

# **ClimaTracker - Instruções para Execução da Atividade Prática**

## **API de Clima:**

- Utilize a API **WeatherAPI** para obter dados meteorológicos de uma cidade digitada pelo usuário.
- A API devolve informações como temperatura, descrição do clima, umidade e vento, e a posição geográfica da cidade consultada.

## **2. Mapas:**

- Utilize a biblioteca **Leaflet** para exibir um mapa interativo da cidade solicitada.
- A cidade inicial será Campo Mourão (código de latitude e longitude: -24.0469, -52.3780), que será exibida por padrão no mapa.

## **3. Funcionalidades:**

- O usuário poderá digitar o nome de uma cidade no campo de texto e, ao clicar no botão de "Buscar", o clima atual dessa cidade será exibido.
- O mapa será atualizado automaticamente para a localização dessa cidade.
- As informações do clima de Campo Mourão serão exibidas em uma área separada no topo do mapa.

# **Procedimento de Execução**

## **1. Preparando o Ambiente**

- Crie um diretório no seu computador para o projeto ClimaTracker.
- Crie três arquivos:
  - `index.html`: Arquivo HTML contendo a estrutura do site.
  - `styles.css`: Arquivo CSS para a estilização da página.
  - `script.js`: Arquivo JavaScript para implementar a lógica do sistema.

## **2. Estrutura HTML**

- No arquivo `index.html`, inclua a estrutura básica do HTML:

- Cabeçalho com o título "ClimaTracker", links para o arquivo CSS e as bibliotecas externas (Leaflet).
- Corpo com um contêiner principal que inclui:
  - Um campo de entrada de texto para digitar a cidade.
  - Um botão para acionar a pesquisa.
  - Áreas para exibição de informações do clima, como temperatura, descrição, umidade e vento.
  - Uma área de erro para exibir mensagens quando a cidade não for encontrada.
  - Um mapa interativo, configurado para exibir o clima da cidade escolhida.

### **3. Estilização CSS**

- No arquivo `styles.css`, defina a aparência do site, com destaque para:
  - Um design clean e moderno com cores suaves.
  - Uso de bordas arredondadas e sombras para dar profundidade ao layout.
  - Animações e transições para suavizar a experiência do usuário.

### **4. Implementação JavaScript**

- No arquivo `script.js`, implemente as seguintes funcionalidades:
  - `initializeMap`: Inicializa o mapa com a localização de Campo Mourão.
  - `getWeather`: Obtém os dados do clima de uma cidade inserida pelo usuário.
  - `getLocalWeather`: Exibe as informações de clima de Campo Mourão.
  - Função de busca: Ao clicar no botão "Buscar", o clima da cidade será exibido e o mapa será atualizado para essa cidade.

### **5. Integração com a API**

- Registre-se em WeatherAPI e obtenha uma chave de API.

- *Use a chave para realizar requisições para a API e obter os dados do clima para a cidade inserida.*

## **6. Testando o Projeto**

- *Após concluir o código, abra o arquivo `index.html` em um navegador.*
- *Digite o nome de uma cidade no campo de entrada e clique no botão "Buscar". As informações climáticas da cidade selecionada serão exibidas.*
- *O mapa será atualizado automaticamente para a nova cidade.*
- *As informações de Campo Mourão também estarão visíveis no topo do mapa.*

## **7. Resolução de Problemas**

- *Erro na cidade: Se o nome da cidade não for encontrado, uma mensagem de erro será exibida.*
- *Erro na requisição da API: Caso ocorra algum erro durante a consulta à API, uma mensagem de erro será exibida e as informações climáticas não serão carregadas.*

## **Observações**

- *Certifique-se de que a chave de API foi configurada corretamente.*
- *O mapa está configurado para exibir a localização inicial de Campo Mourão. Caso deseje, é possível alterar a localização inicial para qualquer outra cidade.*
- *O código está projetado para ser simples e modular, facilitando a manutenção e a personalização.*

## **Ferramentas que vão ser utilizadas na apresentação:**

**Visual Studio Code:** será utilizado para podermos falar sobre a estrutura do código.

**Postman:** também será necessário ter uma conta no postman, para que possamos falar sobre como funciona uma api.