

ESCOLA POLITÉCNICA E DE ARTES

Disciplina: CMP1059 - Tecnologia de Construção de Softwares

Professor: Me. Daniel Corrêa da Silva

Estudo de Caso: Sistema de Autenticação e Logout

Contexto:

Você foi contratado para desenvolver uma aplicação web para gerenciamento de acesso de usuários utilizando tecnologias client-side como HTML, CSS (Bootstrap) e JavaScript. A aplicação deverá ser capaz de gerenciar o processo de login e cadastro de novos usuários, além de validar as informações inseridas pelos mesmos.

Contexto:

1. HTML
2. CSS - biblioteca Bootstrap
3. Javascript

Requisitos:

1. **Página de Login:**
 - Crie uma página `index.html` com um formulário de login.
 - Utilize o framework CSS Bootstrap para estilizar o formulário e a página como um todo.
 - O formulário deve conter:
 - Um campo para o e-mail do usuário.
 - Um campo para a senha.
 - Botões “Entrar” e “Cadastrar-se”.
 - Ao clicar em “Cadastrar-se”, o usuário deve visualizar um modal para inserir suas informações (Nome, E-mail, Senha e Confirmar Senha).
 - Ao clicar em “Entrar”, deve ser feita a validação dos campos e, caso o usuário e senha estejam corretos, ele deve ser redirecionado para a página principal.
2. **Modal de Cadastro:**
 - Crie um modal com o id `registerModal` para receber as seguintes informações:
 - Nome.
 - E-mail.
 - Senha.
 - Confirmar Senha.
 - Valide se a senha e a confirmação de senha coincidem.
 - Adicione um botão de “Cadastrar” para efetivar o cadastro e fechar o modal.
3. **Validações:**
 - Ao submeter o formulário de cadastro, valide:
 - Se todos os campos foram preenchidos.
 - Se a senha e a confirmação de senha coincidem.
 - Se o e-mail não está cadastrado (utilizar array para simular o banco de dados).
4. **Lógica de Login:**
 - No formulário de login, ao clicar no botão “Entrar”:
 - Verifique se o usuário e senha existem no array de usuários cadastrados.
 - Se existir, redirecione para a página principal (`main.html`).

- Caso contrário, exiba uma mensagem de erro abaixo do formulário.

5. Lógica de Negócio:

- Crie um arquivo `auth.js` que implementa toda a lógica de autenticação utilizando a programação orientada a objetos.
- Utilize arrays para armazenar os usuários e simular um banco de dados.

Implementação Sugerida

1. Estrutura de Arquivos

- `index.html` (Página de Login e Cadastro)
- `main.html` (Página Principal)
- `auth.js` (Script de lógica de autenticação)
- `main.js` (Script de lógica de logout)
- `style.css` (Estilização adicional, se necessário)

2. Definição das Classes e Métodos no `auth.js`

1. Classe `User`

- Representa um usuário cadastrado.
- Atributos:
 - `name` (string)
 - `email` (string)
 - `password` (string)
- Métodos:
 - `constructor(name, email, password)` – Inicializa um novo usuário.

2. Classe `Auth`

- Responsável por gerenciar o processo de login e cadastro.
- Atributos:
 - `users` (array) – Lista de usuários cadastrados.
- Métodos:
 - `registerUser(name, email, password)` – Cadastra um novo usuário se o e-mail não existir.
 - `loginUser(email, password)` – Verifica se o e-mail e senha são válidos e autentica o usuário.
 - `isEmailRegistered(email)` – Retorna `true` se o e-mail já estiver registrado.
 - `validatePassword(password, confirmPassword)` – Retorna `true` se as senhas coincidirem.
 - `showError(message)` – Exibe uma mensagem de erro no formulário.