

1. Qual a principal diferença entre REST e SOAP?

REST e SOAP são duas abordagens diferentes para a transmissão de dados online. Em específico, eles definem como as interfaces de programação de aplicações (APIs) são criadas, possibilitando a comunicação dos dados entre aplicações web. REST (Representational State Transfer) ou “transferência de estado representacional” é um conjunto de princípios de arquitetura.

Já SOAP (Simple Object Access Protocol) ou “protocolo de acesso a objetos simples” é um protocolo oficial mantido pela World Wide Web Consortium (W3C). A principal diferença é que SOAP é um protocolo e REST, não. Normalmente, uma API será baseada em REST ou SOAP, dependendo do caso de uso e das preferências do desenvolvedor.

2. Em quais cenários SOAP ainda é utilizado?

- Sistemas bancários e financeiros
- Serviços de seguradoras
- Ambientes corporativos legados
- Integrações B2B que exigem alto nível de segurança, confiabilidade e transações complexas

Isso acontece porque SOAP possui recursos robustos como segurança (WS-Security), transações e contratos bem definidos (WSDL).

3. Quais são as vantagens e desvantagens de usar REST ao invés de SOAP?

REST se destaca por ser uma solução leve e ágil, especialmente por utilizar formatos como JSON, o que torna seu uso mais simples e acessível. É bastante eficiente em aplicações web e mobile, além de ser uma arquitetura flexível e escalável. No entanto, pode não ser a melhor escolha para cenários que exigem transações complexas, possui uma abordagem de segurança mais básica, geralmente limitada ao uso de HTTPS, e não conta com um contrato formal como o WSDL, o que pode resultar em menor padronização entre diferentes implementações.

4. O que é WS-Security e como ele se compara à segurança em APIs REST?

As APIs SOAP usam protocolos integrados chamados Web Services Security (WS Security). Esses protocolos definem um conjunto de regras orientado por confidencialidade e autenticação. As APIs SOAP são compatíveis com os padrões definidos pelos dois principais órgãos de padrões internacionais: a Organização para o Avanço de Padrões em Informação Estruturada (OASIS) e o Consórcio World Wide Web (W3C). Elas usam uma combinação de criptografia XML, assinaturas

XML e tokens SAML para verificar a autenticação e a autorização. Em geral, as APIs SOAP são apreciadas por ter medidas de segurança mais abrangentes.

APIs REST geralmente dependem de HTTPS, OAuth2, JWT e outras práticas modernas de segurança, mas não tem um padrão tão robusto e específico quanto o WS-Security.

5. Explique o modelo de maturidade de Richardson.

O Modelo de Maturidade de Richardson é uma maneira de classificar a complexidade e a maturidade de uma API RESTful. Ele descreve quatro níveis de maturidade, que ajudam desenvolvedores e arquitetos a entender e melhorar o design de suas APIs. Este modelo é frequentemente utilizado para avaliar o quanto uma API se alinha com os princípios REST (Representational State Transfer) e como ela pode evoluir para um design mais robusto e escalável.

Nível	Descrição
0	Um único endpoint (RPC, sem REST)
1	Usa recursos (URLs diferentes para cada entidade)
2	Usa métodos HTTP corretamente (GET, POST, etc.)
3	HATEOAS (Hypermedia As The Engine Of Application State) - a resposta da API traz links para ações possíveis

6. O que é GraphQL? Pesquisa e relacione com os conceitos vistos em aula.

- GraphQL é uma linguagem de consulta criada pelo Facebook para APIs.
- Ao invés de ter vários endpoints como REST, o cliente envia uma query e recebe exatamente os dados que pediu.
- Diferente de SOAP e REST, não existe overfetching ou underfetching (dados a mais ou faltando).
- Mais eficiente para front-end modernos (React, mobile), reduz chamadas desnecessárias.

Enquanto REST expõe endpoints e SOAP define operações, o GraphQL permite ao cliente definir o que ele precisa e como.