

# Tutorial Técnico

## Estrutura do Projeto

O projeto é composto por três arquivos principais:

pichu2.html: Estrutura HTML da aplicação (interface do usuário)

style.css: Estilização (design, cores, layout responsivo)

Pichu3.js: Lógica de interação (adicionar tarefas, marcar como concluído, temas)

## HTML (pichu2.html)

### Estrutura Básica do Documento

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Projeto Pichu - Protótipo</title>
7   <link rel="stylesheet" href="style.css">
8 </head>
```

Componentes Fundamentais:

<!DOCTYPE html>: Declaração do tipo de documento HTML5

<html lang="pt-BR">: Elemento raiz com definição de idioma (português brasileiro)

Meta Tags:

-charset="UTF-8": Suporte a caracteres especiais e acentos

-viewport: Configuração para responsividade em dispositivos móveis

Título: "Projeto Pichu - Protótipo" aparece na aba do navegador

CSS: Link para o arquivo de estilos style.css

## Componentes Principais:

### Botão de Alternar Tema

```
11 <button id="toggleTheme">Alternar Tema</button>
```

Posicionamento: Fixo no canto superior direito (controlado pelo CSS)

Função: Alterna entre os temas claro e escuro (implementado em JavaScript)

Identificador: id="toggleTheme" para seleção no JavaScript

### Cabeçalho (Header)

```
<header>
  <h1>Projeto Pichu - Protótipo</h1>
</header>
```

Propósito: Exibe o título principal da aplicação

Estilização: Centralizado e com margem inferior (definido no CSS)

### Área Principal (Main)

```
18 <div id="mensagemParabens" style="display: none; margin-top: 20px; font-size: 1.3em; font-weight: bold; color: green;
19   background: #d4edda; padding: 15px; border-radius: 8px; border: 2px solid #c3e6cb;">
20   Parabéns! Você concluiu todas as suas tarefas!
21 </div>
```

Comportamento: Inicialmente oculta (display: none)

Aparece quando: Todas tarefas são marcadas como concluídas (via JavaScript)

Estilo Inline: Cores de sucesso (verde) com bordas arredondadas

### Seção de Tarefas

```
21 <section class="tarefas">
22   <h2>Tarefas</h2>
23   <div style="display: flex; gap: 10px; width: 90%; margin-bottom: 10px;">
24     <input type="text" id="novaTarefa" placeholder="Nova tarefa..." style="flex: 1; padding: 8px; border-radius:
25       6px; border: 1px solid #ccc;">
26     <button id="adicionarTarefa" style="padding: 8px 12px; border-radius: 6px; border: none; background-color:
27       #007bff; color: white;">Adicionar</button>
28   </div>
29   <ul id="listaTarefas" style="width: 90%; text-align: left;"></ul>
30 </section>
```

### Componentes:

- Campo de texto (input) para digitar novas tarefas
- Botão para adicionar a tarefa à lista

-Lista não ordenada (ul) para exibir as tarefas pendentes

Identificadores:

-id="novaTarefa": Seleciona o campo de input no JS

-id="adicionarTarefa": Seleciona o botão no JS

-id="listaTarefas": Seleciona a lista de tarefas no JS

### Seção de Metas Concluídas

```
30 <section class="metas">
31   <h2>Metas Concluídas</h2>
32   <ul id="listaMetas" style="width: 90%; text-align: left;"></ul>
33 </section>
```

Propósito: Exibe tarefas marcadas como concluídas

Dinâmica: Itens são movidos para cá via JavaScript quando marcados

Identificador: id="listaMetas" para manipulação no JS

### Seção de Observações

```
35 <section class="observacoes">
36   <h2>Observações</h2>
37   <textarea placeholder="Adicione suas observações..."></textarea>
38 </section>
```

Funcionalidade: Área de texto livre para anotações gerais

Características:

-Placeholder como texto de orientação

-Redimensionável (controlado pelo CSS)

### Rodapé (Footer)

```
41 <footer> Seguir o link (ctrl + clique)
42   
43   
44 </footer>
```

Sistema de Temas:

-light-only: Mostra girassol no tema claro

-dark-only: Mostra imagem azul no tema escuro

Acessibilidade: Atributos alt para descrição das imagens

Inclusão do JavaScript

```
46 <script src="Pichu3.js"></script>
```

Posicionamento: No final do body para garantir que o DOM esteja carregado

Função: Contém toda a lógica interativa da aplicação

## CSS (style.css)

Estilo Base do Body

```
3 body {  
4     font-family: Arial, sans-serif;  
5     display: flex;  
6     flex-direction: column;  
7     align-items: center;  
8     background-color: #f8f9fa;  
9     padding: 20px;  
10    height: 100vh;  
11    position: relative;  
12 }
```

Propósito: Define o estilo fundamental da página

Características:

- Fonte padrão Arial para todo o documento
- Layout flexbox vertical (coluna) com conteúdo centralizado
- Cor de fundo clara padrão (#f8f9fa)
- Altura fixa igual à viewport (100vh)
- Posição relativa para elementos filhos posicionados absolutamente

## Botão de Alternar Tema

```
15  #toggleTheme {  
16      position: fixed;  
17      top: 15px;  
18      right: 20px;  
19      padding: 8px 12px;  
20      border-radius: 6px;  
21      border: none;  
22      cursor: pointer;  
23      background-color: #007bff;  
24      color: white;  
25      font-weight: bold;  
26      z-index: 1000;  
27  }  
28  
29  .dark-theme #toggleTheme {  
30      background-color: #0d6efd;  
31  }
```

Funcionalidade: Botão fixo no canto superior direito

Detalhes:

- Posicionamento fixo na tela (não rola com a página)
- Estilo visual com bordas arredondadas e sombras
- Z-index alto para garantir que fique acima de outros elementos
- Cor alterada no tema escuro para melhor contraste

## Container Principal (Grid Layout)

```
34  ∨ .container {  
35      display: grid;  
36      grid-template-columns: 1.7fr 1fr;  
37      grid-template-rows: 2fr 1fr;  
38      gap: 20px;  
39      width: 90%;  
40      max-width: 1200px;  
41      height: 80vh;  
42  }
```

Layout: Sistema de grid com 2 colunas e 2 linhas

Proporções:

- Coluna esquerda (tarefas) ocupa 1.7fr
- Coluna direita (metas e observações) ocupa 1fr
- Seção de tarefas ocupa 2 linhas
- Gap de 20px entre elementos

## Estilização das Seções

```
45  ∨ section {  
46      background-color: #ffffff;  
47      display: flex;  
48      flex-direction: column;  
49      align-items: center;  
50      border: 2px solid #ddd;  
51      border-radius: 12px;  
52      text-align: center;  
53      padding: 20px;  
54      box-shadow: 0px 4px 6px rgba(0, 0, 0, 0.1);  
55  }
```

Design:

- Cartões com bordas arredondadas (12px)
- Sombras sutis para efeito de profundidade

- Espaçamento interno consistente (20px)
- Layout flexbox interno para alinhamento vertical

### Posicionamento Específico das Seções

```
72  ∨ .tarefas {  
73    |   grid-row: span 2;  
74  }  
75  
76  ∨ .metas {  
77    |   grid-column: 2;  
78    |   grid-row: 1;  
79  }  
80  
81  ∨ .observacoes {  
82    |   grid-column: 2;  
83    |   grid-row: 2;  
84  }
```

### Organização:

- Tarefas ocupam as duas linhas na primeira coluna
- Metas ficam na primeira linha da segunda coluna
- Observações ficam na segunda linha da segunda coluna

### Sistema de Temas (Claro/Escuro)

```

113  .dark-theme {
114      background-color: #121212;
115      color: #f1f1f1;
116  }
117
118  .dark-theme section {
119      background-color: #1e1e1e;
120      border-color: #333;
121      box-shadow: 0px 4px 6px rgba(255, 255, 255, 0.05);
122  }
123
124  .dark-theme section textarea {
125      background-color: #2a2a2a;
126      color: #f1f1f1;
127      border: 1px solid #444;
128  }
129
130  .dark-theme header h1,
131  .dark-theme footer p {
132      color: #f1f1f1;
133  }

```

Mecanismo:

- A classe .dark-theme no body ativa todas as variações
- Cores escuras para fundo e texto
- Ajustes de contraste para elementos interativos
- Bordas mais escuras para melhor visibilidade

Controle de Imagens por Tema



```
136 ∨ .light-only {  
137   |   display: block;  
138   }  
139  
140 ∨ .dark-only {  
141   |   display: none;  
142   }  
143  
144 ∨ body.dark-theme .light-only {  
145   |   display: none !important;  
146   }  
147  
148 ∨ body.dark-theme .dark-only {  
149   |   display: block !important;  
150   }  
151
```

Lógica:

- Imagens com classe .light-only visíveis apenas no tema claro
- Imagens com classe .dark-only visíveis apenas no tema escuro
- Uso de !important para garantir precedência

## Responsividade (Media Queries)

```
153 ④ @media screen and (max-width: 900px) {
154 ④   .container {
155     grid-template-columns: 1fr;
156     grid-template-rows: auto;
157     height: auto;
158   }
159
160 ④ .tarefas, .metas, .observacoes {
161     grid-column: auto;
162     grid-row: auto;
163     width: 100%;
164   }
165
166 ④ section textarea {
167     height: 150px;
168   }
169 }
170
171 ④ @media screen and (max-width: 600px) {
172 ④   body {
173     padding: 10px;
174   }
175
176 ④   header h1 {
177     font-size: 1.5em;
178   }
179
180 ④   .container {
181     gap: 10px;
182   }
```

Breakpoints:

-900px: Muda para layout de coluna única

-600px: Ajustes adicionais para telas pequenas

Ajustes:

- Redução de tamanhos de fonte
- Diminuição de espaçamentos
- Altura adaptável para áreas de texto

## JS(Pichu3.js)

### Estrutura Geral do Código

```
1 document.addEventListener("DOMContentLoaded", () => {
```

Propósito: Garante que todo o JavaScript execute apenas após o carregamento completo da estrutura HTML, evitando erros de elementos não encontrados.

### Seleção de Elementos

```
2 const inputTarefa = document.getElementById("novaTarefa");
3 const botaoAdicionar = document.getElementById("adicionarTarefa");
4 const listaTarefas = document.getElementById("listaTarefas");
5 const listaMetas = document.getElementById("listaMetas");
6 const botaoTema = document.getElementById("toggleTheme");
```

Elementos Controlados:

- inputTarefa: Campo de texto para inserir novas tarefas
- botaoAdicionar: Botão que aciona a adição de tarefas
- listaTarefas: UL que contém as tarefas pendentes
- listaMetas: UL que mostra tarefas concluídas
- botaoTema: Botão de alternância entre temas claro/escuro

### Variável de Estado

```
8 let jaMostrouParabens = false; // 🐞 controla se o alerta já foi mostrado
```

Função: Controla se a mensagem de parabéns já foi exibida ao usuário, evitando repetições indesejadas.

## Adição de Tarefas

### Event Listener do Botão

```
10  botaoAdicionar.addEventListener("click", () => {
11      const texto = inputTarefa.value.trim();
12  12  if (texto !== "") {
13      adicionarTarefa(texto);
14      inputTarefa.value = "";
15      jaMostrouParabens = false; // 🐞 reseta o alerta ao adicionar nova tarefa
16  }
17  });
18
```

### Fluxo:

- Captura o texto do input (removendo espaços extras com trim())
- Verifica se não está vazio
- Chama adicionarTarefa() com o texto
- Limpa o campo de input
- Reseta o controle da mensagem de parabéns

### Função adicionarTarefa()

```
19  function adicionarTarefa(texto) {
20      const li = document.createElement("li");
21
22      li.style.marginBottom = "8px";
23
24      const checkbox = document.createElement("input");
25      checkbox.type = "checkbox";
26      checkbox.style.marginRight = "10px";
27
28      const span = document.createElement("span");
29      span.textContent = texto;
30
31      checkbox.addEventListener("change", () => {
32          if (checkbox.checked) {
33              span.style.textDecoration = "line-through";
34              listaMetas.appendChild(li);
35          } else {
36              span.style.textDecoration = "none";
37              listaTarefas.appendChild(li);
38          }
39      });
40  }
```

### Processo de Criação:

- Cria um novo elemento <li> para a tarefa
- Adiciona um checkbox e um span com o texto
- Configura o estilo básico dos elementos
- Adiciona evento de change no checkbox

### Lógica do Checkbox:

- Quando marcado:
- Aplica estilo de texto tachado
- Move o item para "Metas Concluídas"
- Quando desmarcado:
- Remove o estilo tachado
- Retorna o item para "Tarefas"
- Sempre verifica se todas tarefas foram concluídas

### Verificação de Conclusão

```
47 function verificarConclusao() {
48     const checkboxes = listaTarefas.querySelectorAll('input[type="checkbox"]');
49     const todosMarcados = Array.from(checkboxes).every(cb => cb.checked);
50
51     // Só mostra o alerta se ainda não foi mostrado
52     if (checkboxes.length > 0 && todosMarcados && !jaMostrouParabens) {
53         alert("🎉 Parabéns! Você concluiu todas as suas tarefas!");
54         jaMostrouParabens = true;
55     }
56 }
```

### Funcionamento:

- Seleciona todos os checkboxes de tarefas pendentes
- Verifica se todos estão marcados usando Array.every()

### Condições para exibir alerta:

- Há pelo menos uma tarefa (checkboxes.length > 0)
- Todas estão marcadas (todosMarcados)
- O alerta ainda não foi mostrado (!jaMostrouParabens)
- Exibe alerta e atualiza o controle

## Sistema de Temas

```
58     botaoTema.addEventListener("click", () => {  
59         document.body.classList.toggle("dark-theme");  
60     });  
61 });  
62
```

### Mecanismo:

- Alterna a classe dark-theme no elemento body
- O CSS detecta esta classe e aplica os estilos correspondentes
- As imagens do footer são alternadas automaticamente via CSS