



# Teste Front-End: Dados Climáticos nos Aeroportos

---

**Equipe:** Flávio, Igor, Raphael e Rogério.

---

## Conteúdo do Documento:

- Requisitos.
- Modelagem e Planejamento.
- Desenvolvimento.
- Testes.
- Documentação.
- Conclusão

# Levantamento de Requisitos

Para a criação do Sistema de consulta climática, fizemos um levantamento inicial detalhado dos requisitos e especificações necessárias para a realização do projeto. Nesta etapa, registramos:

- **Objetivo e Escopo do Sistema:**

**Objetivo:** O objetivo do sistema é fornecer informações climáticas em tempo real para diferentes estações meteorológicas utilizando uma API de serviços climáticos do CPTEC/INPE.

**Escopo:** O sistema permitirá ao usuário selecionar uma estação meteorológica a partir de uma lista predefinida e, em seguida, exibir as condições climáticas atuais, incluindo código da estação, nome da estação, temperatura, umidade e pressão atmosférica. O escopo exclui previsões meteorológicas futuras e dados históricos.

- **Requisitos Funcionais:**

1. O sistema deve permitir ao usuário selecionar uma estação meteorológica a partir de um menu dropdown.
2. O sistema deve fazer uma requisição à API do CPTEC/INPE para obter dados climáticos da estação selecionada.
3. O sistema deve exibir o código da estação, temperatura, umidade e pressão atmosférica em uma área de resultados.
4. O usuário deve ter a possibilidade de trocar a fonte do site de acordo com os que foram fornecidos.

- **Requisitos Não Funcionais:**

1. O sistema deve responder às requisições de dados climáticos em no máximo 2 segundos.
2. O sistema deve ser compatível com os navegadores mais populares (Chrome, Firefox, Edge, Safari).
3. O sistema deve exibir mensagens de erro claras em caso de falha na requisição ou dados inválidos.

# Modelagem e Planejamento

Com os requisitos levantados, a próxima etapa é a modelagem e o planejamento do sistema. Nesta etapa fizemos um mockup teste para verificar se o script em JS iria funcionar.

Utilizamos o Bootstrap para estilizar o HTML e ficar responsivo. Um arquivo JS para realizar as funções de pesquisa no XML , trocar a fonte do site e o CSS

Criamos uma área de seleção para gerenciar as opções com os estados escolhidos que ativa a função de consulta e uma div com o id 'resultado' destinada aos resultados dos dados buscados no XML fornecido pela API.

Finalizando, adicionamos a fonte, um botão para limpar a div com os dados e um footer simples.

## Dados Climáticos - Aeroportos

Selecione a Estação Meteorologica do aeroporto desejado

RJ ▾

Enviar

Digite o código do aeroporto

Enviar

Fontes de Dados: API "CPTEC/INPE", disponível em:<http://servicos.cptec.inpe.br/XML/>

Direitos Reservados - 2024

# Desenvolvimento

O site foi desenvolvido de acordo com o pedido do cliente. Inicialmente criamos um HTML teste para testar as requisições XML da API, feito isso criamos o arquivo principal para o nosso site, e nele inclui-se uma seção que mostra os dados climáticos e uma imagem de acordo com o clima da estação (Ex: Imagem de chuva, se o clima da estação estiver chuvoso), isso ocorre com todos os tipos de clima.

Na estilização do site fizemos um mockup no canva para criar algo mais bem definido de acordo com nossas ideias, após isso usamos o bootstrap para estilizar totalmente o site e criar uma identidade visual para “SkyWeather”.

Depois de ter um site padrão, adicionamos uma função para trocar de CSS de acordo com o clima da estação e logo após uma função para trocar a fonte do site

## Testes

Este é o nosso html de teste para verificar se está tudo certo

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>Jasmine</title>
6   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/jasmine/3.9.0/jasmine.min.css">
7   <script src="https://cdnjs.cloudflare.com/ajax/libs/jasmine/3.9.0/jasmine.min.js"></script>
8   <script src="https://cdnjs.cloudflare.com/ajax/libs/jasmine/3.9.0/jasmine-html.min.js"></script>
9   <script src="https://cdnjs.cloudflare.com/ajax/libs/jasmine/3.9.0/boot.min.js"></script>
10  <script src="./js/scripts.js"></script>
11  <script src="./js/spec.js"></script>
12 </head>
13 <body>
14 </body>
15 </html>
16
```

Aqui está nossos testes da estação, para verificar se o nome da estação está de acordo com o código

```
1 describe("Testes para a função getEstacaoNome", function() {
2   it("deve retornar 'Santa Maria - SE' quando o código for 'SBAR'", function() {
3     expect(getEstacaoNome('SBAR')).toBe('Santa Maria - SE');
4   });
5
6   it("deve retornar 'Guarulhos - SP' quando o código for 'SBGR'", function() {
7     expect(getEstacaoNome('SBGR')).toBe('Guarulhos - SP');
8   });
9
10  it("deve retornar 'Uberlândia - MG' quando o código for 'SBUL'", function() {
11    expect(getEstacaoNome('SBUL')).toBe('Uberlândia - MG');
12  });
13
14  it("deve retornar 'Pampulha - MG' quando o código for 'SBBH'", function() {
15    expect(getEstacaoNome('SBBH')).toBe('Pampulha - MG');
16  });
17
18  it("deve retornar 'Cabo Frio - RJ' quando o código for 'SBCB'", function() {
19    expect(getEstacaoNome('SBCB')).toBe('Cabo Frio - RJ');
20  });
21
22  it("deve retornar 'Estação Desconhecida' para um código desconhecido", function() {
23    expect(getEstacaoNome('XXXX')).toBe('Estação Desconhecida');
24  });
25 });
```

# Documentação

Função responsável por criar uma url, buscar as informações usando API e apresentá-las no HTML. Ele obtém o select e o valor selecionado e depois determina o nome da estação com base nesses valores.

```
1 function consultaClima() {
2     var clima = document.getElementById('clima').value;
3     var nomeEstacao = '';
4     switch (clima) {
5         case 'SBAR':
6             nomeEstacao = 'Santa Maria - SE';
7             break;
8         case 'SBGR':
9             nomeEstacao = 'Guarulhos - SP';
10            break;
11        case 'SBUL':
12            nomeEstacao = 'Uberlândia - MG';
13            break;
14        case 'SBBH':
15            nomeEstacao = 'Pampulha - MG';
16            break;
17        case 'SBCB':
18            nomeEstacao = 'Cabo Frio - RJ';
19            break;
20        default:
21            nomeEstacao = 'Estação Desconhecida';
22            break;
23    }
```

Após isso ele exibe a div onde aparecerá os resultados de acordo a url da estação selecionada.

```
1 var div = document.getElementById('resultado');
2 var url = 'http://servicos.cptec.inpe.br/XML/estacao/' + clima + '/condicoesAtuais.xml';
3 console.log(url);
4
5 var xmlhttp;
6 if (window.XMLHttpRequest) {
7     xmlhttp = new XMLHttpRequest();
8     xmlhttp.overrideMimeType("application/xml;charset=utf-8");
9 } else {
10     xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
11 }
12
13 xmlhttp.onerror = function(){
14     div.innerHTML = 'VALOR INVÁLIDO!';
15 }
```

Função responsável por processar a resposta da requisição e verificar se o XML foi recebido corretamente verificando também se o elemento <metar> está presente no XML

```
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    }
17 }
```

Exibe os dados na div “resultado” ou exibe erros se houver algum


```
1      div.innerHTML = `<div class="d-flex justify-content-center">
2          Código: ${codigo}<br>
3          Aeroporto: ${nomeEstacao}<br>
4          Atualização: ${atualizacao}<br>
5          Temperatura: ${temperatura}°C<br>
6          Sigla: ${sig}<br>
7          Descrição: ${descricao}<br>
8          Pressão: ${pressao} mb <br>
9          Umidade: ${umidade}%<br>
10         </h5></div>
11     `;
12     } else {
13         div.innerHTML = 'Dados não disponíveis para a estação.';
14     }
15     } else {
16         div.innerHTML = 'Erro ao processar a resposta XML.';
17     }
18     } else {
19         div.innerHTML = 'Erro ao realizar a requisição: ' + xmlhttp.status;
20     }
21     };
```

Função responsável por trocar o css do site.

```
1  function styleCSS(css){
2      document.getElementById("css").setAttribute("href",css);
3  }
```



Função responsável por limpar a seleção.



```
1 function limpar(){
2     var el = document.getElementById("resultado");
3     el.innerHTML = "";
4     styleCSS("./css/estilo.css")
5 }
```

Função responsável por limpar a seleção.



```
1 function consultaFonte(){
2     // Obtém o select e o valor selecionado
3     var fonte = document.getElementById('fonte').value;
4     // Obtém o select e o valor selecionado
5     switch (fonte) {
6         case 'padrao':
7             document.body.style.fontFamily = 'Segoe UI';
8             break;
9         case 'Helvetica':
10            document.body.style.fontFamily = 'Helvetica';
11            break;
12        case 'Verdana':
13            document.body.style.fontFamily = 'Verdana';
14            break;
15        case 'CourierNew':
16            document.body.style.fontFamily = 'Courier New';
17            break;
18        default:
19            break;
20    }
21 }
22
```

# Conclusão Final

**Projeto:** criar um sistema de informações climáticas em tempo real para diferentes estações meteorológicas utilizando uma API de serviços climáticos do CPTEC/INPE.

Data de conclusão: 06/06/2024

**Conclusão:** o projeto de desenvolvimento do site climático foi concluído com sucesso, atendendo a todos os objetivos estabelecidos inicialmente. O site está agora em operação, proporcionando uma API eficaz para a empresa de serviços climáticos do CPTEC/INPE, visando informar seus usuários e promover seus serviços climáticos.

**Tarefas realizadas pelos membros da equipe:**

- Flávio: Planejamento, Mockup (Inicial), Documentação, Mockup (Final), CSS
- Igor: Planejamento Mockup (Inicial), Documentação, Mockup (Final), CSS
- Raphael: Planejamento Testes, Script JS, Documentação, Mockup(Final)
- Rogério: Planejamento, Script JS, Mockup (Final), CSS de cada clima