**Documentação do Projeto: Calculador de Horários de Remédio**

**Visão Geral**

Este projeto consiste em um web application desenvolvido utilizando HTML, CSS e JavaScript. O objetivo principal é auxiliar usuários a calcular e visualizar os horários para tomar seus medicamentos, com base na frequência prescrita e no horário inicial de administração.

O projeto é estruturado em três arquivos principais:

1. **index.html:** Contém a estrutura HTML da página, incluindo os formulários para entrada de dados dos medicamentos, os botões de interação e a tabela onde os horários calculados serão exibidos.
2. **style.css:** Define toda a estilização visual da página, garantindo uma apresentação agradável e responsiva em diferentes dispositivos (celulares, tablets e desktops).
3. **script.js:** Implementa a lógica de programação, incluindo a capacidade de adicionar múltiplos medicamentos, o cálculo dos horários de tomada e a manipulação do DOM (Document Object Model) para exibir os resultados na tabela.

**1. Código HTML (index.html)**

O arquivo index.html fornece a base estrutural do nosso web app.

HTML

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Calculador de Horários de Remédio</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<h1>Calculador de Horários de Remédio</h1>

<div id="medicamentos-container">

<div class="medicamento-form">

<h2>Novo Medicamento</h2>

<label for="nome">Nome do Medicamento:</label>

<input type="text" id="nome" name="nome" required>

<label for="frequencia">Frequência (em horas):</label>

<input type="number" id="frequencia" name="frequencia" min="1" required>

<label for="horario-inicial">Horário Inicial:</label>

<input type="datetime-local" id="horario-inicial" name="horario-inicial" required>

</div>

</div>

<button id="adicionar-medicamento">Adicionar Outro Medicamento</button>

<button id="calcular-horarios">Calcular Horários</button>

<h2>Horários de Tomada</h2>

<table id="horarios-container">

<thead>

<tr>

<th>Dia do Mês</th>

<th>Dia da Semana</th>

<th>Horário</th>

<th>Medicamento</th>

</tr>

</thead>

<tbody>

</tbody>

</table>

<script src="script.js"></script>

</body>

</html>

**Estrutura e Elementos Principais:**

* **<head>:** Contém metadados sobre o documento, define o título da página e linka o arquivo style.css para aplicar os estilos visuais.
* **<h1>Calculador de Horários de Remédio</h1>:** Título principal da página.
* **<div id="medicamentos-container">:** Container que armazena os formulários para cada medicamento. Inicialmente, um formulário já está presente.
* **<div class="medicamento-form">:** Representa um formulário individual para inserir os detalhes de um medicamento (nome, frequência e horário inicial).
  + **<label>:** Rótulos descritivos para cada campo de entrada.
  + **<input type="text">:** Campo para o nome do medicamento.
  + **<input type="number">:** Campo para a frequência em horas (com validação min="1").
  + **<input type="datetime-local">:** Campo para selecionar a data e a hora inicial da primeira dose.
* **<button id="adicionar-medicamento">:** Botão que, ao ser clicado, utiliza JavaScript para adicionar dinamicamente mais formulários de medicamentos.
* **<button id="calcular-horarios">:** Botão que, ao ser clicado, aciona a função JavaScript para calcular os horários de tomada e exibi-los na tabela.
* **<h2>Horários de Tomada</h2>:** Título da seção onde a tabela de horários será exibida.
* **<table id="horarios-container">:** Tabela HTML para exibir os horários calculados.
  + **<thead>:** Cabeçalho da tabela com as colunas "Dia do Mês", "Dia da Semana", "Horário" e "Medicamento".
  + **<tbody>:** Corpo da tabela, que será preenchido dinamicamente com os horários calculados através do JavaScript.
* **<script src="script.js"></script>:** Link para o arquivo JavaScript que contém a lógica do aplicativo. É colocado no final do <body> para garantir que todos os elementos HTML sejam carregados antes da execução do script.
* **Revisão**

A seção do formulário em index.html foi atualizada para incluir o campo de "Duração (em dias)":

HTML

<div id="medicamentos-container">

<div class="medicamento-form">

<h2>Novo Medicamento</h2>

<label for="nome">Nome do Medicamento:</label>

<input type="text" id="nome" name="nome" required>

<label for="frequencia">Frequência (em horas):</label>

<input type="number" id="frequencia" name="frequencia" min="1" required>

<label for="horario-inicial">Horário Inicial:</label>

<input type="datetime-local" id="horario-inicial" name="horario-inicial" required>

<label for="duracao">Duração (em dias):</label>

<input type="number" id="duracao" name="duracao" min="1" required>

</div>

</div>

* Um novo <label for="duracao"> e <input type="number" id="duracao" name="duracao" min="1" required> foram adicionados para permitir que o usuário especifique a duração do tratamento em dias.

**2. Código CSS (style.css)**

O arquivo style.css define a aparência da página, incluindo o layout, as cores, as fontes e a responsividade para diferentes tamanhos de tela.

CSS

/\* style.css \*/

body {

font-family: sans-serif;

margin: 15px; /\* Reduzindo um pouco a margem padrão \*/

}

.medicamento-form {

border: 1px solid #ccc;

padding: 10px; /\* Reduzindo um pouco o padding \*/

margin-bottom: 10px;

border-radius: 5px;

}

.medicamento-form h2 {

font-size: 1.2em; /\* Reduzindo o tamanho da fonte do título \*/

margin-top: 0;

}

.medicamento-form label {

display: block;

margin-bottom: 3px; /\* Reduzindo um pouco a margem inferior \*/

font-weight: bold;

font-size: 0.9em; /\* Reduzindo o tamanho da fonte da label \*/

}

.medicamento-form input[type="text"],

.medicamento-form input[type="number"],

.medicamento-form input[type="datetime-local"] {

width: calc(100% - 10px); /\* Ajuste para a borda e padding \*/

padding: 5px;

margin-bottom: 8px; /\* Reduzindo um pouco a margem inferior \*/

border: 1px solid #ddd;

border-radius: 3px;

box-sizing: border-box; /\* Garante que padding e border não aumentem a largura total \*/

font-size: 0.9em; /\* Reduzindo o tamanho da fonte dos inputs \*/

}

button {

padding: 8px 12px; /\* Reduzindo um pouco o padding dos botões \*/

background-color: #007bff; /\* Cor azul padrão para botões \*/

color: white;

border: none;

border-radius: 5px;

cursor: pointer; /\* Indica que o elemento é clicável \*/

font-size: 0.9em; /\* Reduzindo o tamanho da fonte dos botões \*/

margin-right: 5px; /\* Adicionando um pouco de espaço entre os botões \*/

}

button:hover {

background-color: #0056b3; /\* Cor azul mais escura ao passar o mouse \*/

}

#horarios-container {

margin-top: 15px; /\* Reduzindo um pouco a margem superior \*/

border-collapse: collapse; /\* Combina as bordas da tabela em uma única \*/

width: 100%; /\* Faz a tabela ocupar a largura total do container \*/

font-size: 0.8em; /\* Reduzindo o tamanho da fonte da tabela \*/

}

#horarios-container th,

#horarios-container td {

border: 1px solid #ccc;

padding: 6px; /\* Reduzindo um pouco o padding das células da tabela \*/

text-align: center; /\* Alinha o texto das células ao centro \*/

}

.medicamento-destaque {

font-weight: bold;

color: green; /\* Cor verde para destacar o nome do medicamento \*/

}

/\* Media Query para telas menores que 768px (tablets e celulares) \*/

@media (max-width: 768px) {

body {

margin: 10px; /\* Reduzindo ainda mais a margem em telas menores \*/

}

.medicamento-form {

padding: 10px;

}

.medicamento-form label {

font-size: 0.85em;

}

.medicamento-form input[type="text"],

.medicamento-form input[type="number"],

.medicamento-form input[type="datetime-local"] {

font-size: 0.85em;

}

button {

font-size: 0.85em;

padding: 6px 10px;

margin-bottom: 5px; /\* Adicionando margem inferior entre os botões em telas menores \*/

}

#horarios-container {

font-size: 0.75em;

}

#horarios-container th,

#horarios-container td {

padding: 4px;

}

}

/\* Media Query para telas ainda menores (celulares a partir de 375px) \*/

@media (max-width: 375px) {

h1 {

font-size: 1.5em; /\* Reduzindo o tamanho da fonte do título principal em telas muito pequenas \*/

}

.medicamento-form h2 {

font-size: 1.1em;

}

}

/\* Media Query para telas maiores (acima de 1200px - desktops grandes) \*/

@media (min-width: 1200px) {

body {

max-width: 1200px; /\* Limita a largura máxima do conteúdo em telas grandes \*/

margin: 20px auto; /\* Centraliza o conteúdo na tela \*/

}

}

**Explicação das Regras de Estilo:**

* **body:** Define a fonte padrão (sans-serif) e as margens da página.
* **.medicamento-form:** Estiliza o container de cada formulário de medicamento com uma borda, padding, margem inferior e bordas arredondadas.
* **.medicamento-form h2:** Estiliza o título "Novo Medicamento" dentro de cada formulário.
* **.medicamento-form label:** Estiliza os rótulos dos campos de entrada para serem exibidos como blocos, com margem inferior, negrito e um tamanho de fonte menor.
* **.medicamento-form input[type="text"], .medicamento-form input[type="number"], .medicamento-form input[type="datetime-local"]:** Estiliza os campos de entrada de texto, número e data/hora para ocupar a largura total do container, com padding, margem inferior, borda e bordas arredondadas. A propriedade box-sizing: border-box; garante que o padding e a borda não aumentem a largura total do elemento.
* **button:** Estiliza os botões com padding, cor de fundo azul, texto branco, sem borda, bordas arredondadas, cursor de "pointer" (para indicar que são clicáveis) e um tamanho de fonte menor.
* **button:hover:** Define o estilo do botão quando o mouse passa por cima, alterando a cor de fundo para um azul mais escuro.
* **#horarios-container:** Estiliza a tabela de horários com margem superior, colapso das bordas (para uma aparência de grade única) e largura total. O tamanho da fonte da tabela também é reduzido.
* **#horarios-container th, #horarios-container td:** Estiliza as células de cabeçalho e de dados da tabela com uma borda, padding e alinhamento de texto centralizado.
* **.medicamento-destaque:** Aplica um estilo de negrito e cor verde ao texto para destacar o nome do medicamento na tabela de horários.

**Media Queries (Responsividade):**

As Media Queries são usadas para aplicar estilos diferentes dependendo da largura da tela do dispositivo, garantindo a responsividade do layout:

* **@media (max-width: 768px):** Aplica estilos específicos para telas com largura máxima de 768 pixels (tablets e celulares). Reduz ainda mais as margens, paddings e tamanhos de fonte para melhor se adaptar a telas menores. Adiciona margem inferior entre os botões para melhor espaçamento em layouts verticais.
* **@media (max-width: 375px):** Aplica estilos para telas ainda menores (celulares a partir de 375px). Reduz o tamanho da fonte dos títulos principais e dos títulos dos formulários.
* **@media (min-width: 1200px):** Aplica estilos para telas maiores (desktops grandes). Limita a largura máxima do body para 1200 pixels e centraliza o conteúdo na tela, evitando que o layout fique excessivamente esticado em monitores grandes.

**3. Código JavaScript (script.js)**

O arquivo script.js contém a lógica principal do nosso web app, responsável por adicionar dinamicamente formulários de medicamentos (funcionalidade que permanece, embora a forma de interação tenha mudado), persistir os dados localmente utilizando localStorage e JSON, calcular os horários de tomada e exibir os resultados na tabela.

JavaScript

// script.js

document.addEventListener('DOMContentLoaded', function() {

const medicamentosContainer = document.getElementById('medicamentos-container');

const adicionarMedicamentoBotao = document.getElementById('adicionar-medicamento');

const calcularHorariosBotao = document.getElementById('calcular-horarios');

const horariosContainer = document.getElementById('horarios-container').getElementsByTagName('tbody')[0];

let contadorMedicamentos = 1; // Contador para gerar IDs únicos para os novos formulários

// Função para criar um novo formulário de medicamento

function criarFormularioMedicamento() {

contadorMedicamentos++;

const novoFormulario = document.createElement('div');

novoFormulario.classList.add('medicamento-form');

novoFormulario.innerHTML = `

<h2>Novo Medicamento</h2>

<label for="nome-${contadorMedicamentos}">Nome do Medicamento:</label>

<input type="text" id="nome-${contadorMedicamentos}" name="nome-${contadorMedicamentos}" required>

<label for="frequencia-${contadorMedicamentos}">Frequência (em horas):</label>

<input type="number" id="frequencia-${contadorMedicamentos}" name="frequencia-${contadorMedicamentos}" min="1" required>

<label for="horario-inicial-${contadorMedicamentos}">Horário Inicial:</label>

<input type="datetime-local" id="horario-inicial-${contadorMedicamentos}" name="horario-inicial-${contadorMedicamentos}" required>

<label for="duracao-${contadorMedicamentos}">Duração (em dias):</label>

<input type="number" id="duracao-${contadorMedicamentos}" name="duracao-${contadorMedicamentos}" min="1" required>

`;

return novoFormulario;

}

// Adiciona um novo formulário de medicamento ao clicar no botão

adicionarMedicamentoBotao.addEventListener('click', function() {

const novoFormulario = criarFormularioMedicamento();

medicamentosContainer.appendChild(novoFormulario);

});

// Adiciona um ouvinte de evento para o botão de calcular horários

calcularHorariosBotao.addEventListener('click', calcularEExibirHorarios);

// Função que será chamada ao clicar no botão "Calcular Horários"

function calcularEExibirHorarios() {

// Coleta os dados dos medicamentos dos formulários ATUAIS

const medicamentosFormulario = [];

const formulariosMedicamentos = medicamentosContainer.querySelectorAll('.medicamento-form');

formulariosMedicamentos.forEach(formulario => {

const nomeMedicamento = formulario.querySelector('[name^="nome"]').value;

const frequencia = parseInt(formulario.querySelector('[name^="frequencia"]').value);

const horarioInicialInput = formulario.querySelector('[name^="horario-inicial"]').value;

const duracaoDias = parseInt(formulario.querySelector('[name^="duracao"]').value);

if (nomeMedicamento && !isNaN(frequencia) && horarioInicialInput && !isNaN(duracaoDias)) {

medicamentosFormulario.push({

nome: nomeMedicamento,

frequencia: frequencia,

horarioInicial: horarioInicialInput,

duracao: duracaoDias

});

}

});

// Carrega os medicamentos salvos do localStorage

const medicamentosSalvosJSON = localStorage.getItem('medicamentos');

let medicamentosSalvos = [];

if (medicamentosSalvosJSON) {

medicamentosSalvos = JSON.parse(medicamentosSalvosJSON);

}

// Combina os medicamentos do formulário com os salvos

const todosMedicamentos = [...medicamentosSalvos, ...medicamentosFormulario];

// Salva todos os medicamentos no localStorage (incluindo os novos)

if (todosMedicamentos.length > 0) {

localStorage.setItem('medicamentos', JSON.stringify(todosMedicamentos));

} else {

localStorage.removeItem('medicamentos'); // Remove se não houver medicamentos

}

// Limpa a tabela de horários antes de exibir novos resultados

horariosContainer.innerHTML = '';

const todosHorarios = [];

todosMedicamentos.forEach(medicamento => {

const { nome, frequencia, horarioInicial, duracao } = medicamento;

if (nome && !isNaN(frequencia) && horarioInicial && !isNaN(duracao)) {

const horarioInicialDate = new Date(horarioInicial);

const dataFinal = new Date(horarioInicialDate);

dataFinal.setDate(horarioInicialDate.getDate() + duracao - 1);

let horarioAtual = new Date(horarioInicialDate);

while (horarioAtual <= dataFinal) {

todosHorarios.push({

medicamento: nome,

horario: new Date(horarioAtual)

});

horarioAtual.setTime(horarioAtual.getTime() + frequencia \* 60 \* 60 \* 1000);

}

}

});

// Ordena os horários cronologicamente

todosHorarios.sort((a, b) => a.horario - b.horario);

// Exibe os horários na tabela

todosHorarios.forEach(item => {

const linha = horariosContainer.insertRow();

const diaMesCelula = linha.insertCell();

const diaSemanaCelula = linha.insertCell();

const horarioCelula = linha.insertCell();

const medicamentoCelula = linha.insertCell();

const data = item.horario;

const diaMes = data.getDate();

const diaSemana = obterDiaSemana(data.getDay());

const horarioFormatado = formatarHorario(data);

diaMesCelula.textContent = diaMes < 10 ? `0${diaMes}` : diaMes;

diaSemanaCelula.textContent = diaSemana;

horarioCelula.textContent = horarioFormatado;

medicamentoCelula.textContent = item.medicamento;

medicamentoCelula.classList.add('medicamento-destaque');

});

if (todosHorarios.length === 0 && todosMedicamentos.length > 0) {

alert('Por favor, verifique se os dados dos medicamentos salvos estão corretos.');

} else if (todosMedicamentos.length === 0) {

alert('Por favor, adicione pelo menos um medicamento para calcular os horários.');

}

// Limpa o formulário inicial após o cálculo

if (formulariosMedicamentos.length > 0) {

formulariosMedicamentos[0].querySelector('[name^="nome"]').value = '';

formulariosMedicamentos[0].querySelector('[name^="frequencia"]').value = '';

formulariosMedicamentos[0].querySelector('[name^="horario-inicial"]').value = '';

formulariosMedicamentos[0].querySelector('[name^="duracao"]').value = '';

// Remove os formulários adicionais, mantendo apenas o primeiro vazio

for (let i = formulariosMedicamentos.length - 1; i > 0; i--) {

medicamentosContainer.removeChild(formulariosMedicamentos[i]);

contadorMedicamentos--;

}

}

}

// Função auxiliar para obter o dia da semana por extenso

function obterDiaSemana(dia) {

const dias = ['Domingo', 'Segunda-feira', 'Terça-feira', 'Quarta-feira', 'Quinta-feira', 'Sexta-feira', 'Sábado'];

return dias[dia];

}

// Função auxiliar para formatar a hora no formato HH:MM

function formatarHorario(data) {

const horas = data.getHours().toString().padStart(2, '0');

const minutos = data.getMinutes().toString().padStart(2, '0');

return `${horas}:${minutos}`;

}

// Carrega os dados dos medicamentos do localStorage ao carregar a página e calcula os horários

function carregarMedicamentosSalvosEExibir() {

const medicamentosSalvosJSON = localStorage.getItem('medicamentos');

// Os dados serão carregados e usados diretamente na função calcularEExibirHorarios

calcularEExibirHorarios(); // Chama para exibir os horários com base nos dados carregados (e possíveis novos)

}

// Chama a função para carregar e exibir ao carregar a página

carregarMedicamentosSalvosEExibir();

});

**Explicação do Código JavaScript:**

1. **document.addEventListener('DOMContentLoaded', function() { ... });:** Garante que o script seja executado após o carregamento completo do HTML.
2. **Seleção de Elementos do DOM:** Obtém referências aos elementos HTML necessários.
3. **contadorMedicamentos e criarFormularioMedicamento():** Funcionam da mesma forma, permitindo a criação dinâmica de novos formulários de medicamentos.
4. **Eventos de Clique:** Os listeners de evento para os botões "Adicionar Outro Medicamento" e "Calcular Horários" permanecem, com o botão de calcular acionando a função calcularEExibirHorarios.
5. **calcularEExibirHorarios() (Atualizada):**
   * **Coleta de dados dos formulários atuais:** Itera sobre os formulários visíveis e coleta os dados de nome, frequência, horário inicial e duração, armazenando-os em medicamentosFormulario.
   * **Carregamento dos dados do localStorage:** Tenta obter os dados salvos com a chave 'medicamentos' e os converte de JSON para um array JavaScript (medicamentosSalvos).
   * **Combinação dos dados:** Combina os medicamentos recém-inseridos nos formulários (medicamentosFormulario) com os medicamentos já salvos (medicamentosSalvos) em um array todosMedicamentos.
   * **Salvamento no localStorage:** Salva o array todosMedicamentos (que agora inclui tanto os dados antigos quanto os novos) de volta no localStorage.
   * **Cálculo dos horários:** Itera sobre cada medicamento em todosMedicamentos, calcula os horários de tomada dentro do período de duração especificado e os armazena no array todosHorarios.
   * **Ordenação dos horários:** Ordena o array todosHorarios cronologicamente.
   * **Exibição na tabela:** Limpa a tabela e, em seguida, preenche-a com os horários calculados, mostrando o dia do mês, dia da semana, horário e o nome do medicamento em destaque.
   * **Alertas:** Exibe alertas se não houver medicamentos para calcular ou se os dados estiverem incompletos.
   * **Limpeza do formulário:** Após o cálculo e exibição, o formulário inicial é limpo e quaisquer formulários adicionais são removidos, deixando a interface pronta para a inserção de um novo medicamento, se desejado.
6. **Funções Auxiliares (obterDiaSemana, formatarHorario):** Permanecem as mesmas, responsáveis por formatar a saída dos dados na tabela.
7. **carregarMedicamentosSalvosEExibir() (Atualizada):** Esta função agora simplesmente tenta obter os dados do localStorage (que serão utilizados na chamada subsequente a calcularEExibirHorarios()) e sempre chama calcularEExibirHorarios() para que os horários sejam exibidos com base nos dados carregados (e quaisquer novos dados inseridos no formulário antes do clique em "Calcular Horários").
8. **Chamada de carregarMedicamentosSalvosEExibir():** Garante que a lógica de carregamento e exibição seja iniciada quando a página é carregada.