

# API

## O que é uma API? [↗](#)

API significa **Application Programming Interface** (Interface de Programação de Aplicações).

É um conjunto de regras e definições que permite que **diferentes sistemas se comuniquem entre si**, funcionando como um intermediário entre softwares.

### Exemplo: [↗](#)

Um aplicativo de clima usa uma **API** para buscar as informações de previsão do tempo de um serviço externo.

---

## Vantagens de usar APIs [↗](#)

- **Reutilização de funcionalidades:** não é necessário programar tudo do zero.
  - **Padronização:** facilita a comunicação entre diferentes sistemas.
  - **Integração entre tecnologias:** conecta sistemas escritos em diferentes linguagens.
  - **Segurança e controle de acesso:** é possível aplicar autenticação e autorização.
  - **Escalabilidade:** facilita a expansão e manutenção de sistemas.
- 

## Monólito vs. Microserviços [↗](#)

### Arquitetura Monolítica [↗](#)

- Toda a aplicação está em um único bloco de código.
- Mais simples no início, mas difícil de manter com o crescimento.
- Alterações em um módulo podem impactar todo o sistema.

### Arquitetura de Microserviços [↗](#)

- A aplicação é dividida em **vários serviços pequenos e independentes**.
  - Cada serviço tem sua própria lógica e, possivelmente, seu próprio banco de dados.
  - Facilita manutenção, escalabilidade e deploy contínuo.
  - Ideal para projetos maiores e equipes distribuídas.
- 

## O que é uma API REST? [↗](#)

**REST** (Representational State Transfer) é um estilo de arquitetura que usa o protocolo HTTP.

APIs REST são **leves, padronizadas** e amplamente utilizadas na web.

### Características principais: [↗](#)

- Utiliza **verbos HTTP** (GET, POST, PUT, etc.)
  - Usa **URLs** para identificar recursos
  - Retorna dados geralmente em **JSON**
  - É **stateless**: cada requisição é independente
-

## Verbos HTTP mais usados [🔗](#)

Verbo	Função
<b>GET</b>	Buscar dados
<b>POST</b>	Criar novos dados
<b>PUT</b>	Atualizar dados inteiramente
<b>PATCH</b>	Atualizar dados parcialmente
<b>DELETE</b>	Remover dados

---

## Principais Códigos de Status HTTP [🔗](#)

Código	Significado
<b>200 OK</b>	Requisição bem-sucedida
<b>201 Created</b>	Recurso criado com sucesso
<b>204 No Content</b>	Sucesso, mas sem conteúdo de retorno
<b>400 Bad Request</b>	Erro na requisição (cliente)
<b>401 Unauthorized</b>	Requisição sem autenticação válida
<b>403 Forbidden</b>	Acesso negado
<b>404 Not Found</b>	Recurso não encontrado
<b>500 Internal Server Error</b>	Erro interno no servidor

---