



# Teste de front-end projeto aeroporto

Equipe: Camylle, Eduardo, Hadson, Urick

- . Objetivos
  - . Fases do projeto
  - . Conclusão do projeto
- 

**MOCKUP**

Buscamos ícones de aviões até encontrar o que está presente na mockup. Em seguida, criamos a mockup com base no ícone escolhido, suas cores e demais elementos. Posteriormente, desenvolvemos o HTML correspondente.

Utilizamos o Bootstrap para implementar a navbar. Inserimos o nome selecionado e criamos uma caixa de opções com os estados escolhidos.

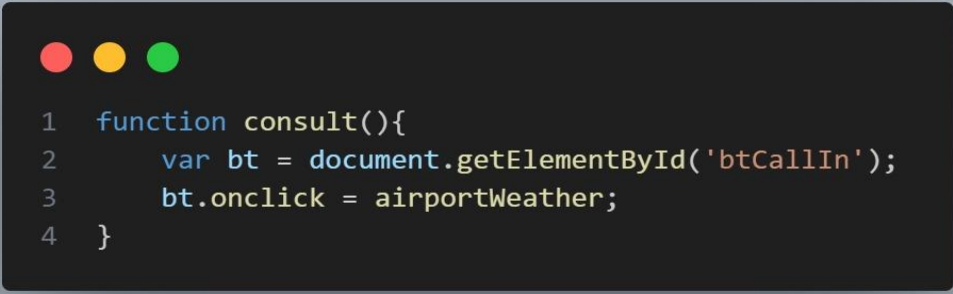
Adicionamos um botão de consulta e uma div vazia destinada aos resultados dos dados buscados no XML.

Por fim, configuramos a fonte, adicionamos um botão para limpar as informações e incluímos um footer simples.

Feito por Eduardo e Hadson.


## SCRIPTS

Inicialmente, foi criada uma função que, ao clicar no botão "Consultar" gerado no HTML, direcionava para a função principal do site, denominada função “consult”.



```
1  function consult(){
2      var bt = document.getElementById('btCallIn');
3      bt.onclick = airportWeather;
4  }
```

A função direcionada é "airportWeather", que retorna o valor dos estados selecionados na caixa de opções e altera o nome do estado com base nessa seleção.

A screenshot of a code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The code is written in JavaScript and is as follows:

```
1 function airportWeather(){
2     var state = document.getElementById('state').value;
3     var result = document.getElementById('result');
4     var url = 'http://servicos.cptec.inpe.br/XML/estacao/' + state + '/condicoesAtuais.xml';
5     console.log(url);
```

Foi adicionada a URL do XML fornecido para a realização da atividade, permitindo a alteração dos dados do XML e sua exibição na div de resultados.

```

1  xmlhttp.onload = function() {
2      if (xmlhttp.status === 200) {
3          var xmlDoc = xmlhttp.responseXML;
4          if (xmlDoc) {
5              var metar = xmlDoc.getElementsByTagName("metar")[0];
6              if (metar) {
7                  var codigo = metar.getElementsByTagName("codigo")[0].textContent;
8                  var temperatura = metar.getElementsByTagName("temperatura")[0].textContent;
9                  var atualizacao = metar.getElementsByTagName("atualizacao")[0].textContent;
10                 var descricao = metar.getElementsByTagName("tempo_desc")[0].textContent;
11                 var umidade = metar.getElementsByTagName("umidade")[0].textContent;
12                 var pressao = metar.getElementsByTagName("pressao")[0].textContent;
13                 var sig = metar.getElementsByTagName("tempo")[0].textContent;
14
15
16                 result.innerHTML = `<div style='text-align: center'>
17                     Código: ${codigo}<br>
18                     Aeroporto: ${nameState}<br>
19                     Atualização: ${atualizacao}<br>
20                     Temperatura: ${temperatura}°C<br>
21                     Sigla: ${sig}<br>
22                     Descrição: ${descricao}<br>
23                     Pressão: ${pressao} mb <br>
24                     Umidade: ${umidade}%<br>
25                     </h5></div>
26                 `;

```

Além disso, foi implementada uma função de limpeza, acionada pelo botão criado no HTML, que, ao ser clicado, redefine o texto da div de resultados para uma caixa vazia.

Feito por Hadson.

## TESTES

Foi realizado um teste para verificar se a URL estava sendo chamada corretamente e se os erros estavam sendo exibidos adequadamente.

No CSS, o body foi estilizado com um fundo azul (#191970), letras brancas, e alinhamento centralizado para todos os itens.

A navbar foi configurada com um fundo azul, letras pretas, conteúdos centralizados, margem de 5 e borda dupla. O texto da navbar recebeu a cor branca, fonte Verdana-Geneva e foi definido sem decoração de texto.

Para o título, foi utilizada a mesma fonte Verdana, fundo azul, alinhamento centralizado, tamanho de fonte grande e cor branca.

O botão de seleção foi estilizado com uma borda azul que aparece ao passar o mouse, utilizando um estilo de borda dupla.

O footer foi posicionado de forma fixa, com letras pretas, fundo azul, fonte Verdana-Geneva em estilo negrito oblíquo, e a mesma borda utilizada nos demais elementos. O fundo do footer foi definido como azul e margem zero. O botão no footer foi estilizado com a fonte Times New Roman.

No HTML, foi ajustado o alinhamento dos textos de resultado para o centro, garantindo que os dados apareçam centralizados ao realizar uma pesquisa.

Feito por Camylle e Eduardo

## **CONSIDERAÇÕES E CONCLUSÕES FINAIS DO PROJETO**

Foram feitas alterações no idioma do projeto. Todas as funções, IDs e nomes foram traduzidos para o inglês. Esta mudança visa padronizar a linguagem utilizada e facilitar a compreensão e colaboração internacional.

Documentação foi feita por Urick.