

# TRABALHO SENAI

## APIS WEB COM REST

Professor Rodrigo

x x x x x  
x x x x x  
x x x x x  
x x x x x



# Tópicos

01

**INTRODUÇÃO**

02

**O QUE É UMA API?**

03

**O QUE É UM REST?**

04

**VANTAGENS E  
DESVANTAGENS**

05

**COMO FUNCIONA?**

06

**EXEMPLO PRÁTICO**

# 01

## Introdução



# API REST

API REST é uma interface de programação de aplicações (API) que segue os princípios de design da arquitetura REST. REST é a sigla para "Representational State Transfer" (transferência de estado representacional), um conjunto de regras e diretrizes sobre como você deve criar uma API web.

As APIs REST se comunicam por meio de solicitações HTTP para executar funções padrão de banco de dados, como criar, ler, atualizar e excluir registros (também conhecidos como CRUD) em um recurso.

# 02

## O QUE É UMA API?

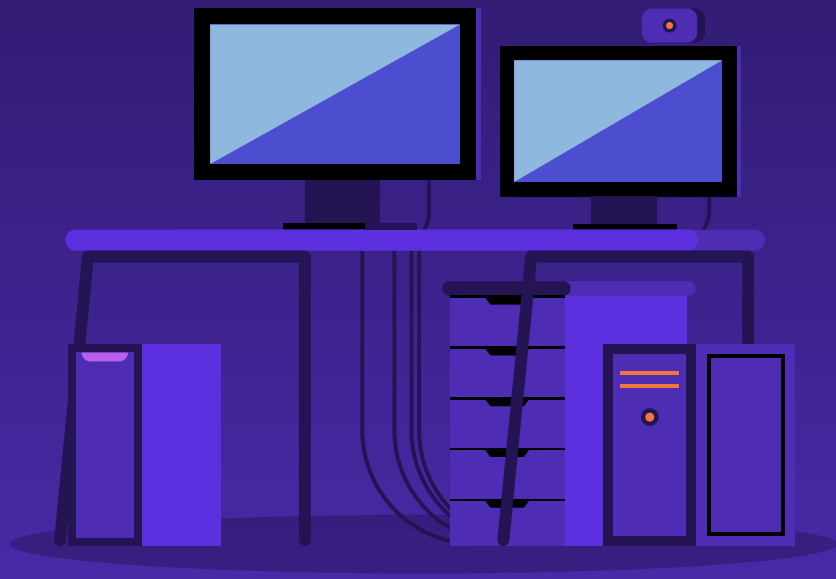
---



# O QUE É UMA API?

Uma API é um conjunto de protocolos e rotinas que permite que diversos sistemas e aplicativos se comuniquem sem os desenvolvedores ou usuários finais saberem como foram implementados. Avaliando as APIs, os desenvolvedores podem fácil e rapidamente reutilizar aplicativos de software e web.

Existem vários tipos de APIs que servem diversos casos de uso. APIs REST são amplamente usadas para desenvolver APIs baseadas em web e são uma parte importante do desenvolvimento de web moderno.



# 03

## O QUE É UMA REST?



# O QUE É UM REST?

REST significa "transferência de estado representacional" (Representational State Transfer). É um estilo arquitetural para desenvolver serviços de web que fornece uma forma padronizada para diversos sistemas de computadores se comunicarem pela internet. A REST é baseada em um conjunto de princípios de design em vez de um protocolo ou uma tecnologia específica e é amplamente usada para desenvolver APIs baseadas em web.

Um dos principais recursos da REST é o uso de métodos HTTP como GET, POST, PUT e DELETE para representar diferentes operações que podem ser realizadas com recursos. Esses recursos podem ser qualquer tipo de dados ou funcionalidade exposta pelo serviço de web e, geralmente, são representados usando um formato de dados padronizado, como JSON.





04

## VANTAGENS E DESVANTAGENS

# VANTAGENS DA API REST:

APIs REST incluem os seguintes benefícios:

**Dimensionável:** As APIs REST são projetadas para serem dimensionáveis, o que significa que elas são capazes de lidar com grandes números de pedidos e acomodar picos de tráfego. Elas usam uma arquitetura leve e simples que facilita a adição de novos recursos e funcionalidades sem a necessidade de rearquitetar todo o sistema.

**Flexível:** As APIs REST são flexíveis e podem ser usadas com uma ampla variedade de idiomas e tecnologias de programação. Elas não estão vinculadas a plataformas ou idiomas específicos, o que significa que podem ser usadas por clientes com dispositivos e sistemas operacionais diferentes.

**Sem estado:** O fato de as APIs REST serem sem estado significa que elas são mais confiáveis e fáceis de armazenar em cache, ajudando a reduzir a latência e melhorar o desempenho.

# VANTAGENS DA API REST:

**Intuitiva:** As APIs RESTful são intuitivas e podem ser acessadas usando métodos HTTP padrão, tornando-as acessíveis a desenvolvedores de todos os níveis de habilidade. Elas fornecem uma forma padronizada para diversos sistemas conseguirem se comunicar, ajudando a reduzir o tempo de desenvolvimento, melhorar a produtividade e minimizar o tempo para comercialização.

**Segurança:** As APIs RESTful podem ser obtidas usando uma variedade de técnicas, como criptografia SSL/TLS e autenticação OAuth. Isto ajuda a garantir que a API está protegida contra acesso não autorizado e que os dados dos clientes estão mantidos em segurança.

APIs REST têm uma taxa de adoção de 93,4%, o que a tornando a arquitetura de API mais usada entre os desenvolvedores.

Fonte: Nordic APIs

# DESVANTAGENS DA API REST:

Embora as REST APIs sejam preferidas por muitos desenvolvedores e para APIs públicas, há algumas limitações e desvantagens. Por exemplo, a criação de uma nova API REST é complexa e não há um padrão de como ela pode ser construída e implementada. Portanto, se um novo desenvolvedor assumir a tarefa sem documentação, pode ser desafiador compreender como a API funciona e se foi criada com princípios REST.

Em última análise, a única maneira de saber como funciona qualquer REST API é lendo a documentação ou o código-fonte se você puder acessá-lo. Mesmo REST APIs que parecem semelhantes podem ter métodos diferentes, então você terá que aprender sobre novos sempre que usá-los. Além disso, alguns clientes podem ter preocupações com o armazenamento de dados, uma vez que essas APIs não mantêm informações.

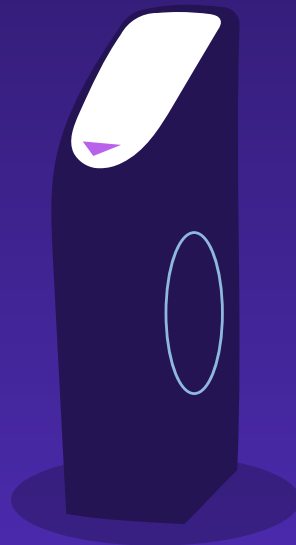


# 05

## COMO FUNCIONA?



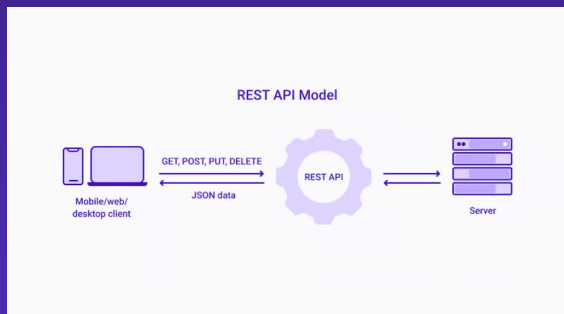
You can enter a subtitle here if you need it



# COMO FUNCIONA UMA API REST?

As APIs REST se comunicam por meio de solicitações HTTP para executar funções padrão de banco de dados, como criar, ler, atualizar e excluir registros (também conhecidos como CRUD) em um recurso.

Por exemplo, uma API REST usaria uma solicitação GET para recuperar um registro. Uma solicitação POST cria um novo registro. Uma solicitação PUT atualiza um registro, e uma solicitação DELETE exclui um. Todos os métodos HTTP podem ser usados em chamadas de API. Uma API REST bem projetada é semelhante a um site em execução em um navegador web com funcionalidade HTTP integrada.



# COMO FUNCIONA UMA API REST?

O estado de um recurso em um determinado instante, ou registro de data e hora, é conhecido como a representação do recurso. Essas informações podem ser entregues a um cliente em praticamente qualquer formato, incluindo JavaScript Object Notation (JSON), HTML, XML, Python, PHP ou texto simples. JSON é popular porque pode ser lido por humanos e máquinas – e é independente da linguagem de programação.

Cabeçalhos e parâmetros de solicitação também são importantes em chamadas de API REST, pois incluem informações identificadoras importantes, como metadados, autorizações, identificadores uniformes de recursos (URIs), cache, cookies e mais. Cabeçalhos de solicitação e resposta, juntamente com os códigos de status HTTP convencionais, são usados em APIs REST bem projetadas.

# PRÁTICAS RECOMENDADAS DA API REST

**Use métodos HTTP corretamente:** Use os métodos HTTP adequados (GET, POST, PUT, PATCH, DELETE etc.) para realizar diversas operações nos recursos, conforme a especificação HTTP.

**Use URLs significativos:** Use URLs de fácil leitura e entendimento que fornecem uma indicação clara do recurso sendo acessado.

**Use JSON de acordo com o formato dos dados:** Apesar de as APIs RESTful aceitarem e responderem a pedidos de API em diversos formatos de dados, JSON é o padrão moderno para enviar e receber dados de APIs.

**Não seja criativo com os códigos de resposta:** Use códigos de resposta HTTP padrão para indicar o estado de um pedido, como 200 para sucesso, 404 para não encontrado e 500 para erros de servidor.

**Implemente controle de versão:** Isto garante que as alterações à API não quebrem aplicativos existentes forçando-os a migrar para a nova versão.



# 06

## Exemplo prático de API REST

---



# EXEMPLO PRÁTICO:

Para ilustrar a criação de uma API REST simples com Python, considere um exemplo básico em que desenvolvemos uma API para gerenciar um banco de dados de livros. Usando o Flask, uma estrutura web leve, podemos configurar endpoints para adicionar, recuperar, atualizar e excluir registros de livros. Aqui está um trecho de como uma solicitação GET é tratada no Flask:

```
do frasco importe Flask, jsonify

app = Flask(__name__)
livros = [{ 'id': 1, 'título': 'Python Essentials', 'autor': 'Jane Doe' }]

@app.route('/books', methods=['GET'])
def get_books():
    retorne jsonify({'books': livros})

se __name__ == '__main__':
    app.run(debug=True)
```

# Participantes:

**Fellipe Eduardo nº5**

**Gabriel Fortunato nº9**

**João Pedro Nardin nº17**

**Nicholas Petros nº24**

**Renan Pereira nº26**