

## Lista de Exercícios

1. Explique o conceito de Web Service segundo a definição da Gartner.

Um **Web Service** é como um “serviço” disponível na internet que permite que **diferentes sistemas conversem entre si**, mesmo que sejam de tecnologias diferentes.

2. Explique a função do protocolo SOAP e descreva a composição de seu envelope.

O **SOAP** (Simple Object Access Protocol) é um protocolo de comunicação baseado em XML que serve para **trocar informações entre aplicações pela internet**. Ele garante que a mensagem enviada de um sistema para outro seja **padronizada, estruturada e independente da linguagem de programação ou plataforma**.

### Estrutura do Envelope SOAP

O **Envelope** é a estrutura principal da mensagem SOAP, escrita em XML. Ele envolve toda a comunicação e é formado por quatro partes:

1. **Envelope**
  - o Elemento raiz.
  - o Define que aquele documento é uma mensagem SOAP.
2. **Header (opcional)**
  - o Contém **informações adicionais** sobre a mensagem, como autenticação, segurança, controle de transações, etc.
3. **Body**
  - o Parte principal, onde ficam os **dados da requisição ou resposta**.
  - o Exemplo: se você pede a cotação do dólar, o Body carrega esse pedido e a resposta.
4. **Fault (opcional, dentro do Body)**
  - o Usado para **relatar erros** na comunicação.
  - o Exemplo: se o serviço não estiver disponível, a mensagem de erro vem aqui.

3. Quais são os três elementos principais de uma arquitetura de Web Service?

### 1. Provedor de Serviço (Service Provider)

- É quem **implementa e disponibiliza** o Web Service.
- Hospeda o serviço na internet e responde às requisições dos clientes.
- Exemplo: um servidor que oferece a cotação do dólar via Web Service.

### 2. Requisitante de Serviço (Service Requester / Client)

- É quem **consome o Web Service**, ou seja, o sistema ou aplicação que faz a chamada ao serviço.
- Pode ser um site, aplicativo mobile ou outro sistema que precisa dos dados.
- Exemplo: um app de finanças que consulta o serviço de cotação.

### 3. Registro de Serviço (Service Registry)

- É como um **catálogo** ou **lista** onde os Web Services podem ser **publicados e descobertos**.
- Normalmente usa UDDI (Universal Description, Discovery, and Integration).
- Exemplo: um diretório onde o provedor registra o serviço para que o cliente saiba como acessá-lo.

4. O que é WSDL e quais são os elementos básicos que o compõem?

O **WSDL** (*Web Services Description Language*) é um **documento em XML** que descreve detalhadamente **o que um Web Service faz e como utilizá-lo**.

5. Qual é a função do UDDI em Web Services?

O **UDDI** (*Universal Description, Discovery and Integration*) é um **padrão de diretório** para Web Services.

#### Principais funções do UDDI:

1. **Publicar** – o provedor registra seu Web Service no diretório.
  2. **Descobrir** – o cliente procura no diretório por um serviço que atenda sua necessidade.
  3. **Integrar** – após encontrar, o cliente usa as informações do WSDL para consumir o serviço.
6. O que é REST e quem o criou?

**REST** (*Representational State Transfer*) é um **estilo de arquitetura** para comunicação entre sistemas distribuídos na web. Ele define um conjunto de **princípios e restrições** para criar APIs simples, escaláveis e baseadas no protocolo **HTTP**.

O **REST foi criado por Roy Fielding**, em sua tese de doutorado na **Universidade da Califórnia, Irvine (2000)**.

7. Qual é a função de um recurso (Resource) no REST e como ele é identificado?






No **REST**, um **Recurso** representa **qualquer entidade ou dado que pode ser manipulado** pelo sistema. É o **objeto central da API** — pode ser um usuário, produto, pedido, documento, etc.

Cada recurso é identificado de forma **única por uma URL (URI)**.

8. O que é virtualização e quais são seus principais tipos de recursos virtualizados?

A **virtualização** é uma tecnologia que permite **criar versões virtuais de recursos físicos**, como servidores, sistemas operacionais, redes ou armazenamento. Em vez de depender de um hardware físico único para cada aplicação, a virtualização permite dividir ou simular recursos, tornando o uso da infraestrutura mais **eficiente, flexível e escalável**.

## Principais tipos de recursos virtualizados






1. **Virtualização de Servidores** 
  - Permite rodar **vários servidores virtuais** em um único servidor físico.
  - Cada servidor virtual pode ter seu próprio sistema operacional e aplicações.
2. **Virtualização de Sistemas Operacionais** (ou de Desktop) 
  - Cria **máquinas virtuais** que simulam sistemas operacionais completos.
  - Usuários podem rodar diferentes SOs (ex.: Windows e Linux) no mesmo computador.
3. **Virtualização de Armazenamento** 
  - Une diversos dispositivos de armazenamento em um único sistema lógico.
  - Facilita a gestão e aumenta a disponibilidade dos dados.
4. **Virtualização de Rede** 
  - Separa a infraestrutura de rede física em **redes virtuais** independentes.
  - Exemplo: VLANs e SDN (Software Defined Networking).
5. **Virtualização de Aplicações** 
  - Permite executar aplicativos em ambientes virtuais, sem precisar instalá-los diretamente no sistema do usuário.
  - Exemplo: Citrix, VMware App.

## 9. O que é um container Docker e o que ele empacota?

Um **container Docker** é uma **unidade leve e portátil** que empacota uma aplicação junto com tudo o que ela precisa para rodar (bibliotecas, dependências, configurações).

Ele garante que a aplicação funcione da **mesma forma em qualquer ambiente** — seja no seu computador, em um servidor ou na nuvem.

Um container empacota:

1. **Código da aplicação**  – o programa em si.
2. **Dependências**  – bibliotecas, frameworks, pacotes necessários.
3. **Configurações**  – variáveis de ambiente, arquivos de configuração.
4. **Runtime**  – tudo que é preciso para executar o código (ex.: Java, Node.js, Python).
5. **Sistema de arquivos isolado**  – garante que o container seja independente do host.

