



---

# SISTEMAS DISTRIBUÍDOS

---

SOA, SOAP e Arquitetura REST



7 DE MARÇO DE 2026  
PROF. DR. LINCOLN SPOSITO  
Universidade São Judas Tadeu – São Bernardo do Campo

## Sumário

SOA, SOAP e a Ascensão do REST: O Contrato como Base da Distribuição .....	2
1. SOA: O Paradigma da Reutilização de Negócio .....	2
2. SOAP: O Protocolo do Rigor Corporativo .....	3
3. REST: A Simplificação via Protocolo HTTP .....	3
4. Affordance Visual: O Fluxo de uma Requisição RESTful.....	4
Exercícios de Fixação: SOA, SOAP e REST .....	5
Parte I: Múltipla Escolha .....	5
Parte II: Exercícios Descritivos (Análise Técnica) .....	5
Gabarito e Respostas Esperadas.....	6
Parte I: Múltipla Escolha .....	6
Parte II: Descritivos.....	6

# SOA, SOAP e a Ascensão do REST: O Contrato como Base da Distribuição

Acesso o portal acadêmico, [clique aqui](#)

*Nota: Esta aula foi estruturada para ser o guia definitivo sobre a transição das arquiteturas baseadas em serviços (Service-Oriented Architecture) para a onipresença das APIs modernas (REST). O material integra a visão de gestão do Dr. em Administração com o rigor técnico do Analista de Sistemas, focando no que o mercado exige: escalabilidade, baixo acoplamento e contratos de interface robustos.*

Na evolução dos sistemas distribuídos, a necessidade de integrar sistemas heterogêneos — que rodam em linguagens, sistemas operacionais e infraestruturas distintas — levou ao surgimento da **Arquitetura Orientada a Serviços (SOA)**. Conforme **Coulouris et al. (2013)**, SOA não é uma tecnologia, mas um estilo arquitetural onde as funcionalidades de um sistema são disponibilizadas como serviços independentes, que se comunicam através de protocolos de rede.

## 1. SOA: O Paradigma da Reutilização de Negócio

A grande contribuição da SOA, sob a ótica de **Monteiro (2020)**, foi permitir que a lógica de negócio fosse encapsulada. Se o **Colégio Mirassol** possui um serviço de "Consulta de Mensalidade", este pode ser consumido tanto pelo Portal do Aluno quanto por um App mobile ou um sistema de auditoria externa, sem que cada um precise reimplementar a lógica de cálculo.

**Quadro 1: Os Pilares da SOA**

Pilar	Descrição Técnica	Impacto no Projeto (Affordance)
<b>Desacoplamento</b>	O cliente não precisa saber como o serviço é implementado internamente.	Reduz o "efeito cascata": mudar o banco de dados do serviço não quebra o cliente.
<b>Contrato de Serviço</b>	O serviço adere a um documento de comunicação (WSDL ou Swagger).	Garante que ambas as partes falem a mesma "língua" técnica.
<b>Composicionalidade</b>	Serviços pequenos podem ser combinados para criar processos complexos.	Permite orquestrar um fluxo de "Matrícula" chamando os serviços de "Financeiro" e "Secretaria".

## 2. SOAP: O Protocolo do Rigor Corporativo

O **SOAP (Simple Object Access Protocol)** foi, por anos, o padrão de ouro para SOA. Ele é baseado exclusivamente em **XML** e é conhecido por sua verbosidade e rigor. Segundo **Tanenbaum & Van Steen (2007)**, o SOAP brilha em ambientes corporativos que exigem alta segurança (WS-Security) e transações complexas (WS-AtomicTransaction).

### Anatomia de uma Mensagem SOAP:

- **Envelope:** A raiz do documento que define o início e o fim da mensagem.
- **Header (Cabeçalho):** Contém informações de controle, autenticação e roteamento.
- **Body (Corpo):** Onde reside o "payload" ou a carga útil (ex: os dados do aluno).
- **Fault:** Uma seção opcional para reportar erros de processamento.

---

## 3. REST: A Simplificação via Protocolo HTTP

Enquanto o SOAP é um protocolo, o **REST (Representational State Transfer)** é um estilo arquitetural que utiliza as propriedades nativas do protocolo HTTP. Proposto por Roy Fielding em sua tese de doutorado, o REST foca em **Recursos**, identificados por URIs (Uniform Resource Identifiers).

No REST, as operações são mapeadas diretamente nos **Verbos HTTP**:

- GET: Recuperar um recurso (ex: GET /alunos/123).
- POST: Criar um novo recurso.
- PUT/PATCH: Atualizar um recurso existente.
- DELETE: Remover um recurso.

### Tabela Comparativa: SOAP vs. REST (A Escolha do Analista)

Característica	SOAP (Rigidez e Segurança)	REST (Agilidade e Web)
Formato de Dados	Apenas XML.	JSON (mais leve), XML, HTML, Texto.
Protocolo	Independente (HTTP, SMTP, TCP).	Estritamente HTTP/HTTPS.
Estado	Pode ser <i>Stateful</i> (guarda estado).	Obrigatoriamente <i>Stateless</i> (sem estado).
Curva de Aprendizado	Alta (exige ferramentas complexas).	Baixa (usa a semântica natural da Web).
Cache	Difícil de implementar.	Nativo e eficiente (essencial para escala).

## 4. Affordance Visual: O Fluxo de uma Requisição RESTful

Imagine o sistema de perícia judicial do **Dr. Lincoln**. Para buscar um laudo, o fluxo segue este padrão:

1. **Cliente (Navegador/App):** Envia GET `https://sposito.pro/api/laudos/45`.
  2. **Interface (Contrato):** O servidor valida se o ID 45 existe e se o usuário tem permissão.
  3. **Processamento:** O componente de negócio busca no banco de dados.
  4. **Resposta:** O servidor retorna um código **HTTP 200 OK** e o corpo em **JSON**. Se não existir, retorna **HTTP 404 Not Found**.
-

## Exercícios de Fixação: SOA, SOAP e REST

### Parte I: Múltipla Escolha

**1. Por que o REST é considerado "Stateless" (sem estado), de acordo com a teoria de Fielding e reforçado por Monteiro (2020)?**

- A) Porque o servidor armazena todos os dados do usuário em memória para a próxima requisição.
- B) Porque cada requisição do cliente deve conter todas as informações necessárias para ser processada, sem depender de contexto armazenado no servidor.
- C) Porque o protocolo HTTP não permite o envio de senhas.
- D) Porque ele funciona apenas quando o servidor está desligado.

**2. Em um cenário de integração entre um Banco Central e um Banco Comercial, onde a segurança extrema e o suporte a transações complexas são vitais, qual tecnologia seria mais recomendada pela bibliografia de Tanenbaum?**

- A) REST com JSON simples.
- B) SOAP com extensões WS-Security.
- C) Compartilhamento de arquivos TXT via FTP.
- D) Chamadas diretas de banco de dados via porta 3306.

**3. Qual é a principal vantagem do formato JSON sobre o XML em sistemas distribuídos de alta performance?**

- A) O JSON é mais seguro contra vírus.
- B) O JSON é mais leve (menos caracteres), reduzindo a latência de rede e o custo de parsing.
- C) O XML não é aceito por navegadores modernos.
- D) O JSON permite criar interfaces gráficas automaticamente.

---

### Parte II: Exercícios Descritivos (Análise Técnica)

**1. O Paradigma do Verbo:** Um desenvolvedor júnior criou uma API onde para deletar um usuário ele usa: POST /excluirUsuario?id=10. Explique por que esta abordagem viola os princípios do REST e como você, como Analista de Sistemas, corrigiria essa rota seguindo o padrão correto.

**2. A Visão de Negócio (SOA):** Como a adoção de uma arquitetura SOA pode auxiliar uma instituição de ensino (como o Colégio Mirassol) a reduzir custos de TI a longo prazo? Utilize os conceitos de *reutilização de serviços* e *interoperabilidade* para fundamentar sua resposta.

---

# Gabarito e Respostas Esperadas

## Parte I: Múltipla Escolha

1. **B** (A natureza stateless permite que qualquer servidor de um cluster processe a requisição, facilitando a escalabilidade horizontal).
2. **B** (O SOAP foi desenhado com o "Enterprise" em mente, suportando protocolos de transação e segurança que o REST básico não cobre nativamente).
3. **B** (Menos bytes na rede significam menor tempo de resposta, fator crítico em SD).

## Parte II: Descritivos

**1. Resposta Esperada:** A abordagem viola o REST porque utiliza o verbo POST (destinado a criação) para uma exclusão e usa uma ação no nome da URI (/excluirUsuario). No REST, a URI deve identificar o **Recurso** (substantivo) e o verbo HTTP deve indicar a **Ação**. A correção seria: DELETE /usuarios/10.

**2. Resposta Esperada:** SOA permite que o colégio crie serviços atômicos (ex: "Serviço de Login"). Quando o colégio decidir lançar um novo portal de notas ou uma catraca inteligente, ele não precisará programar o login novamente, apenas consumirá o serviço já existente. Isso gera economia de horas de desenvolvimento, facilita a manutenção centralizada e garante a consistência dos dados entre diferentes departamentos.