**REST** e **SOAP**, são abordagens diferentes para transmissão de dados e que definem como as interfaces de programação de aplicações (APIs) são criadas, possibilitando a comunicação de dados entre aplicações.

**O que é SOAP?**

SOAP (**Simple Object Access Protocol**) é um protocolo com requisitos específicos, com o uso de XML e HTTP para a comunicação entre sistemas, permitindo a troca de informações entre aplicações em diferentes plataformas com arquiteturas, mecanismo de comunicação e formato de mensagem diferentes.

**O que é REST?**

REST (**Representational State Transfer**) é um conjunto de restrições de arquitetura que oferece uma implementação flexível, que define como os sistemas devem ser projetados para se comunicarem. As restrições do REST se basea no protocolo HTTP, que é utilizado como meio de comunicação entre os sistemas

Com essas regras, é possível desenvolver uma aplicação com interface claramente definida, rotinas padronizadas e facilmente representadas. Isso ajuda a simplificar a comunicação entre máquinas e usuários.

**REST X SOAP**

A principal diferença entre REST e SOAP, REST é um conjunto de restrições de arquitetura, enquanto SOAP é um protocolo específico. Isso significa que REST oferece mais flexibilidade e liberdade na implementação de uma API, enquanto o SOAP é mais rígido em suas exigências.

Muitos sistemas legados ainda usam SOAP, REST surgiu e tornou-se cada vez mais popular nos últimos anos, especialmente no desenvolvimento de APIs baseadas em web, devido à sua simplicidade e facilidade de uso. No entanto, ambos ainda são usados em muitos sistemas, dependendo do caso de uso e das preferências do desenvolvedor.

* API REST não possui um padrão oficial, já que é um estilo de arquitetura. API SOAP, possui um padrão oficial porque é um protocolo.
* REST usa vários padrões como HTTP, JSON, URL e XML, enquanto SOAP é baseada em HTTP e XML;
* Como REST implementa vários padrões, são necessários menos recursos e largura de banda em comparação com o SOAP que usa XML para a sua criação, resultando em um arquivo de tamanho grande;
* As maneiras pelas quais as duas APIs expõem as lógicas de negócios também são diferentes. REST aproveita a exposição da URL como @path (“/ WeatherService”), enquanto o uso da API SOAP de interfaces de serviços como @WebService;
* REST usa a linguagem de descrição de aplicações da Web e SOAP usa a linguagem de descrição de serviços da Web para descrever as funcionalidades oferecidas pelos serviços da web;
* REST é compatível com JavaScript e pode ser implementado facilmente. Já SOAP também é conveniente com JavaScript, mas não suporta uma implementação maior.

**Vantagens do REST sobre SOAP**

Nos últimos anos, desenvolvedores têm preferido utilizar REST para criar APIs. Não é difícil encontrar exemplos, especialmente porque grandes sites de mídias sociais disponibilizam APIs REST para que os desenvolvedores possam integrar suas aplicações às plataformas.

Essas APIs são fornecidas com documentação detalhada que contém todas as informações necessárias para extrair dados da API. Abaixo estão alguns benefícios:

* REST permite uma variedade maior de formatos de dados, enquanto SOAP permite somente XML;
* REST, juntamente com JSON (que funciona melhor com dados e oferece análise mais rápida), é geralmente considerado mais fácil de trabalhar;
* Devido ao JSON, REST oferece melhor suporte;
* REST oferece desempenho superior, principalmente por meio do armazenamento em cache de informações que não são alteradas nem dinâmicas;
* É o protocolo mais utilizado em grandes serviços como Facebook, Microsoft, Amazon e até mesmo o Google.

REST normalmente é mais rápido e consome menos largura de banda. Também é mais fácil de integrar com sites existentes, sem a necessidade de refatorar a infraestrutura do sistema. Isso permite que os desenvolvedores trabalhem mais rápido e gastem menos tempo reescrevendo um sistema do zero, podendo simplesmente adicionar novas funcionalidades.

**O que é RESTful?**

Muitos confundem os termos REST e RESTful, no entanto, ambos representam o mesmo princípio. Em outras palavras, sistemas que utilizam os princípios REST são chamados de RESTful.

Para uma API ser considerar RESTful deve atender aos padrões propostos pelo estilo de arquitetura REST, podendo ser caracterizada com uma API RESTful

Assim, quando uma API é do tipo RESTful, significa que esse sistema consegue aplicar os princípios propostos por Roy Fielding em sua tese REST.

Veja algumas restrições de arquiteturas que se caracterizam uma API RESTful:

* **Arquitetura cliente-servidor:** Baseada em recursos, em que as solicitações são feitas via protocolo HTTP. Isso implica em uma independência entre o cliente e o servidor, de forma que as mudanças feitas por cada um não afetam o outro imediatamente.
* **Comunicação stateless:** Em uma REST API, a comunicação entre cliente e servidor não deve armazenar informações entre solicitações. Cada solicitação deve conter todos os dados necessários para ser atendida, sem depender de informações armazenadas em outras sessões.
* **Cache:** Uma API REST deve permitir o armazenamento em cache de dados para otimizar as solicitações e respostas entre cliente e servidor.
* **Interface uniforme:** Uma interface uniforme é fundamental em uma API REST, pois permite o desenvolvimento independente da aplicação entre usuário e servidor. Ela oferece uma comunicação padronizada e a manipulação de recursos através de representações é uma das condições necessárias para a sua implementação.
* **Sistema de camadas:** Em um sistema, cada camada deve ter uma função específica e ser responsável por uma etapa diferente dos processos de requisição e resposta. Apesar de serem ordenadas hierarquicamente e separadas, as camadas interagem entre si.