



# Rest/Restful

Álamo Nunes  
Dyannah Hanna

**Antes de entender sobre a abordagem REST...**

**Como funciona a forma tradicional  
de construir e consumir serviços web?**

# Web Services

A Arquitetura de Web Services surgiu para permitir a interoperabilidade entre aplicações rodando em diferentes plataformas. Foi desenvolvido com base em um protocolo que encapsula as mensagens (SOAP – Simple Object Access Protocol) e em uma linguagem que descreve as interfaces dos serviços, conhecida como WSDL (Web Services Linguagem de descrição).

Porém, um grande problema era observado por usuários desse tipo de arquitetura: a perda de desempenho das aplicações. O que acontece é que, ao utilizar o protocolo SOAP, é gerada uma camada de abstração que envolve o protocolo HTTP – quando na verdade essa camada não precisaria existir.

**E é aí que entra a arquitetura REST...**

# REST

Concebido como uma abstração da arquitetura da web, trata-se de um conjunto de princípios e definições necessários para a criação de um projeto com interfaces bem definidas. A utilização da arquitetura REST, portanto, permite a comunicação entre aplicações.

...

Os Web Services que adotam REST são mais leves e perfeitos na busca da metodologia ágil. Outro diferencial é a flexibilidade, sendo possível escolher o formato que melhor se encaixa para as mensagens do sistema. Os mais utilizados, além do texto puro, são JSON e XML, dependendo da necessidade de cada momento.

...

A utilização da arquitetura REST tem tornado mais simples e eficientes muitas aplicações que antes eram implementadas de maneira tradicional – baseadas no protocolo SOAP.

As principais aplicações que se beneficiam da utilização desta arquitetura são aquelas nas quais cada requisição tem um custo relativamente alto, como por exemplo, serviços remotos acessados via celular e PDA's.

Enquanto SOAP é um protocolo de mensagens,

**REST é um estilo de arquitetura de software para sistemas distribuídos com hipermídia**



# No RESTful, são conhecidos como verbos os métodos do protocolo HTTP:

- POST (cria um novo recurso);
- GET (lê um recurso);
- PUT (atualiza um recurso);
- DELETE (deleta um recurso);
- HEAD (recupera o header da URI passada);
- OPTIONS (recupera os métodos que são possíveis);
- TRACE (recupera informações de debug).

# SOAP versus REST

As APIs REST são leves e ideais para contextos mais modernos, como a Internet das Coisas (IoT), desenvolvimento de aplicações mobile e computação sem servidor. Os serviços web SOAP oferecem segurança integrada e transações em conformidade que atendem a muitas necessidades empresariais, mas que também os deixam mais pesados. Além disso, muitas APIs públicas, como a do Google Maps, seguem as diretrizes REST.

# Referências

Arquitetura REST: Saiba o que é e seus diferenciais. TOTVS, 23 de mar. de 2020.

Disponível em: <<https://www.totvs.com/blog/developers/rest/>>.

Acesso em: 10 de nov. de 2020.

Arquitetura REST: Uma alternativa para construção de Serviços Web.

DEVMEDIA, 2009. Disponível em:

<<https://www.devmedia.com.br/arquitetura-rest-uma-alternativa-para-construcao-de-servicos-web/15150>>. Acesso em: 10 de nov. de 2020.

Desenvolvendo Web Services RESTFUL utilizando a API JAX-RS 2.0 e Jersey.

DEVMEDIA, 2013. Disponível em:

<<https://www.devmedia.com.br/desenvolvendo-web-services-restful-utilizando-a-api-jax-rs-2-0-e-jersey/27141>>. Acesso em: 10 de nov. de 2020.

What's the difference between SOAP and REST. Red Hat. Disponível em: <https://www.redhat.com/pt-br/topics/integration/whats-the-difference-between-soap-rest>>. Acesso em: 10 de nov. de 2020.