



# SW-II API's

---

Prof. Anderson Vanin

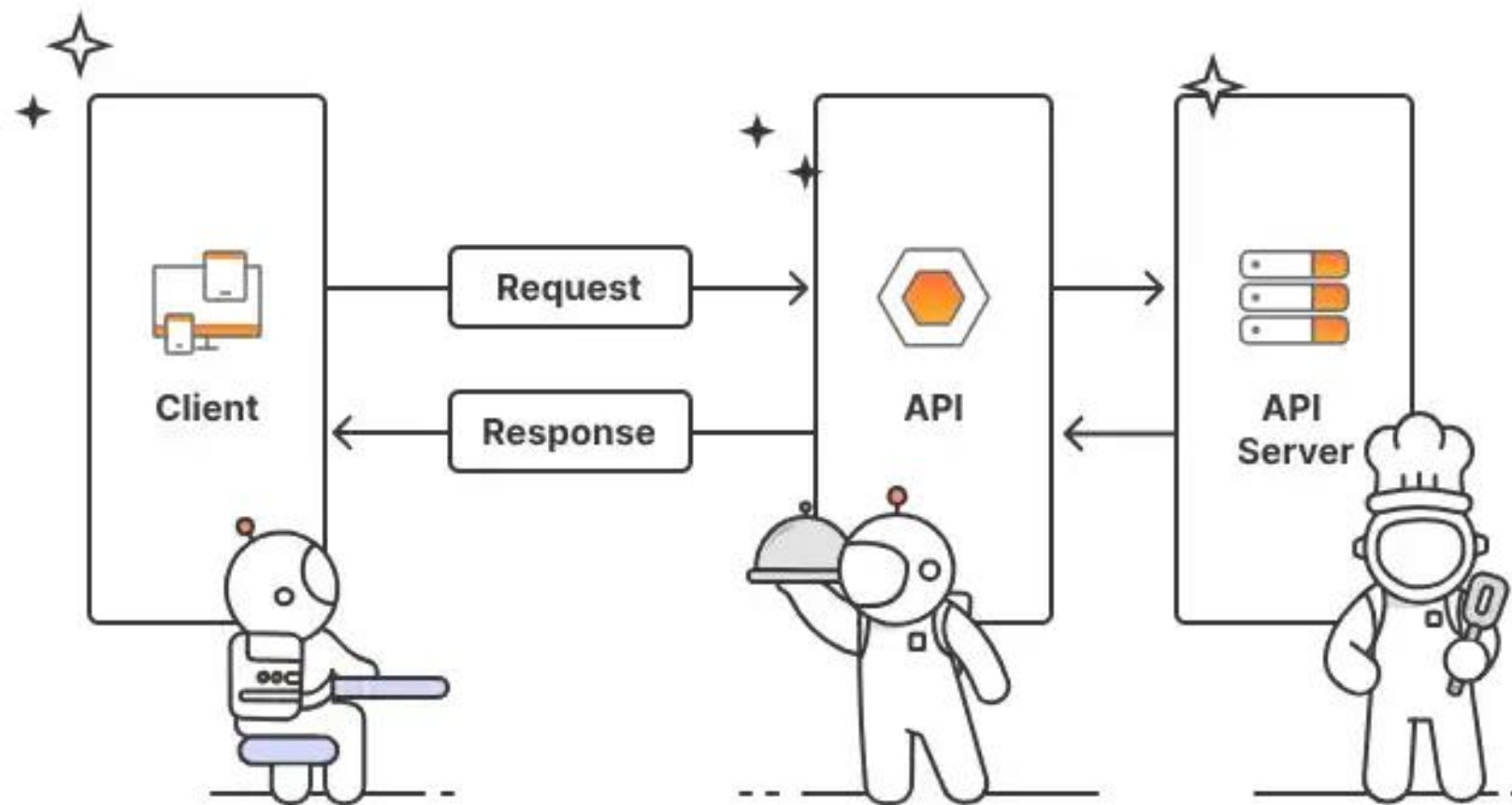
# Cenário atual das aplicações web

Atualmente, vivemos em uma fase na qual a maioria dos usuários utilizam diversos tipos de devices para se conectarem à internet. Os mais populares são smartphones, tablets e notebooks. Desenvolver sistemas para diversos tipos de devices requer o trabalho de construir **web services**, também conhecidos pelo nome de **APIs** (***Application Program Interface***).

# API

Basicamente, essas **APIs são sistemas back-end** que têm o objetivo de trabalhar apenas com dados, de forma centralizada, permitindo que sejam desenvolvidos, separadamente, aplicações clientes que possuem interfaces para o usuário final. Essas aplicações clientes geralmente são: ***mobile apps, aplicações desktop ou webapps.***

# Funcionamento



# Funcionamento

- A API atua como intermediário entre um aplicativo e um servidor.
- É uma coleção de código de programação onde os dados são trocados entre um produto de software e outro.
- Esta comunicação tem duas partes – uma é o **cliente** e a outra é o **servidor**.

# Funcionamento

Então, no relacionamento **cliente-servidor**:

- O **cliente** é quem faz a solicitação ou envia a chamada da API ao servidor.
- O **servidor** hospeda a API e expõe um **conjunto de terminais ou interfaces** com os quais os clientes podem interagir.
- Este tipo de transmissão de dados funciona com base no princípio de **solicitação e resposta**. Resumindo, o cliente envia a solicitação e o servidor responde.

# Para que serve uma API?

- A API tem como objetivo permitir que diferentes sistemas possam se comunicar de forma mais eficiente, possibilitando a integração entre plataformas e aplicativos. Dessa forma, é possível otimizar processos e aumentar a eficiência das operações.
- As APIs também são muito utilizadas para permitir o acesso a dados e informações de terceiros, como redes sociais, serviços de geolocalização, ferramentas de análise de dados e muitos outros.

# Para que serve uma API?

- Dessa maneira, os desenvolvedores conseguem criar aplicativos mais completos e integrados para oferecer uma melhor experiência para os usuários.
- Além disso, uma API pode ser utilizada para criar novos serviços e produtos, como os aplicativos de delivery, que utilizam as APIs de serviços de mapas e geolocalização para rastrear e entregar os pedidos de seus consumidores.
- Em sua essência, o objetivo de uma API é permitir que um sistema específico possa recorrer às funcionalidades dos demais **sem que seja necessário ao usuário migrar entre as plataformas.**



# Como funciona uma API - Exemplos

Suponha que você esteja desenvolvendo um aplicativo de previsão do tempo e deseja obter informações atualizadas de uma fonte confiável. Em vez de escrever todo o código para acessar essas informações, você pode utilizar uma API de previsão do tempo. Essa API geralmente terá uma documentação que descreve como se comunicar com ela, incluindo quais parâmetros enviar e como interpretar a resposta. Por exemplo, ela pode permitir que você especifique a localização para a qual deseja obter a previsão do tempo, como uma cidade ou um código postal. Em seguida, a API retorna um conjunto de dados estruturados contendo informações como a temperatura atual, as condições do tempo e as previsões para os próximos dias.

# Como funciona uma API - Exemplos

Outro exemplo de uso comum de APIs é no setor financeiro. Empresas de serviços financeiros podem usar APIs para acessar dados de mercado atualizados em tempo real, que podem ser usados para tomar decisões de investimento informadas. Esses dados podem incluir preços de ações, taxas de câmbio, preços de commodities e muito mais.

# Como funciona uma API - Exemplos

No e-commerce, isso pode ser notado na aba de pagamento: o consumidor está no site da sua loja preferida, escolhe seu produto ou serviço, coloca no carrinho e finaliza a compra com seus dados bancários no mesmíssimo domínio, sem precisar ir até o site do banco. Essa comunicação entre o e-commerce e a rede bancária se dá por meio de uma API.

# Tipos de API

De um modo geral, vemos vários modelos de APIs se desenvolvendo para que as companhias possam impulsionar sua presença online e se conectarem com outras empresas e desenvolvedores.

# API Pública

A API pública é aquela que qualquer pessoa pode acessar e usar livremente. Esse tipo de API é frequentemente usado para fins de marketing e para incentivar outros desenvolvedores a criarem aplicativos que se integrem com a plataforma da empresa.

As APIs públicas são comumente encontradas em plataformas de redes sociais, serviços de nuvem e sites de comércio eletrônico.

# API Privada

A API privada é uma interface de programação de aplicativos que é usada dentro de uma organização. Essas APIs são usadas principalmente para conectar sistemas e aplicativos internos, como banco de dados, sistemas de gerenciamento de conteúdo e aplicativos empresariais.

As APIs privadas são acessíveis apenas por pessoas autorizadas dentro de uma instituição, como devs e gerentes de TI.

# API Privada

A API privada é uma interface de programação de aplicativos que é usada dentro de uma organização. Essas APIs são usadas principalmente para conectar sistemas e aplicativos internos, como banco de dados, sistemas de gerenciamento de conteúdo e aplicativos empresariais.

As APIs privadas são acessíveis apenas por pessoas autorizadas dentro de uma instituição, como devs e gerentes de TI.

# Quando fazer a integração via API e por que isso é importante?

Basicamente, uma empresa deve pensar em fazer integrações via API quando precisa conectar seus sistemas internos ou externos a outros sistemas, aplicativos ou serviços.

Afinal, são as APIs permitem que diferentes sistemas conversem entre si, compartilhem informações e funcionem de maneira integrada.



# Quando fazer a integração via API e por que isso é importante?

A integração é especialmente importante em empresas que possuem diferentes sistemas ou aplicativos, como sistemas de gestão empresarial (ERP), sistemas de gerenciamento de relacionamento com o cliente (CRM), sistemas de marketing, plataformas de e-commerce, entre outros. Nesses casos, é comum que os dados estejam armazenados em diferentes sistemas e aplicativos, o que pode dificultar o acesso e o uso dessas informações.

# Quando fazer a integração via API e por que isso é importante?

Dessa maneira, esses sistemas podem ser conectados, permitindo que os dados sejam trocados de maneira automatizada e em tempo real, ajudando a aumentar a eficiência, reduzir custos e melhorar a tomada de decisões da organização.

Além disso, a integração via API também pode ser útil para empresas que desejam conectar seus sistemas a serviços de terceiros, como serviços de pagamento, de logística, de análise de dados, assinatura eletrônica... Vejamos um pouco mais sobre essa última funcionalidade.

# Exemplos de APIs

Existem muitos exemplos de APIs públicas que são amplamente utilizadas por desenvolvedores para criar aplicativos da web e móveis, bem como para integrar serviços de terceiros e fontes de dados em aplicativos de software existentes. A seguir listo alguns exemplos:

# Exemplos de APIs

- **API do Google Maps:** A API do Google Maps permite que os desenvolvedores integrem o Google Maps em seus aplicativos, fornecendo aos usuários recursos como geocodificação, direções e atualizações de tráfego em tempo real.
- **API do Twitter:** A API do Twitter permite que os desenvolvedores acessem dados e funcionalidades do Twitter, como pesquisar tweets, postar tweets e recuperar informações do usuário.

# Exemplos de APIs

- **API do Facebook:** A API do Facebook permite integração da funcionalidade do Facebook em seus aplicativos, como permitir que os usuários façam login com suas contas do Facebook, recuperar informações do usuário e postar nprópria rede social.
- **API OpenWeatherMap:** A API OpenWeatherMap fornece aos desenvolvedores acesso a dados meteorológicos de todo o mundo, incluindo condições atuais do clima e tempo, previsões e dados históricos.
- **API do GitHub:** A API do GitHub permite que o acesso às funcionalidades do GitHub, como gerenciamento de repositórios, criação de requisições e recuperação de informações do usuário.

<https://www.youtube.com/watch?v=3LHSyha0xN0>

# API Web

É um conjunto de endereços (**endpoints**) relativamente aos quais as aplicações fazem requisições HTTP (**HTTP requests**) e recebem respostas (**HTTP responses**).

As respostas vêm em formatos padrão. Os mais comuns são **XML** e **JSON**.

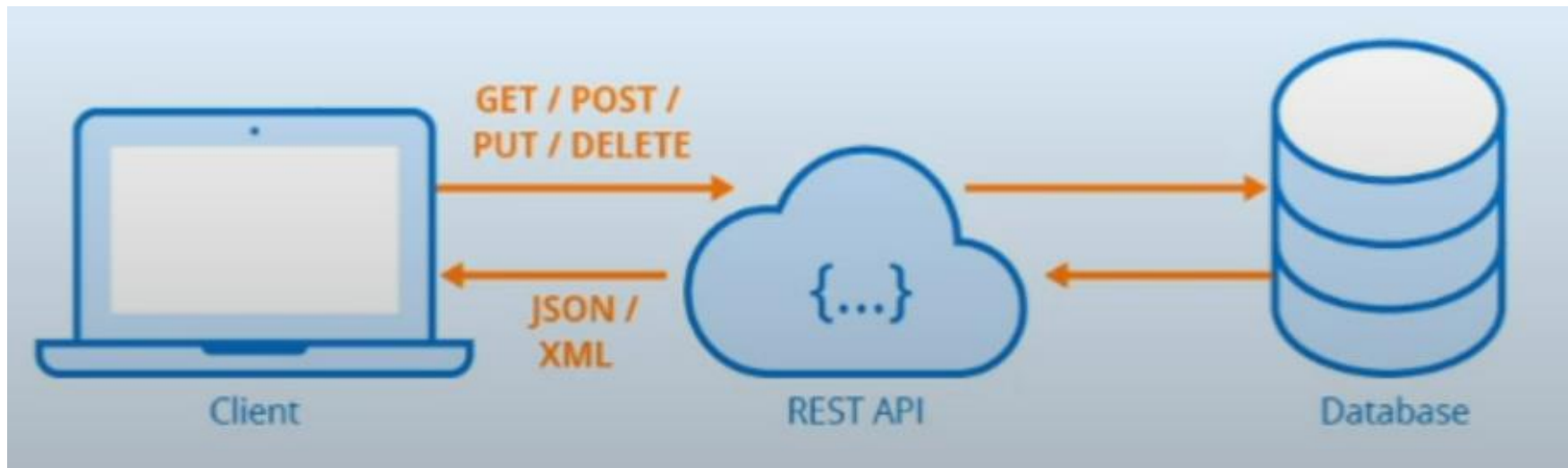
Existem dois modelos que se destacam no tempo: SOAP e REST.

**SOAP** – Simple Object Access Protocol (XML)

**REST** – Representational State Transfer (JSON)

# REST

REST significa ***Representational State Transfer***, é uma abordagem de design estrutural para criação de aplicações baseadas em **requisições HTTP**.





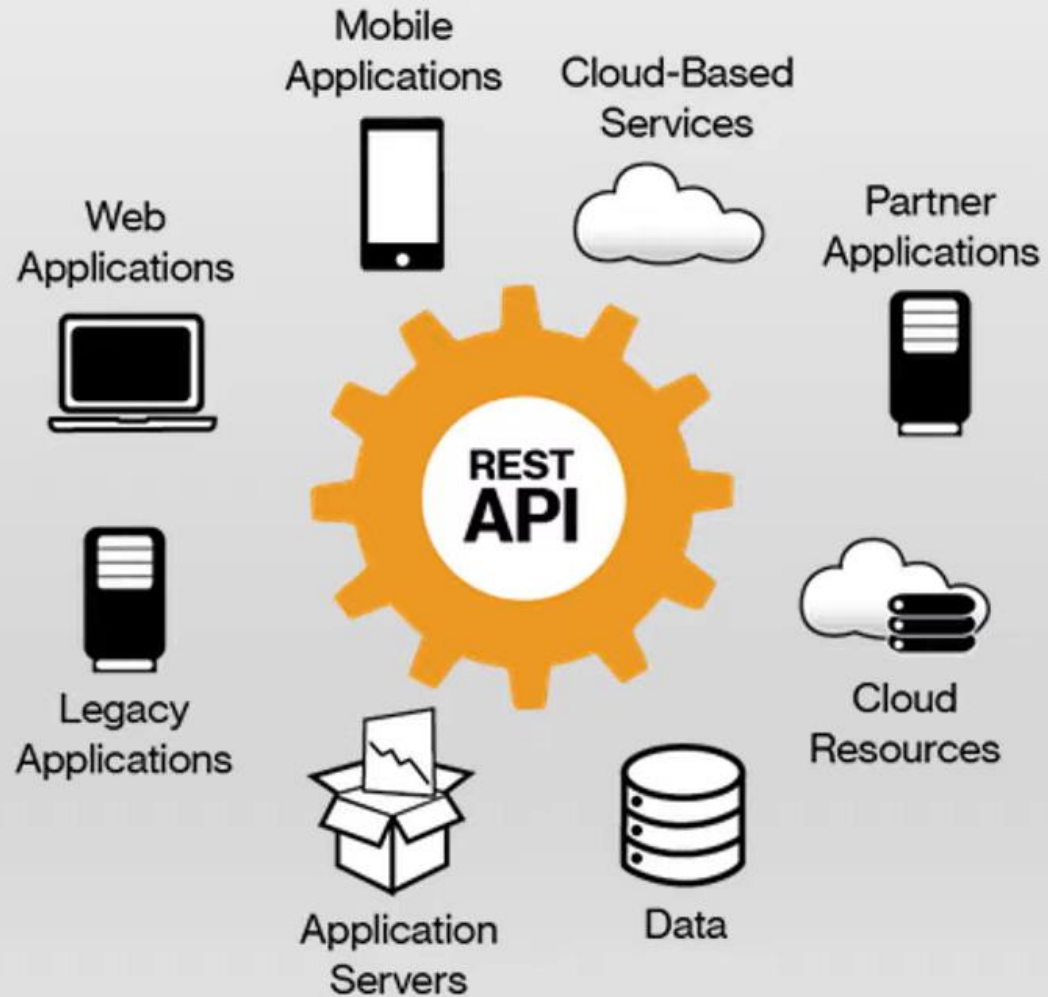
# REST - funcionamento

- Através de requisições HTTP (requests)
- Repostas tradicionalmente em formato JSON
- Sem estado entre comunicações
- Deve procurar facilitar cache de conteúdo no cliente
- Deve ser completamente independente entre API e cliente
- Pode permitir o uso de camadas (escalabilidade e segurança)

# REST X RESTFull

- **REST**: A arquitetura de desenvolvimento.
- **RESTFull**: Uma API que aplica por completo todas as regras da arquitetura REST.

# REST API



# Exemplo prático

Para demonstrar o funcionamento e a construção de uma API, utilizaremos a linguagem PHP para a construção da API e de uma pequena Aplicação Web que irá consumir esta API acessando algumas informações estáticas.