



Projeto Clima

Projeto Base

<https://openweathermap.org/>

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Clima - FUNCIONANDO</title>
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <h1>Clima</h1>

  <form class="busca">
    <input type="search" id="searchInput" />
    <button>Buscar</button>
  </form>

  <div class="resultado">
    <div class="titulo">--</div>

    <div class="info">
      <div class="temp">
        <div class="tempTitulo">Temperatura</div>
        <div class="tempInfo">-- <sup>°C</sup></div>
      </div>
    </div>
  </div>
</body>
</html>
```

```

        
    </div>
    <div class="vento">
        <div class="ventoTitulo">Vento</div>
        <div class="ventoInfo">-- <span>km/h</span></div>
        <div class="ventoArea">
            <div class="ventoPonto" style="transform: rotate(0deg)"></div>
        </div>
    </div>
</div>

<div class="aviso"></div>

<footer>Criado por Professor Cleber</footer>

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

```

Explicando o código HTML

Vamos analisar este código HTML passo a passo:

1. Estrutura Básica

- < !DOCTYPE html > - Declara que este é um documento HTML5
- < html lang="pt" > - Define o início do documento HTML com português como idioma
- < head > - Contém metadados e links para recursos externos

2. Seção Head

- < meta charset="UTF-8"> - Define a codificação de caracteres como UTF-8
- < meta http-equiv="X-UA-Compatible" > - Garante compatibilidade com o Internet Explorer
- < meta name="viewport"> - Configura a visualização responsiva para dispositivos móveis

- `< link rel="stylesheet">;` - Conecta o arquivo CSS externo

3. Corpo da Página

`< body>` contém os seguintes elementos principais:

3.1. Formulário de Busca

`< form class="busca" >` - Formulário para pesquisa do clima

- Campo de busca com `<input type="search">`
- Botão para submeter a pesquisa

3.2. Área de Resultado

`< div class="resultado" >` contém:

- Título do resultado (inicialmente vazio com "--")
- Seção de temperatura com:
- Título "Temperatura"
- Valor da temperatura em °C
- Ícone do clima

3.3. Informações do Vento

A seção `< div class="vento" >` inclui:

- Título "Vento"
- Velocidade do vento em km/h
- Área com indicador de direção do vento (ventoPonto)

3.4. Elementos Finais

- `< div class="aviso" >` - Área para mensagens de aviso
- `< footer >` - Rodapé com créditos
- `< script src="script.js" >` - Link para o arquivo JavaScript que controla a funcionalidade

Este código cria uma interface para exibir informações meteorológicas, com áreas específicas para mostrar temperatura, velocidade e direção do vento, além de um ícone representativo do clima atual.

Código CSS

```
* {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
}  
  
body {  
  display: flex;  
  min-height: 100vh;  
  justify-content: center;  
  align-items: center;  
  flex-direction: column;  
  background-color: #333;  
  font-family: Arial, Helvetica, sans-serif;  
}  
  
h1 {  
  color: #FFF;  
  margin: 20px 0;  
}  
  
.busca {  
  display: flex;  
  width: 300px;  
}  
  
.busca #searchInput {  
  flex: 1;  
  padding: 10px;  
  outline: none;  
  font-size: 15px;  
  border-radius: 5px;  
  border: 0;  
}
```

```
.busca button {
  padding: 10px;
  font-size: 15px;
  border: 0;
  background-color: #555;
  color: #FFF;
  border-radius: 5px;
  margin-left: 10px;
  cursor: pointer;
}

.busca button:hover {
  background-color: #444;
}

.resultado {
  width: 300px;
  display: none;
  border: 1px dotted #FFF;
  margin-top: 20px;
  border-radius: 5px;
  background-color: rgba(0, 0, 0, 0.2);
}

.titulo {
  width: 100%;
  color: #FFF;
  font-size: 25px;
  font-weight: bold;
  text-align: center;
  padding: 20px;
}

.info {
  display: flex;
  width: 100%;
}

.info .temp {
  flex: 1;
```

```
    text-align: center;
}
.info .vento {
    flex: 1;
    text-align: center;
}
.tempTitulo,
.ventoTitulo {
    color: #777;
    font-size: 16px;
    margin-bottom: 5px;
}
.tempInfo,
.ventoInfo {
    color: #FFF;
    font-size: 30px;
    font-weight: bold;
}
.tempInfo sup {
    font-size: 15px;
    font-weight: normal;
}
.ventoInfo span {
    font-size: 15px;
    font-weight: normal;
}
.ventoArea {
    width: 50px;
    height: 50px;
    border: 1px solid #FFF;
    border-radius: 50%;
    margin: auto;
    margin-top: 20px;
    padding-left: 25px;
    padding-top: 25px;
}
.ventoPonto {
    width: 15px;
```

```
height: 1px;  
background-color: #FFF;  
transform-origin: left;  
}  
  
.aviso {  
color: #FFF;  
padding-top: 20px;  
}  
  
footer {  
margin-top: 20px;  
color: #FFF;  
font-size: 13px;  
}
```

Explicando o código CSS

1. Reset e Configurações Globais

O seletor * aplica as seguintes propriedades a todos os elementos:

- margin: 0 - Remove todas as margens externas
- padding: 0 - Remove todos os preenchimentos internos
- box-sizing: border-box - Inclui padding e border no cálculo total da largura/altura

2. Estilo do Body

O body é estilizado como um container flex com as seguintes propriedades:

- display: flex - Ativa o layout flexbox
- min-height: 100vh - Altura mínima de 100% da viewport
- justify-content: center - Centraliza horizontalmente
- align-items: center - Centraliza verticalmente
- flex-direction: column - Organiza elementos em coluna

- background-color: #333 - Fundo cinza escuro

3. Área de Busca

A classe .busca define o formulário de pesquisa:

- display: flex - Layout flexbox para alinhar input e botão
- width: 300px - Largura fixa do formulário

O campo de busca (#searchInput) tem:

- flex: 1 - Ocupa espaço disponível
- padding: 10px - Espaçamento interno
- border-radius: 5px - Cantos arredondados

4. Área de Resultado

A classe .resultado define o container de resultados:

- display: none - Inicialmente oculto
- border: 1px dotted #FFF - Borda pontilhada branca
- background-color: rgba(0, 0, 0, 0.2) - Fundo semi-transparente

5. Informações de Temperatura e Vento

As classes .info, .temp e .vento organizam as informações meteorológicas:

- display: flex - Layout flexbox para organizar dados
- text-align: center - Centraliza textos
- Diferentes tamanhos de fonte e cores para hierarquia visual

6. Indicador de Vento

As classes .ventoArea e .ventoPonto criam o indicador de direção do vento:

- border-radius: 50% - Forma circular
- transform-origin: left - Define ponto de origem para rotação

7. Elementos de Feedback

Classes .aviso e footer para elementos informativos:

- Cores claras (#FFF) para contraste com o fundo escuro

- Espaçamentos adequados para legibilidade

Código JavaScript

```
document.querySelector('.busca').addEventListener('submit', async (event)
⇒ {
    event.preventDefault();

    let input = document.querySelector('#searchInput').value;

    if(input !== '') {
        clearInfo();
        showWarning('Carregando...');

        let results = await fetch(`https://api.openweathermap.org/data/2.5/we
ather?q=
        ${encodeURIComponent(input)}&units=metric&lang=pt_br&appid=d06cdb298fafc
83c520d5ab677fc477e`);
        let json = await results.json();

        if(json.cod === 200) {
            showInfo({
                name: json.name,
                country: json.sys.country,
                temp: json.main.temp,
                templcon: json.weather[0].icon,
                windSpeed: json.wind.speed,
                windAngle: json.wind.deg
            });
        } else {
            clearInfo();
            showWarning('Não encontramos esta localização.');
        }
    } else {
        clearInfo();
    }
});
```

```

function showInfo(obj) {
    showWarning('');

    document.querySelector('.titulo').innerHTML = `${obj.name}, ${obj.country}`;

    document.querySelector('.tempInfo').innerHTML = `${obj.temp} <sup>°C</sup>`;
    document.querySelector('.ventoInfo').innerHTML = `${obj.windSpeed} <span>km/h</span>`;

    document.querySelector('.temp img').setAttribute('src',
        `http://openweathermap.org/img/wn/${obj.tempIcon}@2x.png`);

    document.querySelector('.ventoPonto').style.transform = `rotate(${obj.windAngle-90}deg)`;

    document.querySelector('.resultado').style.display = 'block';
}

function clearInfo() {
    showWarning('');
    document.querySelector('.resultado').style.display = 'none';
}

function showWarning(msg) {
    document.querySelector('.aviso').innerHTML = msg;
}

```

Explicação Detalhada do Código JavaScript

1. Event Listener do Formulário

O código começa adicionando um listener ao formulário de busca:

- `document.querySelector('.busca').addEventListener('submit', ...)` - Escuta o evento de envio do formulário

- `event.preventDefault()` - Impede o comportamento padrão de recarregar a página
- `async (event) ⇒ {...}` - Função assíncrona que permite usar `await` para requisições

O `await` é uma palavra-chave em JavaScript que só pode ser usada dentro de funções assíncronas (`async`). Ela tem algumas características importantes:

- Pausa a execução da função assíncrona até que uma Promise seja resolvida
- Retorna o valor resolvido da Promise
- Torna o código assíncrono mais legível, evitando callbacks aninhados

Por exemplo, quando fazemos:

```
let results = await fetch(url);
let json = await results.json();
```

O código espera a requisição `fetch` terminar antes de continuar para a próxima linha. Sem `await`, precisaríamos usar `.then()` encadeados:

```
fetch(url)
  .then(results ⇒ results.json())
  .then(json ⇒ {
    // usar os dados
  });
```

Benefícios do `await`:

- Código mais limpo e fácil de entender
- Melhor tratamento de erros usando `try/catch`
- Evita o "callback hell" (callbacks aninhados)
- Facilita o debug do código assíncrono

Lembre-se que `await` só funciona dentro de funções marcadas com `async`, caso contrário, resultará em erro de sintaxe.

2. Captura e Validação do Input

O código obtém e valida o valor digitado pelo usuário:

- Captura o valor do input com `document.querySelector('#searchInput').value`
- Verifica se o input não está vazio (`if(input !== '')`)
- Se houver valor, limpa informações anteriores (`clearInfo()`) e mostra "Carregando..."

3. Requisição à API

Faz uma chamada à API OpenWeatherMap:

- Usa `fetch` para fazer requisição HTTP
- `encodeURIComponent(input)` - Codifica a cidade digitada para URL
- Parâmetros incluem: unidades métricas, idioma português e chave da API
- `await results.json()` - Converte a resposta para objeto JavaScript

4. Processamento da Resposta

Verifica e processa a resposta da API:

- `if(json.cod === 200)` - Verifica se a requisição foi bem-sucedida
- Em caso de sucesso, chama `showInfo()` com os dados do clima
- Em caso de erro, limpa as informações e mostra mensagem de erro

5. Função `showInfo()`

Responsável por exibir as informações do clima na interface:

- Atualiza o título com nome da cidade e país
- Mostra temperatura em Celsius
- Exibe velocidade do vento
- Carrega ícone do clima
- Ajusta a direção do indicador de vento
- Torna visível a div de resultado

6. Funções Auxiliares

Duas funções de suporte:

- `clearInfo()` - Limpa as informações da tela e oculta o resultado

- • `showWarning()` - Exibe mensagens de aviso para o usuário

Este código cria uma aplicação web completa que consulta uma API de clima e exibe os resultados de forma organizada e interativa para o usuário.