# Interface de programação de aplicações (API) e Web Services

## Criando um Web

Service De que forma um Web Service é criado? Toda vez que alguém publica o back-end de uma aplicação em um servidor web para que ele possa ser acessado por outros sistemas através de protocolos, ele se torna um Web Service.

É um serviço de aplicação que permite a transferência de dados entre plataformas, independentemente da linguagem de programação utilizada

Dessa forma, as informações de uma determinada aplicação são disponibilizadas em um servidor web, e o aplicativo se torna um Web Service. Essas informações armazenadas no servidor web são:

* Processamento de formulários.
* Estrutura de programação de um banco de dados.
* Sistemas de gerenciamento de conteúdo.
* Outros aplicativos da web do lado do servidor usando PHP, JSP, Ruby, ASP.NET, Java e outras linguagens de programação.

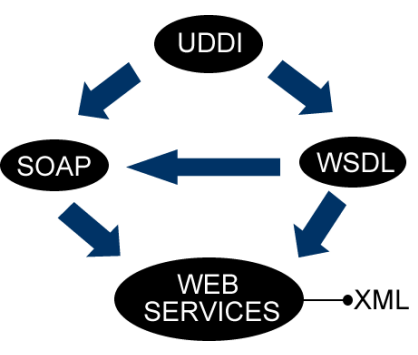
## Modelos de arquiteturas e padrões de Web Services

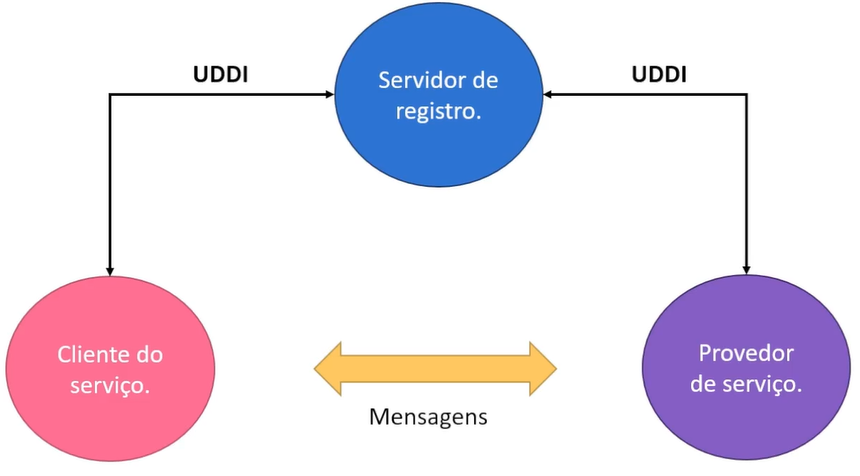
Em Web Services, existem os modelos de arquiteturas e padrões, que possuem, basicamente, o objetivo de detalhar e proteger as informações importantes de uma arquitetura. Existem quatro modelos: modelo orientado a mensagem, modelo orientado a serviço, modelo orientado a recursos e modelo de policiamento. Entenderemos o funcionamento de cada um deles.

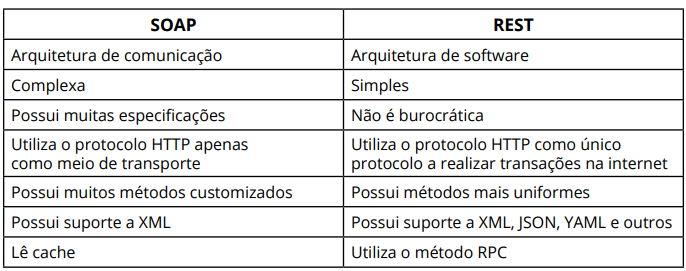
* **Modelo orientado a mensagem**: também conhecido como message oriented model, tem como objetivo priorizar detalhes importantes da arquitetura, os quais, necessariamente, estão relacionados com o processamento de mensagens, por exemplo, sua estrutura, o modo de transporte, entre outros aspectos relevantes (MEDYK, 2006).
* **Modelo orientado a serviço**: também conhecido como service oriented model, este modelo prioriza detalhes importantes relacionados ao serviço e à ação da arquitetura.
* **Modelo orientado a recursos**: também conhecido como resource oriented model, este modelo prioriza detalhes importantes relacionados aos recursos da arquitetura.
* **Modelo de policiamento**: também conhecido como policy model, este modelo prioriza detalhes importantes relacionados ao comportamento dos agentes da arquitetura, sendo aplicado sobre agentes que necessitam acessar diretamente os recursos da arquitetura. O modelo de policiamento se relaciona com outros aspectos, tais como políticas de segurança, gerenciamento e aplicações.

## Tecnologias utilizadas em Web Services

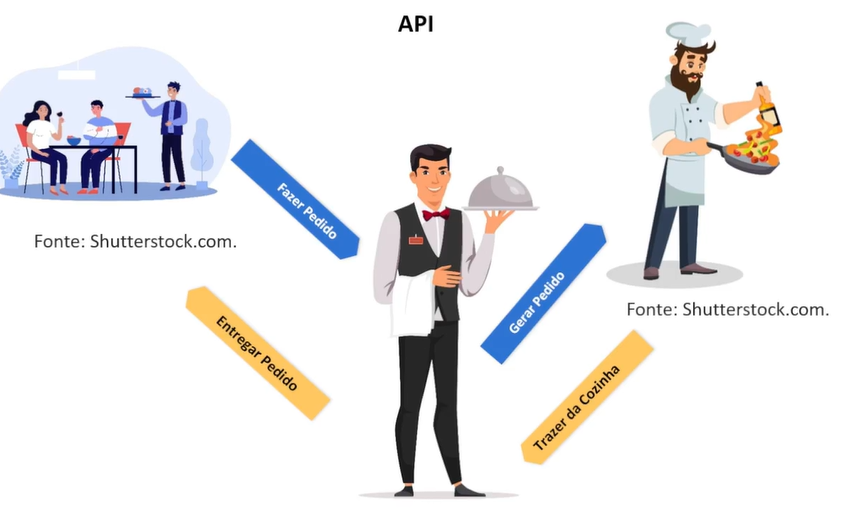
Para o bom funcionamento de um Web Service, são utilizadas algumas ferramentas, as quais auxiliam no processo de segurança, comunicação e administração. São exemplos delas: HTTP, XML, WSDL e UDDI. 9

* HTTP: protocolo responsável por transportar dados.
* XML: linguagem de marcação que facilita o compartilhamento de informações em Web Services.
* WSDL: descreve as interfaces de um Web Service e como as mensagens são formatadas quando são usados os protocolos. É baseada em XML.
* UDDI: permite que os comerciantes possam encontrar Web Services de forma rápida, fácil e dinâmica, além de interagir uns com os outros.
* SOAP: é uma arquitetura de Web Services.
* REST: é uma arquitetura de Web Services.

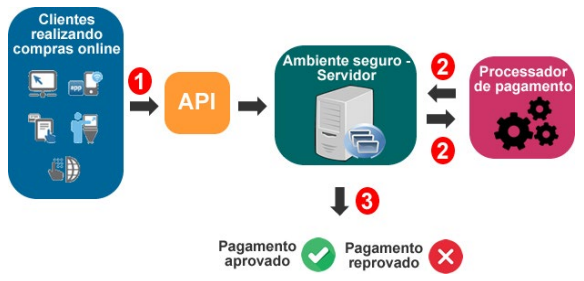




## Introdução aos APIs



A Application Programming Interface (Interface de Programação de Aplicativos), mais conhecida como API, é classificada como um conjunto de rotinas e padrões de programação que possuem o objetivo de acessar aplicativos de software ou plataformas baseados na web. De forma direta, APIs são várias ferramentas, podendo ser métodos de desenvolvimento e protocolos, as quais, juntas, facilitam as comunicações de programas e aplicações.

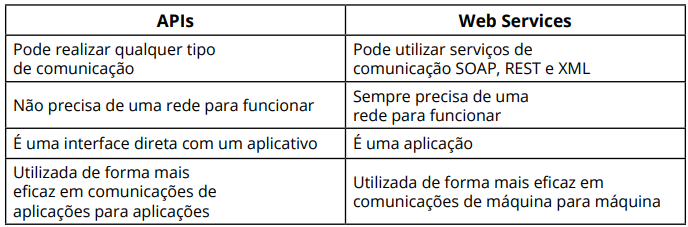


Um exemplo da atuação de API: um aplicativo de fotos desenvolvido para celular, no qual o usuário teria acesso à câmera através de uma API já existente, sem necessidade de criar uma interface de câmera; o aplicativo de mensagens instantâneas WhatsApp, que consegue integrar os contatos do telefone através da API; a rede social Instagram, que utiliza uma API para postar a mesma publicação no Facebook.

Para que uma API possa ser criada, é necessário possuir um sistema de fácil integração, segura, estável e com bom desempenho.

## Diferença entre APIS e Web Services

É correto afirmar que todo Web Service é uma API, mas nem toda API é um Web Service, porque tanto o Web Service quanto as APIs realizam a comunicação entre aplicações, porém a forma como são utilizados é totalmente diferente.



# Analisando projetos de arquiteturas de API Web

O desenvolvimento de qualquer tipo de Web Service deve ser muito bem estruturado, para que ele possa funcionar da maneira correta. Temos que lembrar que nem sempre todo Web Service é uma API, neste caso, uma API web é um Web Service.

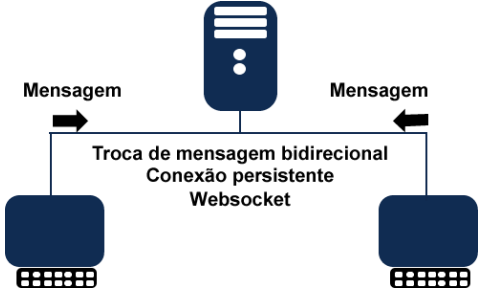
A construção de um projeto de arquitetura de uma API web não se limita à preocupação da utilização de URLs no protocolo de transporte HTTP, na transmissão de dados ou em cabeçalhos. Realizar a implementação de uma API web é ter a preocupação com o seu funcionamento, pois são muitos processos e elementos que precisam ser levados em consideração.

## Identificando o projeto de arquitetura de API web

Se você deseja construir uma API web, a primeira coisa que deve fazer é identificar qual seria a sua finalidade. Este processo é muito importante para que seu projeto possa ser implementado da maneira correta. Visualizando a construção de um projeto de arquitetura de uma API Web, podemos verificar, por exemplo, que, além do protocolo de transmissão de HTTP, é possível utilizar outros protocolos de comunicação, como o **WebSockets, o XMPP e o MQTT.**

Caso você não defina bem o seu projeto de estrutura de API web, pode ser que tenha problemas no andamento dele, porque cada protocolo é utilizado para um tipo de implementação. O protocolo HTTP é um dos principais protocolos de comunicação no meio web, porém, em algumas aplicações, ele não desempenha uma função eficaz, porque não realiza uma comunicação simultânea, o que acaba sobrecarregando um servidor web.

## WebSockets

O protocolo WebSockets atua tanto em navegadores quanto em servidores web, realizando uma comunicação bidirecional, ou seja, existe uma comunicação, um transporte de mensagem bilateral, realizado em tempo real. Este tipo de protocolo é muito utilizado em aplicações de jogos on-line, nos quais existem vários jogadores interagindo em tempo real, websites ou aplicativos de chat, links de conteúdos esportivos, reuniões e outros eventos que ocorrem ao vivo, assim como em aplicações que realizam atualizações de informações em tempo real, como redes sociais.

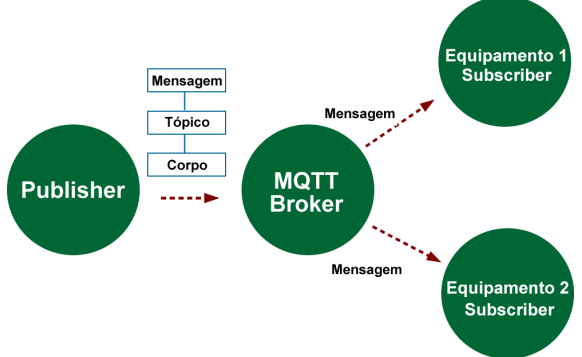
*É uma tecnologia que permite a comunicação em tempo real entre um navegador e um servidor.*



## XMPP

O protocolo XMPP é baseado em XML, sendo muito utilizado em aplicativos que realizam a comunicação de troca de mensagens instantâneas. Este protocolo tem como recursos a detecção de presença, realizando o controle de status de um usuário, verificando se ele está online ou off-line; permite que o próprio usuário defina seu status; permite chamada de vídeos e voz. A maior parte dos aplicativos de mensagens instantâneas que existem atualmente utiliza o protocolo XMPP, por exemplo, WebEx, GoToMeeting, Facebook Messenger, WhatsApp e Telegram.

## MQTT

MQTT é um protocolo de troca de mensagens entre máquinas muito utilizado na construção de APIs web que realizam a troca de pequenas mensagens. Logo, suas características principais são: protocolo muito leve e de baixo consumo de hardware. Ele utiliza a arquitetura publishsubscribe (publicação e assinatura). Este tipo de protocolo é muito utilizado em aplicações voltadas para IoT (internet das coisas), Arduino (plataforma que armazena componentes elétricos e programas) e na comunicação via bluetooth.

Dessa forma, fica clara a necessidade de identificar o tipo de projeto que você construirá, para poder utilizar uma arquitetura de API web que funcione para a finalidade que deseja. Se você deseja utilizar uma API web, pode ser que precise utilizar um desses protocolos citados, por isso, verifique as possibilidades da utilização deles em diferentes tipos de projetos de arquiteturas de APIs.

# Tecnologias para a implementação de uma API Web