UNIVERSIDADE SALVADOR

CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Alison Braz Santos - RA 1272116871
Carlos Alberto do Bomfim São Luiz Moreira - RA 1272118121
Gilson Sousa Silva - RA 1272118998
Lucas Vital Rocha - RA 1272116103
Thiago dos Santos -RA 1272117023

Usabilidade, desenvolvimento web, mobile e jogos

Salvador-BA
2023.

Relatório 2 Aplicação Web de Séries

STREAMBOX

Projeto apresentado como requisito parcial de avaliação da Unidade Curricular *Usabilidade, desenvolvimento web, mobile e jogos* do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Salvador - UNIFACS.

Docentes: Wellington Lacerda S da Silva.

Galdir Damasceno Reges Junior.

Salvador-BA

2023.1

RESUMO

Este relatório especifica as funcionalidades da aplicação WEB StreamBOX - Séries. Por meio deste, será possível tomar ciência de como funcionará a plataforma, quais telas foram criadas, suas funcionalidades e uma breve explicação do código.

Apresentaremos de forma resumida e objetiva todas as telas criadas, especificando as funções de cada uma delas.

O objetivo deste relatório tem como listar as funcionalidades solicitadas, deixando de forma clara e objetiva os requisitos. O presente relatório visa ser escrito de forma abrangente e clara para que possa ser utilizado por todos que compõem o time e ainda ser compreendido por aqueles que tenham interesse .

SUMÁRIO

1 INTRODUÇÃO

O StreamBOX é uma plataforma de gerenciamento de séries criada para proporcionar o melhor em aplicação WEB já desenvolvida no mercado. Na plataforma é possível gerenciar suas séries, de forma intuitiva, tendo em vista proporcionar uma experiência incrível para os seus usuários.

Por meio desta plataforma é possível gerenciar suas séries quando quiser, de onde estiver, por meio do celular, tablet ou desktop, você também pode visualizar o que está assistindo, quais séries já assistiu, além de adicionar a uma lista de desejo quais pretende assistir, é incrível poder organizar aquilo que você gosta. Não acha?

StreamBOX, uma experiência fascinante feita pra você.

2 Back End

O back-end está sendo construído no framework express.js para utilização do Node.js, sendo escrito na linguagem JavaScript, utilizando o SQLite para construção do Banco de Dados, tendo a biblioteca Lib Cors onde é configurado o uso da API para ter acesso em outros locais.

Localização: Entrega2> Backend

2.1 Plataforma

A plataforma é o local de cadastro das plataformas de streaming onde possui diversas funções que são armazenadas no banco de dados SQLite na variável ***'./database/data/streambox.db'*** .

Localização: Entrega2> Backend> controller> plataforma.js

2.1.1 Funções:

2.1.2.1 Inserir;

Nesta função são inseridos as plataformas de streamings, onde é possível cadastrar imagens e o nome.

2.1.1.2 Atualizar;

Por meio desta função é possível realizar qualquer alteração nas plataformas de streamings cadastrados e tem como parâmetro um id para localizar o cadastro.

2.1.1.3 Listar;

A função de listar dá retorno a todas as plataformas de streamings cadastrados em formato de lista.

2.1.1.4 Deletar;

Para que tenha a possibilidade de exclusão de um cadastro, foi criado a função Deletar, por meio desta é possível realizar a exclusão de um cadastro passando como parâmetro um id.

2.1.1.5 Filtrar Item;

Também é possível realizar a filtragem por item, passando um id da plataforma é localizado em filtro o cadastro sem a necessidade de listar todos.

2.2 Séries

Em ***serie.js*** são inseridos as séries pelo usuário administrador da aplicação StreamBox, possui diversas funções que são armazenadas no banco de dados SQLite na variável './database/data/streambox.db' .

Localização: Entrega2> Backend> controller> serie.js

2.2.1 Funções:

2.2.1.1 Inserir;

Por meio desta função são inseridos as séries, onde é possível cadastrar imagens e o nome.

2.2.1.2 Atualizar;

Em atualizar é possível realizar qualquer alteração nos cadastrados das séries e tem como parâmetro um id para localizar o cadastro.

2.2.1.3 Listar;

A função de listar dá retorno a todas as séries cadastrados em formato de lista.

2.2.1.4 Deletar;

Para que tenha a possibilidade de exclusão de uma série cadastrada, foi criado a função Deletar, por meio desta é possível realizar a exclusão de um cadastro passando como parâmetro um id.

2.2.1.5 Filtrar Item;

Também é possível realizar a filtragem por item, passando um id da série para realizar o filtro no cadastro sem a necessidade de listar todos.

2.3 Usuário

Para ter acesso a aplicação StreamBox é necessário possuir um cadastro, por meio de funções aqui presentes é possível salvar os cadastros em banco de dados, alterar e consultar ou listar os usuários cadastrados.

Localização: Entrega2> Backend> controller> usuario.js

2.3.1 Funções:

2.3.1.1 Inserir;

Nesta função são inseridos os dados dos usuários, como: Nome, E-mail, Senha e por Tipo que é um parâmetro para separar o usuário administrador de um usuário comum.

2.3.1.2 Listar;

Nesta função é realizada a listagem dos usuários cadastrados usando como parâmetro o ***id_usuario***.

2.3.1.3 Alterar Status de Série;

Nesta função é realizada a alteração dos status das séries utilizando algumas variáveis como id do usuário e status das séries.

2.3.1.4 Obter Status de Série;

Já nessa função é possível obter o status das séries, onde são levantados os status de assistindo ou concluídos.

2.4 Funções Gerais:

Foi criado uma classe ***imagen.js*** responsável por organizar todas as imagens vinculadas ao código do Back-End, contém o pacote ***multer*** para lidar com o upload dos arquivos de imagens .

A função MIDD é responsável por validar os usuários comuns e usuário administrador, quando os mesmos entram com credenciais de acesso por E-mail e Senha.

Em index.js na pasta do routes são importados as classes ***plataforma.js, serie.js, imagem.js e usuario.js,*** esta pasta é utilizada para criar as rotas para cada uma das classes citadas, usado no método http.

Assim, o código fornece um módulo Node.js que lida com operações relacionadas a usuários e séries em um banco de dados SQLite. Ele exporta várias funções que são usadas em outros módulos e rotas da aplicação.

inserir(req, res, next): Esta função é responsável por inserir um novo usuário no banco de dados. Ela recebe os dados do usuário do corpo da solicitação (**req.body**) e os insere na tabela "***usuario***". Se a inserção for bem-sucedida, ela responde com

uma mensagem de sucesso e os dados do usuário inserido. Caso contrário, ela responde com uma mensagem de erro.

listar(req, res, next): Essa função recupera todos os usuários do banco de dados e os retorna como resposta. Ela consulta a tabela "***usuario***" e transforma cada linha em um objeto contendo os campos "id", "nome" e "email". Em seguida, responde com uma mensagem de sucesso e a lista de usuários.

alterarStatusSerie(req, res, next): Essa função é usada para atualizar o status de uma série para um usuário específico. Ela primeiro verifica se o status da série já existe para o usuário na tabela "***usuario_serie***". Se existir, atualiza o status com os valores fornecidos no corpo da solicitação. Caso contrário, insere um novo registro com o status da série. A função responde com uma mensagem de sucesso e o novo status da série.

obterStatusSerie(req, res, next): Essa função recupera o status de uma série para um usuário específico. Ela consulta a tabela "***usuario_serie***" usando o ID da série fornecido na solicitação e o ID do usuário obtido a partir do objeto **res.locals.usuario**, em seguida, responde com uma mensagem de sucesso e o status da série.

obterSerie(req, res, next): Esta função recupera a lista de séries associadas a um usuário específico. Ela consulta a tabela "***usuario_serie***" e faz um join com a tabela "***serie***" para obter os detalhes da série. Em seguida, retorna uma lista de objetos contendo informações sobre cada série, como ID, status do usuário e detalhes da série (nome e imagem). As funções **alterarStatusSerieDB**, **inserirStatusSerieDB**, **obterStatusSerieDB** e **obterSerieDB** são funções auxiliares usadas internamente pelas funções principais. Elas lidam diretamente com as consultas SQL ao banco de dados.

Assim, as funções fornecem uma interface para realizar operações de CRUD (criação, leitura, atualização e exclusão) relacionadas a usuários e séries no banco de dados SQLite.

3 Front End

O front-end foi construído em React com a utilização de AXIOS para fazer as consultas API, está sendo utilizado o react-router-dom para roteamento da aplicação, e o react-bootstrap onde são usados os componentes como buttam. A aplicação inclui diferentes páginas HTML com estilos CSS e funcionalidades JavaScript.

O código HTML com Estilos e Scripts Externos, representa uma página HTML que faz uso de estilos CSS externos e scripts JavaScript externos. Os estilos CSS externos são importados através de links **<link>** no elemento **<head>**. Esses estilos estão localizados em arquivos separados, como "***estilogeral.css***" e "t***oast.css***". Os scripts JavaScript externos também são importados através de tags **<script>** no elemento **<head>**. Essa página também faz uso de fontes do Google, que são importadas através de links **<link>** para a API do Google Fonts. Além disso, são usadas bibliotecas externas, como os ícones do Bootstrap, importados via link **<link>**.

HTML com Estilos Embutidos e Scripts Internos. A página HTML utiliza estilos CSS embutidos no elemento **<style>** dentro do elemento **<head>**. Os estilos são definidos diretamente no HTML, em vez de serem importados de arquivos externos. Os scripts JavaScript são definidos internamente na página, dentro da tag **<script>** no elemento **<head>**, também são usadas fontes do Google, importadas..

Código HTML com Função de Pré-Visualização de Imagem, é página HTML que permite ao usuário selecionar uma imagem de seu dispositivo e exibi-la em tempo real, inclui um elemento **<input>** do tipo "***file***" e uma imagem **<img>** com um atributo **id** para identificação. Um evento de escuta é adicionado ao elemento **<input>** para detectar quando o usuário seleciona um arquivo de imagem.

Quando o arquivo é selecionado, um objeto **FileReader** é usado para ler o conteúdo do arquivo e, em seguida, o resultado é definido como o atributo **src** da imagem, atualizando-a para exibir a visualização da imagem selecionada.

Ferramentas e Tecnologias Utilizadas:

HTML: Linguagem de marcação usada para estruturar o conteúdo da página.

CSS: Linguagem de estilo usada para aplicar estilos visuais à página.

JavaScript: Linguagem de programação usada para adicionar interatividade e funcionalidades dinâmicas à página.

Bootstrap: Framework CSS popular usado para estilização responsiva e componentes pré-estilizados.

Google Fonts: API do Google que permite importar fontes personalizadas para uso na página.

FileReader: Uma API JavaScript que permite ler o conteúdo de arquivos selecionados pelo usuário.

3.1 Telas Web e Mobile:

Localização: Entrega1

3.1.1 Usuário/Cliente

As telas do Usuário/Cliente é composta por telas de cadastro, login, menu, minha lista de séries, sobre e concluídas.

3.1.1.1 Cadastrar;

Na tela de cadastro o usuário/cliente precisará fornecer o nome, um e-mail, e construir uma senha conforme imagens.

3.1.1.2 Login;

Para se logar na plataforma é simples, basta entrar com o e-mail e a senha cadastrada.

3.1.1.3 Menu

Após logar o usuário/cliente terá acesso a um menu, aos cards com opções de adicionar à minha lista de séries e o ícone de tv ou card se direcionando para a página sobre.

3.1.1.4 Minha Lista de Séries;

Já em minha lista de séries o usuário/cliente tem a facilidade de construir uma lista de desejos salvando suas séries favoritas, por lá também é possível acessar a tela

Sobre, ou até excluí-la da lista se desejar.

3.1.1.5 Sobre;

Na tela Sobre, tem informações detalhadas da série, como ano de lançamento, país de origem, quantidade de temporadas, número de episódio total, além de status como assistindo, finalizados, em produção e descontinuados.

3.1.1.6 Assistindo;

Nesta tela o usuário/cliente verá quais as séries estão assistindo, terá a opção de ir para a tela Sobre e também marcar como já concluídas.

3.1.1.7 Concluídas;
Por esta tela é possível desmarcar como concluída e também acessar a tela Sobre.

3.2.2 Usuário/Administrador

3.1.2.1 Login;

Após se logar, é possível visualizar o menu de navegação com as opções de ver as séries cadastradas, realizar o cadastro de novas séries, cards com opções de ver sobre, editar e ou apagar as séries cadastradas.

3.1.2.2 Cadastrar Séries;

Nesta tela é possível cadastrar as séries preenchendo os campos solicitados, como o título da série, qual o país de origem, o número de temporadas, a quantidade de episódios, o lançamento, status, onde está disponível e um resumo sobre a série.

3.1.2.3 Atualizar Séries;

Nesta tela o usuário/administrador terá a opção de atualizar uma série cadastrada, podendo modificar qualquer um dos itens cadastrados, representado na parte Cadastrar Séries em 3.1.2.2.

3.1.2.4 Menu de Séries;

Neste menu é possível visualizar todas as séries cadastradas e tem a opção de excluir ou editá-las.

3.1.2.5 Sobre;

Nesta tela o usuário/administrador terá informações detalhadas da série, como ano de lançamento, país de origem, quantidade de temporadas, número de episódios totais, além de opções de atualizar ou excluir a mesma.

3.2 Home

Esta é nossa tela inicial, por ela é possível ter informações dos serviços oferecidos e de valores, opções de se cadastrar e ou logar em nossa plataforma.

3.3 Erro da Página

Quando um usuário tenta acessar uma página não disponível, irá apresentar uma tela 404 informando que a página solicitada não foi encontrada.

4 Banco de Dados

Para criação do banco de dados foi utilizado o ***SQLite***, onde é importado o módulo ***sqlite3*** e o módulo ***mkdirp*** para criar o diretório onde o banco de dados será armazenado.

A função ***start*** é chamada para executar as operações de criação e inserção de dados no banco de dados. Ela chama as funções ***createDatabase***, ***insertUsuario***, ***insertSeries*** e ***insertSeriesUsuario*** em sequência.

A função **createDatabase** é responsável por criar as tabelas necessárias no banco de dados, como "***usuario***", "***usuario_serie***", "***serie***", "***plataforma***" e "***serie_plataforma***". Cada tabela tem suas colunas definidas.

A função ***insertUsuario*** é usada para inserir dados dos usuários com nome, e-mail, senha e tipo.

A função ***insertSeries*** insere dados da série com nome, imagem, data de lançamento, país de origem, número de temporadas, número de episódios, status e uma descrição.

A função ***insertSeriesUsuario***, insere uma única linha, representando a relação entre um usuário e uma série, com os campos ***id_usuario, id_serie, concluido, quero_assistir, assistindo e nao_recomendo.***

A função ***insertPlataformas*** insere dados na tabela "***plataforma***".

É exportada a função ***start*** para que ela possa ser utilizada em outros módulos.
O nosso banco de dados cria tabelas do banco de dados SQLite, inserem dados e estabelecem uma relação entre tabelas. Isso fornece uma base de dados para a aplicação StreamBox.

5 Conclusão

O StreamBOX é uma plataforma de streaming de séries que oferece uma experiência de visualização de alta qualidade aos seus usuários. A plataforma permite que os usuários cadastrem-se, façam login e criem uma lista de séries desejadas.

A interface do usuário/cliente inclui telas de cadastro, login, menu, minha lista de séries, sobre as séries e séries concluídas. Os usuários podem adicionar séries à sua lista de desejos, visualizar informações detalhadas sobre as séries, marcar séries como assistindo e concluídas, além de excluir séries da lista.

Por outro lado, a interface do usuário/administrador permite o login e oferece opções adicionais, como visualizar séries cadastradas, cadastrar novas séries e atualizar as informações existentes. Eles também podem acessar a tela "Sobre" para obter detalhes sobre uma série específica e têm a opção de editar ou excluir as séries.

No caso de um erro de página, quando um usuário tenta acessar uma página não disponível, a plataforma exibe uma tela 404 informando que a página solicitada não foi encontrada.

O Back End do StreamBOX é construído com o framework Express.js e utiliza o Node.js. Ele utiliza o banco de dados SQLite e possui as seguintes funcionalidades:

Em Plataforma e Séries são permitidas Inserir uma cadastrar, Atualizar um cadastro existente, Listar tudo que foi cadastrado, Deletar qualquer cadastro e Filtrar Item de um cadastro. Já em Usuário tem as funções Inserir o cadastro de usuários administrador ou comum, Listar todos os usuários cadastrados, Alterar Status de Série e Obter Status de Série.

Também tem a Classe Imagem que organiza as imagens relacionadas ao Back End e lida com o upload de arquivos de imagens, o MIDD valida os usuários comuns e administradores com base em suas credenciais de acesso. Em Rotas são criadas rotas para as diferentes classes mencionadas acima.

O Front End do StreamBOX é construído em React e utiliza o Axios para fazer consultas à API. Também utiliza o React Router DOM para roteamento e o React Bootstrap para componentes. Ele inclui diferentes páginas HTML com estilos CSS e funcionalidades JavaScript.

O StreamBOX utiliza o banco de dados SQLite para armazenar as informações relacionadas a plataformas de streaming, séries e usuários.

O StreamBOX se apresenta como uma plataforma de streaming de séries com recursos avançados, visando proporcionar uma experiência de visualização envolvente e conveniente para seus usuários.

REFERÊNCIAS

https://www.primevideo.com/offers/nonprimehomepage/ref=dv_web_force_root

https://www.netflix.com

https://www.figma.com

https://wireframe.cc/