# **Documentação do Projeto**

## **Visão Geral**

Este projeto é uma aplicação web de autenticação de usuários, com funcionalidades de login e cadastro. A aplicação permite que os usuários se cadastrem, façam login e acessem uma página inicial. Os dados dos usuários são armazenados localmente no navegador utilizando o (a princípio) localstorage.

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## **Estrutura do Projeto**

A estrutura de pastas do projeto é a seguinte:

**meu-projeto/**

**│**

**├── css/**

**│ ├── signin.css # Estilos da tela de login**

**│ └── signup.css # Estilos da tela de cadastro**

**│**

**├── js/**

**│ ├── signin.js # Lógica da tela de login**

**│ └── signup.js # Lógica da tela de cadastro**

**│**

**├── images/**

**│ ├── bg-signin.png # Imagem de fundo da tela de login**

**│ └── bg-signup.png # Imagem de fundo da tela de cadastro**

**│**

**├── html/**

**│ ├── signin.html # HTML da tela de login**

**│ └── signup.html # HTML da tela de cadastro**

**│**

**└── index.html # Página inicial**

## 

## 

## 

## 

## 

## 

## 

## **Funcionalidades**

### **1. Tela de Login (signin.html)**

* **Formulário de Login**: O usuário pode inserir seu nome de usuário e senha.
* **Validação**: O sistema verifica se as credenciais correspondem a um usuário cadastrado.
* **Mensagem de Erro**: Exibe uma mensagem caso o login falhe.
* **Redirecionamento**: Se o login for bem-sucedido, o usuário é redirecionado para a página inicial.

### **2. Tela de Cadastro (signup.html)**

* **Formulário de Cadastro**: O usuário pode inserir seu nome, nome de usuário, senha e confirmar a senha.
* **Validações**:
  + Verifica se o nome possui no mínimo 3 caracteres.
  + Verifica se o nome de usuário possui no mínimo 5 caracteres.
  + Verifica se a senha possui no mínimo 6 caracteres.
  + Verifica se a senha e a confirmação de senha são iguais.
* **Armazenamento**: Os dados do usuário são salvos no localStorage do navegador.

### **3. Index (index.html)**

* **Controle de Acesso**: A página inicial verifica se o usuário está logado, usando um token armazenado no localStorage.
* **Saída**: O usuário pode sair da sessão, que remove o token e os dados do usuário do localStorage.

## **Tecnologias Utilizadas**

* **HTML**: Para a estrutura das páginas.
* **CSS**: Para estilizar as telas de login e cadastro.
* **JavaScript**: Para a lógica da aplicação, incluindo validações e manipulação do **localStorage.**

## 

## 

## 

## **O Que Falta Fazer**

### **Funcionalidades Pendentes**

1. **Melhorias na UI/UX**:
   * Implementar feedback visual mais claro para validação de formulário.
   * Adicionar animações e transições nas páginas para melhorar a experiência do usuário.
2. **Segurança**:
   * Implementar uma solução de autenticação mais segura em vez de usar localStorage (considerar uso de um backend e banco de dados).
   * Adicionar criptografia para senhas.
3. **Tratamento de Erros**:
   * Melhorar o tratamento de erros para lidar com casos em que o localStorage falha ou não está disponível.
4. **Testes**:
   * Realizar testes de usabilidade e funcionalidade para garantir que todas as partes da aplicação funcionem como esperado.
5. **Documentação**:
   * Atualizar a documentação conforme novas funcionalidades são implementadas e revisadas.
6. **Integração com Laravel**:
   * No futuro, será introduzido o Laravel como framework de backend para gerenciar as operações da aplicação.
   * Um banco de dados será implementado para armazenar os dados dos usuários de forma segura e eficiente, substituindo o uso do localStorage.

## **Considerações Finais**

Este projeto é uma aplicação simples de autenticação, adequada para fins de aprendizado e prática de desenvolvimento web. O uso do localStorage é uma maneira eficaz de armazenar dados localmente, mas não é seguro para aplicações em produção. Para produção, recomenda-se utilizar um banco de dados e implementar autenticação segura com **Laravel.**