**O que é uma API?**

**API** significa **Interface de Programação de Aplicações** (*Application Programming Interface*). Em termos simples, é um **conjunto de regras** que permite que diferentes sistemas **se comuniquem entre si**.

Imagine a API como um **garçom** num restaurante:

* O cliente (front-end) faz o pedido.
* O garçom (API) leva o pedido à cozinha (back-end).
* A cozinha prepara e devolve o prato.
* O garçom entrega ao cliente.

Ou seja, a API **recebe requisições**, **processa dados** e **retorna respostas** — geralmente em **formato JSON**.

**Como a API se integra ao Front-End?**

O **front-end** (HTML, CSS, JavaScript, React, etc.) usa **requisições HTTP** para se comunicar com a API. A integração acontece assim:

1. **Usuário interage com a interface** (ex: clica em "Cadastrar").
2. O front-end faz uma **requisição HTTP** (GET, POST, PUT, DELETE) para a API.
3. A API processa a requisição e responde com os dados.
4. O front-end **exibe os dados** ou atualiza a interface conforme necessário.

1) Realizar uma requisição get utilizando a API Open Weather que criamos a aula passada Buscar clima de Lisboa, Portugal http://localhost:3000/weather?city=Lisboa&country=PT

1. Buscar clima de Rio de Janeiro, Brasil http://localhost:3000/weather?city=Rio de Janeiro&country=BR

|  |  |
| --- | --- |
| const axios = require('axios'); | Importa a biblioteca Axios para fazer requisições HTTP. |
| const baseURL = 'https://api.exemplo.com'; | Define a URL base da API. Pode ser alterada conforme o servidor. |
| axios.get(...) | Realiza uma requisição GET para a rota /multiples. |
| .then(...) | Trata a resposta da API e exibe os dados no console. |
| .catch(...) | Captura e exibe erros caso a requisição falhe. |