

# **Documentação do Sistema de Cadastro de Alunos**

Desenvolvedores: Hian, Pedro e Eduardo

## 1. Introdução

Este sistema tem como objetivo cadastrar alunos, permitindo também editar, remover e buscar registros. É uma aplicação web desenvolvida utilizando as linguagens HTML, CSS e JavaScript, com armazenamento local no navegador (localStorage), dispensando banco de dados.

## 2. Estrutura HTML (index.html)

- Descrição:
- Container centralizado com título, busca, formulário (nome, idade, curso) e tabela.
- Importação de CSS e JavaScript.

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title>Cadastro de Alunos</title>
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <div class="container">
    <h1>Cadastro de Alunos</h1>
    <input type="text" id="buscar" placeholder="Buscar aluno por nome" />
    <form id="form-aluno">
      <input type="text" id="nome" placeholder="Nome" required />
      <input type="number" id="idade" placeholder="Idade" required />
      <input type="text" id="curso" placeholder="Curso" required />
      <button type="submit">Cadastrar</button>
    </form>
    <h2>Lista de Alunos</h2>
    <table id="tabela-alunos">
      <thead>
        <tr>
          <th>Nome</th>
          <th>Idade</th>
          <th>Curso</th>
          <th>Ações</th>
        </tr>
      </thead>
      <tbody></tbody>
    </table>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

### 3. Estilo CSS (style.css)

- Descrição:
- Layout limpo, tabela com efeito zebra, botões coloridos para ações (verde, amarelo, vermelho).

```
body {  
  font-family: 'Segoe UI', sans-serif;  
  background-color: #f2f7fc;  
  margin: 0;  
  padding: 0;  
}  
.container {  
  max-width: 700px;  
  margin: 40px auto;  
  background: #fff;  
  padding: 30px;  
  border-radius: 12px;  
  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);  
}  
h1, h2 {  
  text-align: center;  
  color: #333;  
}  
form {  
  display: flex;  
  flex-direction: column;  
  gap: 10px;  
  margin-bottom: 20px;  
}  
input, button {  
  padding: 10px;  
  font-size: 1rem;  
}  
/* ...restante do CSS... */
```

### 4. Funcionalidade JavaScript (script.js)

- Descrição:
- Adicionar, editar, remover alunos com confirmação.
- Buscar alunos em tempo real.
- Persistência no LocalStorage.

```
let alunos = JSON.parse(localStorage.getItem('alunos')) || [];  
let editIndex = -1;  
  
const form = document.getElementById('form-aluno');  
const tabela = document.getElementById('tabela-alunos').querySelector('tbody');
```

```
const buscar = document.getElementById('buscar');

function atualizarTabela(filtro = '') {
  tabela.innerHTML = '';
  alunos
    .filter(aluno => aluno.nome.toLowerCase().includes(filtro.toLowerCase()))
    .forEach((aluno, index) => {
      const tr = document.createElement('tr');
      tr.innerHTML = `<td>${aluno.nome}</td><td>${aluno.idade}</td><td>${aluno.curso}</td>
<td><button onclick="editarAluno(${index})">Editar</button>
<button onclick="removerAluno(${index})">Remover</button></td>`;
      tabela.appendChild(tr);
    });
}

/* ...funções removerAluno, editarAluno, salvarLocalStorage... */
```

## 5. Conclusão

Este projeto demonstra como é possível construir uma aplicação web completa, funcional e responsiva utilizando apenas HTML, CSS e JavaScript, com dados persistidos localmente.

É uma base sólida para expansão com banco de dados, autenticação, entre outros.