**APLICAÇÃO WEB PARA ORGANIZAÇÃO DE TAREFAS DIÁRIAS**

Irving Higino de Santana Cavalcanti

FLD207825CET

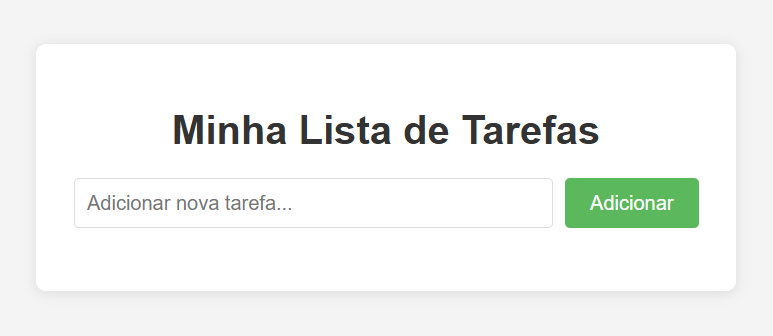
Adriana Neves dos Reis

**RESUMO**

Este trabalho apresenta o desenvolvimento de uma aplicação web voltada ao gerenciamento de tarefas (to do list), com armazenamento local de dados e funcionalidades de exclusão e marcação de tarefas como concluídas. O objetivo do projeto é demonstrar o uso prático de tecnologias web modernas para solucionar um problema cotidiano: o controle eficiente de tarefas pessoais. A metodologia utilizada foi baseada em pesquisa bibliográfica e prática de desenvolvimento incremental. A aplicação permite ao usuário adicionar tarefas, visualizá-las, removê-las e marcar como concluídas, com persistência local dos dados. Os resultados obtidos demonstraram a viabilidade técnica e funcional da solução.

O projeto também é totalmente disponibilizado de maneira digital através do [GitHub](https://github.com/icavalcanti17/To-Do-List), para facilitar acesso de e análise da aplicação desenvolvida: <https://github.com/icavalcanti17/To-Do-List>.

**Palavras-chave**: to do list, aplicação web, banco de dados local.



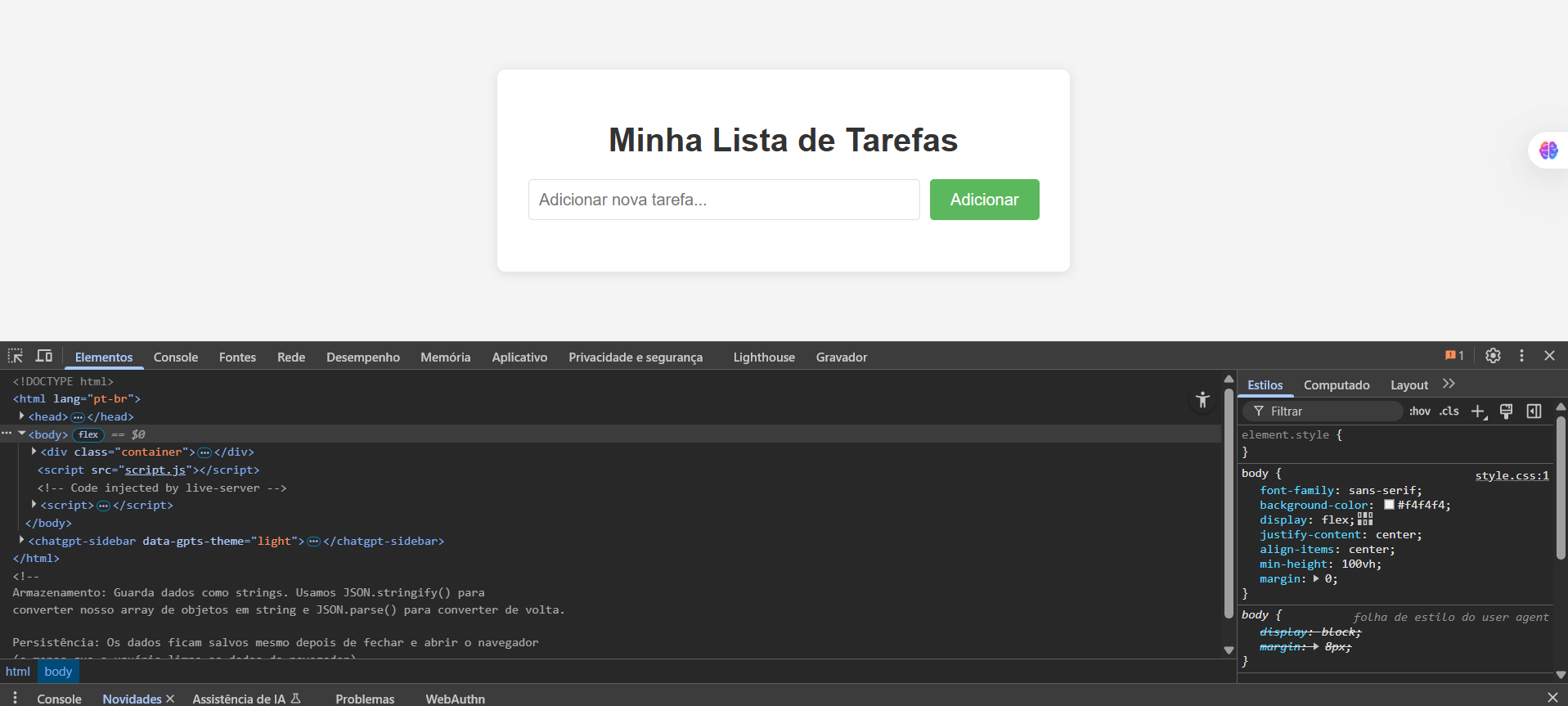
**INTRODUÇÃO**

A organização de tarefas pessoais tornou-se um desafio recorrente em meio ao cotidiano agitado. Neste contexto, ferramentas digitais simples e eficazes tornam-se essenciais. A partir dessa motivação, este trabalho propõe o desenvolvimento de uma aplicação web do tipo "to do list", cujo objetivo principal é permitir o controle pessoal de atividades por meio de um sistema acessível e funcional. A pergunta que norteia este projeto é: como criar uma aplicação web capaz de gerenciar tarefas localmente, de forma intuitiva e eficaz?

**FUNDAMENTAÇÃO TEÓRICA**

**Desenvolvimento Web e Aplicações Interativas**

O desenvolvimento web moderno tem como objetivo criar aplicações interativas, responsivas e acessíveis. Segundo Freeman e Robson (2014), o desenvolvimento web envolve a integração de tecnologias como HTML, CSS e JavaScript para criar experiências ricas no navegador. Aplicações como listas de tarefas (to-do lists) são exemplos clássicos de aplicações de página única (SPA), que oferecem interatividade sem a necessidade de recarregar a página.



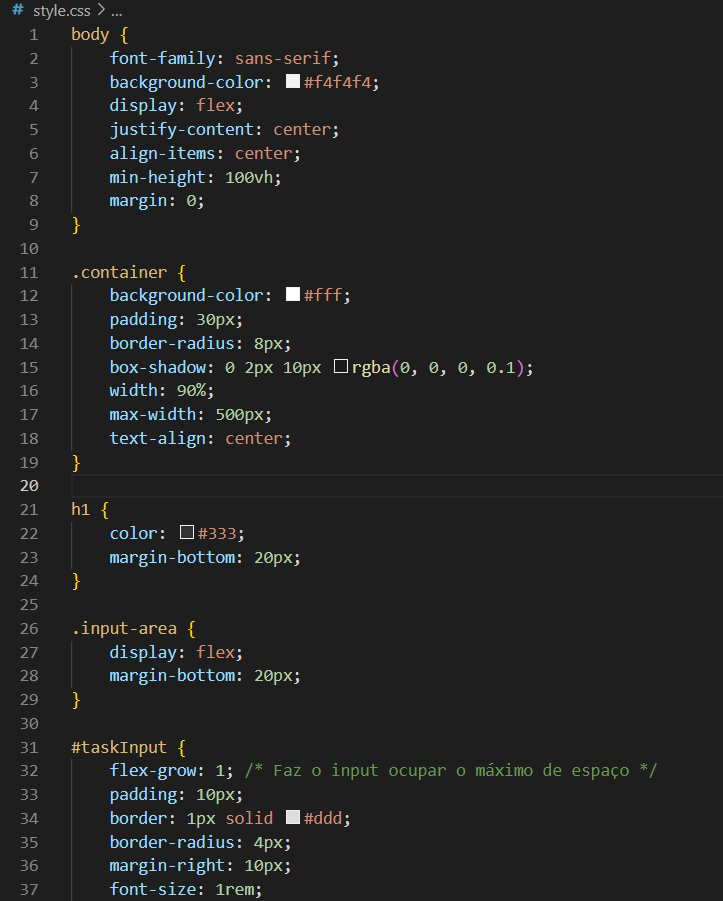
**HTML como Estrutura Semântica**

HTML (HyperText Markup Language) é a espinha dorsal de qualquer aplicação web. De acordo com Duckett (2011), o HTML fornece a estrutura semântica do conteúdo, permitindo que navegadores e tecnologias assistivas compreendam a hierarquia e o propósito de cada elemento. Em uma to-do list, elementos como <ul>, <li>, <input> e <button> são essenciais para estruturar as tarefas e interações do usuário.



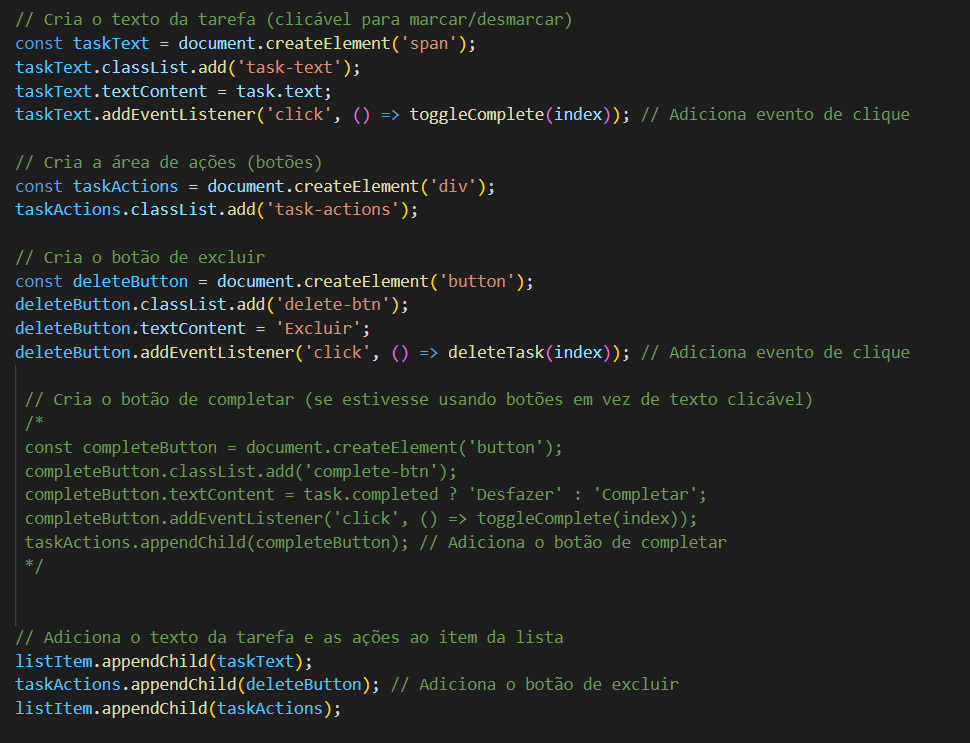
**CSS e a Estilização da Interface**

CSS (Cascading Style Sheets) é responsável pela apresentação visual da aplicação. Conforme Meyer (2016), o CSS permite separar o conteúdo da forma, promovendo manutenibilidade e reutilização de estilos. Em uma lista de tarefas, o CSS pode ser usado para riscar tarefas concluídas com text-decoration: line-through, além de aplicar cores, espaçamentos e animações que melhoram a experiência do usuário.



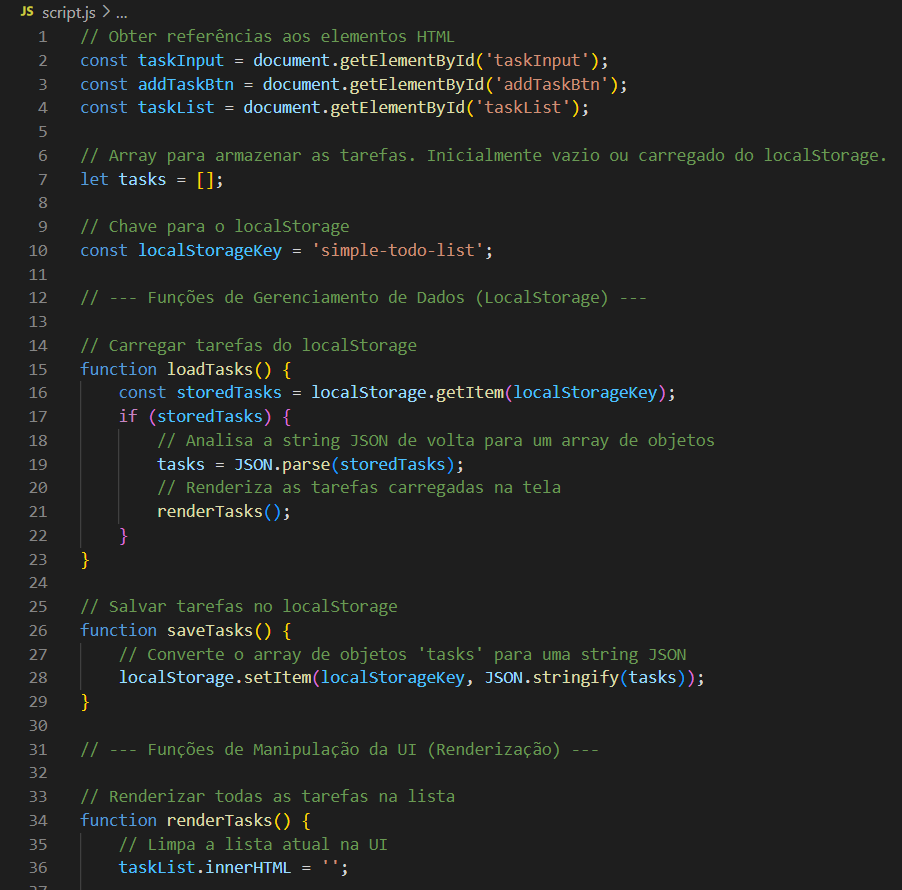
**JavaScript e a Lógica de Interação**

JavaScript é a linguagem de programação que adiciona comportamento dinâmico às páginas web. Segundo Flanagan (2020), JavaScript permite manipular o DOM (Document Object Model), responder a eventos e interagir com APIs do navegador. Em uma to-do list, ele é usado para adicionar, remover e marcar tarefas como concluídas, além de armazenar dados localmente. Neste projeto descrevo cada passo explicando o que está sendo feito, adicionando comentaários (destacados em verde) o que cada linha de código realiza, para melhor didática e entendimento do projeto.



**Armazenamento Local com LocalStorage**

O localStorage é uma API do navegador que permite armazenar dados de forma persistente no lado do cliente. De acordo com a MDN Web Docs, os dados armazenados no localStorage não expiram, permanecendo disponíveis mesmo após o fechamento do navegador. Isso é ideal para aplicações como listas de tarefas, onde o usuário espera que suas informações sejam preservadas entre sessões.



**Segurança e Limitações do LocalStorage**

Embora útil, o localStorage possui limitações. Michel Matos (2024) alerta que os dados armazenados podem ser acessados por qualquer script na página, o que representa um risco de segurança se informações sensíveis forem armazenadas. Portanto, é recomendado usá-lo apenas para dados não confidenciais, como tarefas simples.

**Usabilidade e Experiência do Usuário (UX)**

A usabilidade é um fator crítico no sucesso de aplicações web. Nielsen (1994) define usabilidade como a facilidade com que os usuários podem aprender e utilizar um sistema. Em uma to-do list, isso se traduz em uma interface intuitiva, com botões claros, feedback visual imediato e interações consistentes. A aplicação deve ser acessível e responsiva, adaptando-se a diferentes dispositivos.



**Design Centrado no Usuário (DCU)**

O Design Centrado no Usuário propõe que o desenvolvimento de interfaces deve considerar as necessidades, desejos e limitações dos usuários finais. Segundo Norman (2013), o design eficaz é aquele que antecipa o comportamento do usuário e oferece soluções simples e eficazes. Em uma to-do list, isso pode incluir a possibilidade de editar tarefas, desfazer ações e organizar tarefas por prioridade.

**Acessibilidade e Inclusão Digital**

A acessibilidade garante que pessoas com diferentes habilidades possam utilizar a aplicação. De acordo com a W3C (2023), práticas como uso de aria-labels, contraste adequado de cores e navegação por teclado são fundamentais. Uma to-do list acessível amplia seu alcance e promove inclusão digital.

**Boas Práticas de Desenvolvimento Front-End**

Por fim, o desenvolvimento front-end moderno valoriza a modularidade, reutilização de código e desempenho. Segundo Keith Grant (2018), o uso de componentes reutilizáveis, separação de responsabilidades (HTML para estrutura, CSS para estilo, JS para comportamento) e testes de usabilidade são práticas recomendadas. Isso garante que a aplicação seja escalável, fácil de manter e agradável de usar.

**METODOLOGIA**

O desenvolvimento da aplicação seguiu uma abordagem qualitativa e prática, com pesquisa bibliográfica como base para o entendimento dos conceitos técnicos. A pesquisa é do tipo exploratória, com coleta de dados em fontes confiáveis de desenvolvimento web. A metodologia de construção envolveu etapas como levantamento de requisitos, codificação em HTML, CSS e JavaScript, e testes funcionais. O armazenamento local foi implementado por meio da API LocalStorage. As funcionalidades implementadas incluem adição, exclusão e marcação de tarefas como concluídas.

**CONSIDERAÇÕES**

A construção desta aplicação web permitiu compreender, na prática, conceitos fundamentais de desenvolvimento front-end e manipulação de dados localmente. Além da experiência técnica adquirida, o projeto reforçou a importância da organização de tarefas na rotina diária e como soluções simples podem ter grande impacto na produtividade pessoal.

**REFERÊNCIAS BIBLIOGRÁFICAS**

Este resumo apresenta livros e recursos online recomendados para aprender e pesquisar sobre desenvolvimento front-end (HTML, CSS, JavaScript).

Referências Bibliográficas (Livros Sugeridos):

Duckett, Jon. HTML and CSS (2011). Ótimo para fundamentos de HTML5 e CSS3 de forma visual e didática.

Duckett, Jon. JavaScript and JQuery (2014). Focado em JavaScript essencial, DOM e interatividade.

Robbins, Jennifer Niederst. Learning Web Design (ed. mais recente). Abrangente: HTML, CSS, JS, gráficos, em um volume só.

Haverbeke, Marijn. Eloquent JavaScript. Livro denso sobre JavaScript puro. Disponível online gratuitamente.

Ao usar, cite no formato padrão (ABNT, APA, etc.).

Recursos Online Essenciais para Pesquisa e Aprendizado:

MDN Web Docs (developer.mozilla.org): Documentação oficial, completa e atualizada (HTML, CSS, JS, APIs Web). A "bíblia" do desenvolvimento web. Essencial para detalhes e referência precisa.

W3Schools (w3schools.com): Tutoriais e exemplos interativos (HTML, CSS, JS). Bom para sintaxe básica e testes rápidos.

freeCodeCamp (freecodecamp.org): Currículo prático e gratuito com foco em projetos (HTML, CSS, JS).

The Odin Project (theodinproject.com): Currículo gratuito focado em "aprender fazendo" e usar documentação oficial.

Plataformas de Cursos (Udemy, Coursera, etc.): Cursos estruturados (muitos pagos) para aprendizado guiado.

Conceitos Chave para Pesquisar (Use MDN ou W3Schools):

Manipulação do DOM (Document Object Model)

Tratamento de Eventos (Event Handling)

O que é Desenvolvimento Front-end

Manipulação de Classes CSS com JavaScript (usando classList)