

Relatório

Nome do projeto: WatchMe

Grupo:

Arthur Oliveira Passos - 1272022772

Filipe Souza Alves - 1272023233

Kaled Freire Barreto - 1272023144

WatchMe

Uma plataforma de gerenciamento de séries entre outras plataformas de streaming, que permite aos usuários criar listas personalizadas de séries, aproveitando ao máximo os sistemas de streaming já cadastrados. Com essa solução, os usuários terão a possibilidade de organizar suas séries favoritas em diferentes listas, facilitando o acesso e a catalogação de acordo com o sistema de streaming em que estão disponíveis.

Nosso sistema resolve esse problema ao permitir que os usuários criem sua própria lista de favoritos entre plataformas diferentes de streaming, catalogando-as de acordo com o sistema de streaming em que estão disponíveis. Dessa forma, é possível criar listas específicas para séries disponíveis na Netflix, por exemplo, e outras para séries disponíveis no Amazon Prime Video. Isso proporcionando uma experiência de organização única e simplificada para cada usuário, de acordo com suas preferências e necessidades.

Em nossa plataforma, levamos em consideração princípios fundamentais de design para construir interfaces amigáveis e intuitivas. Utilizamos o princípio do feedback para fornecer aos usuários um retorno imediato sobre suas ações, permitindo que saibam o que está acontecendo em cada etapa. Dessa forma, garantimos que eles tenham uma experiência fluida e satisfatória ao criar suas listas de séries.

Outro princípio que aplicamos é o da consistência, mantendo elementos visuais e de comportamento coerentes em todo o design da plataforma. Isso cria um padrão reconhecível e facilita a compreensão e navegação do usuário. Além disso, seguimos o princípio da simplicidade, mantendo o design limpo e livre de elementos desnecessários, a fim de minimizar a complexidade e tornar a plataforma fácil de entender.

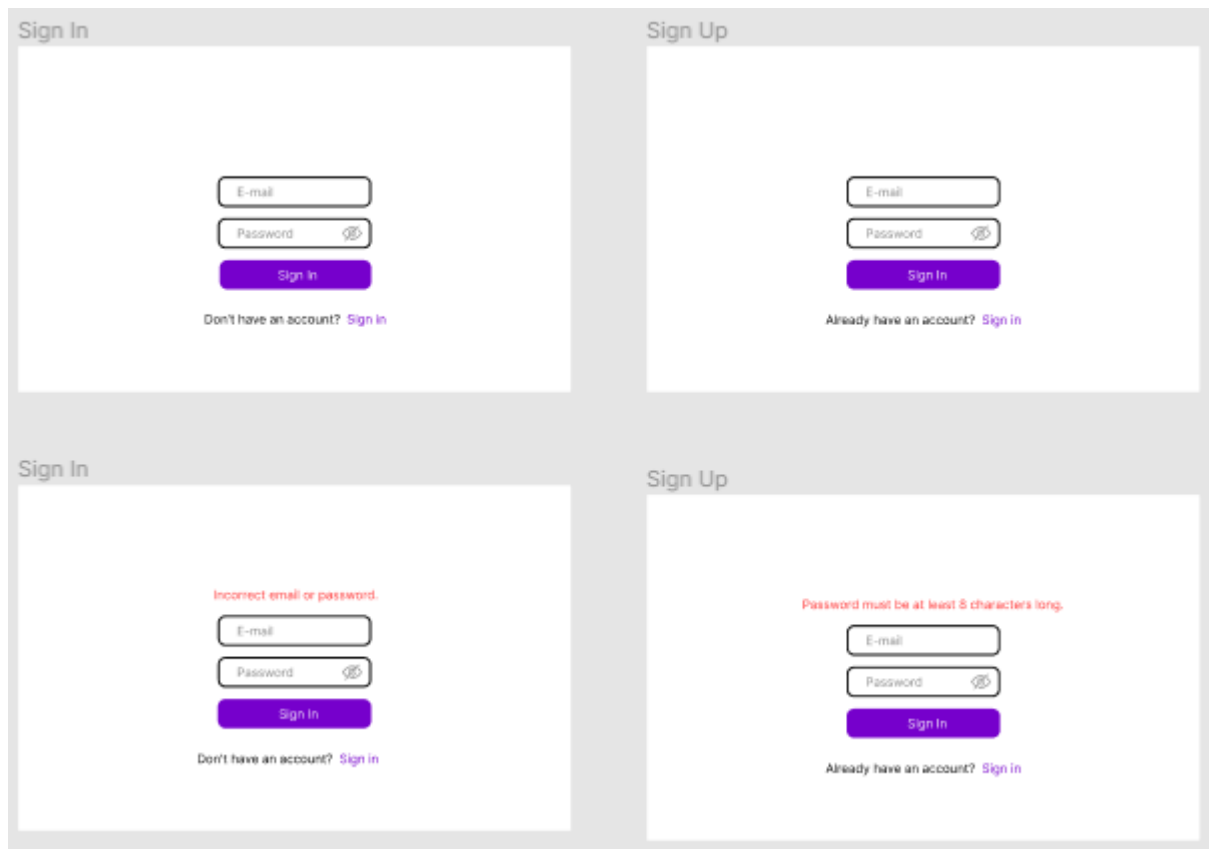
A visibilidade também é uma prioridade em nosso sistema, garantindo que os elementos importantes sejam claramente visíveis para os usuários. Isso permite que eles acessem e utilizem facilmente as funcionalidades da plataforma, sem esforço

adicional. Por fim, buscamos a eficiência, projetando a interface para que os usuários possam realizar suas tarefas de forma eficiente e eficaz, economizando tempo e esforço ao criar e gerenciar suas listas de séries.

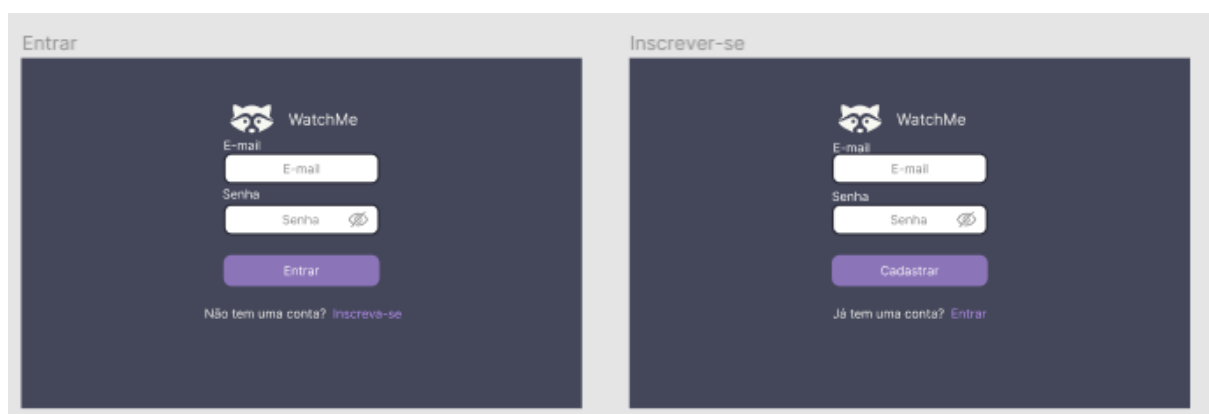
Combinando esses princípios de design com a proposta de nossa plataforma, proporcionamos aos usuários uma experiência completa, onde podem organizar, descobrir e desfrutar de suas séries favoritas de maneira intuitiva, eficiente e personalizada. Nossa equipe está empenhada em oferecer uma interface que seja amigável e atenda às necessidades dos usuários, tornando sua jornada de entretenimento ainda mais prazerosa.

Requisitos e Funcionalidades

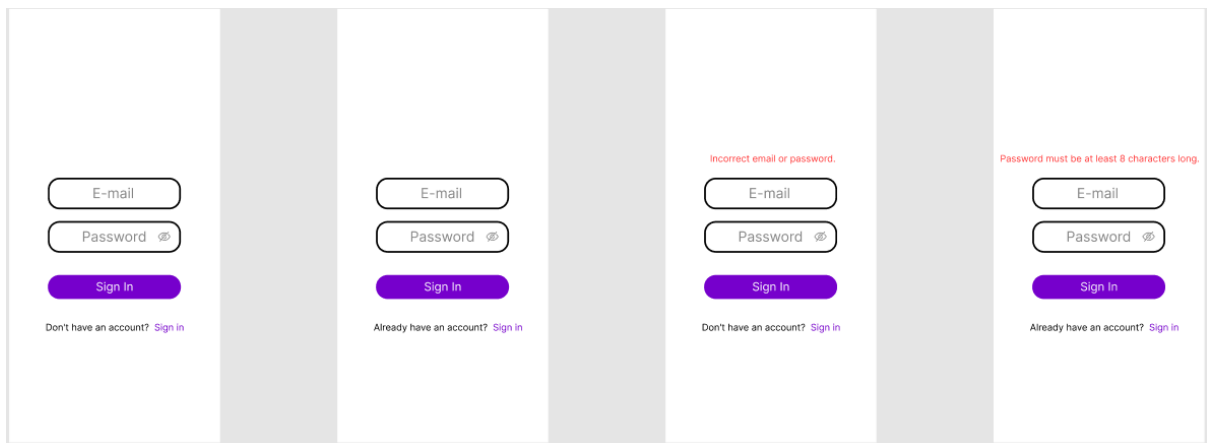
Realizar cadastro e login: O usuário poderá realizar seu cadastro na plataforma, fornecendo seus dados e após a verificação desses dados o seu cadastro será criado, caso já possua uma conta cadastrada o mesmo deve conseguir acessar o sistema.



Durante a construção da tela de login resolvemos alterar o fundo da tela para uma cor mais escura justamente se baseando em sistemas como a Netflix, Disney Plus e Prime Vídeos, e também pois um fundo mais escuro facilita e torna mais agradável a visualização/navegação dentro do sistema, o mesmo se aplica para as telas mobile.



Também adicionamos na tela as descrições dos campos de login e cadastro para que ficasse de fácil entendimento ao usuário o que ele deve preencher.

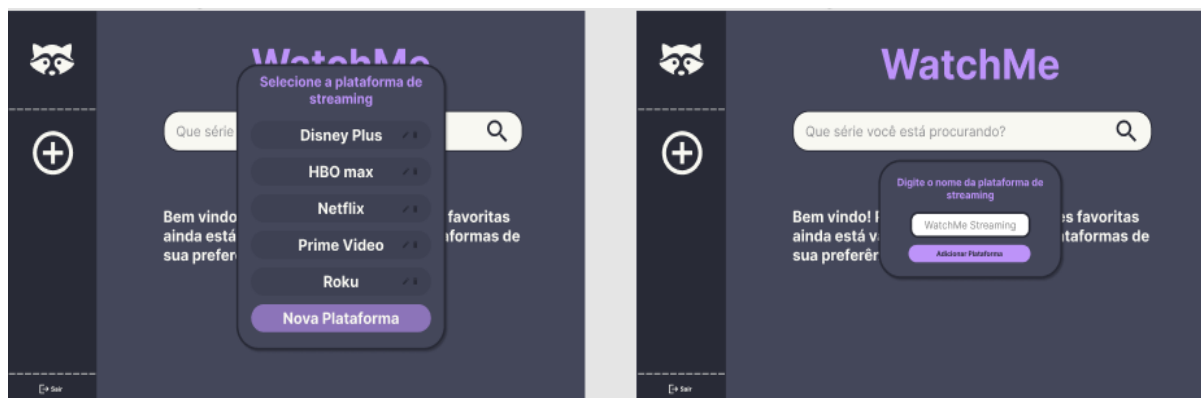


Pesquisar série: O sistema deverá ter na interface da plataforma uma barra de pesquisa onde o usuário poderá pesquisar a série que deseja pelo nome.

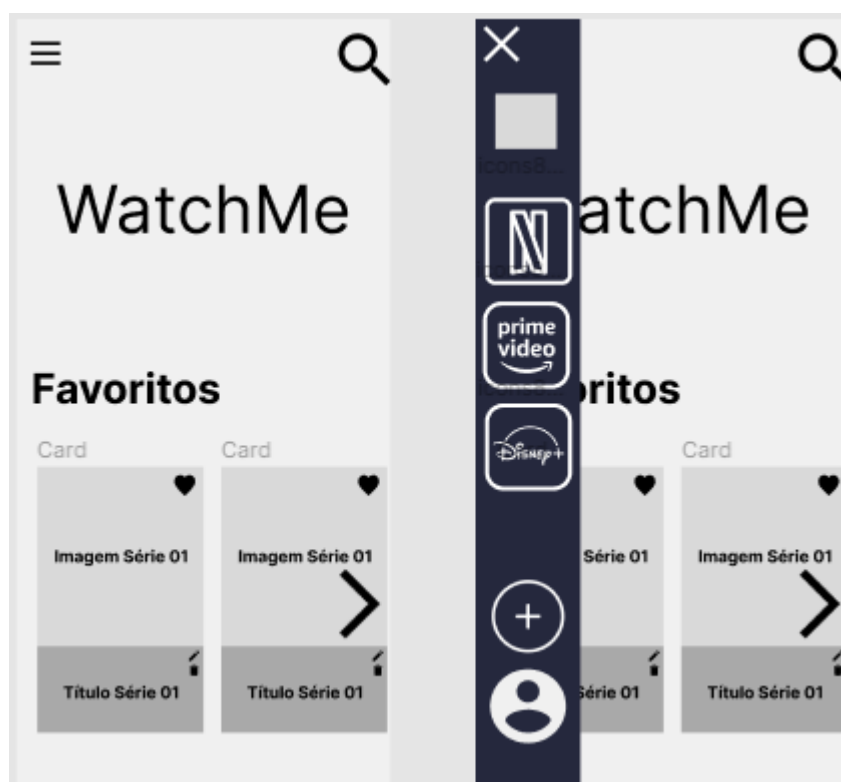
Na criação desta tela primeiramente resolvemos retirar a ícone de de perfil que se localizava no canto inferior esquerdo da tela pois como o no nosso sistema não vai ter sistema de troca de perfil resolvemos trocar por um botão de sair.

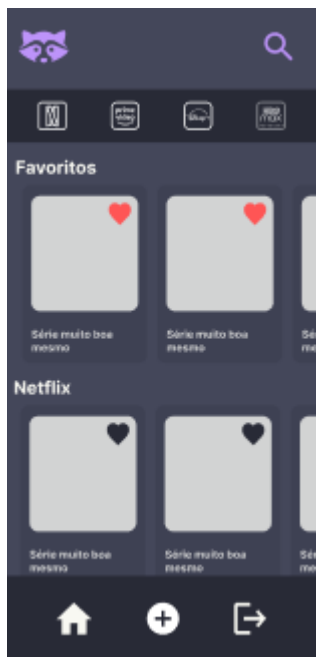


Cadastrar streaming: O sistema deve permitir ao usuário cadastrar uma plataforma de streaming apertando na funcionalidade adicionar nova plataforma..



Na tela de cadastro de streaming/série, criamos um modal no meio da tela para fácil visualização, além de manter um padrão reconhecível e facilita a compreensão e navegação do usuário, além de alterar também a visualização dos streamings cadastrados nas telas mobile, pois, a barra lateral consumia muito espaço de tela então criamos algo se baseando na barra de navegação do Disney Plus alterando para uma barra de rolagem horizontal no meio da tela, pois assim consome menos espaço de tela.

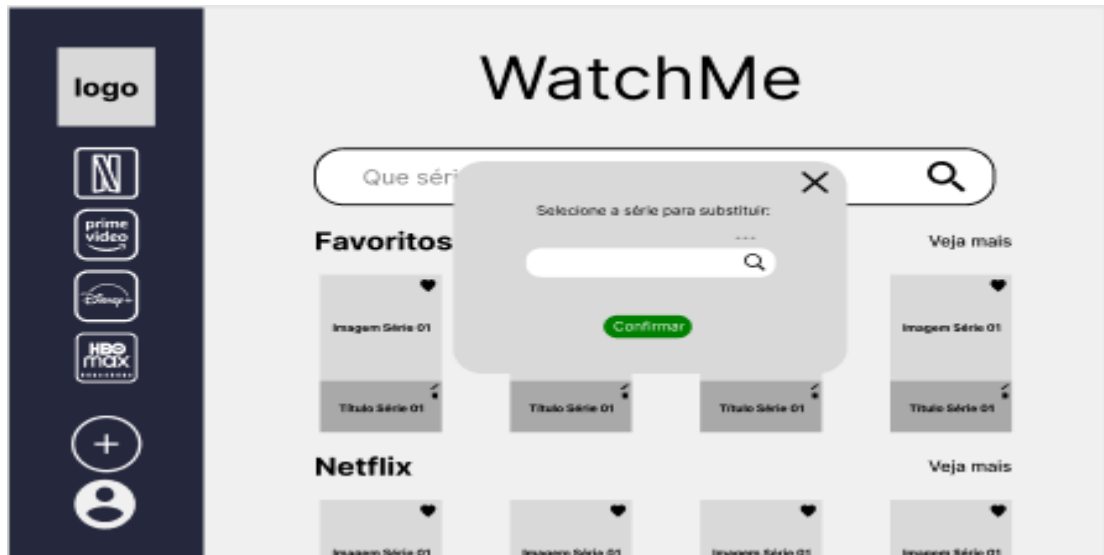




Cadastrar série: O sistema deve permitir ao usuário cadastrar uma nova série, clicando no botão de “Cadastrar nova série”.



Editar série: O sistema deve permitir ao usuário editar uma série já cadastrada, clicando no botão de editar localizada ao lado do nome da série com um símbolo de um lápis.



Alteramos o modal de edição da série devido a uma mudança na funcionalidade em si, pois no modelo atual o usuário pode alterar os dados da série cadastrada e no modelo antigo o mesmo apenas uma série adicionada na lista dele por outra cadastrada no sistema.



Filtrar séries: O sistema deve permitir ao usuário filtrar as séries pelo sistema de streaming selecionado na barra lateral localizada no lado esquerdo da tela.



Adicionar aos favoritos: O usuário pode adicionar as séries aos favoritos clicando no símbolo de coração no canto superior direito da série, para que assim o mesmo tenha suas séries favoritas de fácil acesso em tela assim que entra na plataforma.

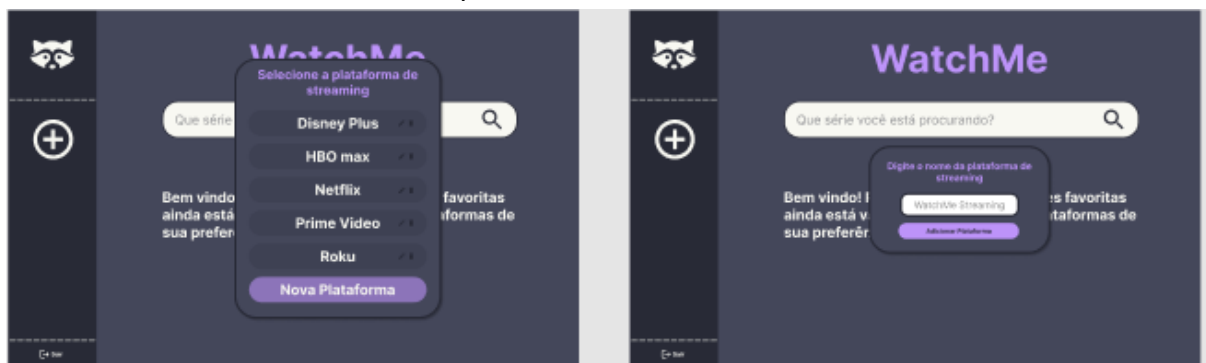


Visualização expandida: O usuário pode expandir a visualização da sua lista clicando na função veja mais que se localiza ao lado oposto do nome da lista.

No wireframe percebemos também que para dar uma noção de continuidade poderíamos colocar as séries saindo um pouco da tela para que o usuário tivesse a impressão de algo contínuo.



Editar streaming: O usuário pode editar a plataforma de streaming clicado no botão de editar com um símbolo de um lápis.

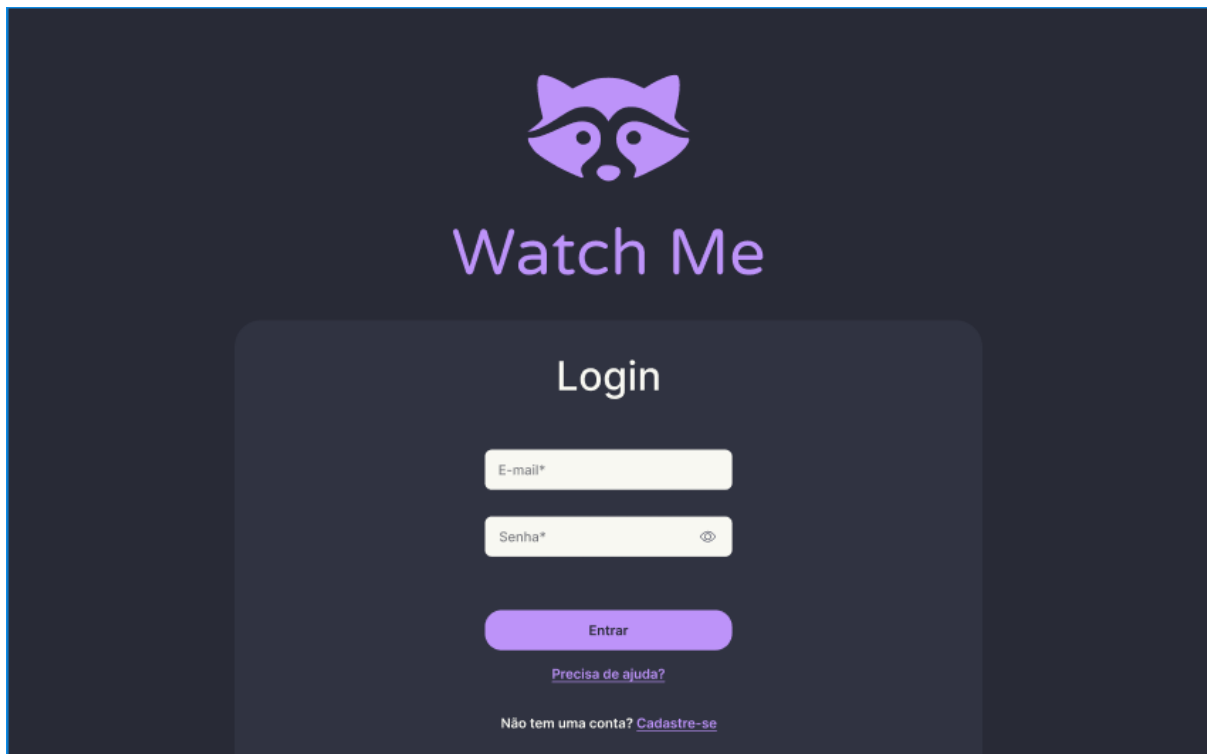


Visualizar série em específico: O usuário pode visualizar mais detalhes da série cadastrada no sistema clicando em cima da área em retângulo onde se localiza imagem da série.



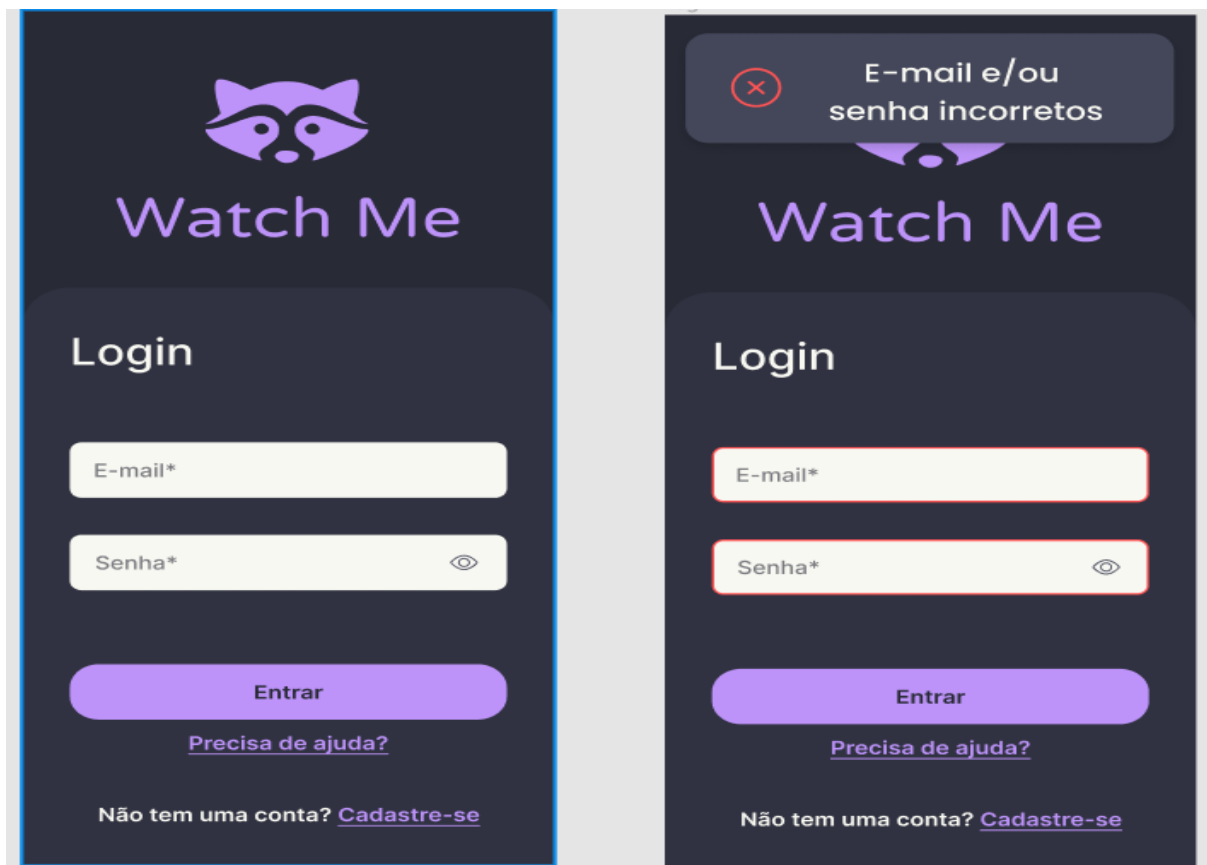
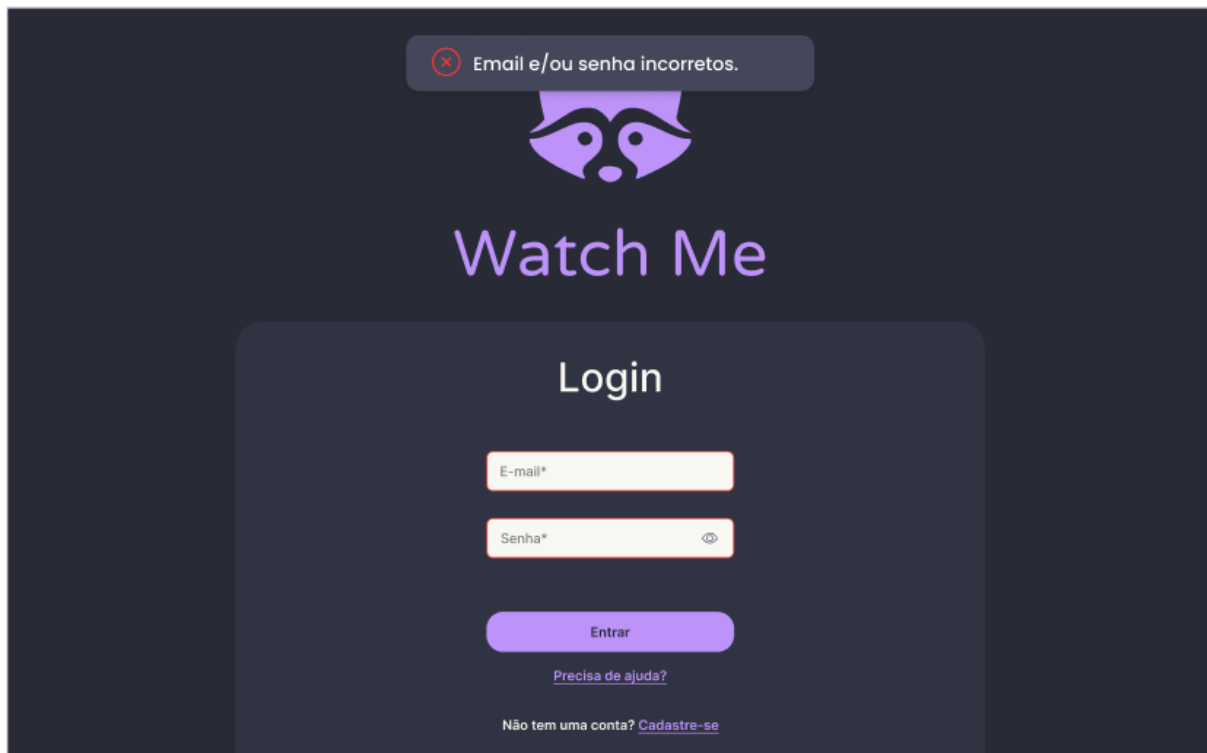
Foi retirado o botão de favoritar com um símbolo de coração pois o mesmo acabou ficando muito pequeno demais em tela(colocar uma heurística que combine com essa explicação).

Depois de primeira entrega do projeto melhoramos o design do projeto pensando em na lista de 10 heurísticas de usabilidade que são amplamente utilizadas no design de interfaces de usuário:

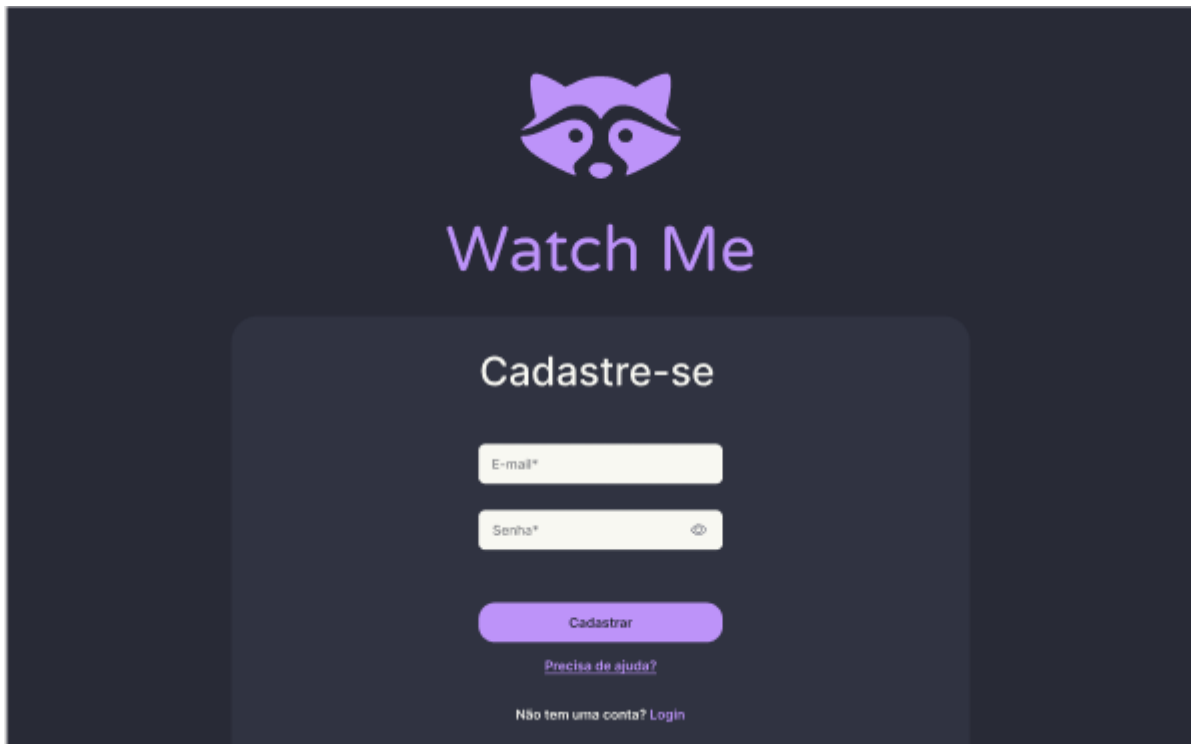


Na tela de login colocamos o login centralizado com esse fundo com uma cor mais clara para deixar mais claro ao usuário que é nesta região onde ele vai interagir com o sistema, além de padronizar as cores dos botões e adicionar uma interação de preciso de ajuda, utilizando é claro da heurística de Ajuda e documentação(Quando necessário, deve haver suporte ao usuário por meio de documentação, ajudas contextuais e recursos de suporte).

Além disso melhoramos o design das mensagens de erro na tela de login, optamos por colocar um pop-up por fica visualmente mais agradável ao usuário e fácil de enxergar em tela, além de tentar obedecer as heurísticas de Estética e design minimalista (as interfaces devem ser esteticamente agradáveis, com elementos de design minimalistas, evitando informações desnecessárias ou confusas) e Prevenção de erros (o sistema deve ser projetado para evitar erros e fornecer mensagens claras e informativas em caso de ocorrência de erros).



Os mesmos motivos foram seguidos na tela de cadastro:



The image shows a registration screen for 'Watch Me'. At the top, there is a purple cat face logo. Below it, the text 'Watch Me' is displayed in a large, light purple font. The main heading for the form is 'Cadastre-se' in white. The form contains two input fields: 'E-mail*' and 'Senha*'. The 'Senha*' field has a small eye icon to its right. Below the fields is a purple 'Cadastrar' button. Underneath the button are two links: 'Precisa de ajuda?' and 'Não tem uma conta? Login'.

Watch Me

Cadastre-se

E-mail*

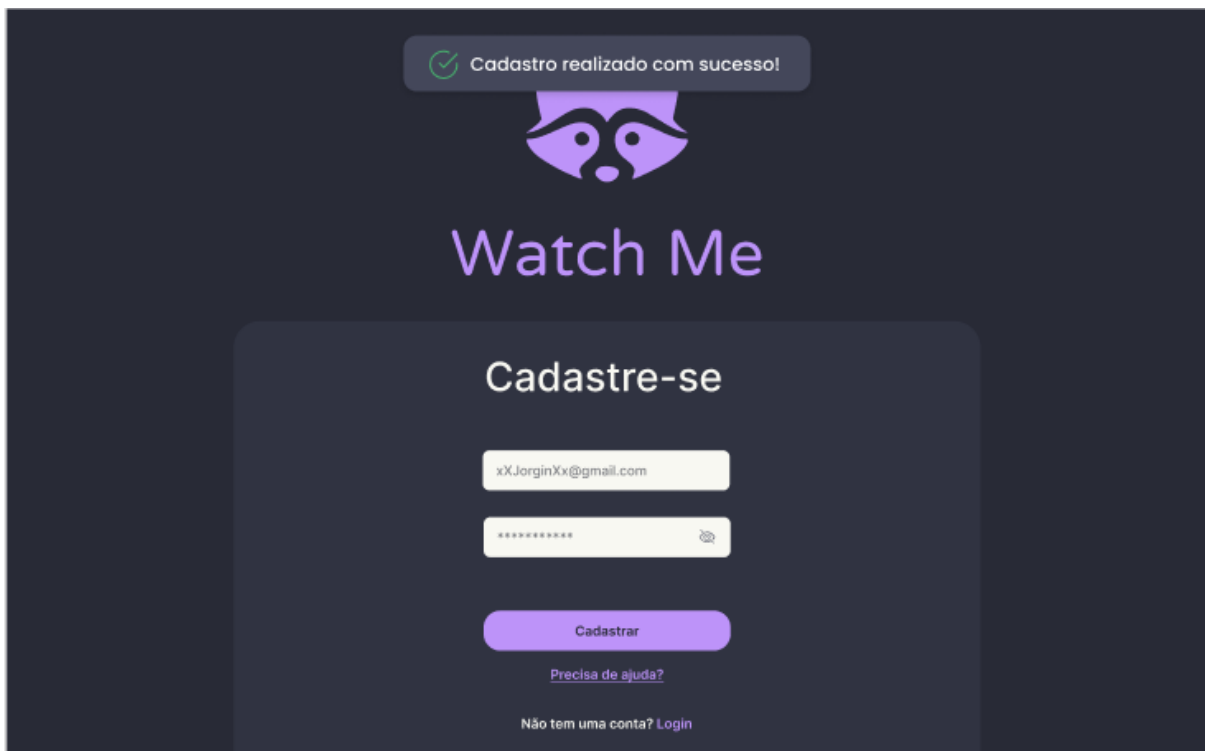
Senha*

Cadastrar

[Precisa de ajuda?](#)

[Não tem uma conta? Login](#)

Tela de sucesso:



The image shows the same registration screen as above, but with a success message at the top. A dark purple banner with a green checkmark icon contains the text 'Cadastro realizado com sucesso!'. The rest of the screen, including the logo, 'Watch Me' text, 'Cadastre-se' heading, input fields, and buttons, remains the same. The 'E-mail*' field now contains the text 'xxJorginXx@gmail.com' and the 'Senha*' field contains a series of asterisks.

Cadastro realizado com sucesso!

Watch Me

Cadastre-se

xxJorginXx@gmail.com


Cadastrar

[Precisa de ajuda?](#)

[Não tem uma conta? Login](#)

Tela de erro:

✖ Este e-mail já está cadastrado.



Watch Me


Cadastre-se

Cadastrar

[Precisa de ajuda?](#)

Não tem uma conta? [Login](#)

✖ A senha precisa ter pelo menos 8 caracteres



Watch Me

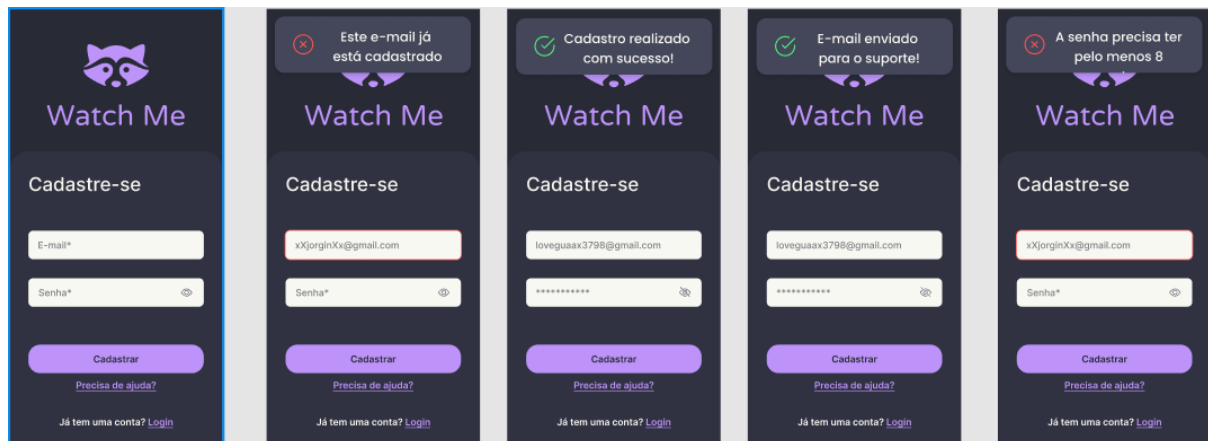
Cadastre-se

Cadastrar

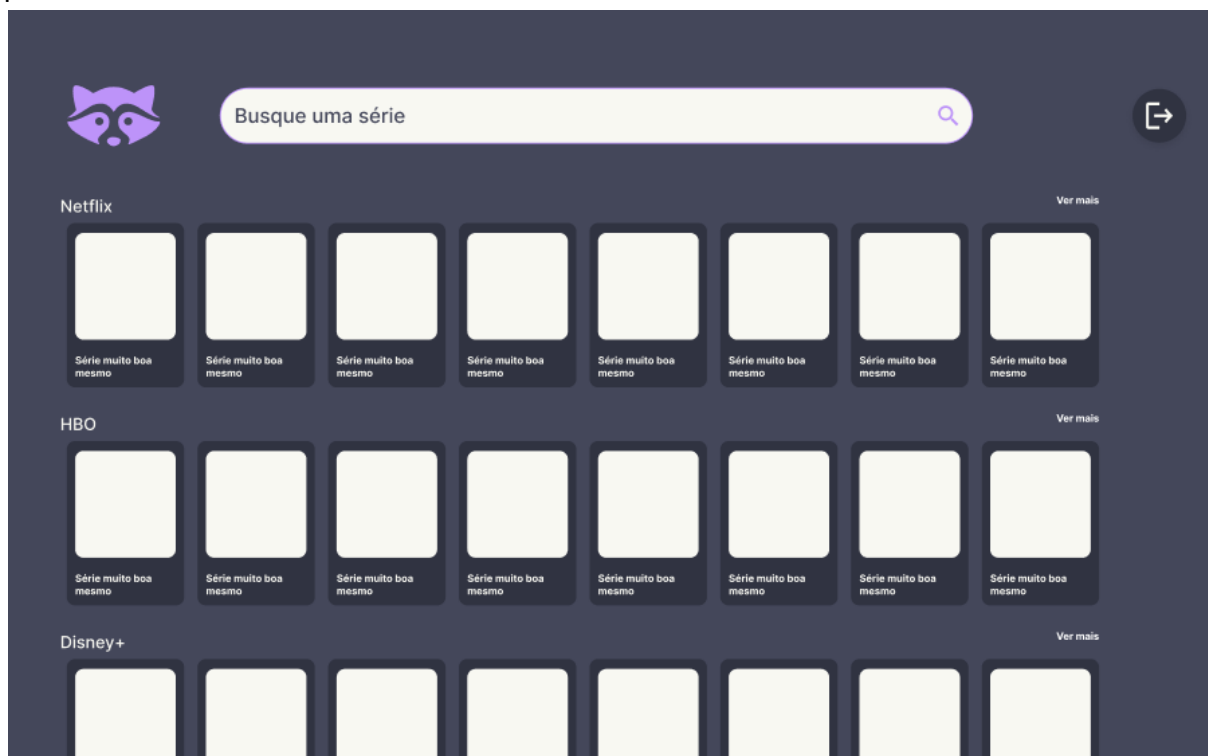
[Precisa de ajuda?](#)

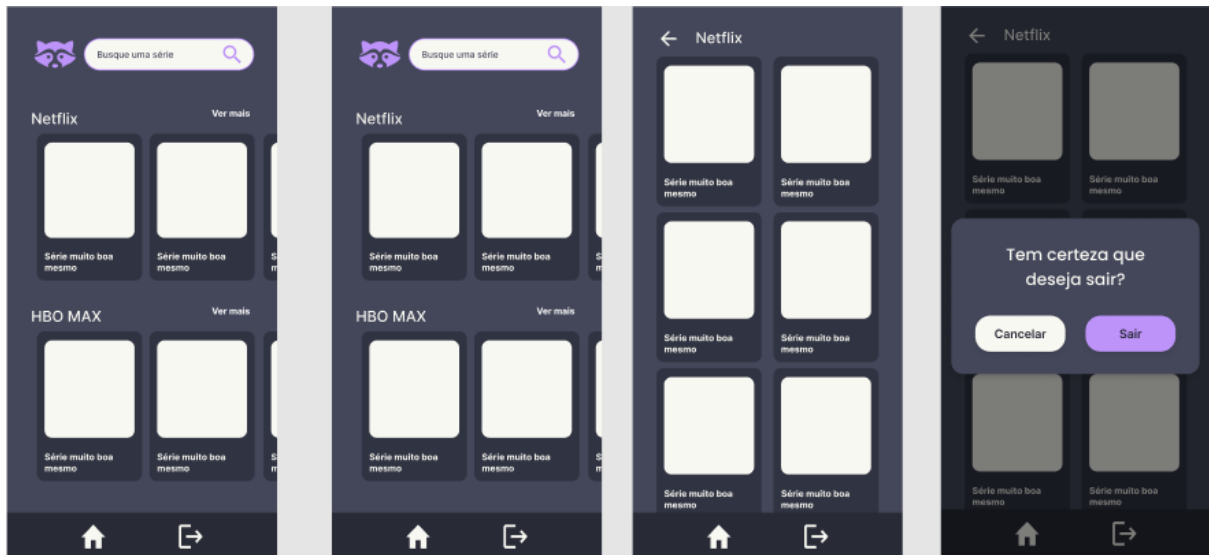
Não tem uma conta? [Login](#)

Mesmo padrão foi seguido no mobile tendo também feedback de cada ação realizada em tela:

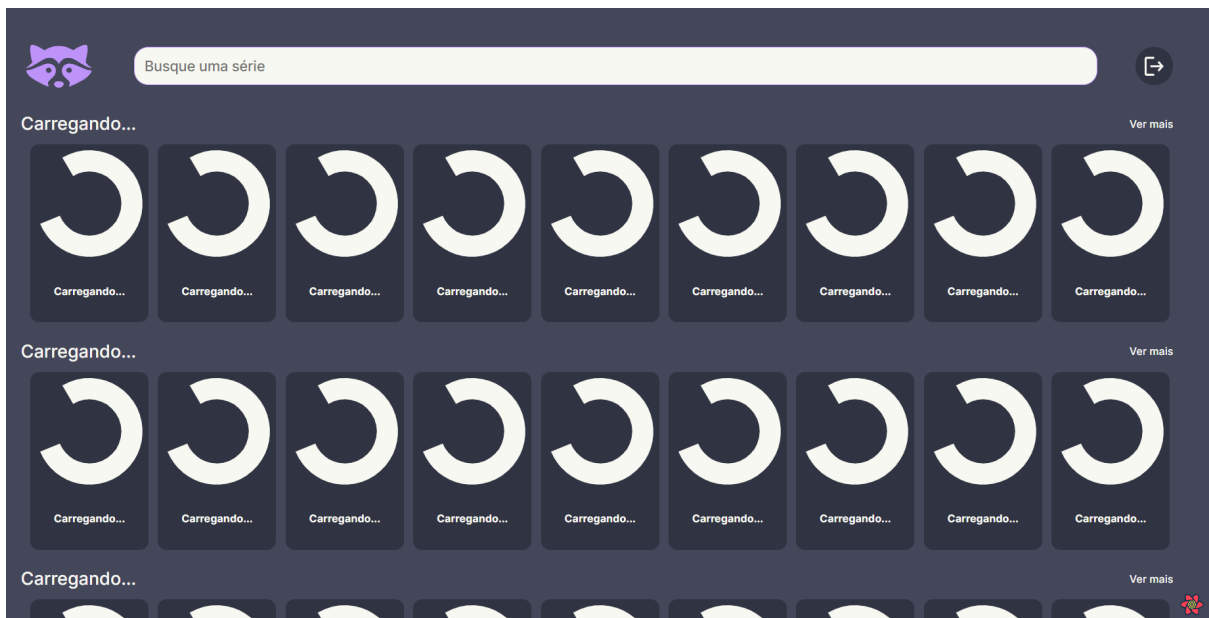


Outra alteração foi na tela inicial sobre a barra lateral, a mesma acabava consumindo muito espaço de tela, o que acabava dificultando a navegação de tela do usuário e por isso preferimos apresentar as listas das plataformas em tela utilizando como padrão as telas de início de sistemas como Netflix e Prime Video.





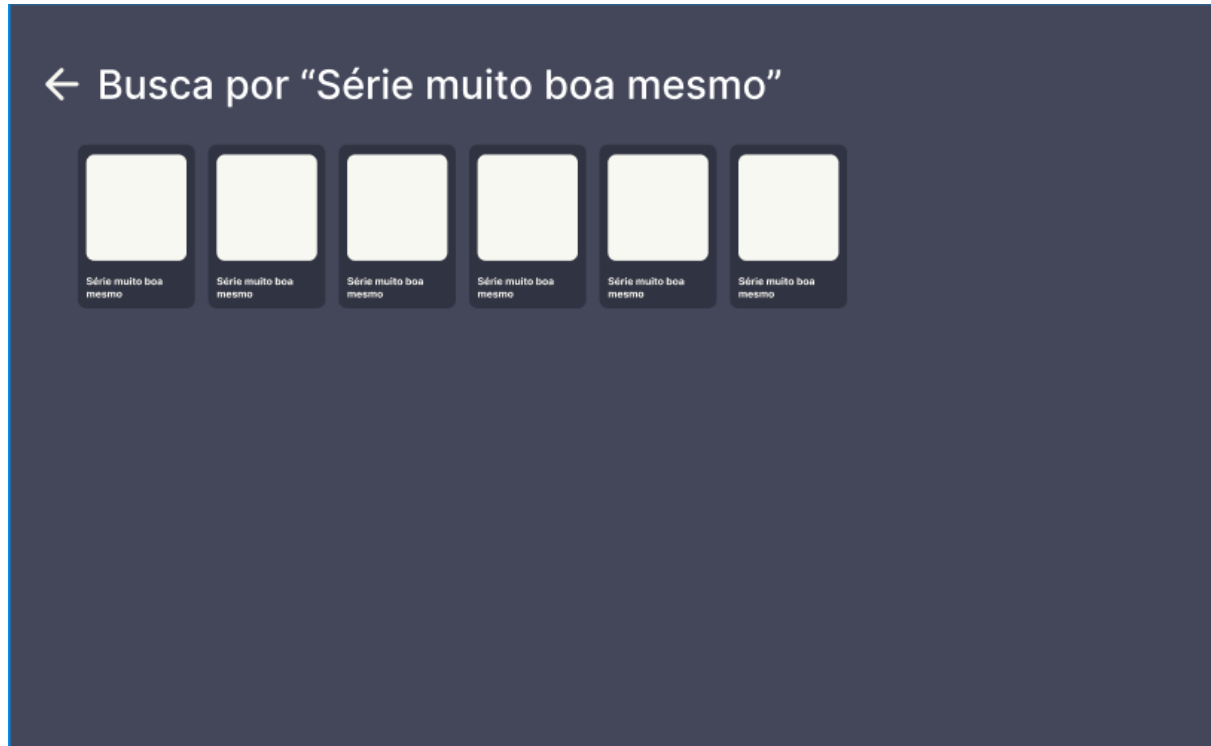
Foi pensado também na heurística de visibilidade do status do sistema(o sistema deve sempre informar aos usuários o que está acontecendo, fornecendo feedback adequado sobre suas ações e mantendo-os informados sobre o status do sistema) que adicionamos esses status de carregamento, fazendo assim com que o usuário não pense que o sistema travou apenas por não mostrar as séries em tela quando estiver carregando a mesma.



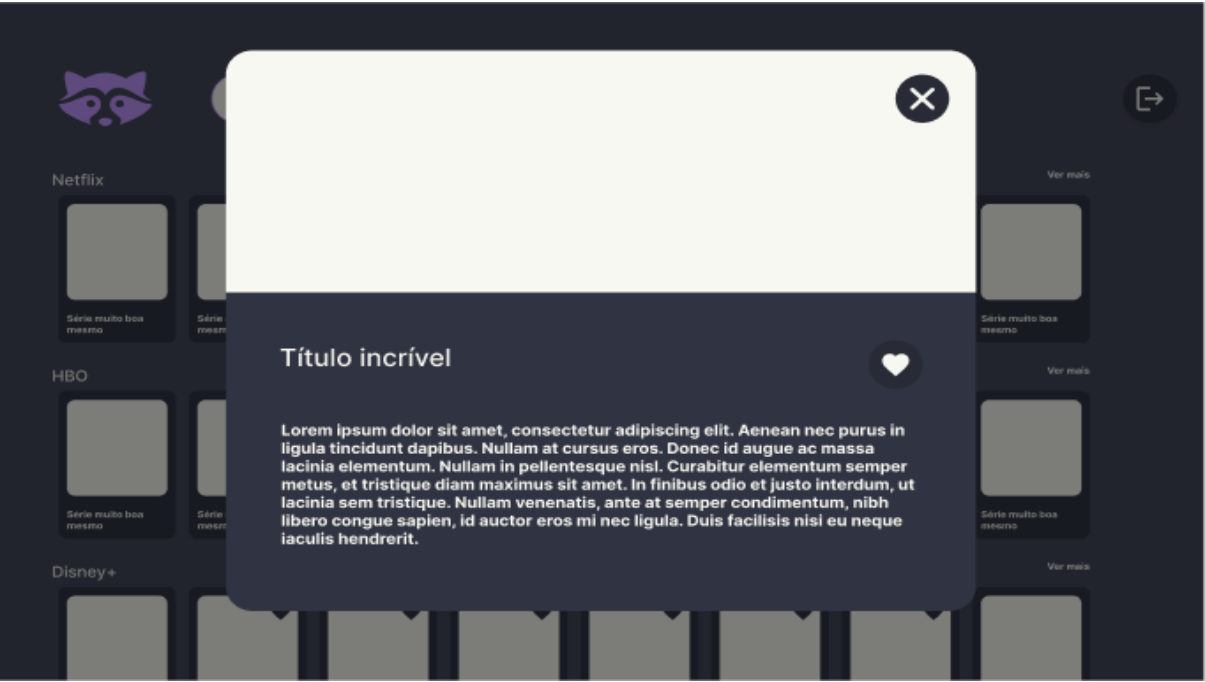
Além disso foi retirado o botão de perfil pois resolvemos não dar a opção de vários perfis no sistema já que a criação das listas é individual, não fazia sentido ter vários perfis por conta e distribuímos botão de sair do sistema para o canto superior direito da tela, e por último foi retirado o botão de favoritar que ficava na foto de cada tela pois o mesmo acabou

ficando muito pequeno devido a proporção de tamanho da tela então mantemos ela apenas quando o mesmo clica na série para abrir a visualização de série.

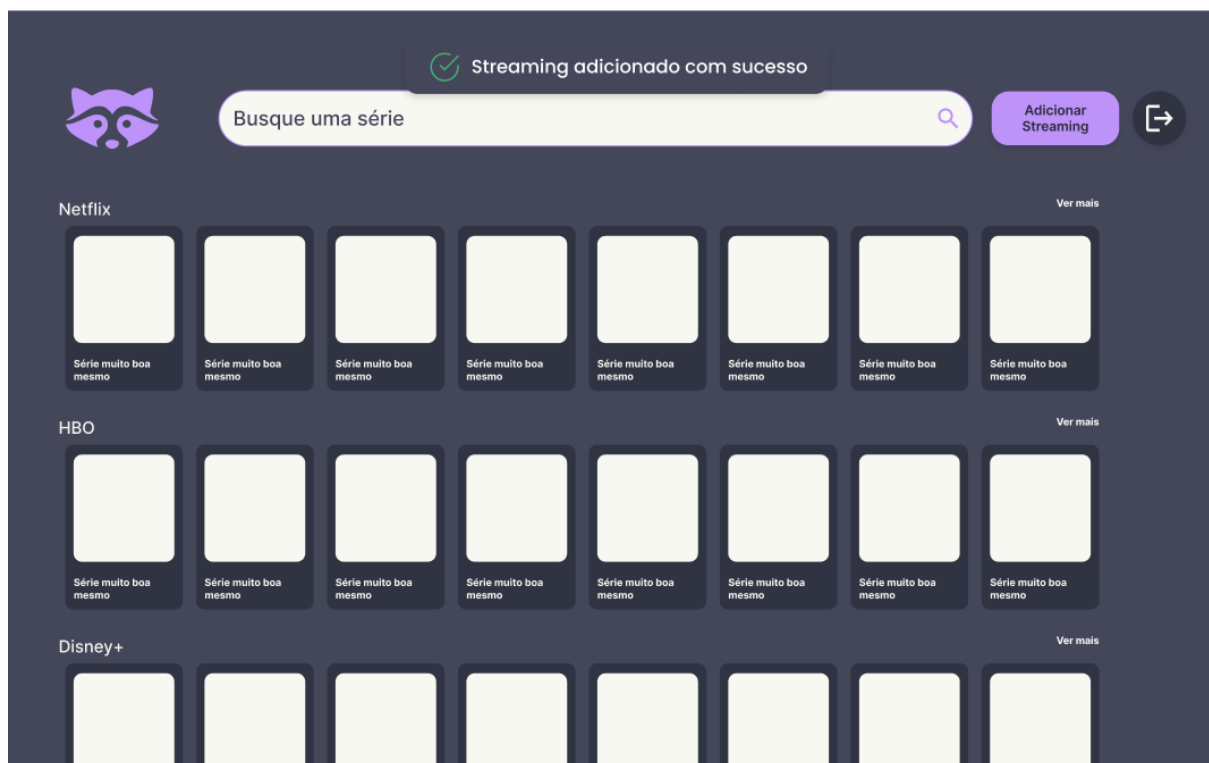
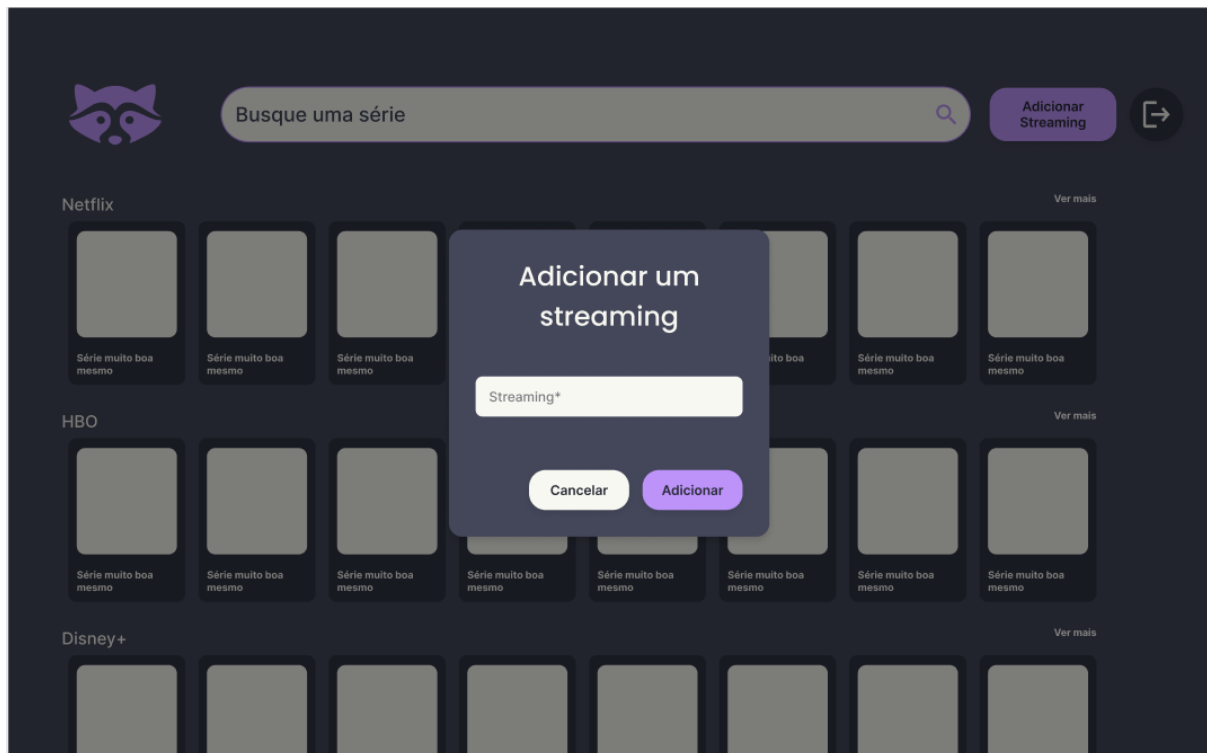
E na tela de pesquisa devido retirada da barra de pesquisa alocados os itens para que eles ficassem mais visualmente organizados em tela:



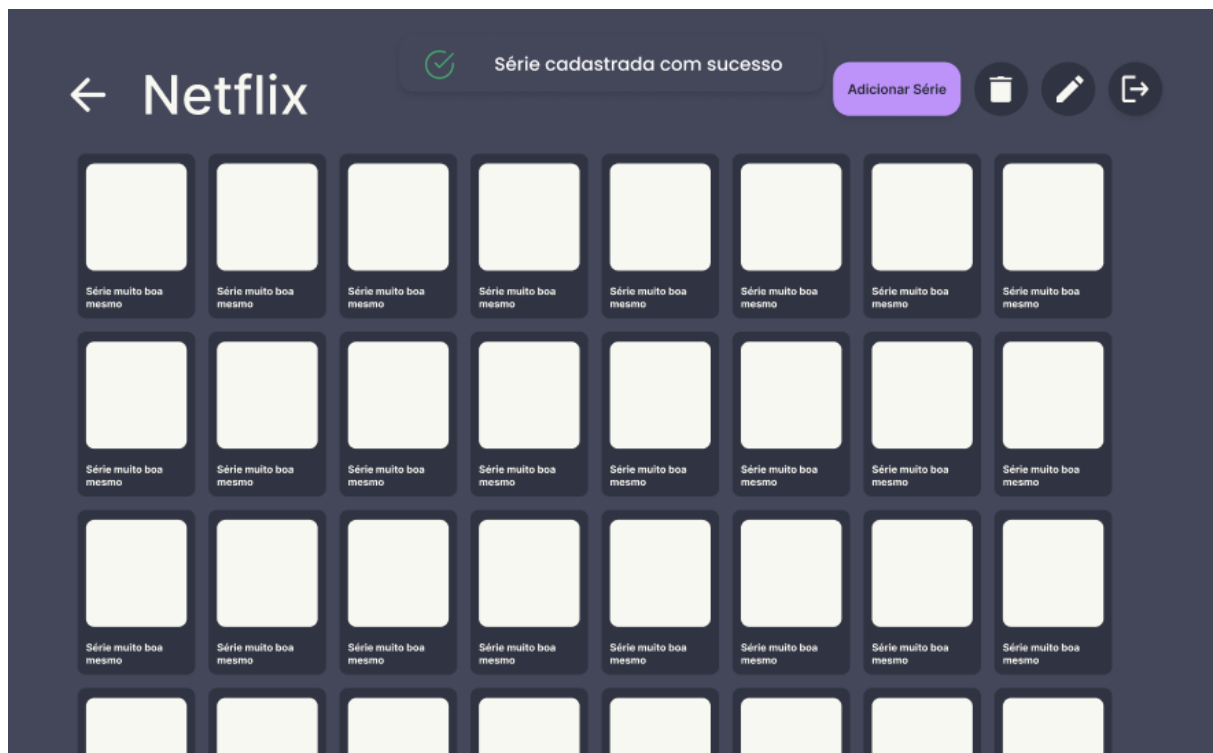
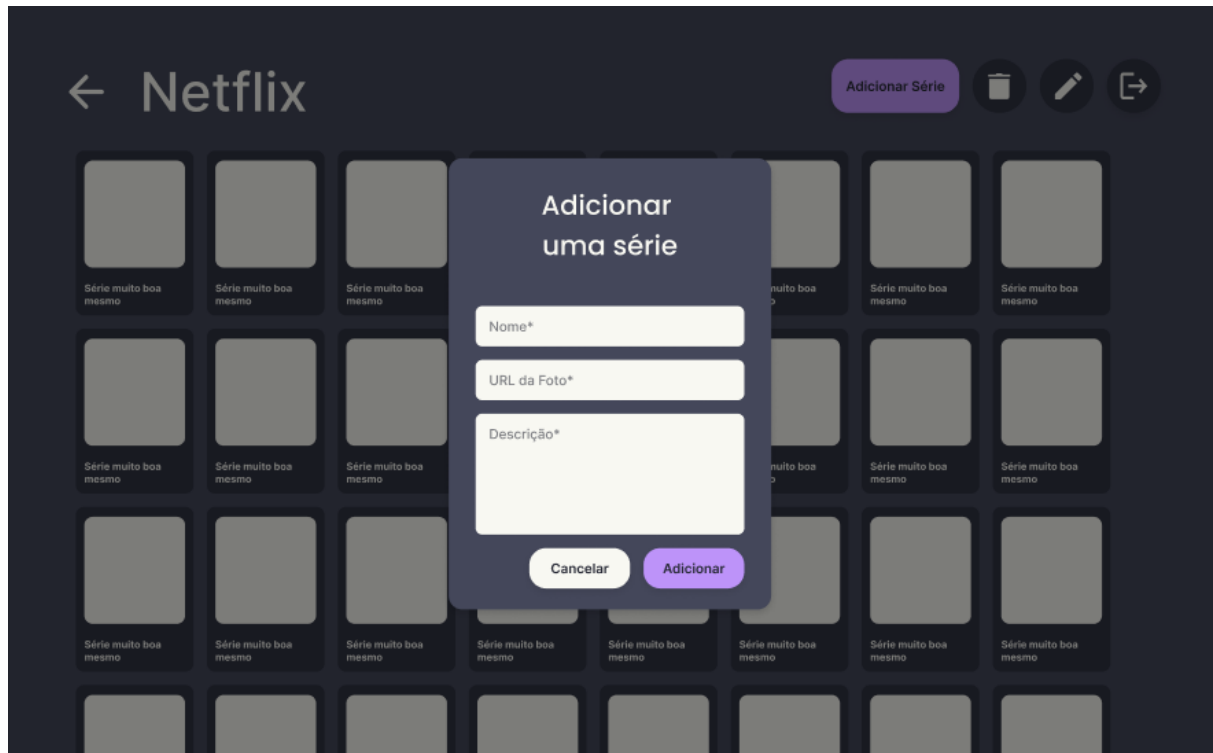
Por último na tela de visualizar série alteramos ela para um modal centralizado em tela com o fundo ao redor acinzentado para dar a impressão de continuidade entre as telas.



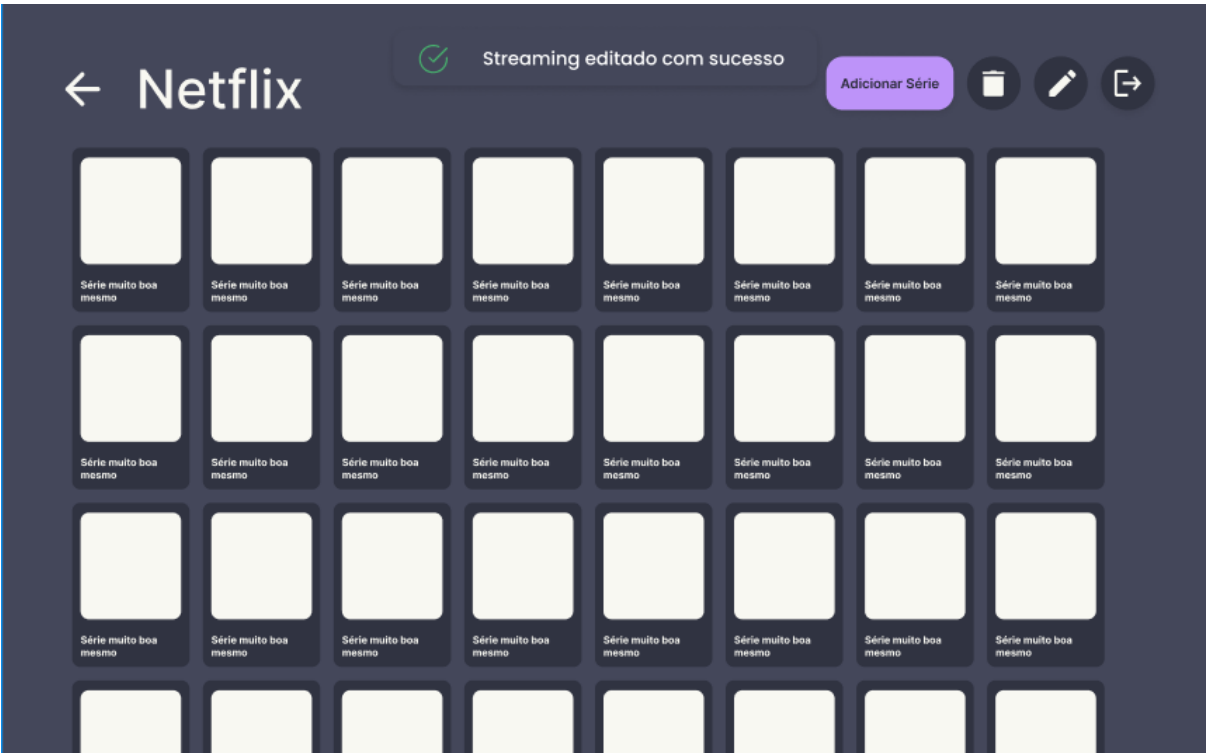
Nas telas de admin criamos o mesmo padrão e das mesmas referências para criar as telas de cadastrar, editar e excluir as séries e as plataformas de streaming:



Cadastrar série:

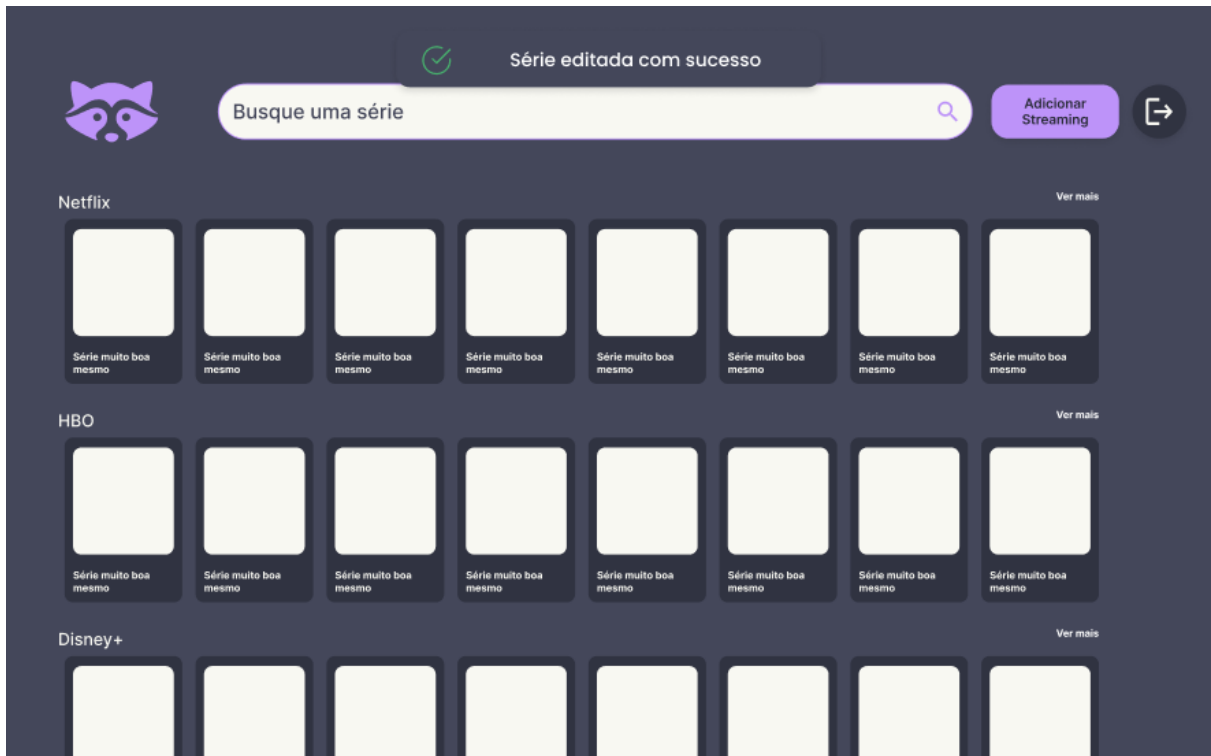


Editar streaming:

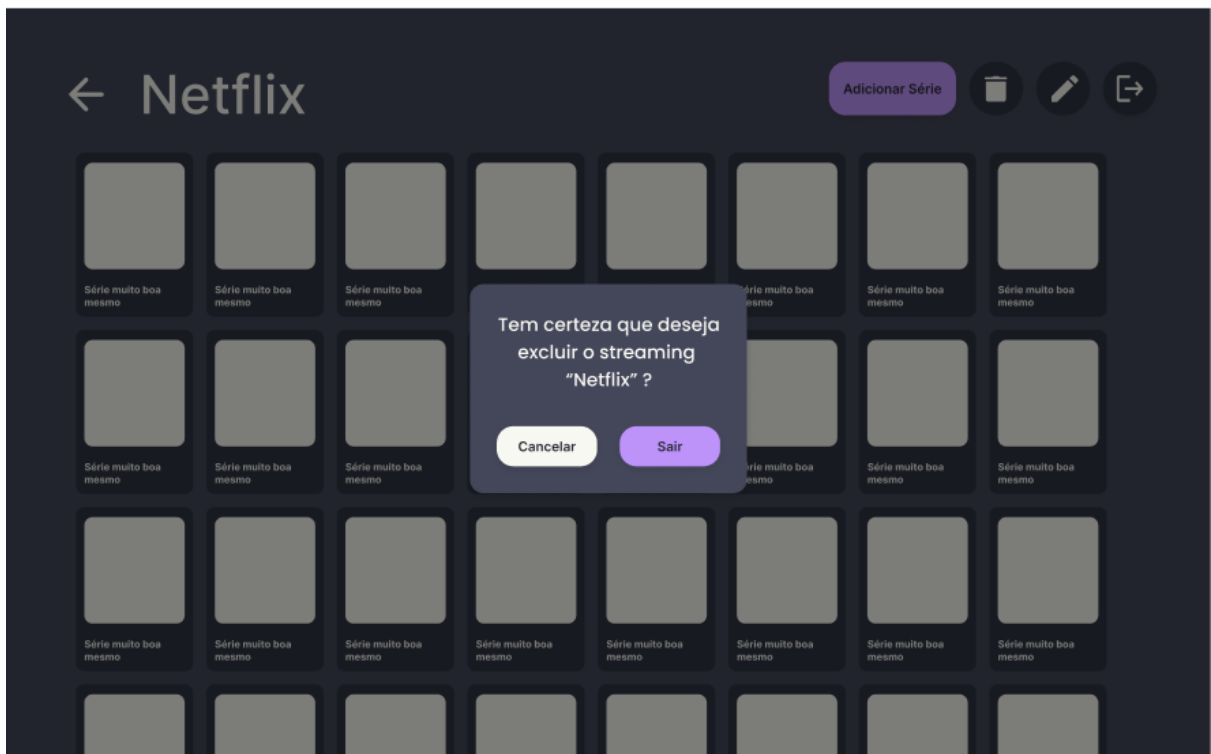


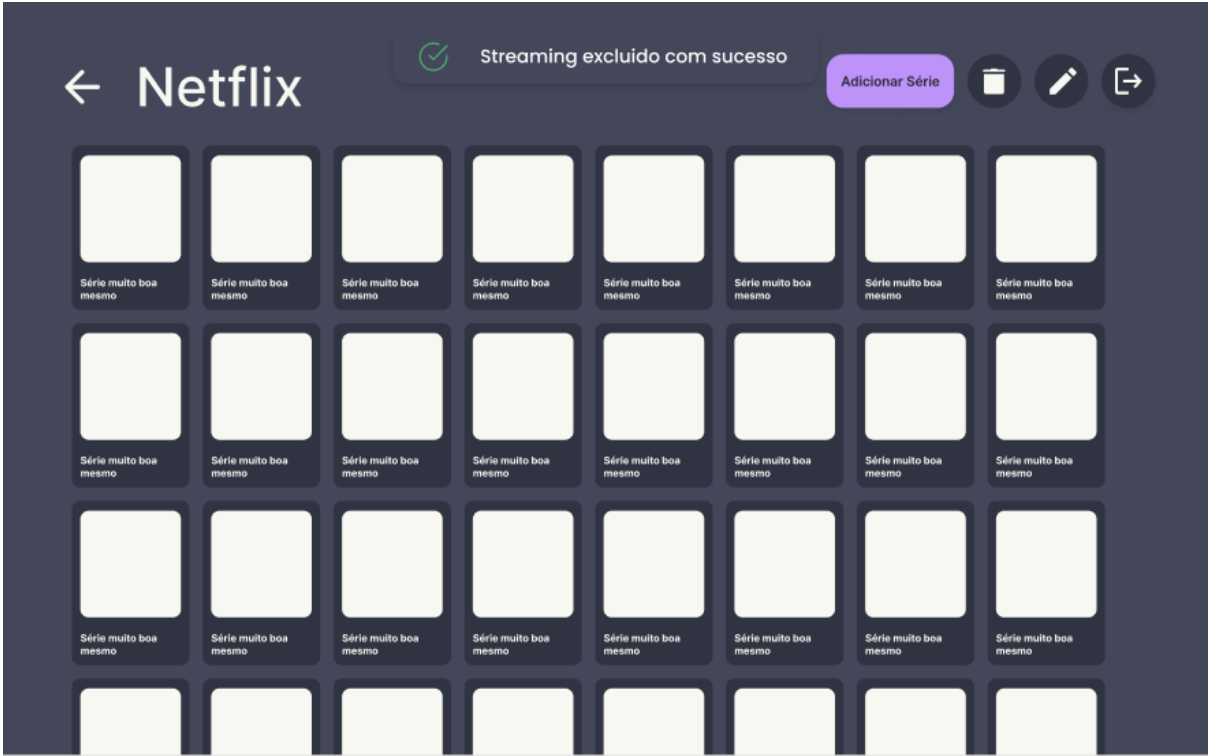
Editar série:



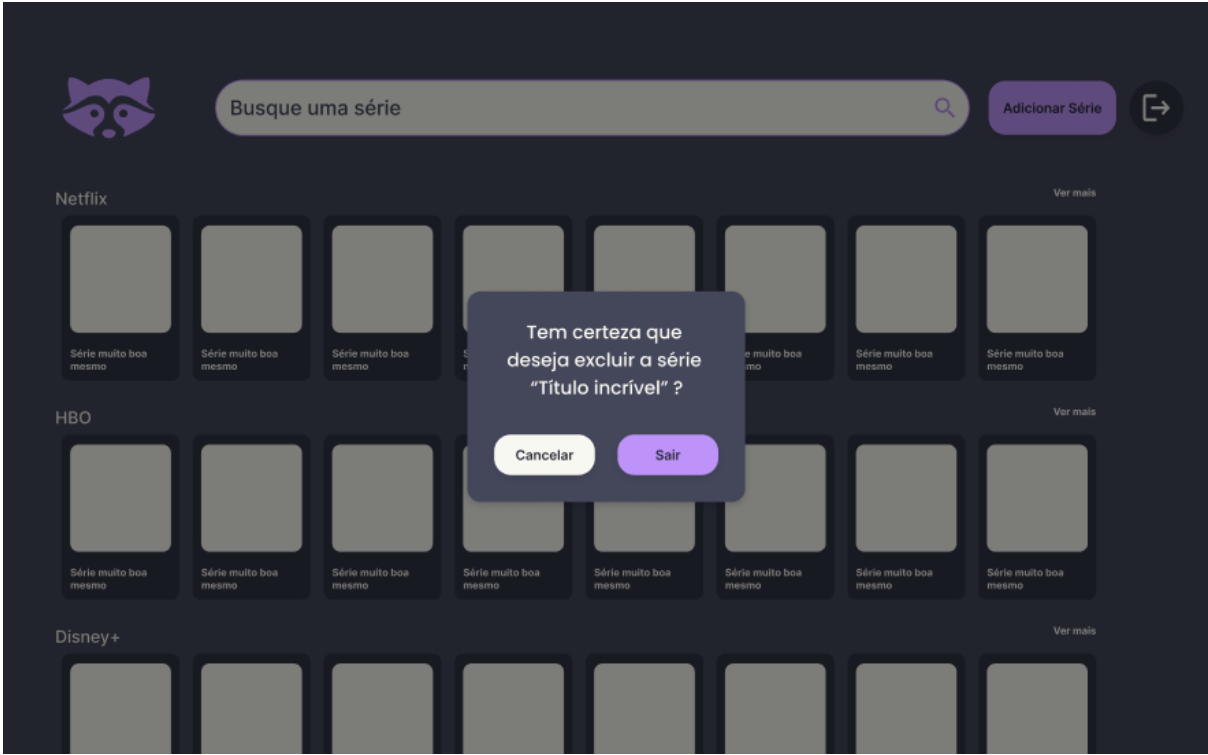


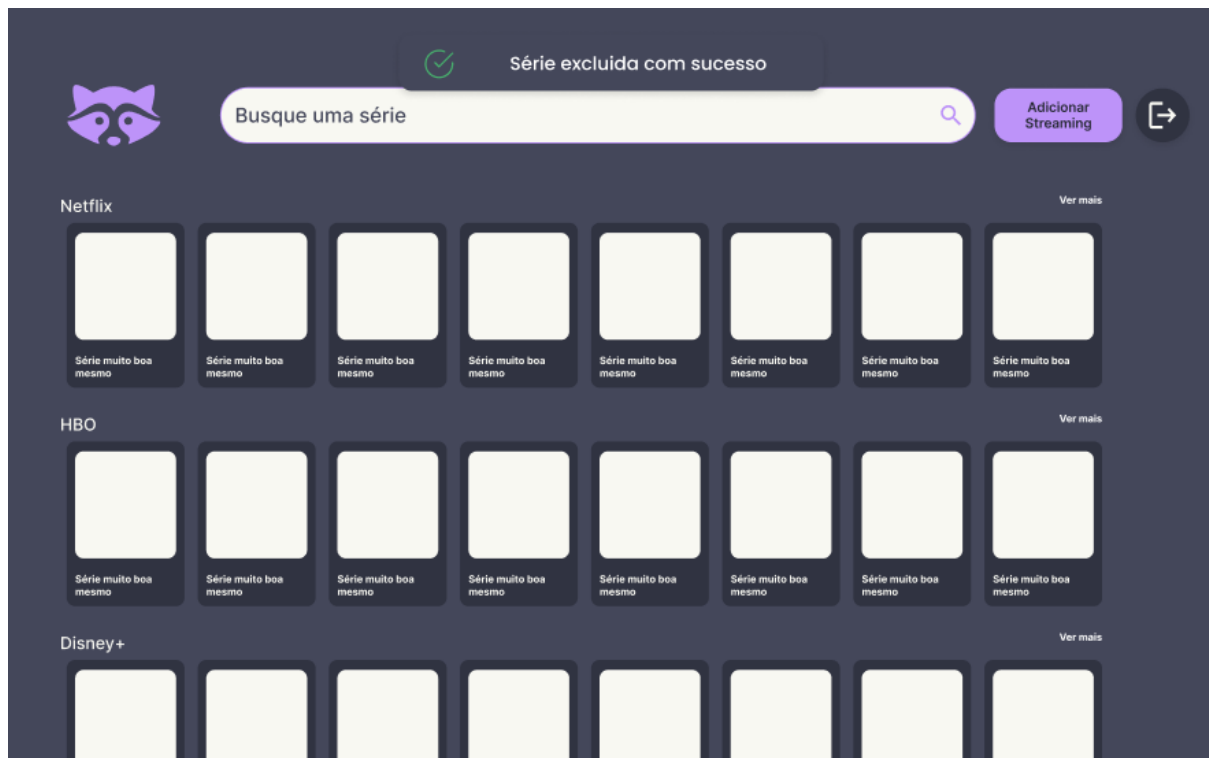
Excluir streaming:





Excluir Série:





No backend, temos diferentes controladores que lidam com as operações relacionadas a usuários, séries, plataformas e favoritos. Cada controlador é responsável por receber as requisições do frontend, executar a lógica de negócio necessária e retornar as respostas apropriadas.

Por exemplo, o controlador de usuários (user) possui o método create, que trata da criação de novos usuários. Ao receber uma requisição para criar um novo usuário, o controlador verifica se os campos obrigatórios, como email e senha, foram fornecidos. Caso algum campo esteja faltando, ele retorna uma resposta de erro com status 400.

```
CREATE TABLE users (  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  email TEXT NOT NULL,  
  password TEXT NOT NULL  
);
```

Em seguida, o controlador verifica se o email já está registrado no sistema. Para isso, faz uma consulta ao banco de dados utilizando o modelo User (que representa a tabela de usuários) e o método getByEmail. Se um usuário com o mesmo email for encontrado, o controlador retorna uma resposta de erro com status 409 indicando que o email já está em uso.

```
SELECT * FROM users WHERE email = 'example@example.com';
```

Caso o e-mail não esteja registrado, o controlador salva o novo usuário no banco de dados utilizando o método `save` do modelo `User`. Em seguida, gera um token de autenticação utilizando o método `generateToken` do controlador de sessão (`session`). O token é associado ao usuário recém-criado e retornado na resposta junto com os detalhes do usuário.

```
INSERT INTO users (email, password) VALUES ('example@example.com', 'password123');
```

Essa é apenas uma parte do fluxo do controlador de usuários. Os outros controladores funcionam de maneira semelhante, recebendo requisições, validando os dados, realizando consultas e atualizações no banco de dados, e retornando as respostas adequadas.

A integração entre o backend e o frontend é feita por meio de requisições HTTP. O frontend envia solicitações para as rotas definidas no backend, passando os dados necessários no corpo da requisição. O backend, por sua vez, recebe essas requisições, utiliza os controladores correspondentes para processar os dados e retornar as respostas apropriadas.

Por exemplo, o frontend pode fazer uma requisição POST para a rota de criação de usuário (`/user/register`), enviando os dados do novo usuário no corpo da requisição. O backend recebe essa requisição, utiliza o controlador de usuários (`user.create`) para criar o usuário no banco de dados e retorna uma resposta com o status 201 (`Created`) indicando que o usuário foi criado com sucesso.

```
SELECT * FROM users WHERE id = 1;
```

O banco de dados utilizado no exemplo é o SQLite, e o acesso a ele é feito por meio da biblioteca do Node.js. O SQLite é um banco de dados relacional leve que armazena os dados em um arquivo local. O backend utiliza consultas SQL para interagir com o banco de dados, realizando operações como criar tabelas, inserir registros, atualizar informações e recuperar dados.