

# Nest.js: desenvolvimento de APIs

# O que é Nest.js?

- <https://nestjs.com/>
- Framework Node Fullstack MVC WEB.
- Lançado em 2017 pelo Kamil Mysliwiec.
- Construído em TypeScript.
- Usa as ideias de arquitetura do Angular.
- Foco em microsserviços.

## Quais as vantagens do Nest.js?

- Coc (Convention Over Configuration).
- Usar TypeScript no desenvolvimento.
- Arquitetura escalável.
- Grande compatibilidade com sistemas:
  - Banco de dados: MySQL, PostgreSQL, MongoDB, etc.
  - Websockets
  - Microsserviços: Redis, RabbitMQ, Kafka, gRPC, CQRS e etc.

# O que o Nest.js não é ou não faz?

O Nest.js não é uma biblioteca front-end.

- O Nest.js é um framework back-end.

O Nest.js não se equipara ao jQuery, Angular, React ou Vue.

- Ele pode usar estas bibliotecas front-end, trabalhar em conjunto com elas.

## Onde o Nest.js é utilizado?

- Adidas
- Decathlon.
- Autodesk.
- Neo4j.

## Cilada ao usar o Nest.js

- Usá-lo sem aprender JavaScript/Node.
- Achar que o Nest.js é o melhor framework Node.
- Usar o Nest.js sem TypeScript
- Não organizar a aplicação utilizando as convenções do framework.
- Achar que resolverá todos os problemas para criação de aplicações WEB ou de microserviços.

# API REST

# Porque API?

- Comunicação entre aplicações (Interoperabilidade).
- Interações entre diferentes plataformas.
- Aplicações como serviços.
- Acessibilidade e disponibilidade de informações.
- Aplicações independentes.



# Exemplos de uso de API

Informações públicas

Redes sociais

Negócios

Serviços

Logística

Mobile

IoT

BIOS

DOM JavaScript

Linguagens de  
programação

# REST

- (R)epresentational (S)tate (T)ransfer.
- Criado por **Roy Fielding** (2000).
- Arquitetura para padronizar o uso do HTTP.

# MAIS UM POUCO SOBRE REST

- Significado para o uso de cada verbo HTTP: GET POST PUT DELETE PATCH OPTIONS, etc.
- Media types.
- Domínio do formato JSON na troca de mensagens.
- Não manter estado.
- Hypermedia.
- Padronização de acesso aos recursos através de URI.
- Padronização de status code e errors.

Endereço da API

Nome da API

Recurso



**www.myapp.com/api/vendas/clientes**

**www.api.myapp.com/vendas/clientes**

**www.api.myapp.com/v1/vendas/clientes**

# API RESTful

VERBO	Descrição	Chamada HTTP
GET	Consultar informações	GET /contas-pagar
POST	Criar um novo recurso	POST /contas-receber
PUT	Atualizar um recurso existente	PUT /contas-receber/2320
DELETE	Excluir um recurso existente	DELETE /contas-pagar/4060
OPTIONS	Consultar informações na API	OPTIONS /contas-pagar



**KEEP  
CALM  
AND  
GO  
PRACTICE**