



## Exercício 3 - Rest e SOAP

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
DE SÃO PAULO**

**CÂMPUS GUARULHOS**

**Programação Orientada a Objetos - Prof. Giovani**

**Gabriel Vinícius Rocha Barboza, GU3042715**

### **1. Qual a principal diferença entre REST e SOAP?**

REST pode utilizar JSON e XML de forma simples e flexível, enquanto SOAP só utiliza XML sendo mais estruturado e verboso.

### **2. Em quais cenários SOAP ainda é utilizado?**

Sistemas legados e aplicações que requerem bastante segurança.

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

REST possui alto desempenho, maior escalabilidade e utiliza cache, porém a segurança depende de implementações.

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

WS-Security é um padrão de segurança que protege os dados no nível da aplicação, sendo mais integrado e sem depender de medidas externas para segurança como APIs REST.

#### **5. Explique o modelo de maturidade de Richardson.**

O modelo de maturidade de Richardson classifica em 4 níveis o quão RESTful é uma API.

Nível 0: Usa HTTP apenas como transporte, sem URIs ou verbos adequados. (Ex: SOAP)

Nível 1: Introduz recursos identificáveis por URLs, mas sem uso correto de métodos HTTP.

Nível 2: Usa recursos e verbos HTTP corretamente (GET, POST, PUT, DELETE, PATCH).

Nível 3 (HATEOAS): Respostas da API incluem links dinâmicos para ações relacionadas, permitindo descoberta sem documentação rígida.

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

GraphQL é uma linguagem de consulta para APIs que permite ao cliente especificar exatamente quais dados deseja receber. Ele resolve problemas como over-fetching (dados excessivos) e under-fetching (dados insuficientes), tornando as APIs mais eficientes e flexíveis.

Na aula, vimos conceitos como métodos HTTP e recursos em REST, que usam múltiplos endpoints para diferentes operações. O GraphQL simplifica isso ao usar um único endpoint para todas as consultas e mutações, permitindo buscar ou modificar dados de várias fontes em uma única chamada.

Além disso, o schema do GraphQL define os tipos de dados disponíveis e gera documentação dinâmica, facilitando o desenvolvimento e a evolução da API sem criar novas versões, algo que REST não faz automaticamente. Isso

conecta com o conceito de HATEOAS, mas de forma mais prática e declarativa.

GraphQL: API intuitiva e flexível para descrever requisitos - Zup

<https://zup.com.br/blog/graphql-api-para-descrever-requisitos>

GraphQL vs REST: como funciona e diferenças - Locaweb

<https://www.locaweb.com.br/blog/temas/codigo-aberto/graphql-vs-rest/>