Estratégia Concursos

Aula 01

Engenharia de Software p/ Prefeitura de Limeira-SP (Téc de Suporte em Informática) - Pós-Edital

Autor:

Diego Carvalho, Equipe Informática e TI, Fábio Alves, Fernando Pedrosa Lopes

08 de Dezembro de 2019

SUMÁRIO	PÁGINA
Apresentação	01
- Arquitetura Orientada a Serviços (SOA)	02
Lista de Exercícios Comentados	59
Gabarito	72

ARQUITETURA ORIENTADA A SERVIÇOS (SOA)

Nesse momento da aula, vamos falar sobre Service Oriented Architecture (SOA)! Antes de adentrarmos nesse tema, é importante que vocês compreendam os conceitos de arquitetura e de serviço. No contexto de engenharia de software, a arquitetura é a organização ou a estrutura dos componentes significativos do sistema de software que interagem por meio de interfaces. *Simples, não é?*

Já um serviço é um mecanismo que permite acessar um conjunto de recursos, no qual o acesso é fornecido por meio de uma interface descrita e exercitada consistentemente de acordo com restrições e políticas especificadas pela descrição do serviço. É oferecido por uma entidade (Provedor de Serviços) para uso de terceiros (Consumidor de Serviços), mas há um detalhe!

Não há necessidade de esses terceiros conhecerem o provedor de serviços, podendo inclusive fazer uso do serviço de forma a extrapolar o escopo original concebido pelo provedor. Aliás, é importante salientar que esses serviços são agnósticos! *Como é, professor? É religião?* Não, esse é o termo utilizado para demonstrar que a lógica de um serviço não depende de outras partes.

De acordo com Thomas Erl:

> *"Dentro de uma solução orientada a serviços, as unidades de lógica (serviços) encapsulam funcionalidades não específicas a nenhum aplicativo ou processo de negócio. Esses serviços são classificados como ativos de tecnologia da informação agnósticos e reusáveis. Serviços agnósticos fornecem um intervalo de funcionalidades genéricas. Qualquer serviço agnóstico pode, portanto, ser adaptado inúmeras vezes para que seja possível automatizar diferentes processos de negócio como parte de diferentes soluções orientadas a serviços. Um serviço é agnóstico quando sua lógica é independente de processos de negócio, aplicativos ou tecnologias proprietárias. Quanto mais agnóstico for um serviço, mais genéricas são suas capacidades. Portanto, serviços agnósticos têm maior potencial de reúso".*

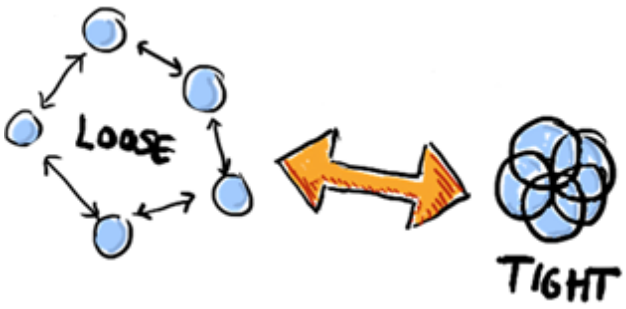
Galera, basta pensar em qualquer empresa de prestação de serviço. Existe um provedor que fornece algum tipo de serviço e o consumidor que consome o serviço fornecido. Eu, por exemplo, contrato uma empresa e pago por um serviço de

fornecimento de energia elétrica para minha casa. *Eu lá quero saber sobre a geração, transmissão e distribuição de energia?* Claro que não, eu quero é a energia!

Eu não ligo para como eles vão fazer essa energia aparecer, eu só quero que ela sempre esteja disponível na minha casa. Thomas Erl destaca também outros benefícios: serviços são reutilizáveis; compartilham um contrato formal; possuem baixo acoplamento; abstraem a lógica; são capazes de se comporem; são autônomos; evitam alocação de recursos por longos períodos; entre outros.

Agora que já sabemos o que é uma arquitetura e o que é um serviço, podemos juntá-los! A OASIS[1] define Arquitetura Orientada a Serviços como um paradigma para organização e utilização de recursos distribuídos que estão sob o controle de diferentes domínios proprietários, permitindo que funcionalidades implementadas sejam disponibilizadas na forma de serviços fracamente acoplados.

Quando falamos em serviços fracamente acoplados, estamos querendo dizer que os serviços são independentes uns dos outros, i.e., se houver uma mudança na implementação de um serviço, em nada afetará o restante. Vocês verão esse conceito de fraco acoplamento dezenas, talvez centenas, de vezes em concursos – sugiro memorizá-lo! Abaixo segue uma lista de definições de SOA que eu já encontrei em provas...



O QUE SOA É...
Um conjunto de princípios e melhores práticas para implementação e execução de processos de negócio automatizados em ambientes de tecnologia da informação heterogêneos.
Uma forma de aproximar a linguagem do negócio e da tecnologia da informação, facilitando a integração de ambientes corporativos por meio de serviços.
Um meio para organizar as soluções que promove o reúso, o crescimento e a interoperabilidade.
Uma abordagem distribuída (não-monolítica) para integração de arquiteturas baseadas no conceito de serviço.
Uma abordagem arquitetural corporativa que permite a criação de serviços de negócio interoperáveis, que podem ser reutilizados e compartilhados entre aplicações e empresas.

[1] OASIS (Organization for the Advancement of Structured Information Standards) é um consórcio global que conduz o desenvolvimento, convergência e adoção de padrões para e-business e web services.

É muito importante visualizar e posicionar a Arquitetura Orientada a Serviços como um modelo de arquitetura que é agnóstico a qualquer plataforma de tecnologia. Ao fazer isso, uma empresa tem a liberdade para perseguir continuamente os objetivos estratégicos associados à computação orientada a serviços, aproveitando os avanços tecnológicos futuros.

No mercado atual, a plataforma de tecnologia mais associada com a realização de Arquitetura Orientada a Serviços são os Web Services. Claro, existem também outras tecnologias, tais como: DCOM, CORBA, RPC, DDS, WCF, etc. Todas elas podem ajudar a implementar a arquitetura orientada a serviços, na medida em que essa arquitetura é uma abordagem, um paradigma, uma forma de organização.

SOA: MOTIVAÇÃO PARA CRIAÇÃO

Pessoal, o desenvolvimento de aplicações em ambientes corporativos ganhou, com o tempo, proporções que não poderiam ser previstas a curto prazo. Esse crescimento todo desordenado criou uma colcha de retalhos, em que cada aplicação era desenvolvida para ligar apenas dois pontos do sistema, sem preocupação com as consequências em outras partes do sistema.

Para entender isso melhor, peço que acompanhem comigo uma pequena história sobre a justificativa para implementação de arquitetura orientada a serviços. Imaginem que João trabalha em um banco e, por conta de novas demandas de negócio, ele precisa de dados específicos, mas esses dados estão espalhados em diferentes aplicações no sistema do banco.

Um programador, então, escreve exclusivamente para João uma aplicação que recupera os dados que ele deseja de outras três aplicações A, B, C e escreve todos esses dados em uma planilha. Como todo usuário, João é extremamente exigente! Ele sabe que precisará desses dados todos os dias de manhã, logo ele não quer ter que gerá-lo na mão, ele quer que a aplicação o faça automaticamente.

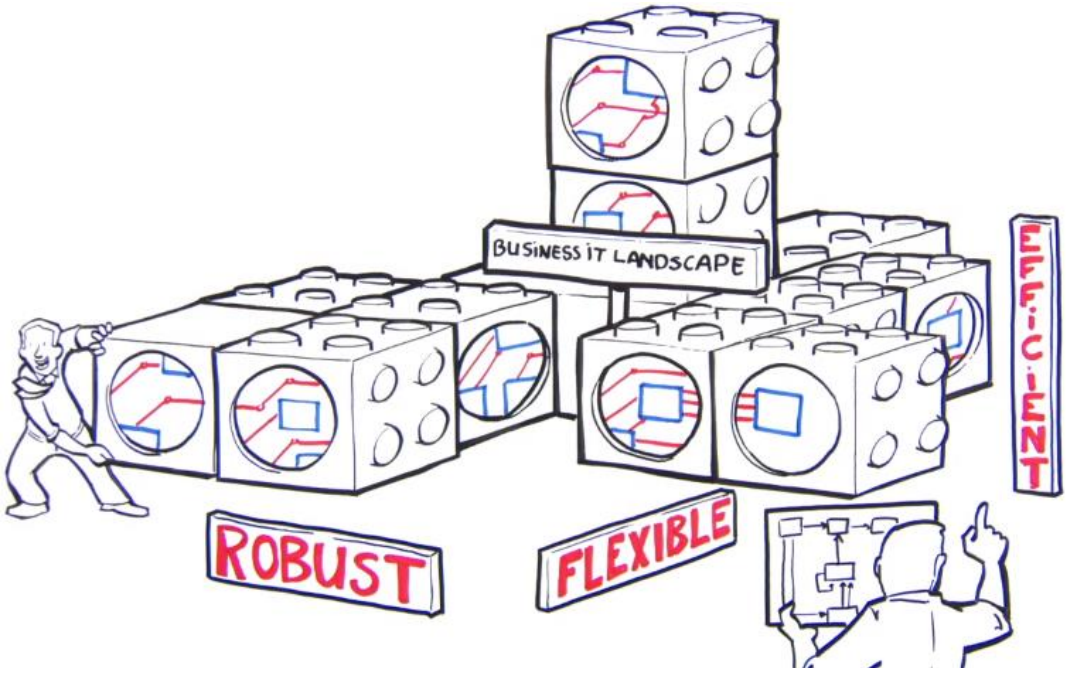
João chama, então, um gerente de infraestrutura para automatizar a geração de sua planilha. O gerente cria um *job* para automatizar pela manhã a criação das planilhas do João. No entanto, percebam que agora nós temos um novo fluxo de aplicação criado especificamente para o tal do João. E mais, esse fluxo depende de três outras aplicações do banco.

Vocês já viram o tamanho de um banco? João não é o único que quer melhorar sua eficiência. Multipliquem essa situação por algumas centenas e nós teremos uma Arquitetura de Bola de Pelo (Hairball Architecture)[2]. Em outras palavras, temos centenas de pelos indo e voltando em tudo que é direção. Ninguém tem controle e ninguém consegue rastrear absolutamente nada.

Imaginem agora que nós temos que substituir uma aplicação imensa do banco porque ela está desatualizada e o fornecedor disse que, em dois anos, ele parará de oferecer suporte. Ninguém sabe quais aplicações dependem dessa aplicação. É uma rede sem fim de dependências que foram criadas uma de cada vez, todas fazendo sentido no momento de sua criação, mas que ficou caótica com o tempo.

[2] Alguns chamam de Spaghetti-Like Architecture, i.e., Arquitetura de Macarronada.

Torna-se impossível mudar algum aspecto sem afetar dezenas de outras coisas, por conta do alto acoplamento. Havia, então, a necessidade de mudar a visão de negócio sobre o parque tecnológico de uma organização de tal forma que as pessoas tenham liberdade para criar melhorias e inovações. Devemos ter uma arquitetura que seja simultaneamente robusta, eficiente e flexível.



Vocês puderam notar dois problemas claros da área de tecnologia da informação desse banco! Alto acoplamento, tendo em vista que os sistemas dependem excessivamente de serviços personalizados; e alta redundância, na medida em que há funcionalidades replicadas em diversos sistemas dentro do próprio banco. Galera, em um nível estratégico, isso significa perda de dinheiro!

Quando se fala em perda de dinheiro, a alta direção dos bancos já começa a tremer! *Bancos gostam de perder dinheiro?* De jeito nenhum! *Professor, por que eles não compram logo uma Arquitetura Orientada a Serviços?* Porque Arquitetura Orientada a Serviços não é um produto que se possa comprar! Eu não posso chegar em um fornecedor e pedir o melhor SOA que eles tiverem...

O QUE SOA NÃO É...		
Não é uma tecnologia.	Não é um produto.	Não é um web service.
Não é um projeto de TI.	Não é um software.	Não é um framework.
Não é uma metodologia.	Não é solução de negócio.	Não é um middleware.
Não é um serviço.	Não é uma ferramenta.	

SOA: CONCEITOS-CHAVES

O Modelo de Referência do SOA apresenta, a partir da perspectiva dinâmica de serviço, três conceitos-chaves dessa arquitetura: a visibilidade entre provedores de serviços e consumidores; a interação entre eles; e os efeitos no mundo real da interação com um serviço. Existem outros conceitos fundamentais, esses são conceitos relacionados à dinâmica dos serviços.

Visibilidade é o relacionamento entre consumidores e provedores que é satisfeito quando eles estão aptos a interagirem entre si. As pré-condições para visibilidade são consciência, concordância e acessibilidade. O iniciante numa interação precisa ter a percepção dos outros parceiros, os participantes precisam estar predispostos a uma interação, e os participantes precisam estar aptos a interagir.

Interação envolve a execução de ações na direção do serviço. Em geral, isto é realizado pelo envio/recebimento de mensagens, mas há modos que não envolvem explicitamente transmissão de mensagens (Ex: uma interação pode ser efetuada pela modificação do estado de um recurso compartilhado). Grosso modo, refere-se à troca de mensagens como modo primário de interação com um serviço.

Quanto ao efeito no mundo real, sempre há um propósito particular associado à interação com um serviço. O efeito é o resultado do uso de um serviço por um cliente que gera a alteração no estado compartilhado entre este e seu consumidor e outras entidades que pertençam ao mesmo domínio. No fim das contas, sempre há de ter um efeito real.

Professor, o que seria esse estado compartilhado? Imagine um sistema de compras online, em que há mais de um serviço interagindo. Ao finalizar uma compra, o estado compartilhado pelo consumidor, provedor, etc será alterado: a conta do comprador terá um débito, o estoque terá sua quantidade reduzida, entre outros. Não se deve focar nos estados internos, mas nas partes compartilhadas. Em suma...

Visibilidade é a capacidade de o provedor e o consumidor de se verem de forma consciente, concordante e acessível – em geral, por meio de descrições; Interação é a execução de ações em si em direção ao serviço – em geral, por meio de troca de mensagens; Efeito no Mundo Real é o resultado prático da interação entre as partes – em geral, por meio de alterações de seu estado compartilhado.

Em geral, as entidades (pessoas e organizações) oferecem competências e atuam como provedoras de serviço. Aqueles com necessidades, que fazem uso dos

serviços, são consumidores de serviço. A descrição permite que consumidores prospectivos decidam se o serviço é conveniente para suas necessidades e estabelece quando um consumidor satisfaz os requisitos do provedor de serviço.

O Modelo de Referência do SOA, como suporte às dinâmicas da interação com serviços, apresenta um conjunto de conceitos que se referem aos próprios serviços: Descrições do Serviço, o Contexto de Execução, e os Contratos e Políticas que estão relacionados aos serviços e aos seus participantes. Vamos ver em um pouco mais de detalhes cada um deles.

A Descrição do Serviço representa a informação necessária para utilizar um serviço. O propósito é facilitar interação e visibilidade, particularmente quando participantes estão em domínios proprietários diferentes. Pelo oferecimento da descrição, ela torna possível que potenciais participantes construam sistemas que usam serviços e até mesmo ofereçam serviços compatíveis.

Uma política representa alguma restrição ou condição sobre o uso, distribuição ou descrição de uma entidade própria definida por qualquer participante. Um contrato, por outro lado, representa um acordo de duas ou mais partes. Assim como as políticas, acordos são também sobre as condições de uso de um serviço; ele pode também restringir os efeitos esperados no mundo real ao usar o serviço.

O Contexto de Execução de uma interação de serviço é o conjunto de elementos de infraestrutura, entidades de processo, afirmações e acordos de políticas que são identificados como parte de uma interação de serviço. Forma-se, então, um caminho entre aqueles que possuem necessidades e aqueles que possuem competências. Em suma...

Descrição do Serviço é um conjunto de informações necessários para utilizar um serviço, facilitando a interação e a visibilidade; Contratos e Políticas são restrições ou condições de uso de um serviço. Formando um conjunto de regras entre os participantes; Contexto de Execução é um caminho estabelecido entre os participantes, constituído por participantes, infraestrutura, entidades, etc.

SOA: OBJETIVOS E BENEFÍCIOS

Thomas Erl afirma que a arquitetura orientada a serviços possui diversos objetivos estratégicos, tais como: maior interoperabilidade intrínseca; maior federação; e maior diversificação de fornecedores. O primeiro objetivo trata simplesmente de estabelecer a interoperabilidade nativa dentro dos serviços, a fim de reduzir a necessidade de integração. *Reduzir a necessidade de integração?* Sim, senhor!

Antigamente, uma organização possuía diversos sistemas escritos em linguagens diferentes, rodando em plataformas diferentes, e utilizando tecnologias diferentes. Para esses sistemas se comunicarem, era necessário investir massivamente em integração para permitir o intercâmbio de dados. No SOA, os serviços já são nativamente interoperáveis, reduzindo – então – a necessidade de integração.

O segundo objetivo trata de federação! *O que é isso, professor?* Um ambiente federado é aquele em que recursos e aplicativos são unidos, entretanto, cada um mantém sua autonomia e autogestão. Isso pode ser alcançado por meio do uso de serviços padronizados e com capacidade de composição, de modo que cada serviço encapsula um segmento da organização e o expressa de maneira consistente.

- O terceiro objetivo trata da capacidade que o cliente tem de analisar diversos fornecedores e serviços, invocando o melhor serviço ou aquele que seja mais condizente com sua demanda. Na tabela abaixo, vocês podem visualizar diversas vantagens e desvantagens (sim, elas também existem!) da utilização da arquitetura orientada a serviços. *Bacana?*

VANTAGENS DA UTILIZAÇÃO DE SOA
Reutilização: o serviço pode ser reutilizado para outras aplicações.
Produtividade: com o reuso, a equipe de desenvolvimento pode reutilizar serviços em outros projetos, diminuindo o tempo de desenvolvimento.
Flexibilidade: isolando a estrutura de um serviço as mudanças são feitas com maior facilidade.
Manutenibilidade: com baixo acoplamento, facilita a manutenção dos serviços.
Alinhamento com o negócio: a área de negócio visualiza os processos alinhados com a tecnologia.

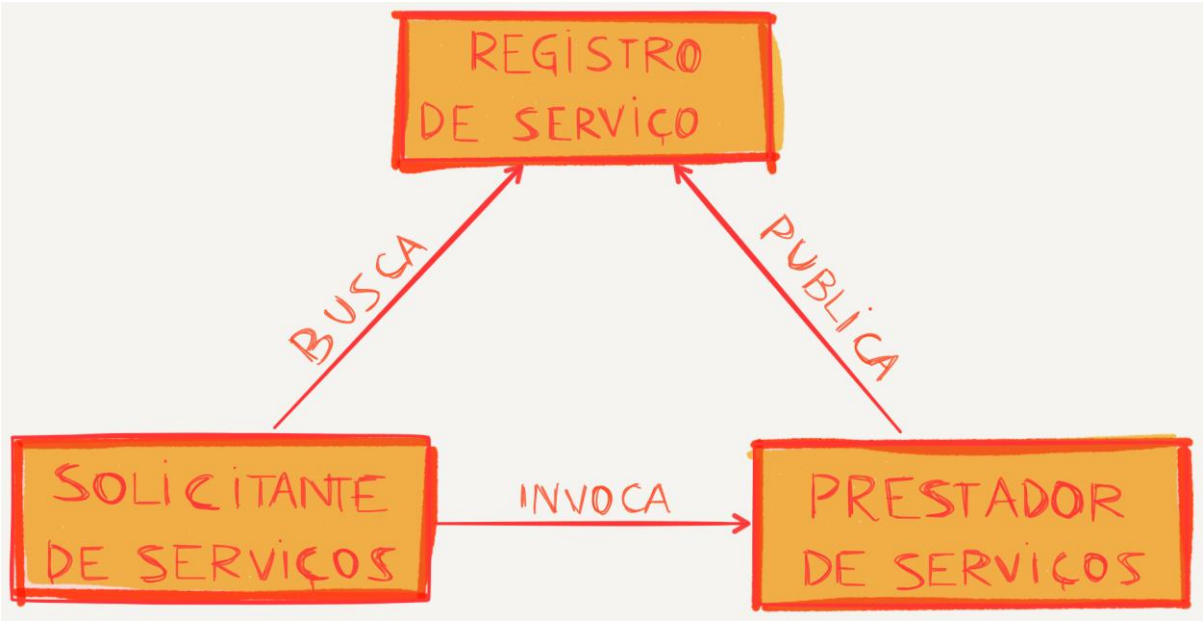
Interoperabilidade: disponibilizar serviços independentemente da plataforma e tecnologia.
Integração: a integração com outros serviços, aplicativos e sistemas legados.
Governança: gerenciamento dos processamentos de negócio.
Padronização: é baseado no uso de padrões (de design, de contratos, protocolos etc).
Abstração: serviço totalmente abstraído da sua implementação.
Riscos: auxilia a mitigação de riscos de negócio.

DESVANTAGENS DA UTILIZAÇÃO DE SOA
Performance: a performance depende do servidor onde o serviço está publicado, como também da rede.
Complexidade: uma grande quantidade de serviços precisa ser gerenciada.
Robustez: caso uma exceção acontecer não tem como reverter o processo.
Disponibilidade: uma queda na rede ou no servidor deixa todos os serviços indisponíveis.
Testabilidade: o debug no serviço é um problema para os desenvolvedores.
Governança: exige-se que haja governança na organização.
Design: grande aumento na complexidade do design dos serviços.
Segurança: os serviços estão disponíveis na rede, qualquer aplicativo pode consumir esse serviço, os dados são trafegados pela rede podendo ser interceptados.

SOA: MODELOS ARQUITETÔNICOS

Galera, nós vimos insistentemente que a arquitetura orientada a serviços é independente de implementação e tecnologia. Em outras palavras, há diversas modelos arquitetônicos de implementação de uma arquitetura orientada a serviços. Veremos agora algumas formas de implementar essa arquitetura em uma organização. Vamos começar pelo modelo mais primitivo...

Antigamente, havia um *Modelo End-To-End*, isto é, os prestadores de serviços notificavam os solicitantes de serviços sobre serviços disponíveis; os solicitantes de serviços, então, invocavam os serviços (imagem acima). Esse modelo foi substituído por um Modelo Triangular, que fornece uma estrutura subjacente para criação, registro, descoberta e composição de serviços distribuídos.

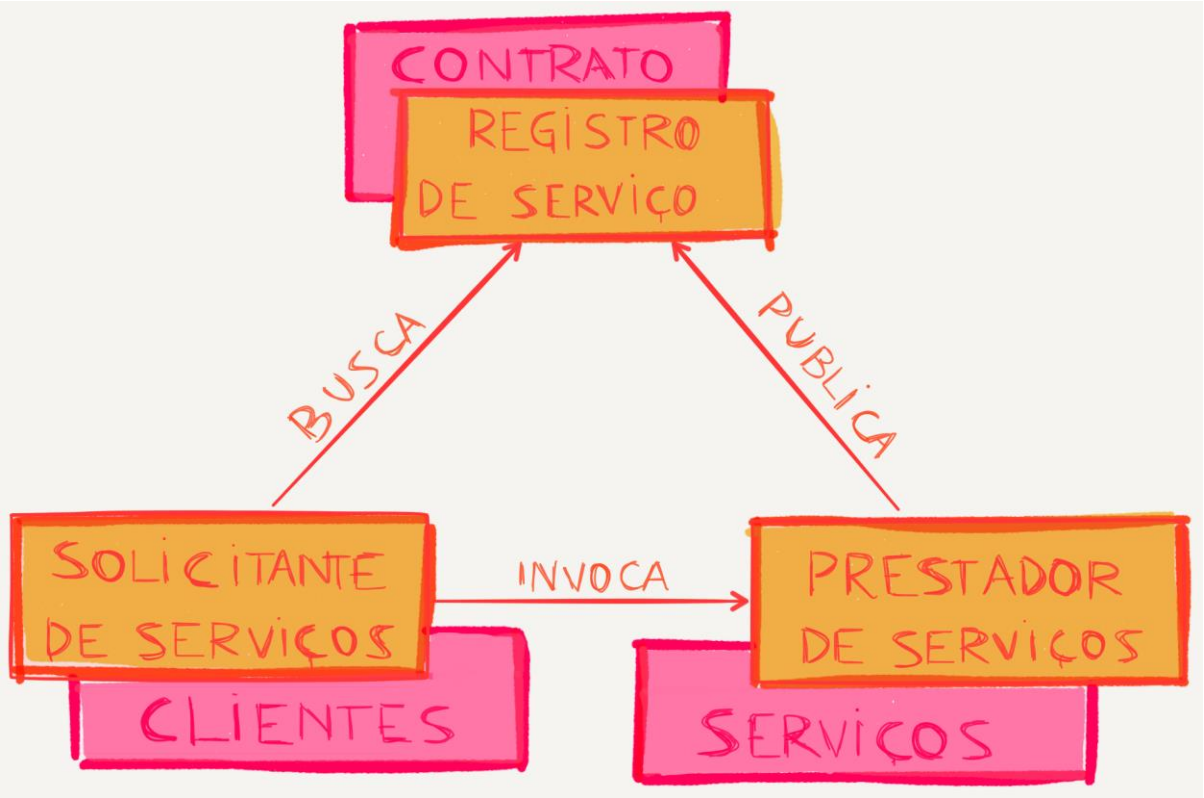


Neste modelo, três papeis são identificados com base em seus comportamentos e responsabilidades sobre um serviço: Prestadores de Serviço, Registro de Serviços e Solicitantes de Serviço. Funcionava assim: um prestador de serviços publica serviços em um registro de serviços; o registro de serviços registra e organiza os serviços publicados e fornece serviços de busca.

Esse registro de serviços geralmente contém um repositório de serviços associado a duas interfaces de acesso: uma interface de publicação que serve aos prestadores de serviços e uma interface de consulta que serve solicitantes de serviços. Dessa forma, um solicitante de serviços consulta o registro de serviços por um serviço de interesse e obtém a localização do prestador correspondente.

O solicitante de serviço, então, se conecta ao prestador de serviço, e remotamente invoca o serviço do prestador. Se um solicitante de serviço estiver ciente de um prestador de serviço apropriado, ele já pode decidir se conectar diretamente ao prestador de serviço sem sequer consultar o registro de serviços. A vantagem do registro é que se pode procurar e descobrir serviços apropriados.

Existem dois modelos bastante similares que implementam o modelo triangular com algumas particularidades. O primeiro é o Modelo Publish-Find-Bind, formado pelo prestador de serviços, solicitante de serviços e o registro de serviços – aqui chamado de Broker de Serviços. O segundo é o Modelo Find-Bind-Execute[3], formado pelo prestador de serviços, solicitante de serviços e o registro de serviços.



Esse último modelo funciona assim: prestadores de serviços publicam serviços no registro de serviço; consumidores de serviços realizam uma busca por algum serviço

[3] Alguns chamam de Find-Bind-Invoke.

de interesse. Caso o consumidor encontre o serviço desejado, é criado um contrato e devolve-se um endereço (comumente chamado de *endpoint*) para utilização do serviço. Após isso, basta executar o serviço contratado...

O registro, então, preocupa-se em catalogar estes serviços dentro de uma estrutura organizada e disponível por um mecanismo de busca. Eles normalmente contêm, associado aos seus repositórios, duas interfaces de acesso: uma de registro e outra de consulta (*Query*). O consumidor efetua consultas diretamente no registro de forma a obter a localização sobre o serviço e o seu provedor.

Em suma: o provedor determina o comportamento daquele que está disponibilizando o serviço, i.e., é considerado o dono do serviço. Ele é o responsável por fornecer toda a infraestrutura de acesso, tipicamente via rede, e é capaz de responder as requisições. O consumidor determina o comportamento daquele que representa o cliente da organização provedora do serviço.

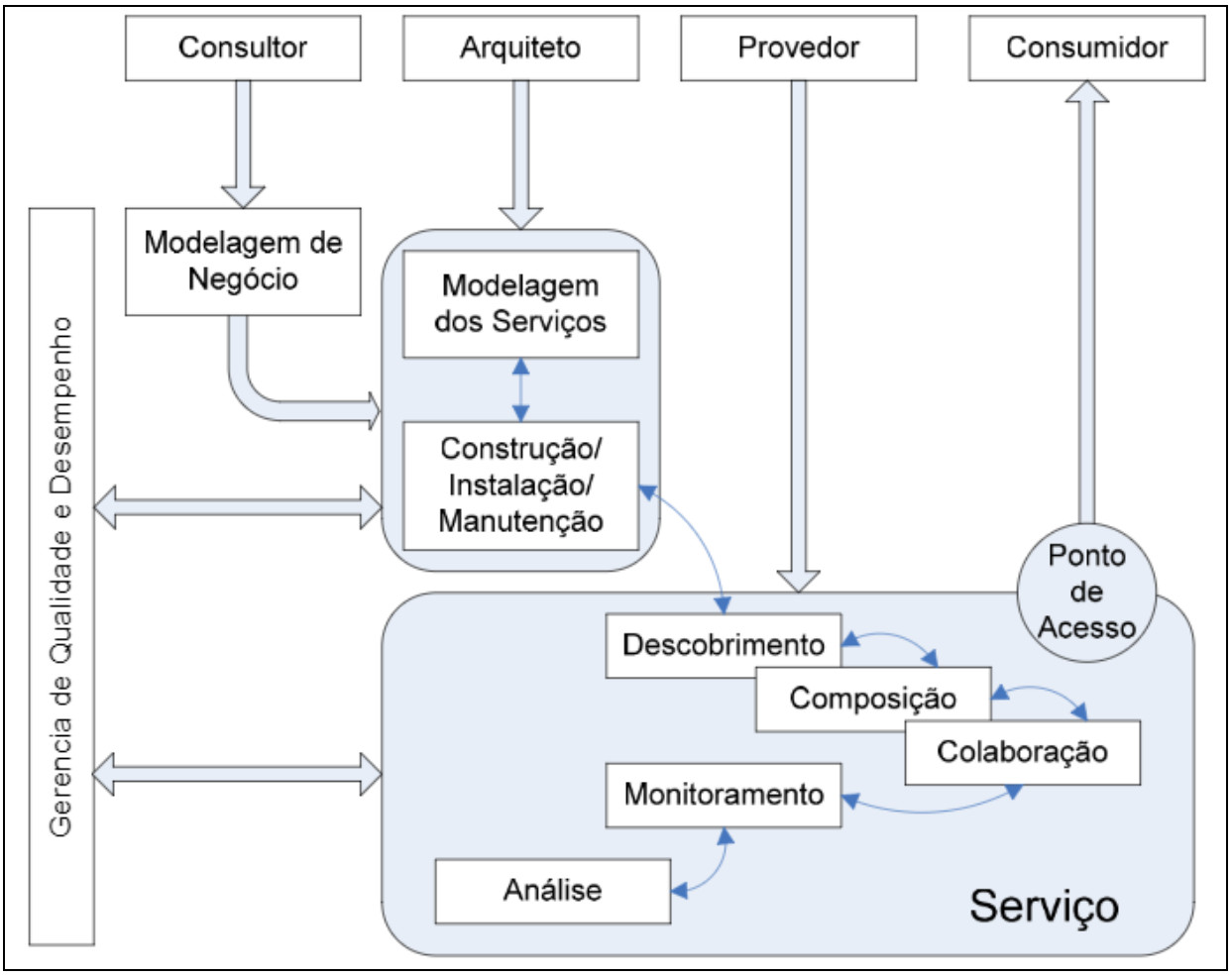
Um consumidor pode ser representado por uma pessoa, uma organização, uma máquina ou um componente de software. Já o registro determina o comportamento que a organização deve ter para divulgar seu serviço e o do cliente que deve proceder para localizar o serviço desejado. Ele gerencia os repositórios que armazenam informações sobre os serviços e entidades que os fornecem.

Galera, algumas questões têm cobrado recentemente os papeis ou atores que compõem o Ciclo de Vida de Soluções SOA. Nesse contexto, abaixo temos alguns papeis considerados essenciais e suficientes, são eles: Consultor de Negócios; Arquiteto SOA; Provedor; e Consumidor – os dois últimos, nós já vimos extensivamente. Vamos ver em mais detalhes na tabela abaixo:

Papel	Descrição
Consultor de Negócios	Papel responsável pelo mapeamento dos processos de negócio da organização.
Arquiteto SOA	Papel responsável pela modelagem dos serviços, além da construção, instalação e manutenção.

Recentemente, algumas questões começaram a cobrar também a modelagem de serviço. A primeira etapa na implantação de uma solução orientada a serviço está relacionada a modelagem do serviço. São responsáveis por identificar os recursos necessários para a construção, o recondicionamento ou reúso de serviços existentes e a integração da solução final.

Normalmente, usa-se as abordagens top-down e bottom-up no processo de modelagem. A modelagem Top-Down determina que a organização identifique, primeiramente, os processos que considerar prioritários. Já a modelagem bottom-up determina que a organização identifique, primeiramente, os processos menos prioritários. *Bacana?*



SOA: PRINCÍPIOS DE DESIGN

Galera, quantas vezes nós já falamos, nessa aula, em Thomas Erl? Pois é, ele é o cara da arquitetura orientada a serviços! Foi do livro dele que retiraram todo o tema da prova discursiva do TCU/2015. Ele destaca em seu livro oito princípios básicos de design que norteiam toda modelagem lógica de serviços e aplicações. Vamos ver agora cada um deles em mais detalhes:

- Contrato de Serviço Padronizado (*Service Contract*):

Todo serviço deve conter um contrato formal padronizado. *O que é um contrato?* Basicamente, trata-se de um documento que descreve o que o serviço faz, dentre outras coisas. Cada contrato de serviços deve estar de acordo com os mesmos padrões de design aplicados a contratos de outros serviços dentro de um inventário de serviços, i.e., deve ser padronizado.

Galera, por que contratos são tão importantes? Porque eles trazem detalhes da funcionalidade provida por um serviço! Ele inclusive auxilia outro princípio que veremos mais a frente – Princípio do Descobrimento (ou Localização). Além disso, um contrato pode conter outros documentos, como um SLA (Service Level Agreement), para constar o nível de serviço acordado.

O contrato de serviço é um dos princípios ou normas de modelagem mais importantes para a visibilidade, à autonomia e o reuso do serviço, pois são neles que estão sendo expostas todas as características, ou seja, todas as funcionalidades que poderão ser usadas por outras aplicações ou serviços. Dessa forma, clientes podem buscar funcionalidades e utilizá-las conforme sua necessidade.

O desenvolvimento do contrato deve ser acompanhado da construção das funcionalidades do serviço, pois este tende a evoluir, e o contrato deve conter todas as características do serviço, considerando possíveis evoluções. Este é um dos maiores riscos encontrados, pois o controle de versão nem sempre é feito. Outra dificuldade é a deficiência de ferramentas para o desenvolvimento.

- Baixo Acoplamento de Serviços (*Service Loose Coupling*):

O que é baixo acoplamento? Na minha época de estudos, eu sempre decorei acoplamento como a independência entre as partes! Se há baixo acoplamento, as partes são pouco dependentes; se há alto acoplamento, as partes são muito

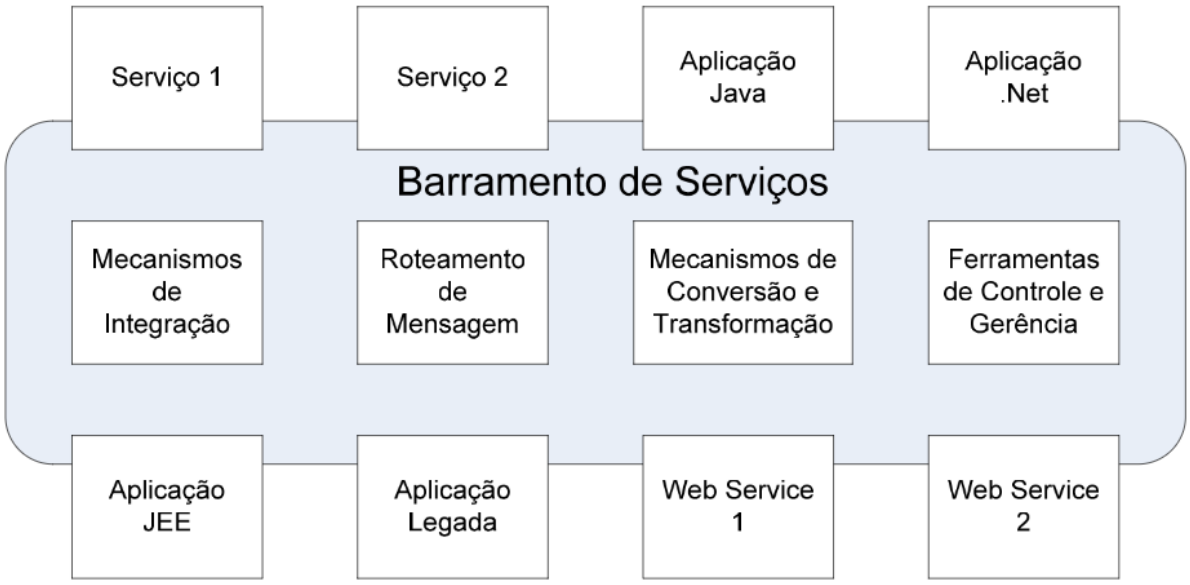
dependentes. O baixo acoplamento de um serviço está relacionado com sua capacidade de ser independente de outros serviços para realizar sua tarefa.

Uma possível consequência do baixo acoplamento é a alta coesão! *De novo, o que é isso?* Eu decorava coesão como a divisão de responsabilidades. Em outras palavras, cada serviço tem sua responsabilidade bem definida e coerente – é o princípio da responsabilidade única. Imaginem um serviço que é responsável por quatro tarefas totalmente diferentes. É uma péssima ideia - gera alto acoplamento!

Um serviço nunca será 100% desacoplado, mas o desafio é achar o grau de desacoplamento que torne o serviço flexível para compor outros serviços e com custo não tão elevado. Para diminuir a dependência com um mundo exterior, utiliza-se a tecnologia chamada ESB (Enterprise Service Bus), implementação conceitual da infraestrutura que abstrai a forma de troca de mensagens feitas pelos sistemas.

Galera, vocês já devem ter pegado ônibus alguma vez na vida (ou ainda pegam). *Como funciona?* Você encontra o ônibus que você quer, você chega nele, e ele te leva onde você quer ir. Você sabe aproximadamente quanto tempo vai demorar para chegar ao seu destino (com exceção de quaisquer atrasos). Se o tráfego estiver intenso, o motorista vai tomar uma rota alternativa para chegar onde você quer.

É parecido com o Enterprise Service Bus (ESB). Ele fornece uma de mover mensagens entre serviços. Embora existam muitos tipos de serviços de mensagens, o ESB é um pacote de estilos diferentes de mensagens, combinadas com serviços de registro e de segurança. É uma forma de as organizações criarem um ambiente SOA escalável sem codificar na mão a forma como os serviços são ligados entre si.



O ESB trata abstrai várias preocupações que o arquiteto não precisa se preocupar. Algumas das capacidades consideradas essenciais para um barramento de serviço corporativo (ESB) são: Resolução de Descrições de Serviços; Transformação de Mensagens; Roteamento Dinâmico de Mensagens (Serviço de conectividade); Tratamento de Exceções; e Monitoramento de Mensagens.

Trata-se, portanto, de uma arquitetura para middlewares com o intuito de integrar serviços distribuídos por meio de diferentes formas de processamento de mensagens e tecnologias (plataformas, hardware, software, protocolos, etc). Por fim, existe ESB sem SOA e SOA sem ESB! Aliás, existe ESB até sem Web Service – pode-se inclusive utilizar outras formas de serviço. Logo, ESB não implementa SOA!

- Abstração de Serviços (*Service Abstraction*):

Vocês se lembram dos Testes Caixa-Preta? São aqueles que são realizados sem a necessidade de entender a forma de implementação! Pois é, serviços devem ser vistos como uma caixa-preta que compõe um sistema. Se eu quiser mudar a implementação, refatorar, manutenir, não há nenhum problema – desde que eu mantenha os padrões de entrada e saída definidos no contrato.

Dessa forma, o consumidor não é afetado! Vejam só: se eu contrato um serviço dos correios para entregar apostilas para os meus alunos, eu não quero saber como eles organizarão a sua logística de entrega! Eles podem mudar o tanto que for, desde que a apostila chegue para os alunos corretamente. Serviços devem ser consumidos independente da linguagem de programação, arquitetura ou outras tecnologias.

Serviços são projetados para limitar informações no contrato de serviços ao que é realmente necessário para que o serviço seja funcionalmente útil a consumidores agora e no futuro. Informações, além das que são publicadas no contratado de serviços, são consideradas privadas e não devem ser disponibilizadas para a criação de consumidores de serviço potenciais.

Muitos dos outros princípios enfatizam a necessidade de publicar mais informações no contrato de serviços. O papel principal desse princípio é manter a quantidade e os detalhes do conteúdo do contrato concisos e equilibrados e evitar o acesso desnecessário a detalhes de serviço adicionais. A abstração de serviço tem como objetivo principal criar uma interface de comunicação com o mundo exterior.

- Reusabilidade de Serviços (*Service Reusability*):

Pessoal, esse é um princípio muito tranquilo! É de se esperar que um serviço seja reutilizável, i.e., eu posso utilizá-lo em diferentes contextos e tecnologias. Como serviços contêm e expressam lógicas, eles podem ser definidos como recursos da organização. Esse princípio é fundamental para aumentar a agilidade do desenvolvimento das aplicações.

É evidente que será necessário mais esforço para construir um serviço mais genérico – que atenda diversas demandas diferentes, abrangendo vários cenários. Observem também que esse princípio é uma consequência do Princípio do Baixo Acoplamento. Para que os serviços sejam reusados, os contratos de serviços devem ter as informações necessárias e bem claras sobre as suas capacidades.

Um serviço reutilizável é aquele que não carrega particularidades técnicas de uma implementação ou regra de negócio específica e é genérico o suficiente para atender outros projetos. Fazer um serviço bastante reutilizável é um investimento de médio prazo e que demanda tempo e dinheiro, e muitas vezes os investidores não enxergam a sua verdadeira e fundamental importância.

Estrategicamente, ele é capaz promover um maior retorno sobre o investimento ao construir componentes de múltiplos propósitos, modelando serviços agnósticos, permitindo a criação de inventários de serviços com alta porcentagem de serviços neutros. Os objetivos por trás da capacidade de reúso de serviço estão diretamente associados a alguns objetivos mais estratégicos da organização.

- Autonomia de Serviços (*Service Autonomy*):

O que é autonomia? É a capacidade de um serviço se auto-administrar. Um serviço autônomo é aquele que independe de um elemento externo para executar sua lógica! *Einh? Como é, professor?* Vamos pensar, por exemplo, no ambiente de execução do serviço! A autonomia de serviços é um princípio que visa fornecer serviços com independência de seu ambiente de execução.

E qual a consequência disso? Maior confiabilidade, na medida em que os serviços podem operar com menos dependência de recursos sobre os quais há pouco ou nenhum controle – a longevidade de um serviço é bastante dependente de sua confiabilidade. Se o serviço não depende de algum elemento externo para executar suas atividades, então ele é mais seguro, confiável e autônomo.

O ato de se autogovernar ou autonomia de serviço pode trazer alguns benefícios claros como, por exemplo, o desempenho. Tendo em vista que, dessa forma, se terá

o mínimo de dependências com outros serviços e as execuções das funcionalidades se tornam mais diretas e consequentemente o número de erros inesperados diminui consideravelmente.

Essa autonomia é medida e disponibilizada nos contratos formais, tendo como finalidade esclarecer o nível de independência aos seus consumidores. Além da maior confiabilidade, a autonomia promove melhor desempenho e previsibilidade comportamental. Além disso, como dissemos, ele aumenta a quantidade de controle que um serviço tem sobre o ambiente em tempo de execução.

- Independência de Estados de Serviços (*Service Statelessness*):

Serviços são explicitamente projetados para minimizar o período durante o qual eles existem em uma condição de dependência de informações de estado. A independência de estado é quando um serviço não precisa reter nenhum dado do estado para que outro serviço processe a sua lógica. Podemos citar como exemplo, o uso do protocolo HTTP.

Ele monta seu cabeçalho e todo o conteúdo da página que é enviada para o navegador, retornando a uma condição de independência de estado em que ele não retém nenhuma solicitação ou memória adicional do navegador. Serviços são projetados para minimizar o período durante o qual eles existem em uma condição de dependência de informação de estado, aumentando a escalabilidade do serviço.

Os serviços devem evitar a alocação de recursos por muito tempo! Recomenda-se a construção de serviços *stateless* (sem estado). Galera, isso não é muito fácil de alcançar! A otimização da alocação de recursos é difícil por conta da natureza distribuída e autônoma dos serviços, além da natureza de mudanças constantes dos ambientes que utilizam essa arquitetura.

- Visibilidade de Serviços (*Service Discoverability*):

O termo em inglês é discoverability! Como esse vocábulo não existe, traduziram como visibilidade. Nós já vimos que serviços são bastante reutilizáveis, mas para que você possa reutilizá-lo, você deve ser capaz de encontrá-lo. Serviços são projetados para serem efetivamente descobertos e interpretados para que, na descoberta, seu propósito e capacidades sejam claramente entendidos.

O objetivo principal é fazer com que o serviço se torne visível a todos, aumentando assim a agilidade, a produtividade e a confiabilidade dos consumidores. Os serviços

são projetados para que sejam encontrados e seus propósitos e capacidades sejam claramente entendidos, ou seja, os contratos de serviços devem ter meta-informações extras que transmitem claramente o que o serviço realmente faz.

A aplicação desse princípio após a implantação de um serviço pode comprometer a qualidade da sua visibilidade, portanto é uma boa prática adicionar as meta-informações antes do lançamento da versão inicial. O mecanismo de descoberta é necessário para evitar o desenvolvimento redundante de uma solução que já está contida em um serviço existente.

- Composição de Serviços (*Service Composability*):

Serviços são projetados para que atuem como participantes eficazes de uma composição, independente do tamanho e da complexidade de composição. Ao discutir os objetivos da composição de serviços, boa parte dos objetivos de reúso de serviços também é aplicável. Isso ocorre porque, muitas vezes, a composição de serviços é uma forma de reúso de serviços.

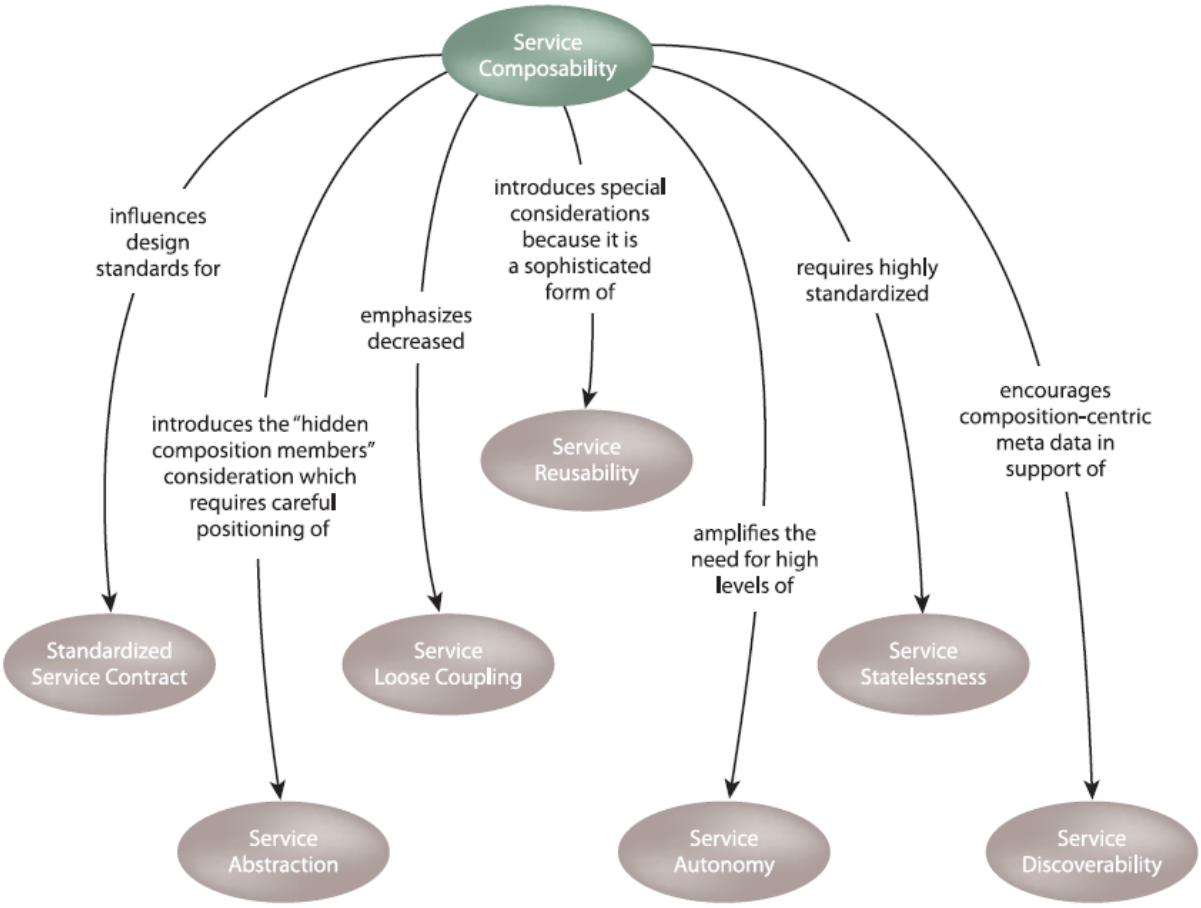
Talvez você se lembre que um dos objetivos que selecionamos para o princípio da reusabilidade era possibilitar a composição de serviços em larga escala. A composição de serviços permite que as capacidades de um serviço sejam combinadas várias vezes com as de outros serviços em novas configurações a fim de resolver problemas diferentes.

Vamos ver um exemplo? Os correios disponibilizam um serviço tal que, dado um CEP, retorna um endereço. Eles também oferecem outro serviço tal que, dado um CEP, retorna os tipos de frete e seus valores. Esses serviços isoladamente resolvem determinados problemas específicos. Quando compostos, esses resolvem outros tipos de problemas. *Simples de entender?*

Ao estabelecer em uma empresa uma lógica representada por um inventário de serviços altamente reusáveis, fornecemos o meio para que uma grande extensão dos futuros requisitos da automação de negócios seja cumprida por meio da composição de serviços. Esse princípio trata de serviços se juntarem e serem acessados de forma a englobar e atender um problema maior.

Estas características se relacionam de uma forma interdependente. Podemos deduzir que o *Baixo Acoplamento* reduz a *Alocação de Recursos*; *Contrato Padronizado* forma as bases para o *Descobrimento*; *Alocação de Recursos* reduzida

maximiza de *Reutilização*; *Descobrimento* promove *Reutilização*; *Autonomia* reduz a *Alocação de Recursos*; *Baixo Acoplamento* permite *Autonomia*.



SOA: COMPOSIÇÃO DE SERVIÇOS

Já que acabamos de falar sobre o princípio da composição de serviços, vamos ver mais alguns detalhes! Conforme vimos anteriormente, uma das principais características de um serviço é a sua capacidade de se compor com outros serviços para a elaboração de um novo serviço de mais alto nível de abstração – como é apresentado na imagem a seguir.

A composição de serviços potencializa os diversos benefícios oferecidos pela arquitetura orientada a serviços. Ela possibilita, por exemplo, que o serviço de mais alto nível realize interações síncronas ou assíncronas, assim como a manutenção de estados. *Professor, você não acabou de passar várias páginas dizendo que os serviços devem ser stateless?* Calma, pequeno gafanhoto...

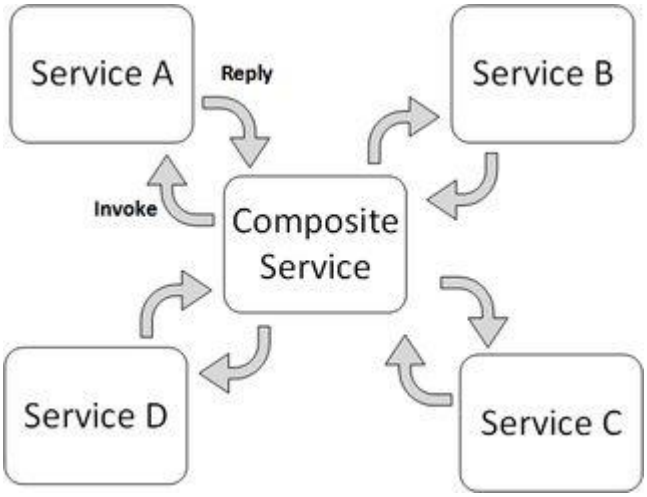
Eu falei que serviços, individualmente, devem ser projetados seguindo princípios de *design* que incluem, por exemplo, a independência de estados. No entanto, quando serviços estão compostos, trabalhando em conjunto, geralmente é necessário que eles sejam capazes de manter seus estados. Galera, existem workflows de coordenação para implementar a composição de serviços. Vejamos os principais:

Orquestração: representa o processo em que vários recursos dentro de uma organização podem ser coordenados para executar um conjunto de lógicas de processo de negócio. Essa tecnologia é mais comumente associada a uma

plataforma central de gerenciamento de atividades que fornecem benefícios como transformação de dados e gerenciamento de transações.

Em outras palavras, a orquestração representa um processo de negócio executável centralizado e único, chamado coordenador, responsável por controlar e coordenar as interações entre diferentes serviços. Ele é o cara que vai invocar e combinar os serviços isolados em serviços compostos. Em uma analogia simplista, a orquestra seria um serviço composto, os músicos seriam serviços individuais e o maestro seria coordenador. Ele diz quando um para e quando o outro começa – ele é a figura central que coordena a interação entre todos os músicos/serviços.

A orquestração inclui o gerenciamento de transações entre serviços individuais e emprega uma abordagem centralizada para a composição do serviço, como é apresentado na imagem abaixo. Além disso, a lógica de processo de negócio, na orquestração, pertence à organização, i.e., é uma lógica privada e intra-organização – em contraste com a coreografia.



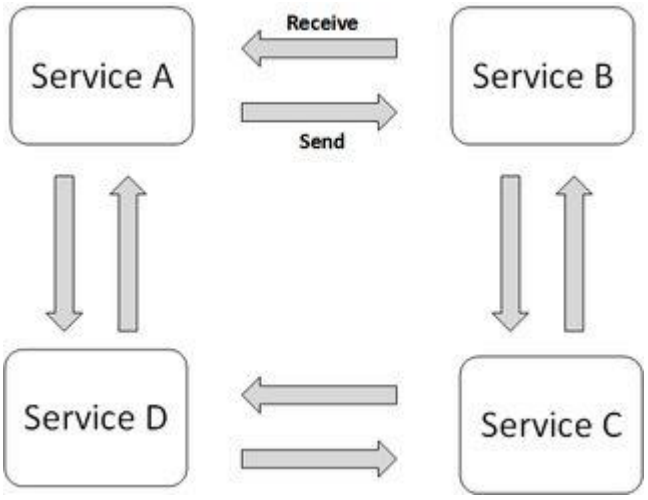
No contexto de web services, existe uma linguagem para facilitar a orquestração das chamadas de serviços: Web Services Business Process Execution Language (WS-BPEL)! Essa é uma linguagem executável, com variáveis, tratamento de exceção, etc

que segue o Padrão OASIS e é utilizada para especificar ações de processos de negócios e orquestrar serviços em um fluxo de processos.

Coreografia: em contraste com a orquestração, não há um coordenador central. Em vez disso, cada serviço envolvido na coreografia sabe exatamente quando executar suas operações e com quem interagir. A coreografia é um esforço colaborativo, distribuído e descentralizado com foco na troca de mensagens em processos de negócio públicos. Pois é, aqui o processo negócio é público e inter-organizações.

Todos os participantes da coreografia precisam estar cientes do processo de negócio, das operações a serem executadas, das mensagens a serem trocadas, e do momento da troca de mensagens. A orquestração é organizada por coordenador, fazendo verificações de pré-condições e pós-condições. Já na coreografia, todos auxiliam no fluxo das operações.

A coreografia descreve as interações entre serviços, ao passo que a orquestração representa o controle do ponto de vista de uma das partes. Isto significa que uma coreografia difere de uma orquestração em relação ao local onde a lógica que controla as interações entre os serviços envolvidos devem residir. Há um grande esforço na ordenação de mensagens e na interação entre os participantes.



No contexto de web services, existe uma linguagem para facilitar a coreografia das chamadas de serviços: Web Services Choreography Description Language (WS-CDL)! Essa é uma linguagem descritiva que descreve colaborações ponto-a-ponto de participantes ao definir, a partir de um ponto de vista global, um comportamento observável comum e complementar.

Por fim, vamos falar um pouco sobre Service Component Architecture (SCA). Trata-se de uma tentativa iniciada por fornecedores de software (Ex: BEA, IBM, Oracle, SAP, etc) de simplificar a construção de uma Arquitetura Orientada a Serviços. Aliás, trata-se de um modelo de programação simples, porém poderoso, para construir aplicativos com base em SOA.

Ele é baseado na ideia de que funções do negócio são providas como uma série de serviços, os quais são combinados a fim de criar soluções que sirvam a uma necessidade particular. Estas aplicações podem conter tanto serviços novos criados especificamente para a aplicação, mas também funções do negócio provenientes de sistemas e aplicações, as quais são reutilizadas como parte da composição.

SCA provê um modelo para composição de serviços e para criação de componentes de serviços, incluindo o reuso de funções de aplicações existentes dentro da composição SCA. Ele é um modelo que tem o objetivo de incluir uma grande variedade de tecnologias para componentes de serviços e para os métodos de acesso que são utilizados para conectá-los.

Para componentes, isto inclui não apenas diferentes linguagens de programação, mas também frameworks, e ambientes geralmente utilizados com estas linguagens. Para métodos de acesso, composições SCA permitem o uso de várias tecnologias de comunicação e acesso a serviços que são usadas em geral, incluindo, por exemplo, Web Services, sistemas de mensagens e Remote Procedure Call (RPC).

O SCA foca em três aspectos: Composição (como empacotar componentes de software para eles serem reutilizados por outras aplicações); Montagem (define como componentes podem ser colocados juntos e tratá-los como um único serviço; e Política (como tratar restrição de acesso ou assinatura digital em uma arquitetura orientada a serviços).

SOA: MANIFESTO SOA

Assim como o Manifesto Ágil, há também o Manifesto SOA. Vejam:

Orientação a Serviço é um paradigma que molda o que você faz. SOA é um tipo de arquitetura que resulta da aplicação de orientação a serviço. Nós temos aplicado orientação para ajudar organizações a, de maneira consistente e sustentável, agregar valor ao negócio, com maior agilidade e efetividade de custos, em alinhamento com a dinâmica das necessidades de negócio. Através de nosso trabalho, priorizamos:

Valor do negócio em relação a **estratégia técnica**;
Objetivos estratégicos em relação a **benefícios específicos de projetos**;
Interoperabilidade intrínseca em relação a **integração personalizada**;
Serviços compartilhados em relação a **implementações de propósito específico**;
Flexibilidade em relação a **otimização**; e
Refinamento evolutivo em relação a **busca da perfeição inicial**.

Isso é, mesmo valorizando os itens à direita, valorizamos mais os itens à esquerda. Nós seguimos os seguintes princípios orientadores:

PRINCÍPIOS ORIENTADORES

1. **Respeitar a estrutura social e de poder da organização.**
2. **Reconhecer que SOA, em última instância, requer mudanças em múltiplos níveis.**
3. **O escopo da adoção de SOA pode variar.**
4. **Manter os esforços gerenciáveis e dentro de limites significativos.**
5. **Produtos e padrões, por si só, não proverão uma SOA nem aplicarão os paradigmas de orientação a serviço por você.**
6. **SOA pode ser realizada através de uma variedade de tecnologias e padrões.**
7. **Estabelecer um conjunto uniforme de padrões e políticas corporativas embasado em padrões da indústria, de facto, e da comunidade.**
8. **Buscar uniformidade no exterior e permitir diversidade no interior.**

9. Identificar serviços através da colaboração entre partes interessadas no negócio e na tecnologia.
10. Maximizar o uso de serviços considerando o escopo de utilização atual e futuro.
11. Verificar que os serviços satisfaçam os requisitos e objetivos de negócio.
12. Evoluir os serviços e sua organização em resposta ao uso real.
13. Separar os diferentes aspectos de um sistema que mudam com diferentes frequências.
14. Reduzir dependências implícitas e publicar todas as dependências externas para aumentar a robustez e diminuir o impacto de mudanças.
15. A cada nível de abstração, organizar cada serviço em torno de uma unidade de funcionalidade coesa e gerenciável.

Esse manifesto foi detalhadamente comentado – parte a parte – por um de seus maiores idealizadores: Thomas Erl. Se quiserem se aprofundar, leiam este artigo:

http://www.soa-manifesto.com/annotated_portuguese.htm

MENSAGERIA E CORBA

Antes de entrar exatamente nesses temas, vamos falar um pouco sobre comunicação síncrona e assíncrona. Em um sistema distribuído, as partes têm que conversar entre si e essa comunicação é realizada, em geral, por meio de trocas de mensagens. Existem duas famosas abordagens de comunicação: síncrona e assíncrona. Vamos ver de maneira bastante lúdica a diferença entre elas...

Minha namorada gostava de me ligar várias vezes durante o dia para tratar de assuntos pessoais. Até que um dia, eu tive que chegar para ela e dizer: "*Amor, eu não aguento mais! Nós precisamos reduzir essa quantidade de comunicação síncrona e substituí-la por outro meio de comunicação assíncrona porque você está me atrapalhando!*".

Tomei um sopapo no pé da orelha e reorganizei meus pensamentos de uma forma que ela entendesse melhor! *O que eu queria dizer?* A comunicação síncrona exige, como o próprio nome diz, sincronia, ou seja, quando ela me telefonava eu tinha que parar o que estava fazendo e respondê-la imediatamente – eu não podia simplesmente ouvir e respondê-la depois.

Então ela dizia: *"Oi, tudo bem?"*; eu respondia: *"Oi, tudo bem! E você?"*; ela dizia: *"Tudo ótimo! Está no trabalho?"*; eu respondia: *"Sim, e você?"*... enfim, a troca de mensagens era síncrona, ela falava algo, eu ouvia e respondia, ela ouvia de volta e respondia de volta e assim por diante. Como eu não podia ouvir uma pergunta dela e não a responder, isso se chamava comunicação síncrona!

Foi então que um gênio colocou no Whatsapp uma nova feature que permitia enviar mensagens de voz. Dessa forma, eu podia ouvir a mensagem dela, ir fazer outra atividade e respondê-la só mais tarde quando eu tivesse um tempo livre, ou seja, a mensagem dela não bloqueava mais as minhas atividades e isso (obrigado, Deus!) se chama comunicação assíncrona.

Quando vamos para o contexto de sistemas, a abordagem assíncrona de comunicação permite que a troca de mensagens seja feita de tal maneira que o sistema que envia a mensagem não precisa esperar o processamento da resposta, ficando livre para continuar sua rotina, diferente de tecnologias de integração com RPC (Remote Procedure Call). Trata-se de uma invocação não-blocante!

É nesse cenário que se insere a Mensageria, i.e., trata-se de uma maneira de resolver problemas complexos de integração de sistemas por meio da comunicação indireta

entre as partes. *E por que dizemos indiretas?* Suponha que tenhamos um problema complexo de integração, i.e., temos dois ou mais sistemas que precisam "conversar", entretanto eles não possuem uma linguagem comum para tal.

Para resolver esse tipo de problema, é comum inserir um middleware que auxiliará a comunicação entre os dois sistemas com o intuito e, por conta desse middleware, diz-se que a comunicação é indireta. A Mensageria permite, portanto, o processamento de requisições de forma assíncrona – além de integrar sistemas desenvolvidos em tecnologias diferentes (Ex: Java e .NET).

Qual a grande vantagem disso? Isso permite reduzir o grau de acoplamento (i.e., dependência) entre os componentes. De mais a mais, a mensageria reduz gargalos dos sistemas, contribui para o aumento da escalabilidade e facilita uma arquitetura flexível e ágil, capaz de responder mais rapidamente a mudanças. Esse é um tópico absurdamente raro de cair em prova!

O CORBA (*Common Object Request Broker Architeture*) é uma arquitetura padrão proposta pelo *Object Management Group* (OMG) para estabelecer e simplificar a troca de dados entre sistemas distribuídos heterogêneos por meio de uma estrutura comum para o gerenciamento de objetos distribuídos que é conhecida como *Object Manager Architecture* (OMA).

O componente fundamental para o CORBA é o ORB (*Object Request Broker*), que é responsável pela interoperabilidade entre objetos. Por meio do IOR (*Interoperable Object Reference*), o ORB consegue localizar os objetos e transmitir informações. O ORB é um módulo intermediário entre cliente/objeto, sendo responsável por aceitar requisições, enviá-las ao objeto correto e entregar a resposta ao cliente.

Além disso, ele utiliza a IDL (Interface Definition Language), uma linguagem utilizada para descrever as interfaces das implementações dos objetos na rede (seria algo similar ao WSDL). Uma interface escrita em IDL especifica o formato da chamada das operações providas pelo objeto e os parâmetros de entrada ou saída necessários para efetuar a operação.

O CORBA é dependente de linguagem de programação? Não só ele é independente de linguagem de programação, como ele é independente de tecnologia (como plataformas, etc). Ele atua de modo que os objetos (componentes de software) possam se comunicar de forma transparente ao usuário, mesmo que para isso seja necessário interoperar com outro software, sistema operacional, ferramentas, etc.

VERSIONAMENTO SOA

O que é o versionamento de serviços? Quando se está inserido num programa SOA, vários projetos de software são criados e realizados em paralelo. À medida que os projetos amadurecem, vários serviços são disponibilizados para uso pelas aplicações, seja para satisfazer algum critério da solução que está sendo criada, seja para disponibilizar tal serviço a um parceiro ou cliente.

Seja qual for o caso, o ciclo dos serviços é iniciado a cada projeto executado. Como já é de se esperar das iniciativas SOA, é comum que alguns serviços sejam necessários em mais de uma aplicação. Neste caso, com os projetos em andamento, os serviços disponibilizados devem suprir expectativas de vários consumidores distintos e, às vezes, estas expectativas acarretam mudanças nos serviços existentes.

A cada vez que um serviço é alterado para acomodar alguma nova necessidade, uma nova versão do serviço é disponibilizada. Nos primeiros estágios de um programa[4] SOA, as mudanças no serviço para acomodar diferentes expectativas são frequentes. Neste caso, é importante que você estabeleça uma forma eficiente de lidar com as mudanças. *Por que, professor?*

Porque se você não o fizer, os consumidores que usarem os serviços já disponibilizados sofrerão as consequências das mudanças a adaptações. Bem, a melhor forma de lidar com as mudanças é estar preparado. Grande parte das práticas relacionadas ao versionamento de serviços são, em verdade, práticas proativas quanto às mudanças.

Orientação a serviços é um modelo de desenvolvimento interessante, mas deve ser muito bem projetado para que ele tenha resultados visíveis. Assim como qualquer projeto de sistemas, a correta aplicação da disciplina de análise e desenho pode ser crucial para o sucesso do projeto. Da mesma forma, se você projetar bem seus serviços, raramente eles lhe trarão problemas quando tiverem que ser alterados.

Além da boa dosagem de análise e desenho em serviços, é importante que você trace claramente um conjunto de políticas relacionadas à gestão dos serviços. Boas práticas neste aspecto envolvem ter um repositório único na organização para manter questões de dependência e rastreabilidade entre artefatos de um serviço numa forma controlada, além de definir o tempo de guarda das versões.

[4] Um programa SOA é a aplicação de SOA em nível organizacional com um fim bem definido.

Ademais, é importante que disponibilizar um mecanismo eficiente para a busca de serviços, na forma de um registro de serviços dinâmico. Se parceiros, clientes e consumidores tiverem um repositório único para acessar detalhes de um serviço como contratos, políticas, modelos de dados, especificações, componentes, etc a mudança e o versionamento tendem a se tornar mais controlados.

Da mesma forma, é importante que toda organização que detenha um portfólio considerável de serviços disponibilize um mecanismo eficiente de consulta, também em nível organizacional. Desta forma, toda vez que consumidores estiverem interessados em alguns serviços, seu mecanismo de registro poderá ajudá-los a obter respostas mais facilmente.

Quando mudanças são eminentes, ter um mecanismo que possibilite a acomodação de mudanças apenas com esforço de configuração e com menos implementação é muito importante. Para o SOA, são importantes que questões de protocolos, APIs, linguagens, conectores e plataformas não sejam problemas grandes para quando você quer mudar o comportamento ou interface de um serviço.

É necessário que você possua uma infraestrutura de integração que acomode questões de mudanças de forma simples e rápida. Neste caso, o uso de um bom barramento de serviços (Enterprise Service Bus) é primordial para lidar com questões de versionamento de serviços. Através de técnicas de roteamento, transformação e mediação de mensagens, é simples acomodar várias versões de um serviço.

Mais ainda, um ESB pode complementar as atividades de um sistema de registros. Em resumo, as estratégias usuais sobre como lidar com versionamento de serviços são: (1) revisar com cuidado aspectos de desenho de serviços; (2) usar repositórios de gerenciamento de artefatos dos serviços; e (3) usar um sistema de registro eficiente para revelar serviços.

SOA: PADRÕES SOA

Padrões SOA são princípios propostos para desenvolver uma solução lógica de serviços dentro de uma Arquitetura Orientada a Serviços. O sucesso do desenvolvimento de software baseado em qualquer paradigma de design particular nunca está garantido. Software desenvolvido sob o paradigma de projeto orientado a serviços carrega riscos ainda maiores.

Isso ocorre porque uma arquitetura orientada a serviços normalmente abrange várias áreas de negócio e requer uma análise inicial considerável. Portanto, SOA desenvolvido sem orientações concretas provavelmente falhará. Para garantir que o movimento em direção a orientação a serviços seja uma mudança positiva, que entrega os benefícios que promete, é útil adotar um conjunto de regras ou padrões.

Os padrões de projeto orientados a serviço são divididos em: contrato de serviço padronizado, serviços fracamente acoplados, abstração de serviços, reusabilidade de serviços, autonomia de serviços, serviços sem estado, descobertabilidade de serviços, e composição de serviços. A aplicação desses princípios pode gerar serviços independentes de tecnologia e interoperáveis.

A aplicação desses princípios ajuda a atingir os objetivos subjacentes relacionados com a adoção de orientação a serviços. Essas metas são de natureza estratégica, ou seja, de longo prazo, olhando para além das necessidades imediatas de uma organização. Estes objetivos estratégicos ajudam a desenvolver uma organização ágil, capaz de responder rapidamente a mudanças de mercar e reduzir esforços.

Soluções orientadas a serviços são, em geral, neutras em relação aos fornecedores, dirigidas por negócio, centradas na organização e orientadas a composição. Em outras palavras, soluções que não dependem de um fornecedor específico, que é totalmente focada nas necessidades de negócio, centradas em uma organização específica e capaz de compor diversos serviços de diversas dimensões.

SOA: SISTEMAS DE GOVERNANÇA SOA

Em geral, o conceito de governança nos diz que se trata de estabelecer e reforçar como pessoas e soluções trabalham em conjunto para alcançar objetivos organizacionais. Observem que se trata de controles diferentes que se distinguem das atividades rotineiras de gestão. Já a definição de Governança de Tecnologia da Informação é:

> *"Conjunto de práticas, padrões e relacionamentos estruturados, assumidos por executivos, gestores, técnicos e usuários de TI de uma organização, com a finalidade de garantir controles efetivos, ampliar os processos de segurança, minimizar os riscos, ampliar o desempenho, aperfeiçoar a aplicação de recursos, reduzirem os custos, suportar as melhores decisões e consequentemente alinhar TI aos negócios".*

Gosto muito dessa última parte da definição, porque a governança pode ser vista como um framework para tomada de decisão. Ela especifica responsabilidades de decisão para ajudar a atingir o comportamento desejado em sua utilização. Em linhas gerais, governança é o estabelecimento e administração de controle sobre um ambiente de modo a influenciar e garantir ações e comportamento.

Com a governança, a corporação torna-se apta a: identificar os ativos que estão sendo produzidos na empresa; ter o controle do ciclo de vida destes ativos; identificar o grau de dependência entre os diferentes ativos; promover reuso e economia; assegurar a execução de um processo; realizar mais facilmente análise de impacto; assegurar cumprimento de políticas de execução e de segurança; etc.

E o que seria a Governança SOA? É um conceito utilizado para atividades relacionadas ao exercício de controle sobre soluções de arquitetura orientada a serviços. Outra definição afirma que é uma forma de garantir e validar que os ativos e artefatos dentro da arquitetura estão agindo como o esperado e mantendo um certo nível de qualidade. Governança SOA estende a Governança de TI!

O SGRM (SOA Governance Reference Model) é um modelo genérico utilizado como linha de base para aplicar o processo em uma organização. Ele define princípios, processos, artefatos, papéis, responsabilidades, tecnologias, entre outros. Vamos continuar nosso papo vendo os princípios que regem a Governança SOA (ou Governança de Serviços).

São eles: promover o alinhamento entre Negócio e TI; estar em conformidade com a governança da organização; haver uma arquitetura de referência SOA; haver

contratos de provedor e consumidor; possuir metadados de serviços; identificar as partes interessadas; deve automatizar os processos de governança; deve implementar um modelo de financiamento; entre outros.

Quem quiser se aprofundar mais, recomendo o site:

https://www.opengroup.org/soa/source-book/gov/index.htm

1. (FCC - 2012 - TRT - 6ª Região (PE) - Técnico Judiciário - Tecnologia da Informação)

Cinco perguntas que você precisa saber antes de investir em SOA

...O que significa efetivamente ter uma governança de SOA?

O tão falado alinhamento da organização é uma das principais preocupações atuais. Um processo unificado de governança faz com que sejam melhorados os negócios da companhia de forma geral. No entanto, não são necessários novos sistemas ou ferramentas que vão melhorar o sistema de gerenciamento a ponto de integrar TI e gestão. A chamada governança de SOA é compartilhar objetivos. O importantes é ter cada stakeholders representado no momento da elaboração de um projeto SOA. Ter algum sistema de gerenciamento de serviço, como ITIL, também colabora para dar uma visibilidade ao cliente.

Sobre SOA, e com base no texto, é correto afirmar que:

a) é essencial que uma empresa adote as melhores práticas da ITIL antes de implantar o SOA.

b) SOA é uma ferramenta de software utilizada no gerenciamento de serviços de TI.

c) SOA, neste contexto, se refere à sigla para Society Of Actuaries, uma organização educacional, profissional e de pesquisa com sede nos Estados Unidos.

d) SOA é uma abordagem de projeto baseada em padrões para a criação de uma infraestrutura de TI integrada capaz de responder rapidamente às mudanças nas necessidades de negócios.

e) a implantação do SOA numa empresa, por si só, é suficiente para garantir o alinhamento dos negócios com TI.

Comentários:

O QUE SOA NÃO É...		
Não é uma tecnologia.	Não é um produto.	Não é um web service.
Não é um projeto de TI.	Não é um software.	Não é um framework.
Não é uma metodologia.	Não é solução de negócio.	Não é um middleware.
Não é um serviço.	Não é uma ferramenta.	

(a) Errado, é uma boa prática, mas não é essencial; (b) Errado, SOA não é uma ferramenta! Ademais, não é utilizada para gerenciar serviços de tecnologia da informação – para tal, existe o ITIL; (c) Errado, é o acrônimo de Service Oriented Architecture; (e) Errado, ela não é capaz de sozinha garantir o alinhamento entre TI e Negócio.

(d) Correto! *SOA é uma abordagem de projeto?* Sim! Ela não é operacional, é uma abordagem que envolve em geral grande investimento e deve ser tratada como tal. *É baseada em padrões para a criação de uma infraestrutura de TI integrada?* Sim, por exemplo: SOAP, WSDL e UDDI. *É capaz de responder rapidamente às mudanças nas necessidades de negócio?* Sim, funcionalidades são fracamente acopladas e altamente coesas, logo responde a mudanças facilmente.

Gabarito: D

---

2. (FCC - 2012 - TRT - 11ª Região (AM) - Analista Judiciário - Tecnologia da Informação) Em SOA:

a) normalmente, são utilizados WSDL para descrever os próprios serviços e SOAP para descrever os protocolos de comunicação.

b) a tecnologia utilizada para prover o serviço, tal como uma linguagem de programação é parte da definição do serviço.

c) orquestração é o processo de sequenciar serviços e prover uma lógica adicional para processar dados, levando em conta a representação de dados.

d) um dado serviço de broker não requer do provedor a necessidade de definição de listas categorizadas dos serviços.

e) um serviço, do ponto de vista da arquitetura, deve funcionar de forma independente do estado de outros serviços, inclusive nos casos de composite services.

Comentários:

(a) Errado, mas a banca deu o gabarito como correto. SOAP não é utilizado para descrever protocolos de comunicação – ele é o protocolo de comunicação; (b) Errado, linguagem de programação não é parte da definição do serviço; (c) Errado, para orquestrar os serviços web, é indiferente a representação de dados; (d) Errado, Broker nada mais é que um registro de serviços, logo ele requer do provedor a definição de listas de serviços; (e) Errado, *composite services* são pequenos serviços agregados, i.e., dependentes entre si.

Gabarito: A

---

3. **(FCC - 2012 - TJ-PE - Técnico Judiciário - Programador de Computador)** Sobre SOA é INCORRETO afirmar:

a) Quando se utiliza SOA, todos os aplicativos desenvolvidos em uma corporação devem ser implementados de forma que possam prover serviços que permitirão a integração de componentes de uma única plataforma.

b) Web Services a grosso modo podem ser classificados como métodos remotos publicados na Web, que através do uso do protocolo SOAP e XML, permitem a exposição de métodos de aplicativos diversos na Web, para consumo por qualquer outro aplicativo ou dispositivo que utilize o HTTP.

c) Uma consideração importante a respeito de versionamento é definir por quanto tempo é necessário manter as diferentes versões de um serviço em funcionamento.

d) O versionamento assume a existência simultânea de versões de um serviço, incluindo as suas operações e suas diferentes implementações.

e) Um conceito importante por trás da arquitetura orientada a serviços está na autonomia; a possibilidade de se distribuir, modificar e manter, independentemente de outros sistemas (consumidores), novas funcionalidades sem causar impactos significativos aos que o utilizam.

Comentários:

É muito importante visualizar e posicionar a Arquitetura Orientada a Serviços como um modelo de arquitetura que é agnóstico a qualquer plataforma de tecnologia. Ao fazer isso, uma empresa tem a liberdade para perseguir continuamente os objetivos estratégicos associados à computação orientada a serviços, aproveitando os avanços tecnológicos futuros.

(a) Errado. Nem todos os aplicativos devem ser implementados como um serviço. Ademais, devem permitir a integração de componentes de quaisquer plataformas – serviços são independentes de tecnologia.

(b) Correto. Não tem muito a acrescentar nesse item – observem que ele apenas cita HTTP (ele não restringe a esse protocolo).

Além da boa dosagem de análise e desenho em serviços, é importante que você trace claramente um conjunto de políticas relacionadas à gestão dos serviços. Boas práticas neste aspecto envolvem ter um repositório único na organização para manter questões de dependência e rastreabilidade entre artefatos de um serviço numa forma controlada, além de definir o tempo de guarda das versões.

(c) Correto. Uma boa prática é definir o tempo de guarda das versões de um serviço em funcionamento.

O que é o versionamento de serviços? Quando se está inserido num programa SOA, vários projetos de software são criados e realizados em paralelo. À medida que os projetos amadurecem, vários serviços são disponibilizados para uso pelas aplicações, seja para satisfazer algum critério da solução que está sendo criada, seja para disponibilizar tal serviço a um parceiro ou cliente.

(d) Correto. Como são projetos criados e realizados em paralelo e que evoluem como serviços, existem também versões (incluindo operações e implementações).

O que é autonomia? É a capacidade de um serviço se auto-administrar. Um serviço autônomo é aquele que independe de um elemento externo para executar sua lógica! Einh? Como é, professor? Vamos pensar, por exemplo, no ambiente de execução do serviço! A autonomia de serviços é um princípio que visa fornecer serviços com independência de seu ambiente de execução.

(e) Correto. A autonomia, como nós vimos em aula, é inclusive um dos princípios de design do SOA.

Gabarito: A

---

4. (FCC - 2011 - TRT - 4ª REGIÃO (RS) - Analista Judiciário - Tecnologia da Informação) Considere:

I. Abordagem arquitetural corporativa que permite a criação de serviços de negócio interoperáveis, que podem ser reutilizados e compartilhados entre aplicações e empresas.

II. As funcionalidades implementadas pelas aplicações devem ser disponibilizadas na forma de componentes e códigos interconectados por alto grau de acoplamento de controle e de dados.

III. É baseada no princípio de processamento centralizado que utiliza o paradigma de dados distribuídos para estabelecer a comunicação entre os sistemas clientes e os sistemas que implementam os serviços.

Quanto às características da arquitetura orientada a serviços - SOA, é correto o que consta em:

a) I, somente.
b) II, somente.
c) I e III, somente.
d) II e III, somente.
e) I, II e III.

Comentários:

O QUE SOA É...
Uma abordagem arquitetural corporativa que permite a criação de serviços de negócio interoperáveis, que podem ser reutilizados e compartilhados entre aplicações e empresas.

(I) Correto. Conforme vimos em aula, está perfeito.

> *Vocês puderam notar dois problemas claros da área de tecnologia da informação desse banco! Alto acoplamento, tendo em vista que os sistemas dependem excessivamente de serviços personalizados; e alta redundância, na medida em que*

há funcionalidades replicadas em diversos sistemas dentro do próprio banco. Galera, em um nível estratégico, isso significa perda de dinheiro!

(II) Errado. Conforme vimos em aula, temos baixo grau de acoplamento;

O QUE SOA É...
Uma abordagem distribuída (não-monolítica) para integração de arquiteturas baseadas no conceito de serviço.

(III) Errado. Conforme vimos em aula, o processamento é distribuído.

Gabarito: A

5. (FCC - 2011 - TRT - 4ª REGIÃO (RS) - Técnico Judiciário - Tecnologia da Informação) Na Arquitetura Orientada a Serviço - SOA, é INCORRETO afirmar que o serviço:

a) responde às requisições encapsulando todo o detalhe do seu processamento.

b) é um componente fortemente acoplado e altamente coeso que implementa uma função reutilizável de negócio.

c) não depende do estado de outros componentes externos para executar um ciclo completo de trabalho.

d) é uma unidade de trabalho oferecida pelo provedor de serviço para atender à demanda requerida por um consumidor de serviço.

e) é invocado por meio de protocolos de comunicação independentes da localização e do suporte tecnológico.

Comentários:

Vocês puderam notar dois problemas claros da área de tecnologia da informação desse banco! Alto acoplamento, tendo em vista que os sistemas dependem excessivamente de serviços personalizados; e alta redundância, na medida em que há funcionalidades replicadas em diversos sistemas dentro do próprio banco. Galera, em um nível estratégico, isso significa perda de dinheiro!

Conforme vimos em aula, é fracamente acoplado.

Gabarito: B

6. (FCC - 2009 - SEFAZ-SP - Agente Fiscal de Rendas - Tecnologia da Informação - Prova 3) A Service-Oriented Architecture - SOA trata-se de

I. um conjunto de produtos para implementar aplicativos dinâmicos e ágeis, do tipo loosely couple.

II. uma meta a ser alcançada, ou seja, disponibilizar uma metodologia de implementação que usa padrões e protocolos de linguagem específicos para execução de aplicativos.

III. soluções que não requerem uma renovação completa de tecnologia e de processo de negócios, que devem ser incrementais e baseadas nos investimentos atuais.

IV. uma abordagem de design de sistemas que orientam como os recursos do TI serão integrados e quais serviços serão expostos para o uso.

Está correto o que consta APENAS em:

a) I e II.
b) I e IV.
c) III e IV.
d) II e III.
e) II e IV.

Comentários:

O QUE SOA NÃO É...		
Não é uma tecnologia.	Não é um produto.	Não é um web service.
Não é um projeto de TI.	Não é um software.	Não é um framework.
Não é uma metodologia.	Não é solução de negócio.	Não é um middleware.
Não é um serviço.	Não é uma ferramenta.	

(I) Errado. Conforme vimos em aula, não se trata de um produto – pelo menos acertou na parte do *loosely coupled*.

(II) Errado. Conforme vimos em aula, não se trata de uma metodologia ou meta. Ademais não utiliza padrões e protocolos específicos.

(III) Correto. Conforme vimos em aula, realmente não requerem renovação completa de tecnologia e processos de negócio – é uma abordagem conceitual.

IV – Correto. Conforme vimos em aula, trata-se de uma abordagem arquitetura corporativa.

Gabarito: C

---

7. **(FCC - 2010 - DPE-SP - Agente de Defensoria - Analista de Sistemas)** Arquitetura orientada a serviço é um novo conceito, no qual cria-se um ambiente de descoberta dinâmico e se faz o uso de Serviços Web através da rede. NÃO é uma tecnologia usada nos serviços Web disponibilizados:

a) WSDL.
b) XML.
c) SOA.
d) SOAP.
e) UDDI.

Comentários:

O QUE SOA NÃO É...		
Não é uma tecnologia.	Não é um produto.	Não é um web service.
Não é um projeto de TI.	Não é um software.	Não é um framework.
Não é uma metodologia.	Não é solução de negócio.	Não é um middleware.
Não é um serviço.	Não é uma ferramenta.	

Conforme vimos em aula, a arquitetura orientada a serviços não é uma tecnologia.

Gabarito: C

---

8. **(FCC - 2011 – TRE/PE - Analista de Sistemas)** Arquitetura padrão proposta pelo Object Management Group (OMG) para estabelecer e simplificar a troca de dados entre sistemas distribuídos heterogêneos por meio de uma estrutura comum para o gerenciamento de objetos distribuídos que é conhecida como Object Manager Architecture (OMA). Trata-se de:

a) IDL.
b) RPC.
c) DCON.
d) CORBA.
e) COM.

Comentários:

O CORBA (Common Object Request Broker Architeture) é uma arquitetura padrão proposta pelo Object Management Group (OMG) para estabelecer e simplificar a troca de dados entre sistemas distribuídos heterogêneos por meio de uma estrutura comum para o gerenciamento de objetos distribuídos que é conhecida como Object Manager Architecture (OMA).

Conforme vimos em aula, trata-se do CORBA!

Gabarito: D

---

9. **(FCC - 2014 – TRF/4 - Analista de Sistemas)** Considere as definições tecnológicas de SOA abaixo.

I. É uma coleção de serviços (barramento de serviços).

II. Utiliza tecnologia de banco de dados para realizar a troca de mensagens.

III. Garante serviços altamente acoplados, fracamente coesos e com alta possibilidade de reutilização.

IV. O serviço, no ponto de vista da arquitetura SOA, é uma função de um sistema computacional que é disponibilizado para outro sistema na forma de um serviço.

V. Um serviço deve funcionar de forma dependente do estado de outros serviços a fim de criar uma interface bem definida, compatível e coerente com o estado do serviço do qual depende.

Está correto o que consta APENAS em:

a) II, III e IV.
b) I, III e IV.

c) I e IV.
d) V.
e) II e V.

Comentários:

(I) Correto, pode-se considerar como uma coleção de serviços; (II) Errado, não utiliza tecnologias de bancos de dados para troca de mensagens – a questão viajou; (III) Errado, garante serviços fracamente acoplados e altamente coesos; (IV) Correto, é realmente uma função do sistema disponibilizada como um serviço; (V) Errado, deve funcionar de forma independente (fracamente acoplada) do estado de outros serviços. Portanto, os itens corretos são I e IV.

Gabarito: C

---

10. **(FCC - 2013 – AL/RN - Analista de Sistemas)** Uma Arquitetura Orientada a Serviços (SOA) é uma forma de arquitetura de sistemas distribuídos que é tipicamente caraterizada pelo seguinte:

I. Visão lógica: O serviço é uma visão abstrata e lógica de programas, bancos de dados, processos de negócio etc. definida em termos de "o que isso faz", carregando em conjunto uma operação de nível de negócio.

II. Orientação de mensagens: O serviço é formalmente definido em termos de mensagens trocadas entre agentes prove- dores e requisitantes.

III. Orientada à descrição: Um serviço é descrito por um metadado que pode ser processado por uma máquina. Essa descrição expõe apenas detalhes que são importantes para o serviço.

IV. Granularidade: Serviços tendem a ser um pequeno número de operações com mensagens relativamente grandes e complexas.

Está correto que é exposto em:

a) I, II, III e IV.
b) III e IV apenas.
c) II e III, apenas.
d) I e II, apenas.
e) I, III e IV, apenas.

Comentários:

(I) Correto, o serviço é definido em termos funcionais, tipicamente implementando operações de negócios. A visão lógica do serviço o abstrai dos programas, bases de dados, processos de negócios, etc, necessários à sua implementação; (II) Correto, o serviço é formalmente definido em termos de mensagens trocadas entre unidades de software provedoras e consumidoras do serviço. A estrutura interna das unidades de software, incluindo sua linguagem de programação, sua estrutura de dados, etc. são deliberadamente abstraídas na SOA. Este baixo acoplamento entre aplicações provedoras e consumidoras de um serviço traz benefícios significativos para sistemas legados: a transparência da implementação interna das aplicações de uma arquitetura SOA permite que qualquer componente de software ou aplicação seja envolvido por uma interface de mensagens e adaptado a uma definição de serviços formal; (III) Correto, um serviço é descrito por uma metalinguagem processável por máquinas. A descrição suporta a natureza pública da SOA: apenas os detalhes do serviço relevantes a seu uso são incluídos na descrição; (IV) Correto, serviços tendem a utilizar um pequeno número de operações com mensagens relativamente grandes e complexas;

Gabarito: A

---

11. (FCC - 2013 – DPE/SP - Analista de Sistemas) A Arquitetura Orientada a Serviços (SOA) possui um modelo de referência que descreve diversas propriedades importantes do SOA. Uma dessas propriedades refere-se ao fato de que a descrição de um serviço deve fornecer dados suficientes para permitir que um consumidor e um provedor de serviços possam interagir entre si. A propriedade descrita recebe a denominação de:

a) funcionalidade do serviço.
b) acessibilidade do serviço.
c) política do serviço.
d) semântica do serviço.
e) conformidade do serviço.

Comentários:

A descrição de um serviço possui cinco propriedades: acessibilidade dos serviços; funcionalidades do serviço; políticas relacionadas com um serviço; interface do serviço; e limites da descrição. A acessibilidade é inerente ao relacionamento de

partes entre os provedores e consumidores de serviço. Contudo, a descrição de um serviço deve incluir dados suficientes para habilitar um consumidor de serviço e um provedor de serviço a interagirem entre si. Isto pode incluir metadados como a localização do serviço e o qual protocolo de informação ele suporta e requer. Ela pode também incluir a informação dinâmica sobre o serviço, como se ele está atualmente disponível.

Gabarito: B

---

12. **(FCC - 2013 – TRT/11 - Analista de Sistemas)** Em relação aos aspectos do projeto de serviços em SOA, é INCORRETO afirmar:

a) O meio de acesso ao serviço é estabelecido no Contrato de Serviço.
b) Os serviços têm controle sobre a lógica que os encapsulam.
c) Serviços são projetados para serem exteriormente descritos, e assim, serem encontrados e avaliados através de mecanismos de descobertas disponíveis.
d) A lógica dos serviços pode exceder ao que está descrito no contrato.
e) A lógica é dividida no serviço com a intenção de reúso.

Comentários:

(a) Correto, meios de acesso (entre outras coisas) é estabelecido no contrato do serviço; (b) Correto, é a alta coesão – cada serviço tem sua responsabilidade sobre a lógica que encapsula; (c) Correto, serviços devem ser "encontráveis" e são projetados para tal; (d) Errado, ela deve ser fidedigna ao contrato; (e) Correto, é um dos princípios da utilização de serviços.

Gabarito: D

---

13. **(FCC - 2008 – METRÔ/SP - Analista de Sistemas)** Enterprise Service Bus - ESB:

a) fortalece o acoplamento entre o serviço chamado e o meio de transporte.

b) implementa arquitetura orientada a serviço (SOA).

c) necessita de Web Services para ser implementado.

d) tem sua base construída a partir da quebra de funções básicas em partes, que são distribuídas onde for preciso.

e) auxilia no aumento de conexões ponto-a-ponto necessárias para permitir a comunicação entre aplicações.

Comentários:

O ESB é uma arquitetura para middlewares com o intuito de integrar serviços distribuídos por meio de diferentes formas de processamento de mensagens e tecnologias (plataformas, hardware, software, protocolos, etc). Por fim, existe ESB sem SOA e SOA sem ESB! Aliás, existe ESB até sem Web Service – pode-se inclusive utilizar outras formas de serviço. Logo, ESB não implementa SOA!

(a) Errado, ele reduz o acoplamento – esse é, inclusive, um dos princípios; (b) Errado, ele não implementa SOA; (c) Errado, não é obrigatório web services; (d) Correto, divide-se funções grandes em pequenas e distribui em forma de serviços; (e) Errado, não há essa relação.

Gabarito: D

---

14. **(FCC - 2012 – TJ/PE - Analista de Sistemas)** O Barramento de Serviços Corporativos (ESB):

I. Fornece um modelo de integração e implantação, permitindo o tráfego de mensagens locais e globais através de componentes de integração, adaptadores configuráveis, protegidos e gerenciados por um sistema integrado de segurança.

II. Pode suportar inúmeras tecnologias como J2EE, SOAP, WSDL, XML, BPEL etc.

III. Herda do SOA o conceito de serviços, mas não é a mesma coisa que SOA, pois não funciona numa filosofia de invocação de serviços (web), e sim de envio de mensagens de controle e dados.

IV. É igual a todas as soluções de integração de aplicações corporativas, onde interfaces dedicadas têm que ser mapeadas, desenhadas e configuradas para cada aplicação e tecnologias envolvidas.

Está correto o que se afirma em:

a) I, II, III e IV.
b) I, II e III, apenas.
c) II e III, apenas.

d) II, III e IV, apenas.
e) I e II, apenas.

Comentários:

O ESB é uma arquitetura para middlewares com o intuito de integrar serviços distribuídos por meio de diferentes formas de processamento de mensagens e tecnologias (plataformas, hardware, software, protocolos, etc). Por fim, existe ESB sem SOA e SOA sem ESB! Aliás, existe ESB até sem Web Service – pode-se inclusive utilizar outras formas de serviço. Logo, ESB não implementa SOA!

(I) Correto, item bem completo; (II) Correto, ele pode suportar infinitas tecnologias; (III) Correto, ESB é diferente de SOA e trabalha com o envio de mensagens e, não, seu consumo; (IV) Errado, não é igual a todas (EAI, por exemplo).

Gabarito: B

---

15. **(FCC - 2012 – TJ/PE - Analista de Sistemas)** Com relação ao Barramento de Serviços Corporativos (ESB) é INCORRETO afirmar:

a) Algumas das capacidades consideradas essenciais para um barramento de serviço corporativo (ESB) são: Resolução de Descrições de Serviços, Transformação de Mensagens e Roteamento Dinâmico de Mensagens.

b) Numa abordagem direcionada a API, o ESB define APIs específicas de plataforma e os fornecedores. Os consumidores utilizam essas APIs para implementar serviços e realizar chamadas. Um exemplo disso são as interfaces Java.

c) Um dos principais objetivos do ESB é prover conectividade para integrar diferentes plataformas de hardware e software, mesmo diante de diferentes middleware e protocolos.

d) Utilizar um ESB em uma arquitetura transforma-a em uma arquitetura orientada a serviços. Isso equivale a dizer que ESB implementa SOA.

e) Numa abordagem direcionada a protocolo, o ESB define um protocolo e os fornecedores. Os consumidores utilizam esse protocolo para enviar e receber mensagens. Um exemplo disso é Web Service utilizando SOAP.

Comentários:

O ESB trata abstrai várias preocupações que o arquiteto não precisa se preocupar. Algumas das capacidades consideradas essenciais para um barramento de serviço corporativo (ESB) são: Resolução de Descrições de Serviços; Transformação de Mensagens; Roteamento Dinâmico de Mensagens (Serviço de conectividade); Tratamento de Exceções; e Monitoramento de Mensagens.

O ESB é uma arquitetura para middlewares com o intuito de integrar serviços distribuídos por meio de diferentes formas de processamento de mensagens e tecnologias (plataformas, hardware, software, protocolos, etc). Por fim, existe ESB sem SOA e SOA sem ESB! Aliás, existe ESB até sem Web Service – pode-se inclusive utilizar outras formas de serviço. Logo, ESB não implementa SOA!

(a) Correto, conforme vimos em aula; (b) Correto, existem duas abordagens: direcionada a API, em que o ESB define APIs específicas de plataforma e os fornecedores e os consumidores as utilizam para implementarem serviços e realizarem chamadas. Um exemplo disso são as interfaces Java; e direcionada a Protocolo, em que ESB define um protocolo e os fornecedores e os consumidores o utilizam para enviar e receber mensagens. Um exemplo disso é Web Service utilizando SOAP; (c) Correto, conforme vimos em aula, ele deve prestar o serviço de conectividade; (d) Errado, conforme vimos em aula, ESB não implementa SOA; (e) Correto, conforme já vimos, também está perfeito.

Gabarito: D

---

16. **(FGV - 2009 - MEC - Analista de Sistemas - Especialista)** A Arquitetura Orientada a Serviços (SOA – Service Oriented Architecture) é uma abordagem arquitetural corporativa que permite a criação de serviços de negócios interoperáveis que podem facilmente ser reutilizados e compartilhados entre aplicações e empresas. Não é considerada característica relevante do SOA:

a) a distribuição.
b) a assincronia.
c) a composição.
d) o reuso "caixa-preta".
e) a heterogeneidade ambiental.

Comentários:

Todas são características do SOA! Heterogeneidade ambiental, i.e., serviços podem funcionar em ambientes completamente diferentes; Reúso Caixa-Preta, i.e., podem ser reutilizá-los sem precisar entender seu funcionamento interno, apenas suas interfaces e funcionalidades; Composição, i.e., podem se integrar com diversos outros componentes; distribuição, i.e., podem ser distribuídos amplamente em um registro de serviços; e, por fim, assincronia, que é a resposta da questão! Isso é um pouco polêmico. Ora, de fato, uma característica relevante do SOA é que seus serviços, em geral, são síncronos. No entanto, aqui é preciso ter "manha" para acertar. Tudo bem, ela implementa serviços assíncronos, mas ele diz "relevante", aí o aluno, infelizmente, tem também que "relevar" algumas coisas para poder acertar.

Gabarito: B

---

17. (FGV - 2009 - MEC - Analista de Sistemas - Especialista) Um Web Service é definido pela W3C como um sistema de software projetado para fornecer interoperabilidade entre máquinas em uma determinada rede. Dentro do contexto dos Web Services assinale a alternativa correta.

a) A UDDI (Universal Description, Discovery, and Integration) é uma linguagem baseada em XML que descreve o que um Web Service pode fazer, onde ele reside e como chamá-lo.

b) SOAP (Simple Object Access Protocol) é um protocolo, baseado em XML, para troca de informação estruturada com Web Services em redes de computadores.

c) A interoperabilidade entre os Web Services e aplicações é garantida devido ao uso obrigatório da linguagem Java na implementação das aplicações.

d) SOA (Simple Object Access) é uma plataforma de arquitetura orientada a serviços, utilizada como base para suportar os Web Services.

e) A WSDL (Web Services Description Language) é uma especificação para publicar e localizar informações sobre Web Services.

Comentários:

(a) Não, UDDI não é uma linguagem; (b) Sim, SOAP é um protocolo baseado em XML para troca de informações estruturadas; (c) Não, Java não é obrigatório; (d) SOA é o acrônimo de Service Oriented Architecture; (e) Não, isso é UDDI!

Gabarito: B

18. (FGV - 2008 - Senado Federal - Analista de Sistemas) Considere as seguintes assertivas sobre uma arquitetura orientada a serviços (SOA):

I. SOA é apenas uma implementação de Serviços Web, possuindo ambas as mesmas características.

II. As mensagens são o principal meio de comunicação entre os provedores e os consumidores de serviços.

III. SOA não prescreve como projetar ou construir a implementação do serviço.

IV. Quando os serviços são disponibilizados na web, eles são identificados por uma URI.

As assertivas corretas são:

a) somente I, II e III.
b) somente II, III e IV.
c) somente I, III e IV.
d) somente I, II e IV.
e) todas.

Comentários:

O QUE SOA NÃO É...		
Não é uma tecnologia.	Não é um produto.	Não é um web service.
Não é um projeto de TI.	Não é um software.	Não é um framework.
Não é uma metodologia.	Não é solução de negócio.	Não é um middleware.
Não é um serviço.	Não é uma ferramenta.	

Interação envolve a execução de ações na direção do serviço. Em geral, isto é realizado pelo envio/recebimento de mensagens, mas há modos que não envolvem explicitamente transmissão de mensagens (Ex: uma interação pode ser efetuada pela modificação do estado de um recurso compartilhado). Grosso modo, refere-se à troca de mensagens como modo primário de interação com um serviço.

(I) Errado, SOA não é WS; (II) Correto, a comunicação ocorre por meio de mensagens; (III) Correto, SOA não é prescritivo – ele não especificará como implementar serviços; (IV) Correto, isso realmente ocorre – o serviço é identificado por uma URI para que possa ser acessado.

Gabarito: B

---

19. **(CESGRANRIO – 2010 – IBGE – Analista de Sistemas)** Sabe-se que SOA é uma abordagem arquitetural corporativa que permite a criação de serviços de negócio interoperáveis que podem facilmente ser reutilizados e compartilhados entre aplicações e empresas. Sobre os princípios básicos da arquitetura SOA é INCORRETO afirmar que:

a) o alto acoplamento entre os serviços é um dos princípios básicos de SOA e define que o consumidor de um serviço deve conhecer os detalhes de sua implementação para que possa reagir de forma rápida quando mudanças ocorrerem.

b) o princípio de dividir para conquistar é muito conhecido há anos e tem como principal objetivo simplificar os problemas encontrados no dia-a-dia. Assim, seguindo esta ideia, os serviços devem ser capazes de se compor e serem acessados de forma a atender um problema maior.

c) os serviços devem ser reutilizáveis, ou seja, não devem carregar particularidades técnicas de uma implementação ou regra de negócio específica e devem ser genéricos o suficiente para atender outros projetos.

d) os serviços devem evitar a alocação de recursos por muito tempo e devem possuir a capacidade de serem encontrados, além de serem autônomos.

e) todo serviço deve ter um contrato formal que descreve o que o serviço faz e, para tal, padrões de mercado são muito utilizados.

Comentários:

- *Princípio do Baixo Acoplamento (Service Loose Coupling):*

O que é baixo acoplamento? Na minha época de estudos, eu sempre decorei acoplamento como a independência entre as partes! Se há baixo acoplamento, as partes são pouco dependentes; se há alto acoplamento, as partes são muito

dependentes. O baixo acoplamento de um serviço está relacionado com sua capacidade de ser independente de outros serviços para realizar sua tarefa.

Conforme vimos em aula, alto acoplamento não é um princípio básico! Pelo contrário, é o baixo acoplamento que é desejável.

Gabarito: A

---

20. (CESGRANRIO – 2009 – CASA DA MOEDA – Analista de Sistemas) Uma das principais características de uma Arquitetura Orientada a Serviços (SOA, na sigla em inglês) é o (a):

a) baixo acoplamento entre os serviços.

b) compartilhamento de sessão entre os serviços que rodam no mesmo servidor.

c) uso predominante de mensagens JMS.

d) exposição dos detalhes internos de cada serviço, facilitando o reúso dos mesmos.

e) ausência de interfaces predefinidas para serviços, já que esses são automaticamente descobertos pelos clientes.

Comentários:

- *Princípio do Baixo Acoplamento (Service Loose Coupling):*

O que é baixo acoplamento? Na minha época de estudos, eu sempre decorei acoplamento como a independência entre as partes! Se há baixo acoplamento, as partes são pouco dependentes; se há alto acoplamento, as partes são muito dependentes. O baixo acoplamento de um serviço está relacionado com sua capacidade de ser independente de outros serviços para realizar sua tarefa.

Conforme vimos em aula, trata-se do baixo acoplamento!

Gabarito: A

---

21. (FIP – 2009 – CMSJC – Analista de Sistemas) Entre as principais características da Arquitetura Orientada a Serviços (Service Oriented Architecture - SOA), é correto afirmar que:

a) a interoperabilidade entre os vários sistemas é feita através da personalização de interfaces proprietárias.

b) as aplicações SOA devem armazenar, localmente, os dados referentes ao estado das transações, enquanto aguardam o processamento realizado por outros serviços.

c) ela permite aumentar a agilidade no desenvolvimento de novos sistemas por meio da composição de serviços já existentes.

d) a criação de inventários de serviços impede o reaproveitamento sistemático de componentes lógicos.

e) o acoplamento fraco aumenta a dependência entre os serviços utilizados.

Comentários:

(a) Errado, quanto mais personalizado, menos interoperável;
(b) Errado, o Princípio da Alocação de Recurso preconiza o inverso;
(c) Certo, conforme preconiza o Princípio da Composição;
(d) Errado, um banco de serviços promove o reaproveitamento de componentes;
(e) Errado, o acoplamento fraco reduz a dependência entre os serviços.

Gabarito: C

---

22. (FGV - 2014 – DPE/RJ – Analista de Sistemas) A arquitetura orientada a serviço (SOA) estabelece um princípio de arquitetura para aplicações distribuídas que fornecem serviços que atendem uma função especifica do negocio. Fred, desenvolvedor de software, começou a implementar sua primeira aplicação adotando SOA. Fred empregará adequadamente os princípios da arquitetura orientada a serviços se:

a) apresentar detalhes técnicos da implementação para facilitar o entendimento por parte de um cliente que deseja utilizar os serviços da sua aplicação.

b) implementar as funcionalidades da sua aplicação na forma de bibliotecas de código que podem ser reutilizadas em outros softwares.

c) utilizar o paradigma funcional para assegurar que os princípios da arquitetura sejam atendidos.

d) garantir que as funções que implementam os serviços recebam requisições das aplicações cliente e as respondam ocultando detalhes do processamento da requisição.

e) utilizar serviços Web e deixar a orquestração dos serviços a cargo das aplicações clientes, pois isso aumenta o acoplamento e diminui a coesão.

Comentários:

(a) Errado, o Princípio da Abstração preconiza que detalhes técnicos sejam ocultos;
(b) Errado, códigos devem ser ocultos.
(c) Errado, não faz o menor sentido essa afirmação;
(d) Certo, de acordo com o Princípio da Abstração;
(e) Errado, o Princípio do Acoplamento preconiza baixo acoplamento.

Gabarito: D

---

23. (CESGRANRIO - 2008 – PETROBRÁS – Analista de Sistemas) A proposta de uma arquitetura orientada a serviços (SOA) prevê uma mudança de foco das aplicações "tradicionais". Este novo paradigma prevê a criação de conjuntos de serviços independentes no lugar de aplicações monolíticas, os quais sejam capazes de interagir entre si e de compor novos serviços de maior granularidade, aumentando a flexibilidade e respondendo de forma mais ágil a mudanças nos cenários de negócio. Qual dos apresentados a seguir NÃO constitui um princípio chave da orientação a serviços?

a) Reuso - a lógica é divida em serviços com a intenção de promover o reuso.

b) Autonomia - os serviços têm controle sobre a lógica que encapsulam.

c) Abstração - o serviço "esconde" do mundo exterior qualquer lógica que não conste de seu contrato.

d) Manutenção de estado - os serviços são projetados para reter o estado entre os acessos de clientes distintos.

e) Baixo acoplamento - os serviços mantêm relacionamentos que minimizam dependências e somente requerem que eles "saibam" da existência dos demais.

Comentários:

- *Princípio da Alocação de Recursos (Service Statelessness):*

Devido ao fato de que um serviço, por sua natureza, deve futuramente ser reutilizado, deve-se tomar o cuidado de não se criar muitas instâncias deste serviço, onerando a infraestrutura que o mantém. Em outras palavras, para um serviço ter disponibilidade, ele deve evitar reter informações específicas a uma determinada atividade, se possível deslocando a sobrecarga para outro componente externo.

Os serviços devem evitar a alocação de recursos por muito tempo! Recomenda-se a construção de serviços stateless (sem estado). Galera, isso não é muito fácil de alcançar. A otimização da alocação de recursos é difícil por conta da natureza distribuída e autônoma dos serviços, além da natureza de mudanças constantes dos ambientes que utilizam essa arquitetura.

Conforme vimos em aula, o Princípio da Alocação de Recursos (ou Manutenção de Estados) preconiza que se evite reter o estado.

Gabarito: D

---

24. (CESGRANRIO - 2010 – ELETROBRÁS - Analista de Sistemas) No CORBA, a linguagem utilizada para definir interfaces para objetos na rede é denominada:

a) C.
b) Assembly.
c) SmallTalk.
d) VB-6.
e) IDL

Comentários:

Além disso, ele utiliza a IDL (Interface Definition Language), uma linguagem utilizada para descrever as interfaces das implementações dos objetos na rede (seria algo

similar ao WSDL). Uma interface escrita em IDL especifica o formato da chamada das operações providas pelo objeto e os parâmetros de entrada ou saída necessários para efetuar a operação.

Conforme vimos em aula, trata-se da IDL.

Gabarito: E

---

25. **(CESGRANRIO - 2011 – FINEP - Analista de Sistemas)** Uma Arquitetura Orientada a Serviços (SOA) é essencialmente uma coleção de serviços que se comunicam entre si. Dessa forma, é preciso que existam mecanismos para conectar tais serviços. Nesse contexto, o middleware responsável por fornecer a infraestrutura para a comunicação entre esses serviços é o:

a) Enterprise Service Bus (ESB)
b) Common Object Request Broker Architecture (CORBA)
c) Remote Procedure Call (RPC)
d) Remote Method Invocation (RMI)
e) Distributed Component Object Model (DCOM)

Comentários:

Trata-se, portanto, de uma arquitetura para middlewares com o intuito de integrar serviços distribuídos por meio de diferentes formas de processamento de mensagens e tecnologias (plataformas, hardware, software, protocolos, etc). Por fim, existe ESB sem SOA e SOA sem ESB! Aliás, existe ESB até sem Web Service – pode-se inclusive utilizar outras formas de serviço. Logo, ESB não implementa SOA!

Conforme vimos em aula, não é um middleware, é uma arquitetura para middleware – por essa razão, a questão foi anulada.

Gabarito: X

ACERTEI	ERREI

LISTA DE EXERCÍCIOS COMENTADOS (DIVERSAS BANCAS)

ARQUITETURA ORIENTADA A SERVIÇOS (SOA)

1. (FCC - 2012 - TRT - 6ª Região (PE) - Técnico Judiciário - Tecnologia da Informação)

Cinco perguntas que você precisa saber antes de investir em SOA

...O que significa efetivamente ter uma governança de SOA?

O tão falado alinhamento da organização é uma das principais preocupações atuais. Um processo unificado de governança faz com que sejam melhorados os negócios da companhia de forma geral. No entanto, não são necessários novos sistemas ou ferramentas que vão melhorar o sistema de gerenciamento a ponto de integrar TI e gestão. A chamada governança de SOA é compartilhar objetivos. O importantes é ter cada stakeholders representado no momento da elaboração de um projeto SOA. Ter algum sistema de gerenciamento de serviço, como ITIL, também colabora para dar uma visibilidade ao cliente.

Sobre SOA, e com base no texto, é correto afirmar que:

a) é essencial que uma empresa adote as melhores práticas da ITIL antes de implantar o SOA.

b) SOA é uma ferramenta de software utilizada no gerenciamento de serviços de TI.

c) SOA, neste contexto, se refere à sigla para Society Of Actuaries, uma organização educacional, profissional e de pesquisa com sede nos Estados Unidos.

d) SOA é uma abordagem de projeto baseada em padrões para a criação de uma infraestrutura de TI integrada capaz de responder rapidamente às mudanças nas necessidades de negócios.

e) a implantação do SOA numa empresa, por si só, é suficiente para garantir o alinhamento dos negócios com TI.

2. (FCC - 2012 - TRT - 11ª Região (AM) - Analista Judiciário - Tecnologia da Informação) Em SOA:

a) normalmente, são utilizados WSDL para descrever os próprios serviços e SOAP para descrever os protocolos de comunicação.

b) a tecnologia utilizada para prover o serviço, tal como uma linguagem de programação é parte da definição do serviço.

c) orquestração é o processo de sequenciar serviços e prover uma lógica adicional para processar dados, levando em conta a representação de dados.

d) um dado serviço de broker não requer do provedor a necessidade de definição de listas categorizadas dos serviços.

e) um serviço, do ponto de vista da arquitetura, deve funcionar de forma independente do estado de outros serviços, inclusive nos casos de composite services.

3. (FCC - 2012 - TJ-PE - Técnico Judiciário - Programador de Computador) Sobre SOA é INCORRETO afirmar:

a) Quando se utiliza SOA, todos os aplicativos desenvolvidos em uma corporação devem ser implementados de forma que possam prover serviços que permitirão a integração de componentes de uma única plataforma.

b) Web Services a grosso modo podem ser classificados como métodos remotos publicados na Web, que através do uso do protocolo SOAP e XML, permitem a exposição de métodos de aplicativos diversos na Web, para consumo por qualquer outro aplicativo ou dispositivo que utilize o HTTP.

c) Uma consideração importante a respeito de versionamento é definir por quanto tempo é necessário manter as diferentes versões de um serviço em funcionamento.

d) O versionamento assume a existência simultânea de versões de um serviço, incluindo as suas operações e suas diferentes implementações.

e) Um conceito importante por trás da arquitetura orientada a serviços está na autonomia; a possibilidade de se distribuir, modificar e manter, independentemente de outros sistemas (consumidores), novas funcionalidades sem causar impactos significativos aos que o utilizam.

4. (FCC - 2011 - TRT - 4ª REGIÃO (RS) - Analista Judiciário - Tecnologia da Informação) Considere:

I. Abordagem arquitetural corporativa que permite a criação de serviços de negócio interoperáveis, que podem ser reutilizados e compartilhados entre aplicações e empresas.

II. As funcionalidades implementadas pelas aplicações devem ser disponibilizadas na forma de componentes e códigos interconectados por alto grau de acoplamento de controle e de dados.

III. É baseada no princípio de processamento centralizado que utiliza o paradigma de dados distribuídos para estabelecer a comunicação entre os sistemas clientes e os sistemas que implementam os serviços.

Quanto às características da arquitetura orientada a serviços - SOA, é correto o que consta em:

a) I, somente.
b) II, somente.
c) I e III, somente.
d) II e III, somente.
e) I, II e III.

5. (FCC - 2011 - TRT - 4ª REGIÃO (RS) - Técnico Judiciário - Tecnologia da Informação) Na Arquitetura Orientada a Serviço - SOA, é INCORRETO afirmar que o serviço:

a) responde às requisições encapsulando todo o detalhe do seu processamento.

b) é um componente fortemente acoplado e altamente coeso que implementa uma função reutilizável de negócio.

c) não depende do estado de outros componentes externos para executar um ciclo completo de trabalho.

d) é uma unidade de trabalho oferecida pelo provedor de serviço para atender à demanda requerida por um consumidor de serviço.

e) é invocado por meio de protocolos de comunicação independentes da localização e do suporte tecnológico.

6. (FCC - 2009 - SEFAZ-SP - Agente Fiscal de Rendas - Tecnologia da Informação - Prova 3) A Service-Oriented Architecture - SOA trata-se de

I. um conjunto de produtos para implementar aplicativos dinâmicos e ágeis, do tipo loosely couple.

II. uma meta a ser alcançada, ou seja, disponibilizar uma metodologia de implementação que usa padrões e protocolos de linguagem específicos para execução de aplicativos.

III. soluções que não requerem uma renovação completa de tecnologia e de processo de negócios, que devem ser incrementais e baseadas nos investimentos atuais.

IV. uma abordagem de design de sistemas que orientam como os recursos do TI serão integrados e quais serviços serão expostos para o uso.

Está correto o que consta APENAS em:

a) I e II.
b) I e IV.
c) III e IV.
d) II e III.
e) II e IV.

7. (FCC - 2010 - DPE-SP - Agente de Defensoria - Analista de Sistemas) Arquitetura orientada a serviço é um novo conceito, no qual cria-se um ambiente de descoberta dinâmico e se faz o uso de Serviços Web através da rede. NÃO é uma tecnologia usada nos serviços Web disponibilizados:

a) WSDL.
b) XML.
c) SOA.
d) SOAP.
e) UDDI.

8. (FCC - 2011 – TRE/PE - Analista de Sistemas) Arquitetura padrão proposta pelo Object Management Group (OMG) para estabelecer e simplificar a troca de dados entre sistemas distribuídos heterogêneos por meio de uma estrutura comum para o gerenciamento de objetos distribuídos que é conhecida como Object Manager Architecture (OMA). Trata-se de:

a) IDL.
b) RPC.
c) DCON.
d) CORBA.
e) COM.

9. (FCC - 2014 – TRF/4 - Analista de Sistemas) Considere as definições tecnológicas de SOA abaixo.

I. É uma coleção de serviços (barramento de serviços).

II. Utiliza tecnologia de banco de dados para realizar a troca de mensagens.

III. Garante serviços altamente acoplados, fracamente coesos e com alta possibilidade de reutilização.

IV. O serviço, no ponto de vista da arquitetura SOA, é uma função de um sistema computacional que é disponibilizado para outro sistema na forma de um serviço.

V. Um serviço deve funcionar de forma dependente do estado de outros serviços a fim de criar uma interface bem definida, compatível e coerente com o estado do serviço do qual depende.

Está correto o que consta APENAS em:

a) II, III e IV.
b) I, III e IV.
c) I e IV.
d) V.
e) II e V.

10. (FCC - 2013 – AL/RN - Analista de Sistemas) Uma Arquitetura Orientada a Serviços (SOA) é uma forma de arquitetura de sistemas distribuídos que é tipicamente caraterizada pelo seguinte:

I. Visão lógica: O serviço é uma visão abstrata e lógica de programas, bancos de dados, processos de negócio etc. definida em termos de "o que isso faz", carregando em conjunto uma operação de nível de negócio.

II. Orientação de mensagens: O serviço é formalmente definido em termos de mensagens trocadas entre agentes prove- dores e requisitantes.

III. Orientada à descrição: Um serviço é descrito por um metadado que pode ser processado por uma máquina. Essa descrição expõe apenas detalhes que são importantes para o serviço.

IV. Granularidade: Serviços tendem a ser um pequeno número de operações com mensagens relativamente grandes e complexas.

Está correto que é exposto em:

a) I, II, III e IV.
b) III e IV apenas.
c) II e III, apenas.
d) I e II, apenas.
e) I, III e IV, apenas.

11. (FCC - 2013 – DPE/SP - Analista de Sistemas) A Arquitetura Orientada a Serviços (SOA) possui um modelo de referência que descreve diversas propriedades importantes do SOA. Uma dessas propriedades refere-se ao fato de que a descrição de um serviço deve fornecer dados suficientes para permitir que um consumidor e um provedor de serviços possam interagir entre si. A propriedade descrita recebe a denominação de:

a) funcionalidade do serviço.
b) acessibilidade do serviço.
c) política do serviço.
d) semântica do serviço.
e) conformidade do serviço.

12. (FCC - 2013 – TRT/11 - Analista de Sistemas) Em relação aos aspectos do projeto de serviços em SOA, é INCORRETO afirmar:

a) O meio de acesso ao serviço é estabelecido no Contrato de Serviço.

b) Os serviços têm controle sobre a lógica que os encapsulam.
c) Serviços são projetados para serem exteriormente descritos, e assim, serem encontrados e avaliados através de mecanismos de descobertas disponíveis.
d) A lógica dos serviços pode exceder ao que está descrito no contrato.
e) A lógica é dividida no serviço com a intenção de reúso.

13. **(FCC - 2008 – METRÔ/SP - Analista de Sistemas)** Enterprise Service Bus - ESB:

a) fortalece o acoplamento entre o serviço chamado e o meio de transporte.

b) implementa arquitetura orientada a serviço (SOA).

c) necessita de Web Services para ser implementado.

d) tem sua base construída a partir da quebra de funções básicas em partes, que são distribuídas onde for preciso.

e) auxilia no aumento de conexões ponto-a-ponto necessárias para permitir a comunicação entre aplicações.

14. **(FCC - 2012 – TJ/PE - Analista de Sistemas)** O Barramento de Serviços Corporativos (ESB):

I. Fornece um modelo de integração e implantação, permitindo o tráfego de mensagens locais e globais através de componentes de integração, adaptadores configuráveis, protegidos e gerenciados por um sistema integrado de segurança.

II. Pode suportar inúmeras tecnologias como J2EE, SOAP, WSDL, XML, BPEL etc.

III. Herda do SOA o conceito de serviços, mas não é a mesma coisa que SOA, pois não funciona numa filosofia de invocação de serviços (web), e sim de envio de mensagens de controle e dados.

IV. É igual a todas as soluções de integração de aplicações corporativas, onde interfaces dedicadas têm que ser mapeadas, desenhadas e configuradas para cada aplicação e tecnologias envolvidas.

Está correto o que se afirma em:

a) I, II, III e IV.

b) I, II e III, apenas.
c) II e III, apenas.
d) II, III e IV, apenas.
e) I e II, apenas.

15. (FCC - 2012 – TJ/PE - Analista de Sistemas) Com relação ao Barramento de Serviços Corporativos (ESB) é INCORRETO afirmar:

a) Algumas das capacidades consideradas essenciais para um barramento de serviço corporativo (ESB) são: Resolução de Descrições de Serviços, Transformação de Mensagens e Roteamento Dinâmico de Mensagens.

b) Numa abordagem direcionada a API, o ESB define APIs específicas de plataforma e os fornecedores. Os consumidores utilizam essas APIs para implementar serviços e realizar chamadas. Um exemplo disso são as interfaces Java.

c) Um dos principais objetivos do ESB é prover conectividade para integrar diferentes plataformas de hardware e software, mesmo diante de diferentes middleware e protocolos.

d) Utilizar um ESB em uma arquitetura transforma-a em uma arquitetura orientada a serviços. Isso equivale a dizer que ESB implementa SOA.

e) Numa abordagem direcionada a protocolo, o ESB define um protocolo e os fornecedores. Os consumidores utilizam esse protocolo para enviar e receber mensagens. Um exemplo disso é Web Service utilizando SOAP.

16. (FGV - 2009 - MEC - Analista de Sistemas - Especialista) A Arquitetura Orientada a Serviços (SOA – Service Oriented Architecture) é uma abordagem arquitetural corporativa que permite a criação de serviços de negócios interoperáveis que podem facilmente ser reutilizados e compartilhados entre aplicações e empresas. Não é considerada característica relevante do SOA:

a) a distribuição.
b) a assincronia.
c) a composição.
d) o reuso “caixa-preta”.
e) a heterogeneidade ambiental.

17. **(FGV - 2009 - MEC - Analista de Sistemas - Especialista)** Um Web Service é definido pela W3C como um sistema de software projetado para fornecer interoperabilidade entre máquinas em uma determinada rede. Dentro do contexto dos Web Services assinale a alternativa correta.

    a) A UDDI (Universal Description, Discovery, and Integration) é uma linguagem baseada em XML que descreve o que um Web Service pode fazer, onde ele reside e como chamá-lo.

    b) SOAP (Simple Object Access Protocol) é um protocolo, baseado em XML, para troca de informação estruturada com Web Services em redes de computadores.

    c) A interoperabilidade entre os Web Services e aplicações é garantida devido ao uso obrigatório da linguagem Java na implementação das aplicações.

    d) SOA (Simple Object Access) é uma plataforma de arquitetura orientada a serviços, utilizada como base para suportar os Web Services.

    e) A WSDL (Web Services Description Language) é uma especificação para publicar e localizar informações sobre Web Services.

18. **(FGV - 2008 - Senado Federal - Analista de Sistemas)** Considere as seguintes assertivas sobre uma arquitetura orientada a serviços (SOA):

    I. SOA é apenas uma implementação de Serviços Web, possuindo ambas as mesmas características.

    II. As mensagens são o principal meio de comunicação entre os provedores e os consumidores de serviços.

    III. SOA não prescreve como projetar ou construir a implementação do serviço.

    IV. Quando os serviços são disponibilizados na web, eles são identificados por uma URI.

As assertivas corretas são:

a) somente I, II e III.
b) somente II, III e IV.
c) somente I, III e IV.

d) somente I, II e IV.
e) todas.

19. (CESGRANRIO – 2010 – IBGE – Analista de Sistemas) Sabe-se que SOA é uma abordagem arquitetural corporativa que permite a criação de serviços de negócio interoperáveis que podem facilmente ser reutilizados e compartilhados entre aplicações e empresas. Sobre os princípios básicos da arquitetura SOA é INCORRETO afirmar que:

a) o alto acoplamento entre os serviços é um dos princípios básicos de SOA e define que o consumidor de um serviço deve conhecer os detalhes de sua implementação para que possa reagir de forma rápida quando mudanças ocorrerem.

b) o princípio de dividir para conquistar é muito conhecido há anos e tem como principal objetivo simplificar os problemas encontrados no dia-a-dia. Assim, seguindo esta ideia, os serviços devem ser capazes de se compor e serem acessados de forma a atender um problema maior.

c) os serviços devem ser reutilizáveis, ou seja, não devem carregar particularidades técnicas de uma implementação ou regra de negócio específica e devem ser genéricos o suficiente para atender outros projetos.

d) os serviços devem evitar a alocação de recursos por muito tempo e devem possuir a capacidade de serem encontrados, além de serem autônomos.

e) todo serviço deve ter um contrato formal que descreve o que o serviço faz e, para tal, padrões de mercado são muito utilizados.

20. (CESGRANRIO – 2009 – CASA DA MOEDA – Analista de Sistemas) Uma das principais características de uma Arquitetura Orientada a Serviços (SOA, na sigla em inglês) é o (a):

a) baixo acoplamento entre os serviços.

b) compartilhamento de sessão entre os serviços que rodam no mesmo servidor.

c) uso predominante de mensagens JMS.

d) exposição dos detalhes internos de cada serviço, facilitando o reúso dos mesmos.

e) ausência de interfaces predefinidas para serviços, já que esses são automaticamente descobertos pelos clientes.

21. (FIP – 2009 – CMSJC – Analista de Sistemas) Entre as principais características da Arquitetura Orientada a Serviços (Service Oriented Architecture - SOA), é correto afirmar que:

a) a interoperabilidade entre os vários sistemas é feita através da personalização de interfaces proprietárias.

b) as aplicações SOA devem armazenar, localmente, os dados referentes ao estado das transações, enquanto aguardam o processamento realizado por outros serviços.

c) ela permite aumentar a agilidade no desenvolvimento de novos sistemas por meio da composição de serviços já existentes.

d) a criação de inventários de serviços impede o reaproveitamento sistemático de componentes lógicos.

e) o acoplamento fraco aumenta a dependência entre os serviços utilizados.

22. (FGV - 2014 – DPE/RJ – Analista de Sistemas) A arquitetura orientada a serviço (SOA) estabelece um princípio de arquitetura para aplicações distribuídas que fornecem serviços que atendem uma função especifica do negocio. Fred, desenvolvedor de software, começou a implementar sua primeira aplicação adotando SOA. Fred empregará adequadamente os princípios da arquitetura orientada a serviços se:

a) apresentar detalhes técnicos da implementação para facilitar o entendimento por parte de um cliente que deseja utilizar os serviços da sua aplicação.

b) implementar as funcionalidades da sua aplicação na forma de bibliotecas de código que podem ser reutilizadas em outros softwares.

c) utilizar o paradigma funcional para assegurar que os princípios da arquitetura sejam atendidos.

d) garantir que as funções que implementam os serviços recebam requisições das aplicações cliente e as respondam ocultando detalhes do processamento da requisição.

e) utilizar serviços Web e deixar a orquestração dos serviços a cargo das aplicações clientes, pois isso aumenta o acoplamento e diminui a coesão.

23. (CESGRANRIO - 2008 – PETROBRÁS – Analista de Sistemas) A proposta de uma arquitetura orientada a serviços (SOA) prevê uma mudança de foco das aplicações "tradicionais". Este novo paradigma prevê a criação de conjuntos de serviços independentes no lugar de aplicações monolíticas, os quais sejam capazes de interagir entre si e de compor novos serviços de maior granularidade, aumentando a flexibilidade e respondendo de forma mais ágil a mudanças nos cenários de negócio. Qual dos apresentados a seguir NÃO constitui um princípio chave da orientação a serviços?

a) Reuso - a lógica é divida em serviços com a intenção de promover o reuso.

b) Autonomia - os serviços têm controle sobre a lógica que encapsulam.

c) Abstração - o serviço "esconde" do mundo exterior qualquer lógica que não conste de seu contrato.

d) Manutenção de estado - os serviços são projetados para reter o estado entre os acessos de clientes distintos.

e) Baixo acoplamento - os serviços mantêm relacionamentos que minimizam dependências e somente requerem que eles "saibam" da existência dos demais.

24. (CESGRANRIO - 2010 – ELETROBRÁS - Analista de Sistemas) No CORBA, a linguagem utilizada para definir interfaces para objetos na rede é denominada:

a) C.
b) Assembly.
c) SmallTalk.
d) VB-6.
e) IDL

25. (CESGRANRIO - 2011 – FINEP - Analista de Sistemas) Uma Arquitetura Orientada a Serviços (SOA) é essencialmente uma coleção de serviços que se comunicam entre si. Dessa forma, é preciso que existam mecanismos para conectar tais serviços. Nesse contexto, o middleware responsável por fornecer a infraestrutura para a comunicação entre esses serviços é o:

a) Enterprise Service Bus (ESB)
b) Common Object Request Broker Architecture (CORBA)
c) Remote Procedure Call (RPC)
d) Remote Method Invocation (RMI)
e) Distributed Component Object Model (DCOM)

GABARITO DOS EXERCÍCIOS COMENTADOS (DIVERSAS BANCAS)

ARQUITETURA ORIENTADA A SERVIÇOS (SOA)

1	2	3	4	5	6	7	8	9	10
D	A	A	A	B	C	C	D	C	A
11	12	13	14	15	16	17	18	19	20
B	D	D	B	D	B	B	B	A	A
21	22	23	24	25	26	27	28	29	30
C	D	D	E	X					

ESSA LEI TODO MUNDO CONHECE:
PIRATARIA É CRIME.

Mas é sempre bom revisar o porquê e como você pode ser prejudicado com essa prática.

1.

Professor investe seu tempo para elaborar os cursos e o site os coloca à venda.

2.

Pirata divulga ilicitamente (grupos de rateio), utilizando-se do anonimato, nomes falsos ou laranjas (geralmente o pirata se anuncia como formador de "grupos solidários" de rateio que não visam lucro).

3.

Pirata cria alunos fake praticando falsidade ideológica, comprando cursos do site em nome de pessoas aleatórias (usando nome, CPF, endereço e telefone de terceiros sem autorização).

4.

Pirata compra, muitas vezes, clonando cartões de crédito (por vezes o sistema anti-fraude não consegue identificar o golpe a tempo).

5.

Pirata fere os Termos de Uso, adultera as aulas e retira a identificação dos arquivos PDF (justamente porque a atividade é ilegal e ele não quer que seus fakes sejam identificados).

6.

Pirata revende as aulas protegidas por direitos autorais, praticando concorrência desleal e em flagrante desrespeito à Lei de Direitos Autorais (Lei 9.610/98).

7.

Concurseiro(a) desinformado participa de rateio, achando que nada disso está acontecendo e esperando se tornar servidor público para exigir o cumprimento das leis.

8.

O professor que elaborou o curso não ganha nada, o site não recebe nada, e a pessoa que praticou todos os ilícitos anteriores (pirata) fica com o lucro.

Deixando de lado esse mar de sujeira, aproveitamos para agradecer a todos que adquirem os cursos honestamente e permitem que o site continue existindo.