

**API**

Luiz Fernando Brogiato da Silva

Nº 22

# O que é uma api?

Uma **API** (Interface de Programação de Aplicações, do inglês *Application Programming Interface*) é um conjunto de definições e protocolos que permite que diferentes sistemas de software se comuniquem entre si. Em termos simples, uma API é uma interface que define como diferentes componentes de software devem interagir. Como analogia, uma API é como uma espécie de contrato entre duas partes: uma delas fornece um serviço ou dados, e a outra deseja utilizar esse serviço ou dados. A API define os métodos e as regras que devem ser seguidos para que essa comunicação ocorra de maneira eficaz. Essas APIs permitem que você envie requisições específicas e receba as respostas desejadas, como dados de um tweet ou a localização de um endereço.

**Existem vários tipos de api's, porém as principais são:**

- **APIs de REST (Representational State Transfer):** Amplamente utilizadas na web. Elas operam sobre o protocolo HTTP, utilizando métodos como GET, POST, PUT, DELETE para realizar operações de leitura, criação, atualização e exclusão de recursos. As APIs REST são conhecidas por serem simples e escaláveis.
- **APIs de SOAP (Simple Object Access Protocol):** Bem antigas e estruturadas que as REST. Elas utilizam XML para transportar dados e são frequentemente utilizadas em sistemas corporativos onde há uma necessidade maior de segurança e transações complexas.
- **APIs de GraphQL:** Uma alternativa às APIs REST, que permite aos clientes solicitarem exatamente os dados que precisam, nada mais, nada menos. Foi desenvolvido pelo Facebook e oferece uma abordagem mais flexível para o desenvolvimento de APIs.
- **APIs de WebSocket:** Muito usadas para comunicação em tempo real, onde a conexão entre o cliente e o servidor permanece aberta e permite o envio de dados de ambas as direções sem a necessidade de reabrir a conexão.

## Como utilizar api's em javascript?

Há várias maneiras de o fazer, porém as mais utilizadas são:

- **Fetch:** Interface moderna que permite fazer requisições HTTP de forma fácil e eficiente. Ela retorna o "Promise", facilitando o uso de operações assíncronas;
- **async/await:** São opções muito recomendadas para tornar o código mais legível, principalmente quando se utiliza as famosas operações assíncronas. Recomendável utiliza-las com o "Fetch";
- **XMLHttpRequest:** Antes do fetch, o XMLHttpRequest era muito utilizado para fazer requisições HTTP. Embora menos utilizado hoje em dia, ainda pode ser necessário em ambientes legados ( legados são sistemas, softwares, ou infraestruturas de TI antigos que ainda estão em uso em uma organização, mas que podem não seguir mais as práticas, tecnologias, ou padrões modernos ).