

Guia de Entrega – Projeto JavaScript Avançado (SPA e DOM)

■ Objetivo

Transformar uma interface HTML/CSS estática em uma aplicação dinâmica e interativa usando JavaScript avançado. O projeto deve demonstrar manipulação de DOM, eventos, armazenamento local (localStorage) e um sistema simples de SPA (Single Page Application).

■ Estrutura de Pastas Recomendada

```
meu-projeto/
  index.html
  css/
    style.css
  img/
    logo.png
  js/
    main.js
    router.js
    templates.js
    formValidation.js
    storage.js
  README.md
```

■ Funcionalidades Obrigatórias

- Manipulação do DOM com sistema SPA básico (sem recarregar a página).
- Sistema de templates JavaScript dinâmicos.
- Validação de formulários com mensagens de erro no DOM.
- Armazenamento de dados usando localStorage.

■ Exemplo de Sistema SPA (router.js)

```
const routes = { '/': 'home', '/form': 'form', '/about': 'about' }; export function
navigateTo(path) { window.history.pushState({}, path, window.location.origin + path);
renderTemplate(routes[path]); } window.onpopstate = () => { const path =
window.location.pathname; renderTemplate(routes[path]); };
```

■ Exemplo de Templates Dinâmicos (templates.js)

```
export function renderTemplate(page) { const app = document.getElementById('app');
const templates = { home: `Bem-vindo!`, form: `Enviar`, about: `Sobre o Projeto` };
app.innerHTML = templates[page] || 'Página não encontrada'; }
```

■ Exemplo de Validação de Formulário (formValidation.js)

```
export function initFormValidation() { const form =
document.getElementById('userForm'); if (!form) return; form.addEventListener('submit',
(e) => { e.preventDefault(); const name = document.getElementById('name').value.trim();
const email = document.getElementById('email').value.trim(); const message =
document.getElementById('formMessage'); if (name === '' || email === '') {
message.textContent = 'Preencha todos os campos!'; message.style.color = 'red'; return;
} if (!email.includes('@')) { message.textContent = 'E-mail inválido!';
message.style.color = 'red'; return; } localStorage.setItem('userData',
JSON.stringify({ name, email })); message.textContent = 'Dados salvos com sucesso!';
message.style.color = 'green'; form.reset(); })); }
```

■ Entrega Final

- Estrutura de pastas bem organizada.
- SPA funcional e templates dinâmicos.
- Formulário com validação e mensagens de erro.
- Uso de localStorage para persistência de dados.
- Link público do GitHub com todos os arquivos.