

# Requisitos do Projeto: Aplicativo Web de Lista de Tarefas

## 1 Introdução

Este documento detalha os requisitos para o desenvolvimento de um aplicativo web simples de lista de tarefas (to-do list), focado em front-end, utilizando apenas HTML, CSS e JavaScript puros. O projeto é direcionado a um desenvolvedor júnior, com ênfase em clareza, funcionalidade e design acessível.

## 2 Funcionalidades Principais

### 1. Adicionar Tarefas:

- O usuário pode inserir uma tarefa em um campo de texto.
- A tarefa é adicionada à lista ao clicar no botão “Adicionar” ou pressionar Enter.

### 2. Exibir Tarefas:

- As tarefas aparecem em uma lista abaixo do campo de entrada, em ordem de adição.

### 3. Marcar Tarefas como Concluídas:

- Cada tarefa possui uma caixa de seleção (checkbox) à esquerda.
- Ao marcar, a tarefa deve ser visualmente indicada como concluída (texto riscado e cor diferente).

### 4. Remover Tarefas:

- Cada tarefa tem um botão “Excluir” (ícone “X” ou lixeira) à direita.
- A tarefa é removida da lista ao clicar no botão.

### 5. Persistência Local:

- As tarefas são salvas no `localStorage` do navegador.
- A lista persiste ao recarregar a página.

## 3 Requisitos de Interface (UI)

### • Layout:

- Cabeçalho com o título “Minha Lista de Tarefas”.
- Campo de entrada de texto e botão “Adicionar” abaixo do cabeçalho.
- Lista de tarefas em uma seção clara e organizada.

### • Estilo Visual:

- Paleta de cores simples e moderna (ex.: tons de azul, branco e cinza).
- Campo de entrada e botão com bordas arredondadas e design limpo.
- Tarefas com fundo claro (ex.: branco ou cinza claro), leve sombra ou borda.

- Tarefas concluídas com texto riscado (`text-decoration: line-through`) e cor suave (ex.: cinza).
- Botão “Excluir” vermelho ou com ícone claro (“X” ou lixeira), com efeito *hover*.
- **Responsividade:**
  - Adaptável a telas de 320px (mínimo) a 1200px.
  - Em telas pequenas, campo de entrada e botão ocupam largura total; lista é rolável, se necessário.

## 4 Requisitos Técnicos

- **Tecnologias:** Apenas HTML, CSS e JavaScript puros, sem frameworks ou bibliotecas externas.
- **Estrutura do Código:**
  - Três arquivos: `index.html`, `styles.css`, `script.js`.
  - Usar comentários para explicar seções principais.
  - Código limpo e indentado.
- **Validações:**
  - Impedir adição de tarefas vazias (exibir alerta se o campo estiver vazio).
  - Confirmar exclusão com `confirm("Deseja excluir esta tarefa?")`.
- **Acessibilidade:**
  - Usar tags semânticas (`<header>`, `<main>`, `<section>`).
  - Navegável por teclado (ex.: Enter para adicionar, foco no campo de entrada).
  - Atributos `aria-label` ou `title` nos botões/ícones.

## 5 Critérios de Sucesso

- Aplicativo intuitivo, usável por leigos sem instruções.
- Interface visualmente consistente e agradável em diferentes tamanhos de tela.
- Todas as funcionalidades (adicionar, marcar concluída, excluir, persistência) funcionando sem erros.
- Código organizado e fácil de entender para manutenção por outros desenvolvedores júnior.

## 6 Prazo e Entrega

- **Prazo:** Protótipo funcional em 5 dias úteis.
- **Entrega:** Arquivo ZIP contendo `index.html`, `styles.css`, `script.js` e `README.txt` com instruções de teste.

## 7 Estrutura do Projeto

```
todo-app/  
  index.html  
  styles.css  
  script.js  
  README.txt
```

## 8 Dicas para o Desenvolvedor

- Estruture o HTML e CSS antes de implementar o JavaScript.
- Use `JSON.stringify()` e `JSON.parse()` para o `localStorage`.
- Teste em navegadores (Chrome, Firefox) e diferentes tamanhos de tela.
- Priorize funcionalidades principais (adicionar/exibir tarefas) antes de secundárias (persistência/acessibilidade).

## 9 Observação Final

Estamos animados para ver o resultado! Para esclarecimentos, contate o stakeholder. Boa sorte!