



# Challenge Web - Março

Tudo bem, pessoal? Espero que vocês tenham começado a trilhar suas carreiras em desenvolvimento web e estejam curtindo e evoluindo muito 🎉

Para aqueles que ainda não conseguiram completar ou não fizeram o primeiro desafio, recomendo que tentem fazê-lo antes já que hoje vamos usar alguns conceitos a mais e é importante reforçar as bases primeiro! O Challenge anterior está no mesmo canal onde você encontrou este Challenge.

Lembrem-se que a comunidade é essencial para todo dev evoluir, então não deixe de trocar ideias com seus colegas do Let's Code Pass para tirar dúvidas, bolar soluções conjuntas e rumar sempre ao próximo nível de evolução em suas carreiras!

## Conhecendo o Challenge

Desta vez, seu desafio será **criar uma aplicação de previsão do tempo!** Muito legal né? Para ter uma ideia geral de como este sistema deverá se comportar, disponibilizamos o link de uma aplicação funcional abaixo:

CodeCloud

Aplicação de previsão do tempo feita com ReactJS

<https://weather-react-plum.vercel.app/>

A aplicação acima foi desenvolvida, utilizando ReactJS. No entanto, você também pode desenvolvê-la utilizando apenas o JavaScript. O visual da aplicação deve ser desenvolvido utilizando CSS, SASS ou alguma biblioteca de estilização como Styled Components.

## Requisitos Funcionais

Fique atento aos requisitos funcionais da aplicação! Como sugestão iremos indicar a seguinte API grátis para pesquisar dados climáticos, mas você pode utilizar alguma outra se preferir, note apenas que as instruções abaixo serão para api documentada em <https://github.com/robertoduessmann/weather-api>

1. **Procurar por cidade:** Sua aplicação deve conter um formulário com um campo de input para que o usuário possa entrar com o nome de uma cidade a ser pesquisada.
2. **Resgatar dados meteorológicos da cidade:** Deve ser possível interagir com a API de clima fazendo uma requisição GET para o endereço:  
<https://goweather.herokuapp.com/weather/>

A documentação da API se encontra em e para utilizá-la basta passar a cidade como parâmetro da seguinte forma  
(<https://goweather.herokuapp.com/weather/{Cidade}>)

Exemplos:

```
https://goweather.herokuapp.com/weather/Campinas  
https://goweather.herokuapp.com/weather/Sao-Paulo
```

```
// Exemplo de uso da fetch api para resgatar dados climáticos de Campin  
const data = fetch("https://goweather.herokuapp.com/weather/Campinas")  
.then(  
  data => data.json()  
)  
.then(  
  data => console.log(data)  
)
```

3. **Exibir clima atual da cidade:** Após resgatar os dados de clima da cidade, devemos exibir o nome da cidade, a temperatura atual e a descrição do clima atual! Para isso, sabemos que a API de tempo escolhida retorna os dados no seguinte formato:

```
temperature: string  
wind: string  
description: CurrentWeather
```

```
forecast: {  
  day: number  
  temperature: string  
  wind: string  
}[]
```

Assim, precisaremos dos atributos temperature e description para concluir esta tarefa!

4. **Exibir previsão para próximos 3 dias da cidade:** Agora vamos utilizar os dados que vem dentro do array forecast para montar alguma estratégia de listagem dos dados e exibir as informações de temperatura e velocidade do vento para os próximos 3 dias! Idealmente exiba o próximo dia como “Amanhã” e os próximos dias pelo nome do dia da semana (ex: Domingo e Segunda-feira).

Não deixe também de usar o pacote react-icons para sua aplicação ficar mais apresentável, com ícones ao lado das informações de tempo e vento!



Embora esses sejam os únicos requisitos funcionais, você pode ficar à vontade para adicionar outras funcionalidades, caso queira. Por exemplo, mudar a cor de fundo da aplicação de acordo com o clima atual da cidade pesquisada. 😊

## Etapas para a Solução do Challenge

- ☐ Iniciar projeto React (duas formas, porém recomendo utilizar o vite)
  - ☐ Para tal devemos rodar um destes comandos abaixo em seu terminal. No Windows, seu terminal chama “CMD” ou “Powershell”.
    - ☐ Create React App (CRA):

```
np create-react-app nome_do_app
```
    - ☐ Vite :

```
npm init vite@latest
```
- ☐ Fazer form com input para pesquisa de cidade
- ☐ Requisição HTTP GET p/ interagir com a API de clima

- ☐ Armazenar estados no React do campo de pesquisa, cidade atual e dados do clima da cidade
- ☐ Utilizar renderização condicional para exibir os dados de clima da cidade, quando presentes
- ☐ Uso de componentes para organizar melhor aplicação
- ☐ Estilização

## Como evoluir a aplicação

- ☐ Possibilitar a usuário armazenar cidades preferidas
- ☐ Possibilitar a usuário escolher cidade padrão que será exibida ao carregar aplicação com useEffect (utilizar localStorage ou cookies)
- ☐ Mostre uma mensagem ou ícone de carregamento ao pesquisar por uma cidade
- ☐ Mudar a cor de fundo da aplicação de acordo com o clima atual da cidade pesquisada
- ☐ Utilizar alguma biblioteca de estilização CSS-IN-JS como styled components
- ☐ Utilizar alguma biblioteca para data-fetching como React Query para revalidar requisições para a API de clima ao voltar para a aba da aplicação
- ☐ Hospedar aplicação na Vercel, Netlify ou Github Pages