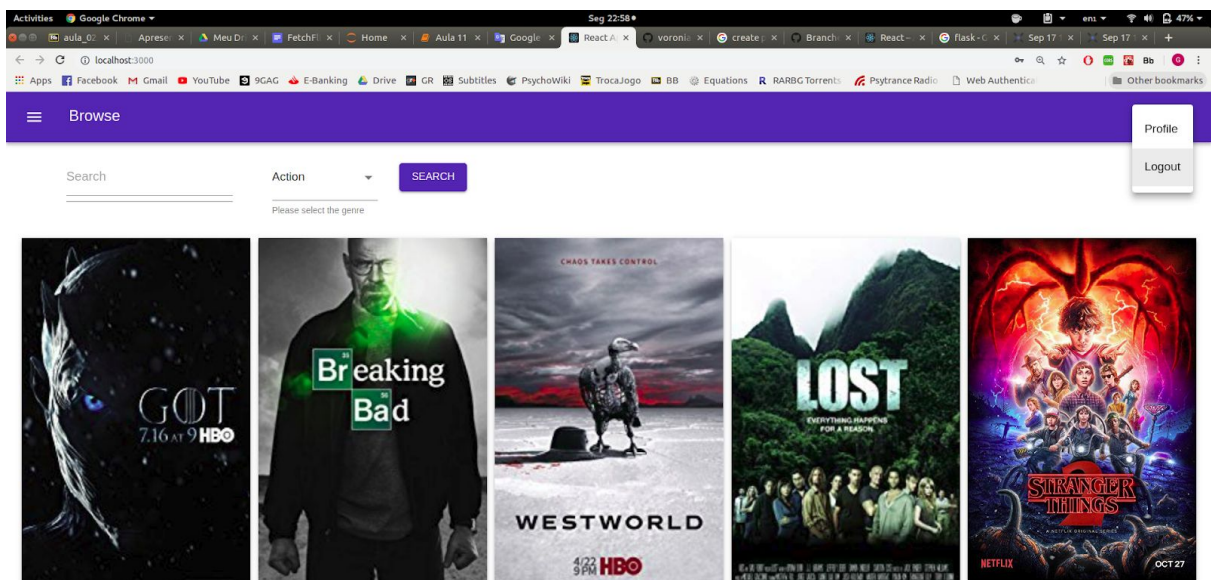
*Ao clicar em um cartaz, informações sobre a série aparecem em um pop-up.*



*Para adicionar uma serie como vista, o usuário deve avaliá-la clicando nas estrelas ao lado de "Your Rating" e clicar no icone ao lado de "CLOSE". Note que ele mudará para o representar que o usuário viu a série.*



*Para ver as séries marcadas como vistas, basta mudar de "Browse" para "Seen" no menu do canto superior esquerdo*



*Para dar logout, usa-se a opção no menu do canto superior direito.*