

**INSTITUTO
FEDERAL**

São Paulo

Câmpus
Birigui

INTRODUÇÃO A API REST COM NODE.JS

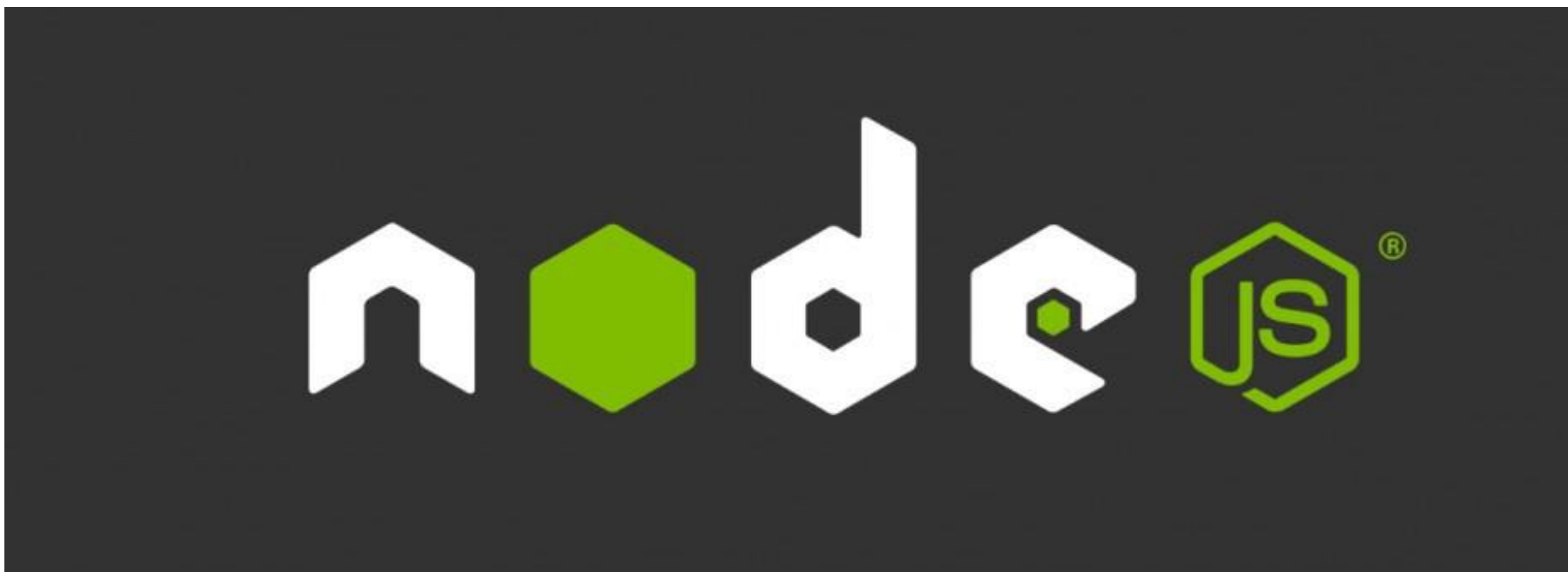
PROF. IGOR SOLFA ROMERO
IGORSOLFAROMERO@GMAIL.COM

Pré-requisitos



**INSTITUTO
FEDERAL**
São Paulo
Câmpus
Birigui

Node.js



Visual Studio Code



Postman



POSTMAN



**INSTITUTO
FEDERAL**
São Paulo
—
Campus
Birigui

Conceitos



**INSTITUTO
FEDERAL**
São Paulo
—
Câmpus
Birigui

Conceitos

- ▶ O que é API?

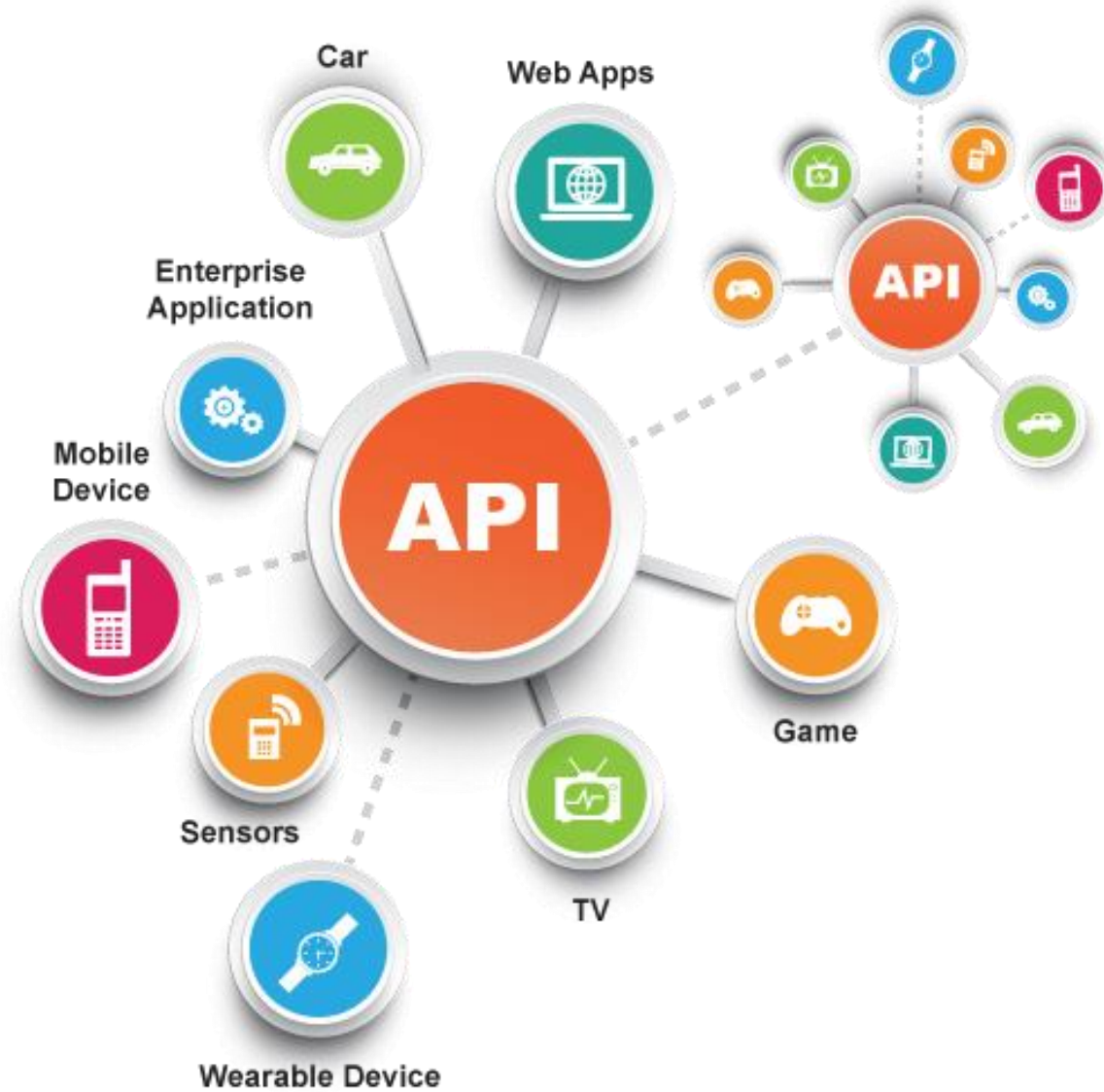


**INSTITUTO
FEDERAL**
São Paulo
Câmpus
Birigui

Conceitos

- ▶ O que é API?
- ▶ Uma API (Interface de programação de aplicações) permite que sua solução ou serviço se comunique com outros produtos e serviços sem precisar saber como eles foram implementados, o que simplifica o desenvolvimento de aplicações.





Fonte: <https://medium.com/gerente-de-produto-ou-product-manager/o-que-%C3%A9-a-economia-da-api-559b70dcb605>

Conceitos

- ▶ O que é REST?



**INSTITUTO
FEDERAL**
São Paulo
Câmpus
Birigui

Conceitos

- ▶ O que é REST?
- ▶ REST é um modelo utilizado para projetar arquitetura de softwares distribuídos que fazem comunicação de dados pela rede.



Conceitos

- ▶ O que é REST?
- ▶ REST (transferência de estado representacional) é um modelo utilizado para projetar arquitetura de softwares distribuídos que fazem comunicação de dados pela rede.
- ▶ No REST a implementação do cliente (front-end) e do servidor (back-end) são feitas separadamente sem nenhuma interferência de um ou outro. Isso significa que os dois podem ser desenvolvidos separadamente.



Conceitos

- ▶ O que é API REST?



**INSTITUTO
FEDERAL**
São Paulo
—
Câmpus
Birigui

Conceitos

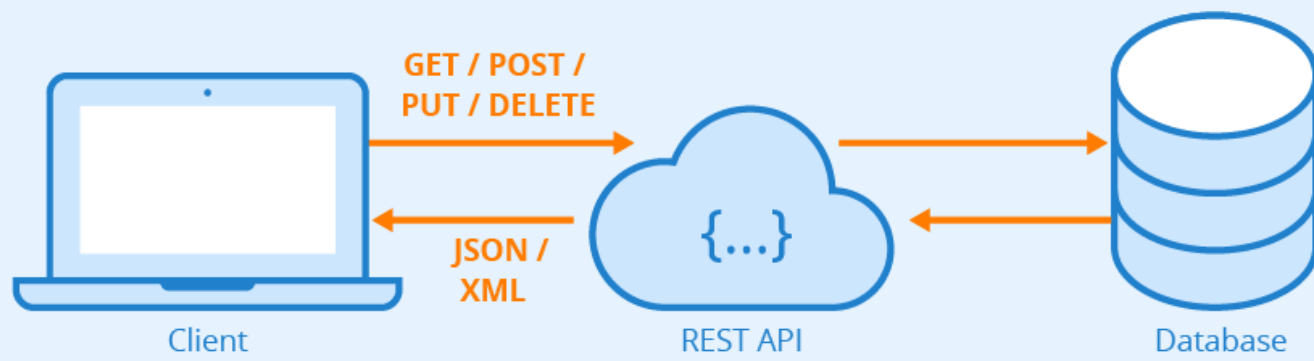
- ▶ O que é API REST?
- ▶ É uma interface de programação de aplicações (API) que está em conformidade com as restrições do estilo de arquitetura REST, permitindo a interação com serviços web.



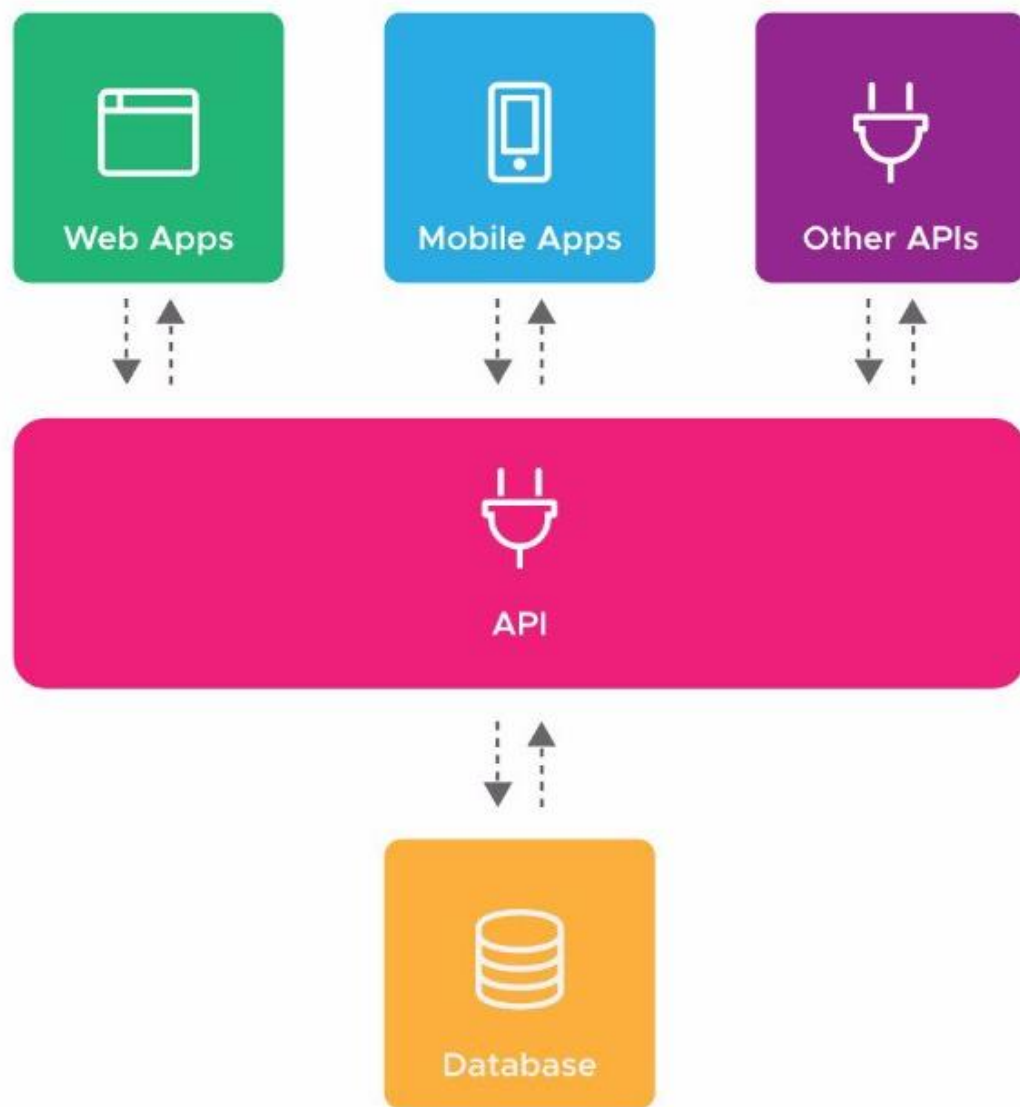
Conceitos

- ▶ O que é API REST?
- ▶ É uma interface de programação de aplicações (API) que está em conformidade com as restrições do estilo de arquitetura REST, permitindo a interação com serviços web.





Fonte: <https://www.astera.com/pt/tipo/blog/definição-de-api/>



Fonte: <https://thiagolima.blog.br/a-anatomia-de-uma-api-restful-80df2aca158e?gi=aac10ecff67d>

Exemplos de APIs



**INSTITUTO
FEDERAL**
São Paulo
—
Câmpus
Birigui

Exemplos de APIs

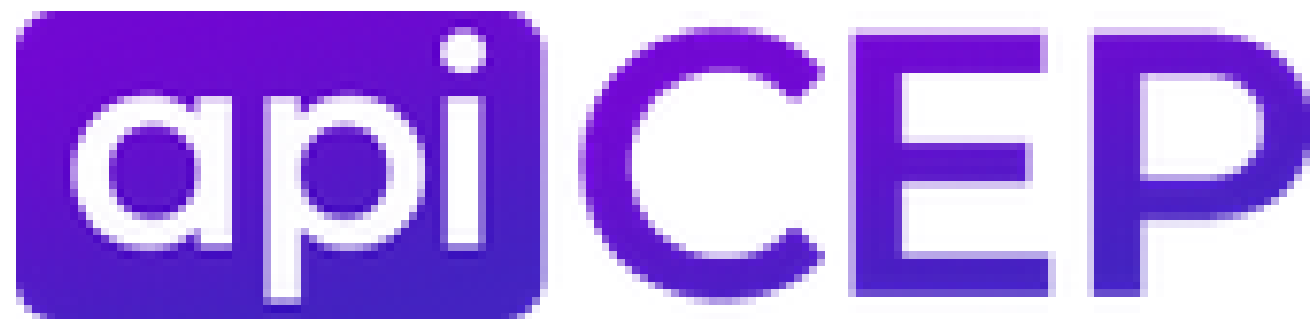


Link: <https://pokeapi.co/>



**INSTITUTO
FEDERAL**
São Paulo
—
Câmpus
Birigui

Exemplos de APIs



Link: <https://apicep.com/api-de-consulta/>

Exemplos de APIs

The logo for CLIMATEMPO, featuring the word "CLIMATEMPO" in white, uppercase, sans-serif font, centered within a solid blue rectangular background.

Link: <https://advisor.climatepo.com.br/>



**INSTITUTO
FEDERAL**
São Paulo
—
Campus
Birigui

Exemplos de APIs



Link: <https://developers.google.com/maps/>

Exemplos de APIs



Link: <https://developer.twitter.com/en/docs>

Tecnologias



**INSTITUTO
FEDERAL**
São Paulo
Câmpus
Birigui

Tecnologias

- ▶ JavaScript



**INSTITUTO
FEDERAL**
São Paulo
—
Campus
Birigui

Tecnologias

- ▶ JavaScript
- ▶ Node.js



**INSTITUTO
FEDERAL**
São Paulo
—
Campus
Birigui

Tecnologias

- ▶ JavaScript
- ▶ Node.js
- ▶ HTTP / HTTPS



Tecnologias

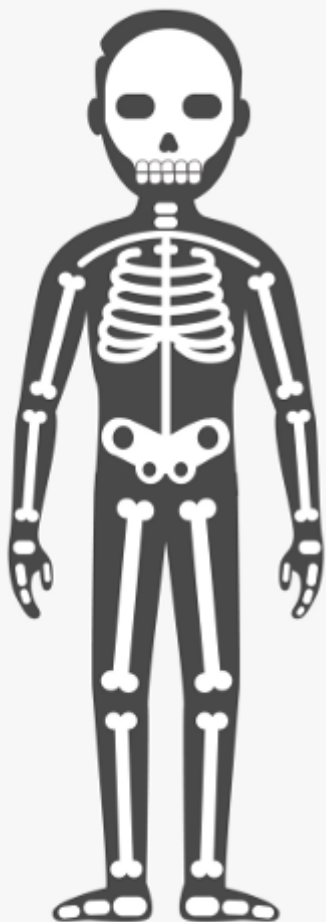
- ▶ JavaScript
- ▶ Node.js
- ▶ HTTP / HTTPS
- ▶ Framework Node (Express)



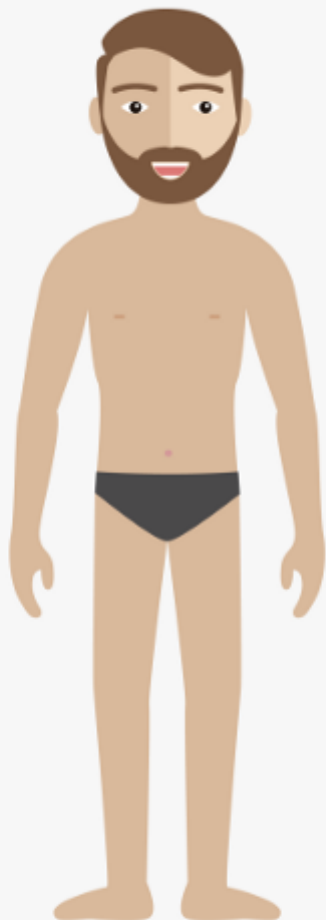
Tecnologias

- ▶ JavaScript
- ▶ JavaScript é uma linguagem de programação que permite a você implementar itens complexos em páginas web. Ele permite criar conteúdo que se atualiza dinamicamente, controlar mídias, imagens animadas, e tudo o mais que há de interessante.





HTML



CSS



Javascript

Fonte: <https://www.alura.com.br/artigos/html-css-e-js-definicoes>



**INSTITUTO
FEDERAL**
São Paulo
Câmpus
Birigui

Tecnologias

- ▶ Node.js
- ▶ O Node.js pode ser definido como um ambiente de execução JavaScript server-side.



Tecnologias

- ▶ Node.js
- ▶ O Node.js pode ser definido como um ambiente de execução JavaScript server-side.
- ▶ Plataforma open-source que permite a execução da linguagem JavaScript do lado do servidor.



Tecnologias

- ▶ Características do Node.js



Tecnologias

- ▶ Características do Node.js
- ▶ O node.js possui uma execução single-thread.



Tecnologias

- ▶ Características do Node.js
- ▶ O node.js possui uma execução single-thread.
- ▶ Em um servidor web utilizando linguagens tradicionais, para cada requisição recebida é criada uma nova thread para tratá-la. A cada requisição, serão demandados recursos computacionais para a criação dessa nova thread. Uma vez que esses recursos são limitados, as threads não serão criadas infinitamente, e quando esse limite for atingido, as novas requisições terão que esperar a liberação desses recursos alocados para serem tratadas.



Tecnologias

- ▶ Características do Node.js
- ▶ No modelo Node.js, apenas uma thread é responsável por tratar as requisições.



Tecnologias

- ▶ Características do Node.js
- ▶ No modelo Node.js, apenas uma thread é responsável por tratar as requisições.
- ▶ Essa thread é chamada de Event Loop, e leva esse nome pois cada requisição é tratada como um evento. O Event Loop fica em execução esperando novos eventos para tratar, e para cada requisição, um novo evento é criado.



Tecnologias

► HTTP

Tecnologias

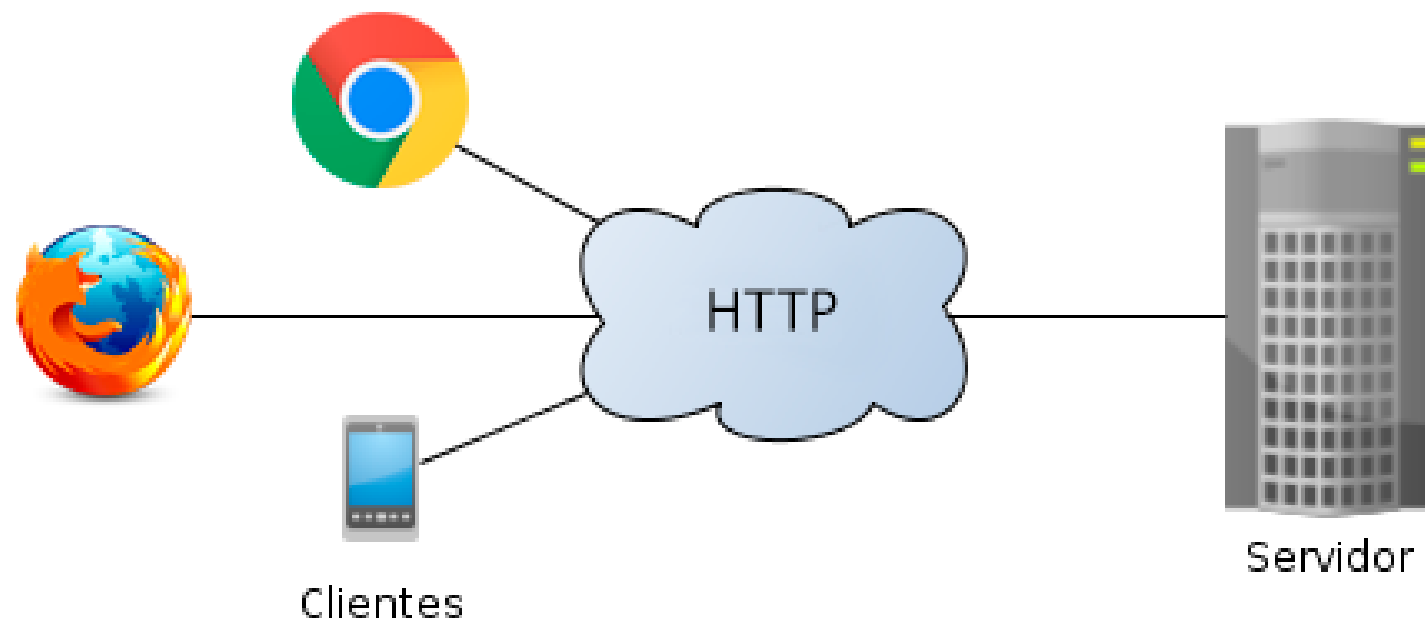
- ▶ HTTP
- ▶ HTTP é uma sigla que quer dizer HyperText Transfer Protocol, ou seja, Protocolo de transferência de dados no formato texto.

Tecnologias

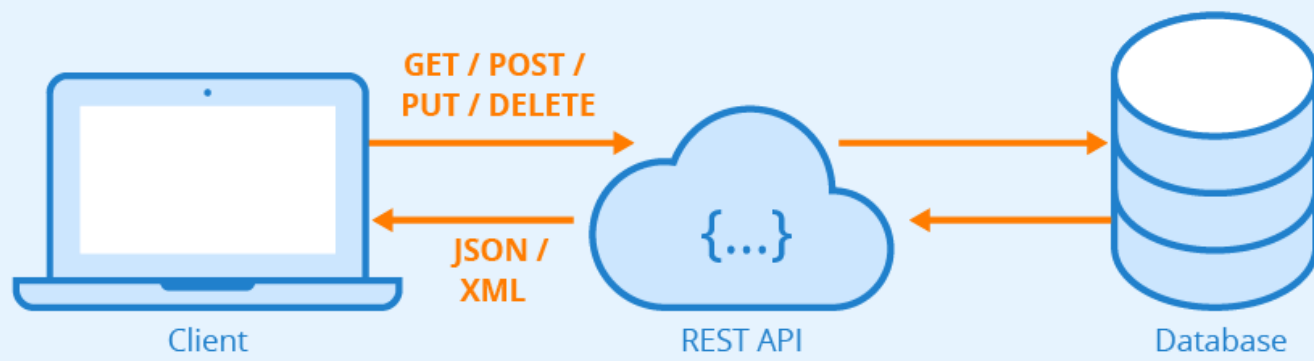
- ▶ HTTP
- ▶ HTTP é uma sigla que quer dizer HyperText Transfer Protocol, ou seja, Protocolo de transferência de dados no formato texto.
- ▶ Seu funcionamento é no modelo **Client-Server** (*Cliente-Servidor*), sendo o **Cliente** sempre quem solicita alguma informação, não importando qual informação, agora o **Servidor** é sempre aquele que disponibiliza e retorna a informação solicitada.

Tecnologias

- ▶ HTTP
- ▶ Na prática o **Cliente** solicita por meio do HTTP um conteúdo para o **Servidor** e caso encontre o que foi solicitado o **Servidor** retorna uma resposta.



Fonte: <https://medium.com/estudonaweb/entendendo-o-protocolo-http-473753cdf66b>



Fonte: <https://www.astera.com/pt/tipo/blog/definição-de-api/>

Tecnologias

- ▶ Exemplos de HTTP

Tecnologias

- ▶ Exemplos de HTTP
- ▶ GET - Buscar dados

Tecnologias

- ▶ Exemplos de HTTP
- ▶ GET - Buscar dados
- ▶ POST - Enviar dados

Tecnologias

- ▶ Exemplos de HTTP
- ▶ GET - Buscar dados
- ▶ POST - Enviar dados
- ▶ PUT - Alterar dados

Tecnologias

- ▶ Exemplos de HTTP
- ▶ GET - Buscar dados
- ▶ POST - Enviar dados
- ▶ PUT - Alterar dados
- ▶ PATCH - Alterar dado

Tecnologias

- ▶ Exemplos de HTTP
- ▶ GET - Buscar dados
- ▶ POST - Enviar dados
- ▶ PUT - Alterar dados
- ▶ PATCH - Alterar dado
- ▶ DELETE - Deletar dado

Tecnologias

- ▶ Códigos HTTP

Tecnologias

- ▶ Códigos HTTP
- ▶ 1xx - Informação

Tecnologias

- ▶ Códigos HTTP
- ▶ 1xx - Informação
- ▶ 2xx - Sucesso

Tecnologias

- ▶ Códigos HTTP
- ▶ 1xx - Informação
- ▶ 2xx - Sucesso
- ▶ 3xx - Redirecionamento

Tecnologias

- ▶ Códigos HTTP
- ▶ 1xx - Informação
- ▶ 2xx - Sucesso
- ▶ 3xx - Redirecionamento
- ▶ 4xx - Erro no Cliente

Tecnologias

- ▶ Códigos HTTP
- ▶ 1xx - Informação
- ▶ 2xx - Sucesso
- ▶ 3xx - Redirecionamento
- ▶ 4xx - Erro no Cliente
- ▶ 5xx - Erro no Servidor

Tecnologias

- ▶ Express - Framework Node.js

Tecnologias

- ▶ Express - Framework Node.js
- ▶ O que é o Express?

Tecnologias

- ▶ Express - Framework Node.js
- ▶ O que é o Express?
- ▶ É o framework Node mais popular e a biblioteca subjacente para uma série de outros frameworks do Node. O Express oferece soluções para:

Tecnologias

- ▶ Express - Framework Node.js
- ▶ Gerenciar requisições de diferentes verbos HTTP em diferentes URLs.

Tecnologias

- ▶ Express - Framework Node.js
- ▶ Gerenciar requisições de diferentes verbos HTTP em diferentes URLs.
- ▶ Integrar "view engines" para inserir dados nos templates.

Tecnologias

- ▶ Express - Framework Node.js
- ▶ Gerenciar requisições de diferentes verbos HTTP em diferentes URLs.
- ▶ Integrar "view engines" para inserir dados nos templates.
- ▶ Definir as configurações comuns da aplicação web, como a porta a ser usada para conexão e a localização dos modelos que são usados para renderizar a resposta.

Tecnologias

- ▶ Express - Framework Node.js
- ▶ Gerenciar requisições de diferentes verbos HTTP em diferentes URLs.
- ▶ Integrar "view engines" para inserir dados nos templates.
- ▶ Definir as configurações comuns da aplicação web, como a porta a ser usada para conexão e a localização dos modelos que são usados para renderizar a resposta.
- ▶ Adicionar novos processos de requisição por meio de "middleware" em qualquer ponto da "fila" de requisições.

Tecnologias

- ▶ E além do Express... ?

Tecnologias



nest

Fonte: <https://nestjs.com/>

Tecnologias



Fonte: <https://adonisjs.com/>

Tecnologias



Fonte: <https://eggjs.org/>

Vamos praticar !

- ▶ Link para o projeto: <https://github.com/igorsromero/minicurso-api-node>

Links Úteis

- ▶ Github: <https://github.com/igorsromero/>
- ▶ LinkedIn: <https://www.linkedin.com/in/igorsromero/>

Referência bibliográfica

- ▶ <https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces>
- ▶ <https://medium.com/trainingcenter/introdução-rest-7ec4cefd16c8>
- ▶ <https://www.astera.com/pt/tipo/blog/definição-de-api/>
- ▶ <https://thiagolima.blog.br/a-anatomia-de-uma-api-restful-80df2aca158e?gi=aac10ecff67d>

Referência bibliográfica

- ▶ <https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>
- ▶ <https://www.alura.com.br/artigos/html-css-e-js-definicoes>
- ▶ <https://www.opus-software.com.br/node-js/>
- ▶ <https://medium.com/estudonaweb/entendendo-o-protocolo-http-473753cdf6b>

Referência bibliográfica

- ▶ <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods>
- ▶ <https://httpstatuses.com/>
- ▶ https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Express_Nodejs/Introduction