

DESENVOLVIMENTO DE APIS

Conceitos básicos de API

Desenvolvimento de APIs

O que é API?

Tipos de API

Http, URL, URN, URI e Recurso

Métodos de requisição Http

JSON

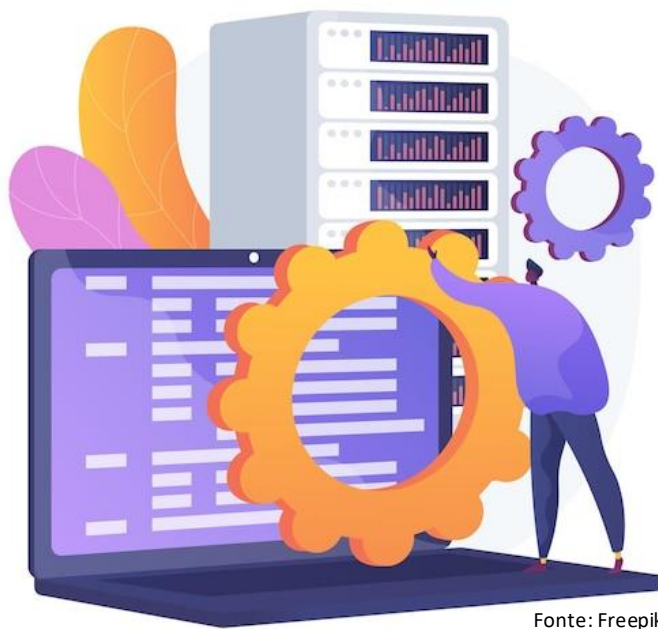
API Rest

Documentação



Fonte: Freepik

O que é API?



Fonte: Freepik

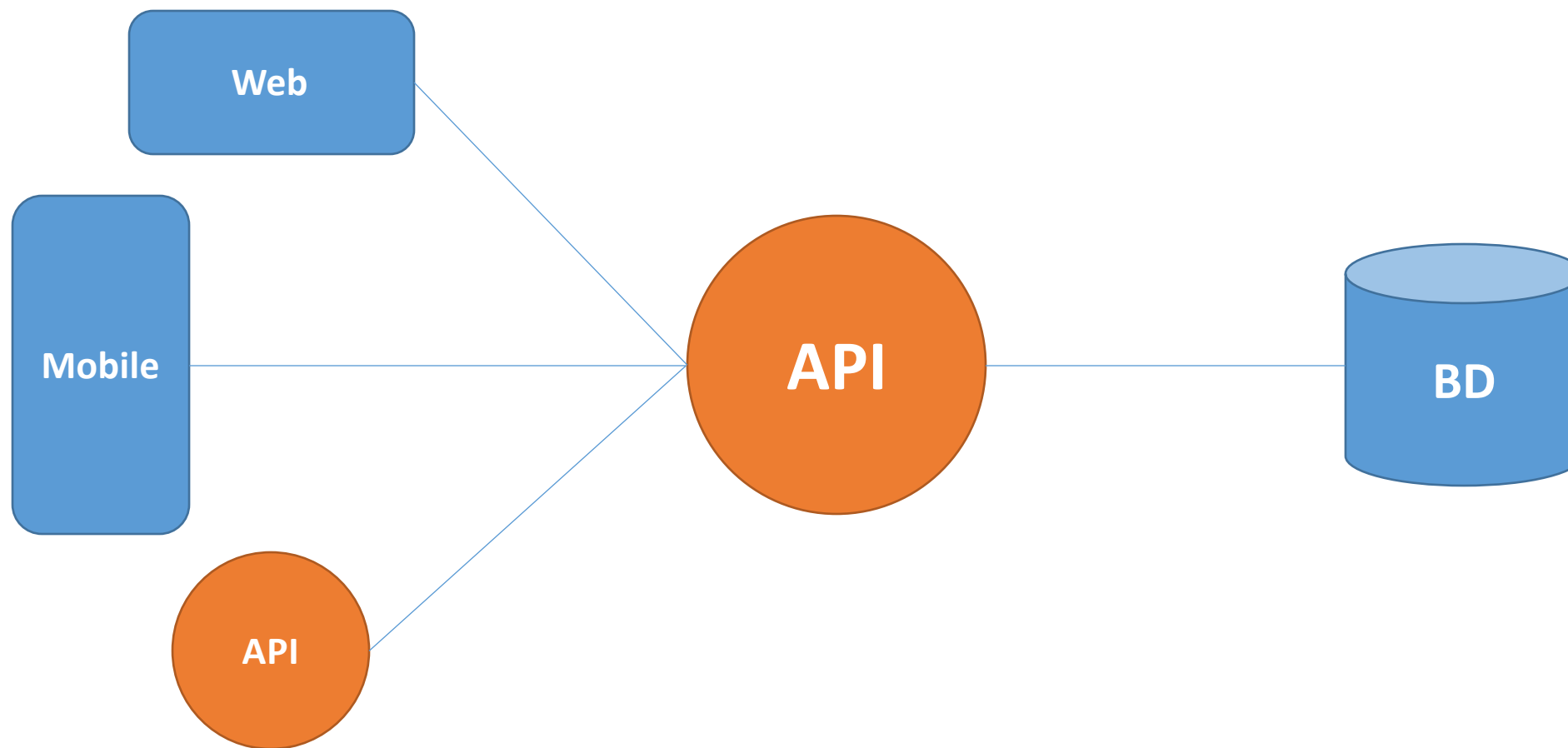
- É um conjunto de rotinas e padrões estabelecidos por um software.
- Mecanismos que permitem que dois componentes de software se comuniquem usando um conjunto de definições e protocolos.

Exemplo:

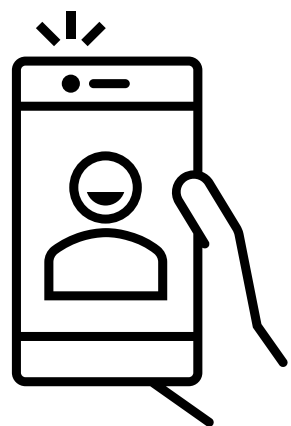
Aplicativos meteorológicos no smartphone

API - Application Programming Interface

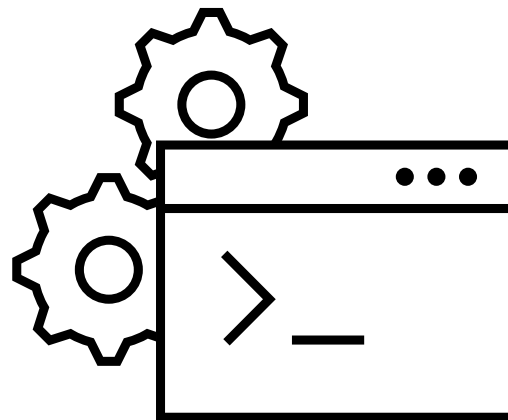
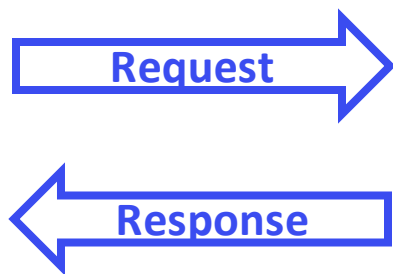
Interface de Programação de Aplicações



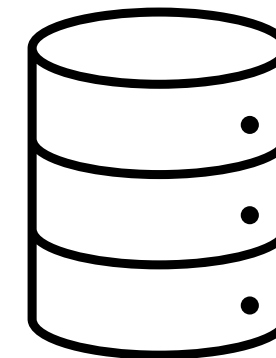
Uma aplicação mobile faz uma requisição para a API, que faz a busca no *server side* e retorna uma resposta.



Front-End



API



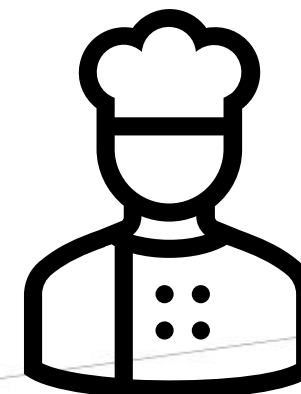
Back-End



Cliente



Garçon



Cozinha

Tipos de APIs



Fonte: Freepik

API aberta/pública

Disponível para
desenvolvedores
externos



Fonte: Freepik

API privada

Uso interno da
empresa



Fonte: Freepik

API de parceiros

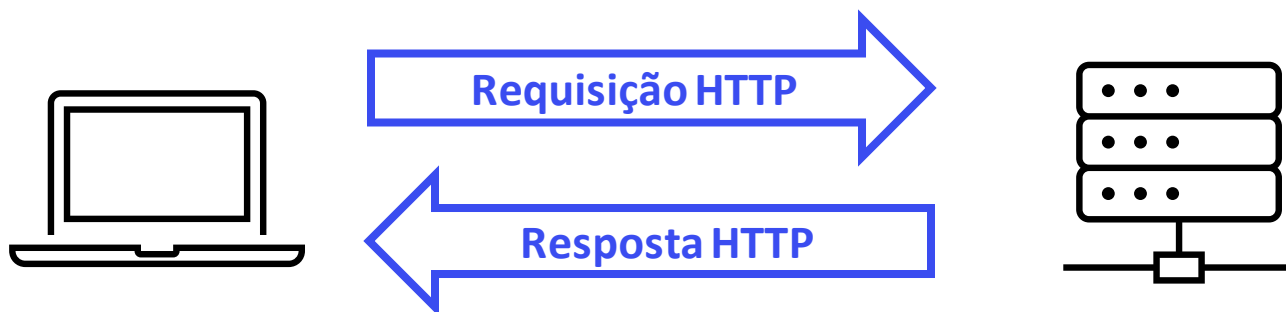
Compartilhada com
parceiros

Fonte: Freepik

HTTP, URL, URN, URI e recurso

Internet → Protocolo HTTP

(cliente e servidor)



Fonte: Freepik

URL = domínio

URN = caminho do recurso

recurso = arquivo acessado

URI = protocolo (http) + domínio (URL) + caminho (URN)

protocolo

URL

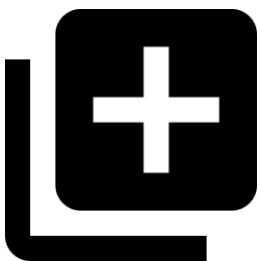
URN

recurso

http://www.sp.senai.br/noticias/27/21809/senaisp-oferece-cursos-online-gratuitos.html

URI

Métodos de requisição HTTP



Fonte: Freepik

POST

criar dado



Fonte: Freepik

GET

obter dado



Fonte: Freepik

PUT

atualizar dado



Fonte: Freepik

DELETE

apagar dado

Fonte: Freepik

JSON

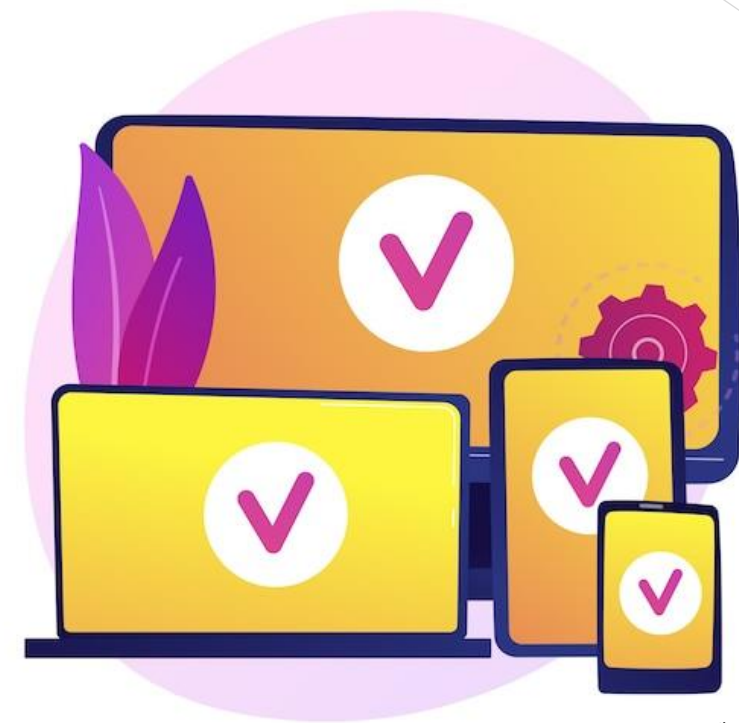
Javascript Object Notation

formato texto → troca de dados entre sistemas

mesmo JSON → diferentes renderizações

pares de dados → nome e valor

```
{  
  "livro": "físico",  
  "título": "API básico",  
  "paginas": 267,  
  "genero": ["técnico", "back-end", "programação",  
  "iniciante"]  
}
```



Fonte: Freepik

JSON

Vantagens

- mais simples
- *parsing* (análise) mais fácil
- suporte a objetos (tipado)
- mais veloz (execução e transporte de dados)
- tamanho reduzido



Fonte: Freepik

API Rest

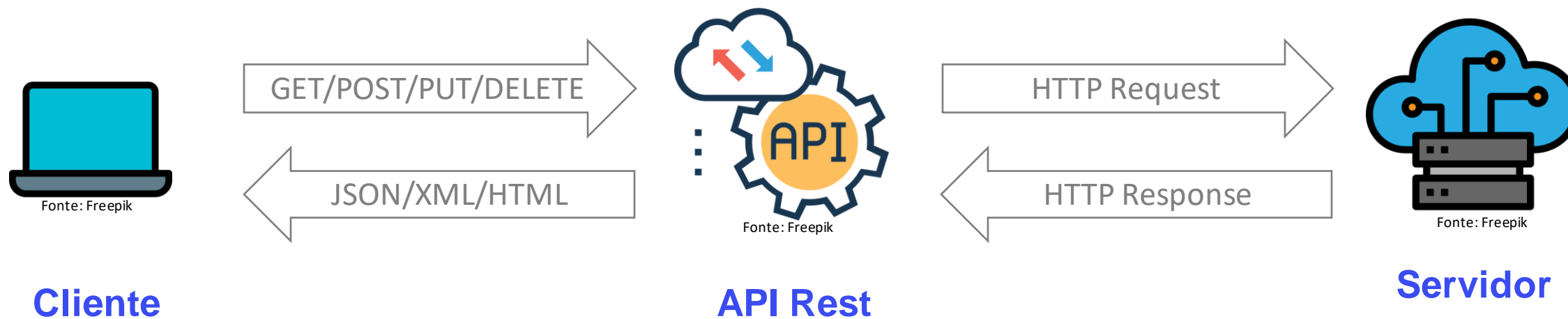
Representational State Transfer

- comunicação de dados → protocolo http
- abstração da arquitetura web
- comunicação entre aplicações



Fonte: Freepik

API Rest Model

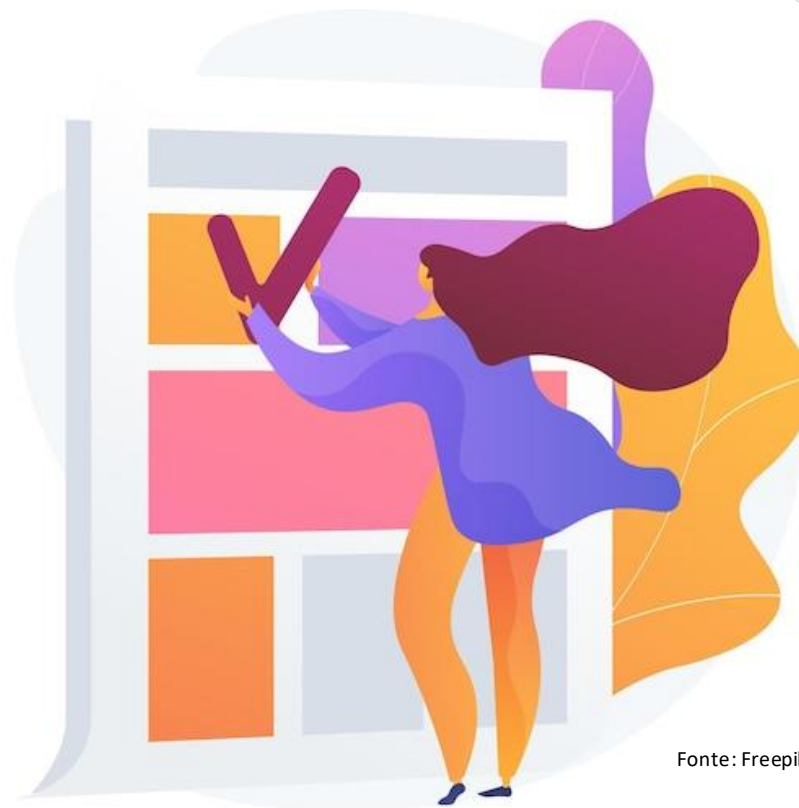


Documentação

Resumo da API

(objetivo + dificuldades e soluções do desenvolvimento)

- Explicar funcionamento
- Listas funcionalidades e recursos
- Serve como base para melhorias futuras



Fonte: Freepik

ViaCEP

Exemplo de API

- Objetivo
- Funcionamento
- Tipos de retorno



Procurando um [webservice](#) gratuito e de alto desempenho para consultar Códigos de Endereçamento Postal (CEP) do Brasil?

Utilize nosso serviço, melhore a qualidade de suas aplicações web e colabore para manter esta base de dados atualizada.

Acessando o webservice de CEP

Para acessar o webservice, um CEP no formato de **{8}** dígitos deve ser fornecido, por exemplo: "01001000".

Após o CEP, deve ser fornecido o tipo de retorno desejado, que deve ser "json" ou "xml".

Exemplo de pesquisa por CEP:

viacep.com.br/ws/01001000/json/

Validação do CEP

Quando consultado um CEP de formato inválido, por exemplo: "950100100" (9 dígitos), "95010A10" (alfanumérico), "95010 10" (espaço), o código de retorno da consulta será um **400** (Bad Request). Antes de acessar o webservice, valide o formato do CEP e certifique-se que o mesmo possua **{8}** dígitos. Exemplo de como validar o formato do CEP em javascript está disponível nos exemplos abaixo.

Quando consultado um CEP de formato válido, porém inexistente, por exemplo: "99999999", o retorno conterá um valor de "erro" igual a "true". Isso significa que o CEP consultado não foi encontrado na base de dados. Veja como manipular este "erro" em javascript nos exemplos abaixo.

Bons estudos!