

Universidade Federal do Piauí - UFPI

Campus Senador Helvídio Nunes Barros - CSHNB

Sistemas de Informação - Sistemas Distribuídos - prof. Rayner Gomes

Aluna: Vandirleya Barbosa da Costa

Tema: Arquiteturas

2º Lista de Exercícios de Fixação

Atenção: Para todas as perguntas, cite o *site*, ou o material utilizado como fonte da pesquisa. Como uma atividade de formação intelectual, não copie e cole, o intuito não é saber sua capacidade de apertar CTRL-C e CTRL-V, mas sua capacidade de resumir, explicar e transmitir um novo conteúdo.

1. O que é *web service*? Qual a motivação para o uso de *web services*?

Um *web service* é um sistema de software projetado para suportar a interoperabilidade entre diferentes máquinas na rede. Ele permite que aplicações se comuniquem e compartilhem dados e serviços entre si, independentemente das plataformas e linguagens de programação utilizadas [1]. A motivação para o uso de *web services* inclui a necessidade de integração de sistemas heterogêneos, a reutilização de funcionalidades existentes e a facilitação da comunicação entre aplicações distribuídas[2].

[1] - [Web Services Tutorial: Introdução às tecnologias Web Services](#).

[2] - [Web service – Wikipédia, a enciclopédia livre](#).

2. Os mecanismos envolvidos na comunicação em um *web service* geralmente utilizam WSDL, SOAP, e XML schemas. Faça um resumo de cada um destes 3 mecanismos, em sua resposta deixe claro: o que é, para que, estrutura, principais características.

- WSDL (Web Services Description Language): É uma linguagem baseada em XML usada para descrever os serviços web e como acessá-los. Ele define a localização do serviço e os métodos disponíveis, permitindo que os clientes saibam como interagir com o serviço [1].
- SOAP (Simple Object Access Protocol): É um protocolo de comunicação baseado em XML que permite a troca de informações estruturadas entre aplicações na web. Ele define uma maneira de empacotar mensagens que podem ser transmitidas via HTTP, SMTP ou outros protocolos [1].
- XML Schemas: São usados para definir a estrutura e os tipos de dados que podem

ser usados em documentos XML. Eles garantem que os dados trocados entre serviços web sejam válidos e sigam um formato predefinido [1].

[1] - [Web Services Tutorial: Introdução às tecnologias Web Services](#).

3. O que é REST? Em relação aos aspectos de *software*, qual a proposição do REST (em sua resposta adicione o conceito de *stateless*).

REST (Representational State Transfer) é um estilo de arquitetura para sistemas distribuídos, que utiliza o protocolo HTTP para comunicação. Ele se baseia em recursos identificados por URIs e usa métodos HTTP padrão (GET, POST, PUT, DELETE) para operações. A proposição do REST é simplificar a comunicação entre sistemas, tornando-a mais eficiente e escalável. O conceito de *stateless* significa que cada requisição do cliente para o servidor deve conter todas as informações necessárias para entender e processar o pedido, sem depender de um estado armazenado no servidor [1].

[1] - [Web Services Tutorial: Introdução às tecnologias Web Services](#).

4. O que é SOA (Service Oriented Architecture)? Qual sua motivação? Qual problema ele propõe resolver?

SOA (Service Oriented Architecture) é um estilo de arquitetura de software que utiliza serviços como unidades básicas de construção. A motivação para SOA é promover a reutilização de serviços, facilitar a integração de sistemas e melhorar a flexibilidade e agilidade dos processos de negócios. SOA propõe resolver problemas de integração complexa e de manutenção de sistemas legados, permitindo que diferentes serviços se comuniquem de maneira padronizada e independente [1].

[1] - [Web Services Tutorial: Introdução às tecnologias Web Services](#).

5. No contexto de SOA o que é acoplamento? Por que o SOA busca um baixo acoplamento? Já que buscamos baixo acoplamento, considerando um sistema distribuído é interessante não ter acoplamento?

Acoplamento refere-se ao grau de dependência entre componentes de um sistema. No contexto de SOA, busca-se um baixo acoplamento para aumentar a flexibilidade e a capacidade de manutenção dos sistemas. Um baixo acoplamento permite que

os serviços sejam modificados ou substituídos sem afetar outros serviços, facilitando a evolução e a escalabilidade do sistema [1]. Em sistemas distribuídos, um acoplamento muito baixo pode ser desejável, mas é importante garantir que os serviços ainda possam se comunicar de maneira eficiente.

[1] - [Web Services Tutorial: Introdução às tecnologias Web Services](#).

6. **Será que existem similaridades com SOA e POO? Você consegue apontar algumas? Na disciplina de POO o professor geralmente menciona que no desenvolvimento buscamos alta coesão e baixo acoplamento! Explique esta afirmação.**

Tanto SOA quanto POO buscam alta coesão e baixo acoplamento. Em POO, alta coesão significa que os métodos de uma classe estão fortemente relacionados e trabalham juntos para realizar uma tarefa específica. Baixo acoplamento significa que as classes têm poucas dependências entre si, facilitando a manutenção e a reutilização. Em SOA, os serviços são projetados para serem coesos, realizando funções de negócio específicas, e fracamente acoplados, permitindo que sejam reutilizados e modificados independentemente [1].

[1] - [Web Services Tutorial: Introdução às tecnologias Web Services](#).

7. **Após saber o que é SOA. Pesquise como:**

a. Quais serviços são relacionados?

Serviços relacionados são aqueles que colaboram para realizar um processo de negócio completo, como serviços de pagamento, autenticação e envio de notificações [1].

b. serviços se comunicam?

Os serviços se comunicam usando protocolos padrão como HTTP, SOAP ou REST, e trocam mensagens em formatos como XML ou JSON [1].

c. Como encontrar um serviço?

Serviços podem ser encontrados usando registries como UDDI (Universal Description, Discovery, and Integration), que permitem a descoberta dinâmica de serviços disponíveis na rede [1].

[1] - [Web Services Tutorial: Introdução às tecnologias Web Services](#).