

## **Aula 00**

*PREVIC (Analista Administrativo -  
Tecnologia da Informação) Engenharia  
de Software*

Autor:  
**Diego Carvalho**

28 de Agosto de 2023

# Índice

1) Apresentação do Prof. Diego Carvalho - Informática .....	3
2) Apresentação Flashcards .....	5
3) SOA .....	7
4) SOA - Conceitos-Chaves .....	12
5) SOA - Objetivos e Benefícios .....	14
6) SOA - Modelos Arquitetônicos .....	16
7) SOA - Princípios de Design .....	20
8) SOA - Composição de Serviço .....	26
9) SOA - Manifesto SOA .....	30
10) SOA - Versionamento de Serviços .....	32
11) SOA - Padrões SOA .....	34
12) SOA - Sistemas de Governança .....	35
13) SOA - Mensageria e CORBA .....	36
14) Questões Comentadas - SOA - CESPE .....	41
15) Questões Comentadas - SOA - FCC .....	54
16) Questões Comentadas - SOA - FGV .....	65
17) Questões Comentadas - SOA - Diversas .....	68
18) Lista de Questões - SOA - CESPE .....	73
19) Lista de Questões - SOA - FCC .....	79
20) Lista de Questões - SOA - FGV .....	88
21) Lista de Questões - SOA - Diversas .....	92



## APRESENTAÇÃO DO PROFESSOR

# PROF. DIEGO CARVALHO

FORMADO EM CIÊNCIA DA COMPUTAÇÃO PELA UNIVERSIDADE DE BRASÍLIA (UNB), PÓS-GRADUADO EM GESTÃO DE TECNOLOGIA DA INFORMAÇÃO NA ADMINISTRAÇÃO PÚBLICA E, ATUALMENTE, AUDITOR FEDERAL DE FINANÇAS E CONTROLE DA SECRETARIA DO TESOURO NACIONAL.

## ESTRATÉGIA CONCURSOS

 PROFESSOR DIEGO CARVALHO - [WWW.INSTAGRAM.COM/PROFESSORDIEGOCARVALHO](https://www.instagram.com/professordiegovalho)



**Sobre o curso: galera, todos os tópicos da aula possuem Faixas de Incidência, que indicam se o assunto cai muito ou pouco em prova.** Diego, se cai pouco para que colocar em aula? Cair pouco não significa que não cairá justamente na sua prova! A ideia aqui é: se você está com pouco tempo e precisa ver somente aquilo que cai mais, você pode filtrar pelas incidências média, alta e altíssima; se você tem tempo sobrando e quer ver tudo, vejam também as incidências baixas e baixíssimas. *Fechado?*

INCIDÊNCIA EM PROVA: BAIXÍSSIMA

INCIDÊNCIA EM PROVA: BAIXA

INCIDÊNCIA EM PROVA: MÉDIA

INCIDÊNCIA EM PROVA: ALTA

INCIDÊNCIA EM PROVA: ALTÍSSIMA

Além disso, essas faixas não são por banca – é baseado tanto na quantidade de vezes que caiu em prova independentemente da banca quanto nas minhas próprias avaliações sobre cada assunto.



#ATENÇÃO

# Avisos Importantes



## O curso abrange todos os níveis de conhecimento...

Esse curso foi desenvolvido para ser acessível a **alunos com diversos níveis de conhecimento diferentes**. Temos alunos mais avançados que têm conhecimento prévio ou têm facilidade com o assunto. Por outro lado, temos alunos iniciantes, que nunca tiveram contato com a matéria ou até mesmo que têm trauma dessa disciplina. A ideia aqui é tentar atingir ambos os públicos - iniciantes e avançados - da melhor maneira possível..



## Por que estou enfatizando isso?

O **material completo** é composto de muitas histórias pessoais, exemplos, metáforas, piadas, memes, questões, desafios, esquemas, diagramas, imagens, entre outros. Já o **material simplificado** possui exatamente o mesmo núcleo do material completo, mas ele é menor e mais objetivo. *Professor, eu devo estudar por qual material?* Se você quiser se aprofundar nos assuntos ou tem dificuldade com a matéria, necessitando de um material mais passo-a-passo, utilize o material completo. Se você não quer se aprofundar nos assuntos ou tem facilidade com a matéria, necessitando de um material mais direto ao ponto, utilize o material simplificado.



## Por fim...

O curso contém diversas questões espalhadas em meio à teoria. Essas questões possuem um comentário mais simplificado porque **têm o único objetivo de apresentar ao aluno como bancas de concurso cobram o assunto previamente administrado**. A imensa maioria das questões para que o aluno avalie seus conhecimentos sobre a matéria estão dispostas ao final da aula na lista de exercícios e **possuem comentários bem mais abrangentes**.



# ESTRATÉGIA FLASHCARDS

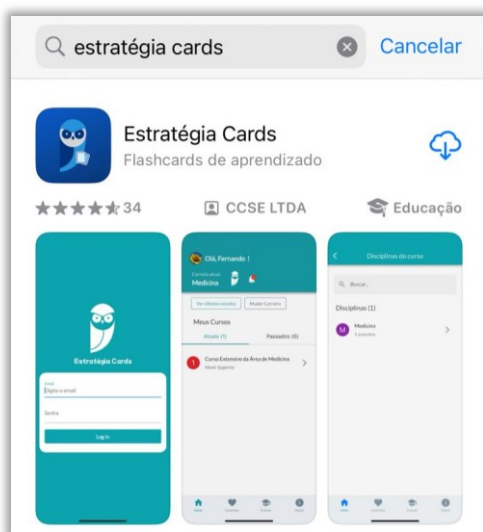
📱 Você tem dificuldade de estudar, memorizar e revisar os conteúdos que estuda em nossas aulas? Então nós temos a ferramenta perfeita para você!

Apresentamos o **Estratégia Cards**: app de flashcards que vai revolucionar sua forma de **estudar** e **revisar** conteúdos de provas de concurso público. Com nossa tecnologia inovadora e interface amigável, você dominará os tópicos mais complexos de maneira eficiente e divertida.

## 🌟 Recursos do Estratégia Cards:

<b>Curadoria de Flashcards</b>	Flashcards criados e revisados por professores especializados em cada área, com qualidade e voltados para concursos públicos.
<b>Flashcards Personalizados</b>	Crie seus próprios flashcards, cobrindo os principais tópicos e matérias dos concursos públicos.
<b>Repetição Espaçada</b>	Técnica de aprendizagem que envolve revisar informações em intervalos crescentes para melhorar a retenção de longo prazo e combater o esquecimento.
<b>Estatísticas Personalizadas</b>	Visualize graficamente o percentual de acertos, erros ou dúvidas dos decks estudados.
<b>Modo Offline</b>	Estude em qualquer lugar, mesmo sem conexão à internet, fazendo o download dos decks.
<b>Estudo por Áudio</b>	<i>Está dirigindo ou fazendo esteira e quer continuar estudando?</i> Basta utilizar a opção “Escutar”.
<b>Decks Favoritos</b>	Você pode escolher decks específicos como favoritos e visualizá-los em uma aba separada do app.
<b>Opções de Estudo</b>	Você poderá estudar todos os cards de um deck; ou apenas os que você errou; ou apenas os que você não estudou ainda; entre outras opções.

## 📱 E como eu consigo baixar?



É muito fácil! Basta pesquisar por “Estratégia Cards” na loja oficial do seu smartphone.

Se você tiver um Android, basta acessar a **Google Play**;



Se for tiver um iPhone, basta acessar a **App Store (iOS)**.



## É para acessar?

Para acessar, basta ter uma conta no Estratégia Concursos. Em seguida, utilize suas credenciais de login e senha para acessar o aplicativo. Por fim, acessa a carreira de Tecnologia da Informação.

## Como utilizar o app:



# ARQUITETURA ORIENTADA A SERVIÇOS (SOA)

## Introdução/Motivação

INCIDÊNCIA EM PROVA: MÉDIA

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 mais um pequeno 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 reuso".*

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, o que eu quero é apenas a energia...

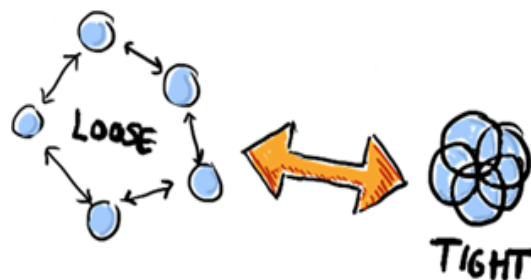
**Eu não ligo para como eles vão fazer essa energia aparecer, eu só quero que ela sempre esteja disponível na minha casa.** Thomas Erl destaca também outros benefícios: serviços são reutilizáveis; compartilham um contrato formal; possuem baixo acoplamento; abstraem a lógica;



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 juntar esses dois conceitos! **A OASIS<sup>1</sup> 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, isto é, 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.

**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.

<sup>1</sup> 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.





E qual foi a motivação para o surgimento da arquitetura orientada a serviços? Bem, 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 comercial qualquer e, por conta de novas demandas de negócio, ele precisa de dados específicos, mas esses dados estão espalhados em diferentes aplicações no sistema de software 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.**

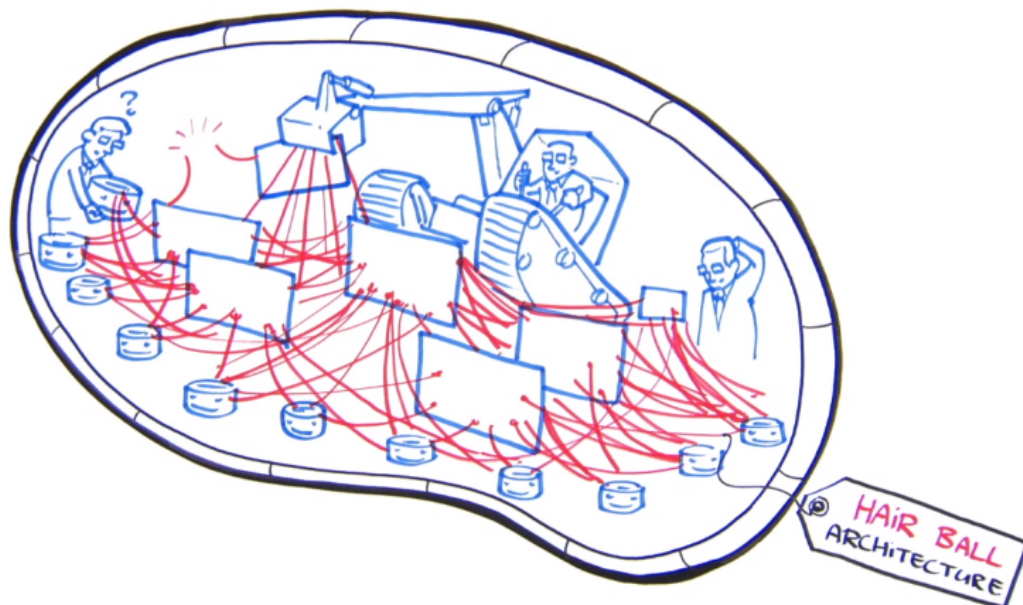


**João chama, então, um gerente de infraestrutura para automatizar a geração de sua planilha. O gerente cria um *job* para automatizar pela manhã a criação das planilhas do João.** Por outro lado, percebam que agora nós temos um novo fluxo de aplicação criado especificamente para o tal do João. E mais, esse fluxo depende de três outras aplicações do sistema de software do banco. *Percebem que está complicando? Pois é...*





Galera, vocês já viram o tamanho de um banco comercial? É claro que João não é o único funcionário que deseja melhorar sua eficiência. **Multipliquem essa situação por algumas centenas e nós teremos uma Arquitetura de Bola de Pelo (Hairball Architecture)<sup>2</sup>.** Em outras palavras, temos centenas de pelos indo e voltando em tudo que é direção. Ninguém tem controle e ninguém consegue rastrear absolutamente nada.



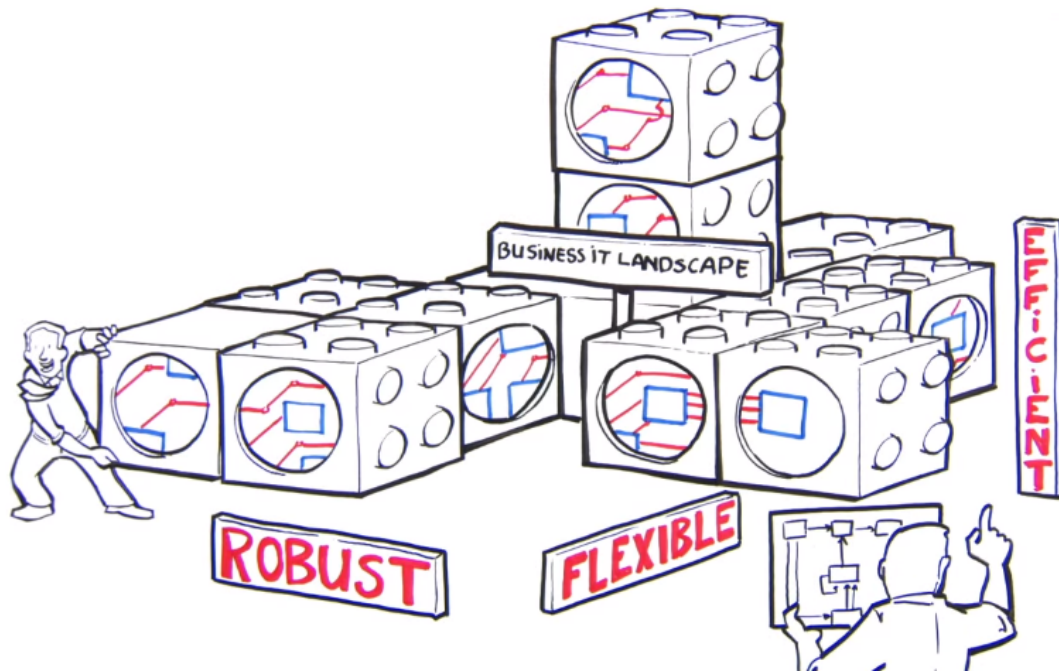
**Imaginem agora que nós temos que substituir uma aplicação imensa do banco porque ela está desatualizada e o fornecedor disse que, em dois anos, ele parará de oferecer suporte.** Ninguém sabe quais aplicações dependem dessa aplicação. É uma rede sem fim de dependências que foram

<sup>2</sup> Alguns chamam de Spaghetti-Like Architecture, isto é, Arquitetura de Macarronada.



criadas uma de cada vez, todas fazendo sentido no momento de sua criação, mas que ficou caótica com o tempo.

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. *Simples, não?*



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?* Ninguém gosta – imaginem um banco! *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 possuem à venda...**

#### 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.	



## Conceitos-Chaves

INCIDÊNCIA EM PROVA: BAIXA

**O Modelo de Referência do SOA apresenta, a partir da perspectiva dinâmica de serviço, três conceitos-chaves dessa arquitetura:** (1) a visibilidade entre provedores de serviços e consumidores; (2) a interação entre eles; (3) e os efeitos no mundo real da interação com um serviço. Existem outros conceitos fundamentais, mas esses são conceitos relacionados à dinâmica dos serviços. *Entendido?*

### CONCEITOS-CHAVE

VISIBILIDADE ENTRE PROVEDORES E CONSUMIDORES DE SERVIÇOS

INTERAÇÃO ENTRE PROVEDORES E CONSUMIDORES DE SERVIÇOS

EFEITOS NO MUNDO REAL DA INTERAÇÃO COM UM SERVIÇO

**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 regra, 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; contexto de execução; e contratos e políticas relacionadas aos serviços e aos seus participantes.**

**A descrição do serviço representa a informação necessária para utilização de 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.

Já contratos e políticas são restrições ou condições de uso de um serviço; e contexto de execução é um caminho estabelecido entre os participantes, constituído por participantes, infraestrutura, etc.



## Objetivos e Benefícios

INCIDÊNCIA EM PROVA: MÉDIA

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.

VANTAGENS DO SOA	DESVANTAGENS DO SOA
<b>Reutilização:</b> o serviço pode ser reutilizado para outras aplicações.	<b>Performance:</b> a performance depende do servidor onde o serviço está publicado, como também da rede.
<b>Produtividade:</b> com o reuso, a equipe de desenvolvimento pode reutilizar serviços em outros projetos, diminuindo o tempo de desenvolvimento.	<b>Complexidade:</b> uma grande quantidade de serviços precisa ser gerenciada.
<b>Flexibilidade:</b> isolando a estrutura de um serviço as mudanças são feitas com maior facilidade.	<b>Robustez:</b> caso uma exceção acontecer não tem como reverter o processo.
<b>Manutenibilidade:</b> com baixo acoplamento, facilita a manutenção dos serviços.	<b>Disponibilidade:</b> uma queda na rede ou no servidor deixa todos os serviços indisponíveis.
<b>Alinhamento com o negócio:</b> a área de negócio visualiza os processos alinhados com a tecnologia.	<b>Testabilidade:</b> o debug no serviço é um problema para os desenvolvedores.
<b>Interoperabilidade:</b> disponibilizar serviços independentemente da plataforma e tecnologia.	<b>Governança:</b> exige-se que haja governança na organização.



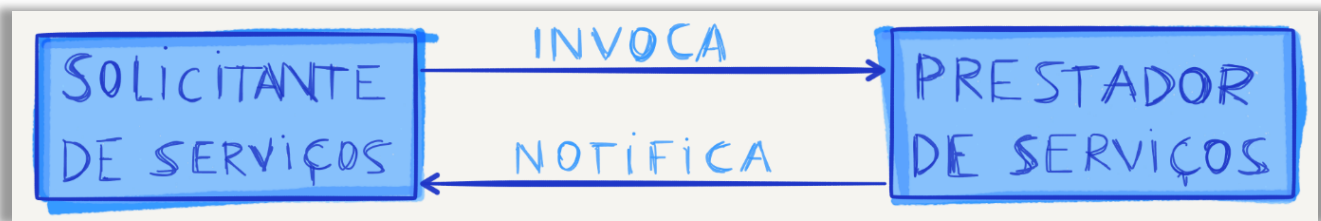
<b>Integração:</b> a integração com outros serviços, aplicativos e sistemas legados.	<b>Design:</b> grande aumento na complexidade do design dos serviços.
<b>Governança:</b> gerenciamento dos processamentos de negócio.	<b>Segurança:</b> os serviços estão disponíveis na rede, qualquer aplicativo pode consumi-los; os dados são trafegados pela rede podendo ser interceptados.
<b>Padronização:</b> é baseado no uso de padrões (de design, de contratos, protocolos, etc).	
<b>Abstração:</b> serviço totalmente abstraído da sua implementação.	
<b>Riscos:</b> auxilia a mitigação de riscos de negócio.	



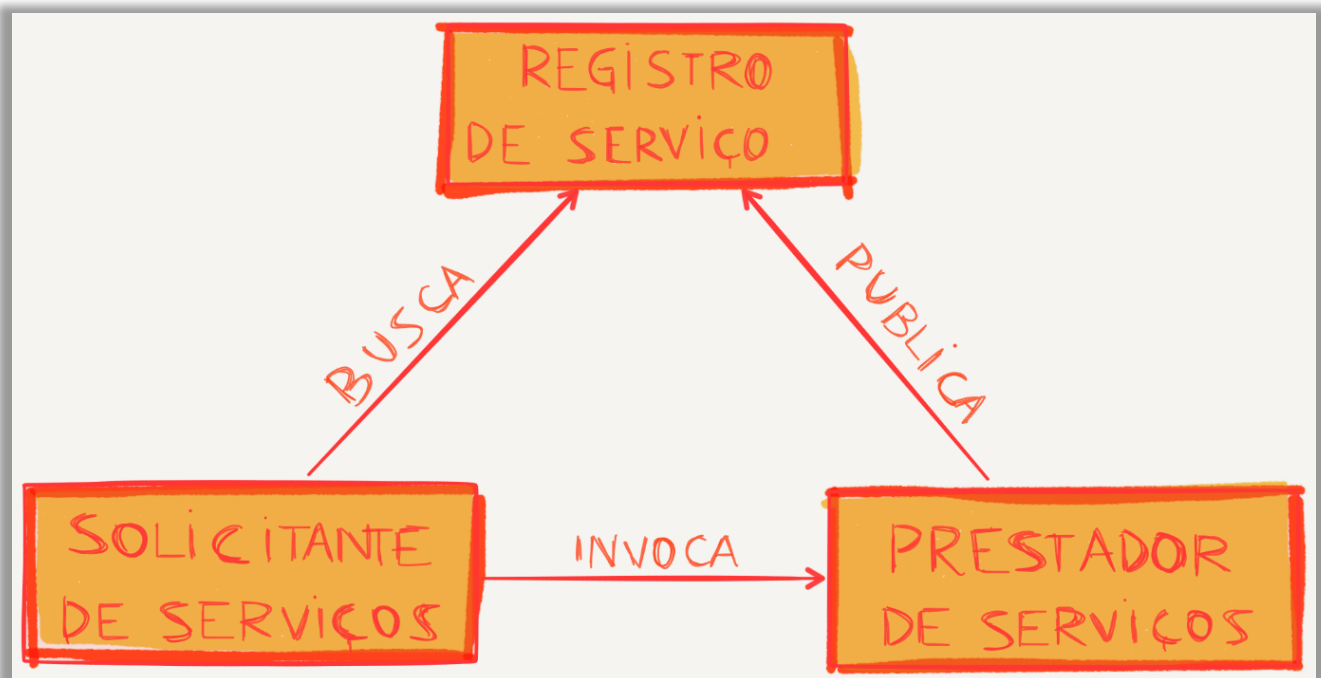
## Modelos Arquitetônicos

INCIDÊNCIA EM PROVA: MÉDIA

Galera... nós vimos insistentemente que a arquitetura orientada a serviços é independente de implementações e tecnologias específicas. Em outras palavras, isso quer dizer que existem diversos 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 papéis são identificados com base em seus comportamentos e responsabilidades sobre um serviço: (1) Prestadores de Serviço, (2) Registro de Serviços e (3) Solicitantes de Serviço. Funcionava assim: um prestador de serviços publica serviços em um



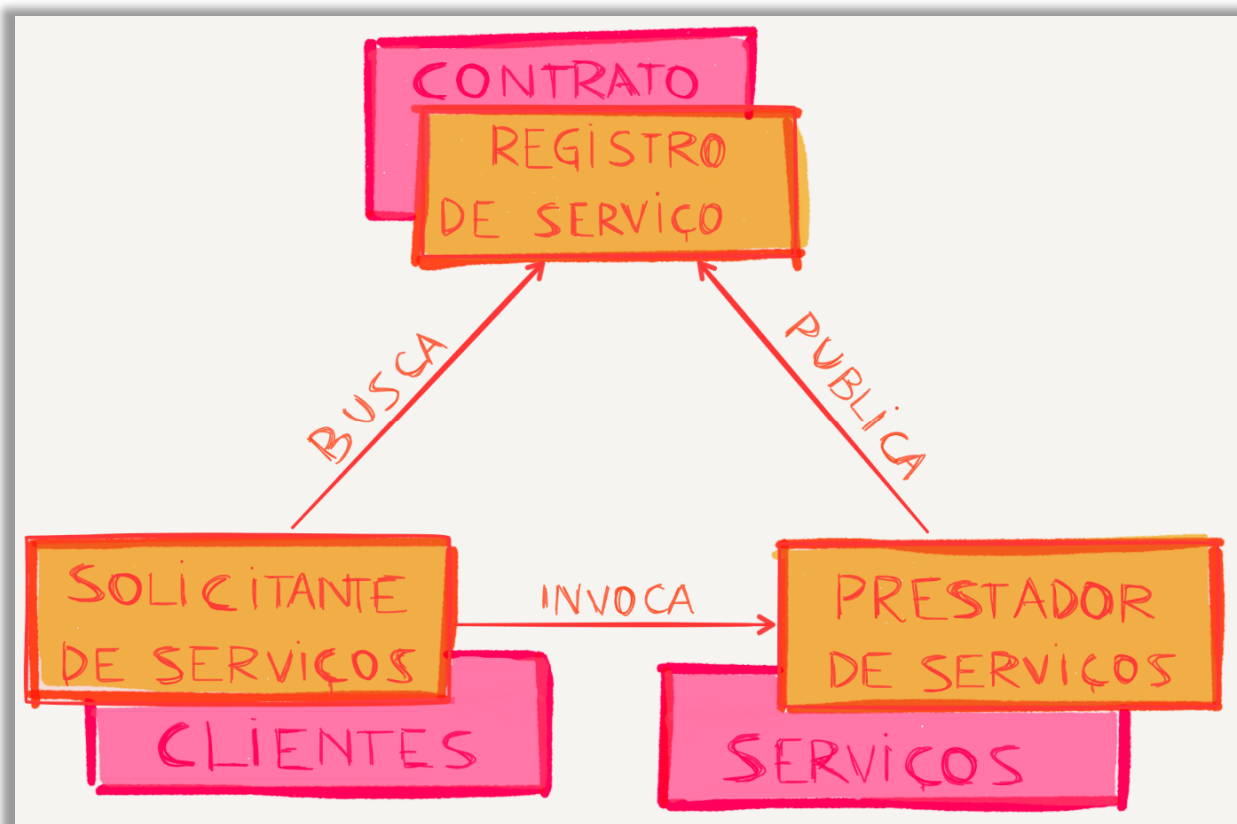


registro de serviços; o registro de serviços registra e organiza os serviços publicados e fornece serviços de busca.

**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<sup>1</sup>, formado pelo prestador de serviços, solicitante de serviços e o registro de serviços.



<sup>1</sup> Alguns chamam de Find-Bind-Invoke.

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

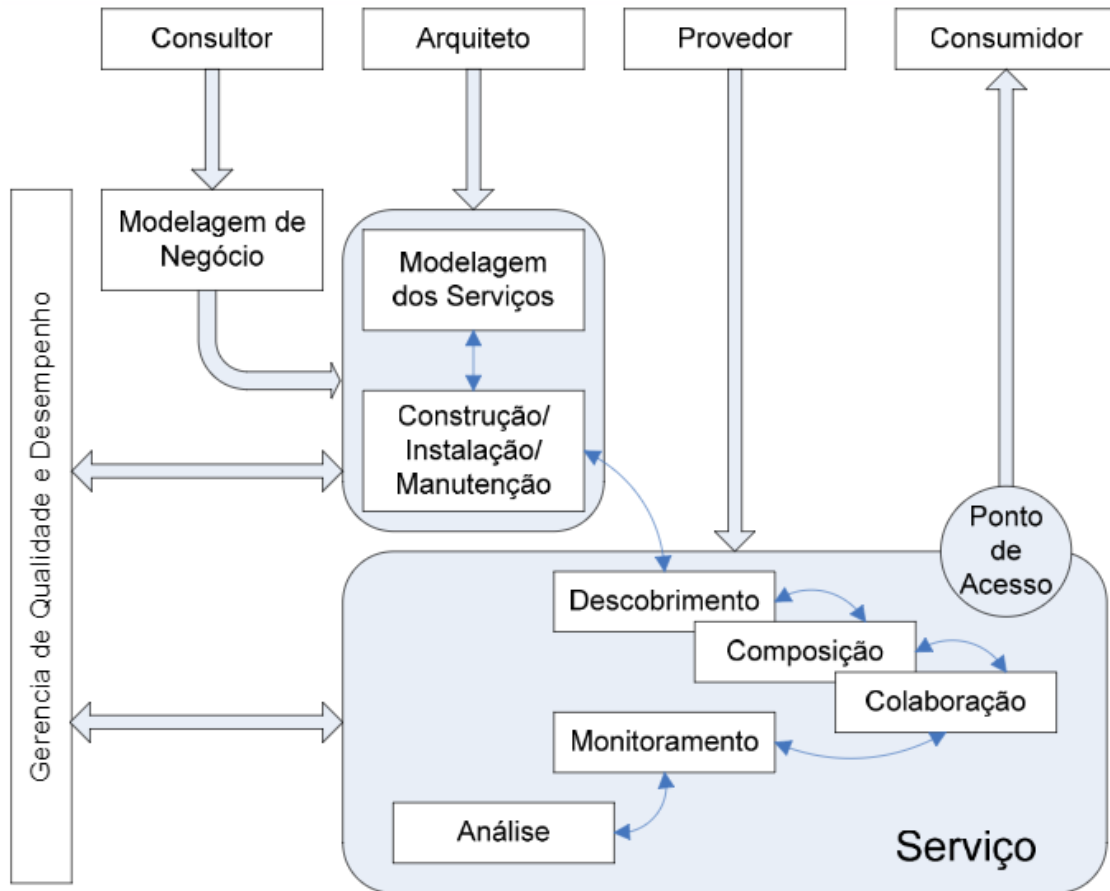
Galera, algumas questões têm cobrado recentemente os papéis ou atores que compõem o Ciclo de Vida de Soluções SOA. **Nesse contexto, abaixo temos alguns papéis considerados essenciais e suficientes, são eles: consultor de negócios; arquiteto SOA; provedor; e consumidor.** Os dois últimos papéis já foram vistos extensivamente, então não vamos detalhá-los. Vamos ver agora os outros dois papéis:

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 acondicionamento 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, processos prioritários; e a modelagem bottom-up é justamente o contrário.



## Princípios De Design

INCIDÊNCIA EM PROVA: ALTA

Galera, quantas vezes nós já falamos, nessa aula, sobre o tal do Thomas Erl? Pois é, ele é o cara da arquitetura orientada a serviços! Foi do livro dele – por exemplo – 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, isto é, 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.

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. Cada serviço tem sua responsabilidade bem definida e coerente – **é o princípio da responsabilidade única.** Imaginem um serviço que é

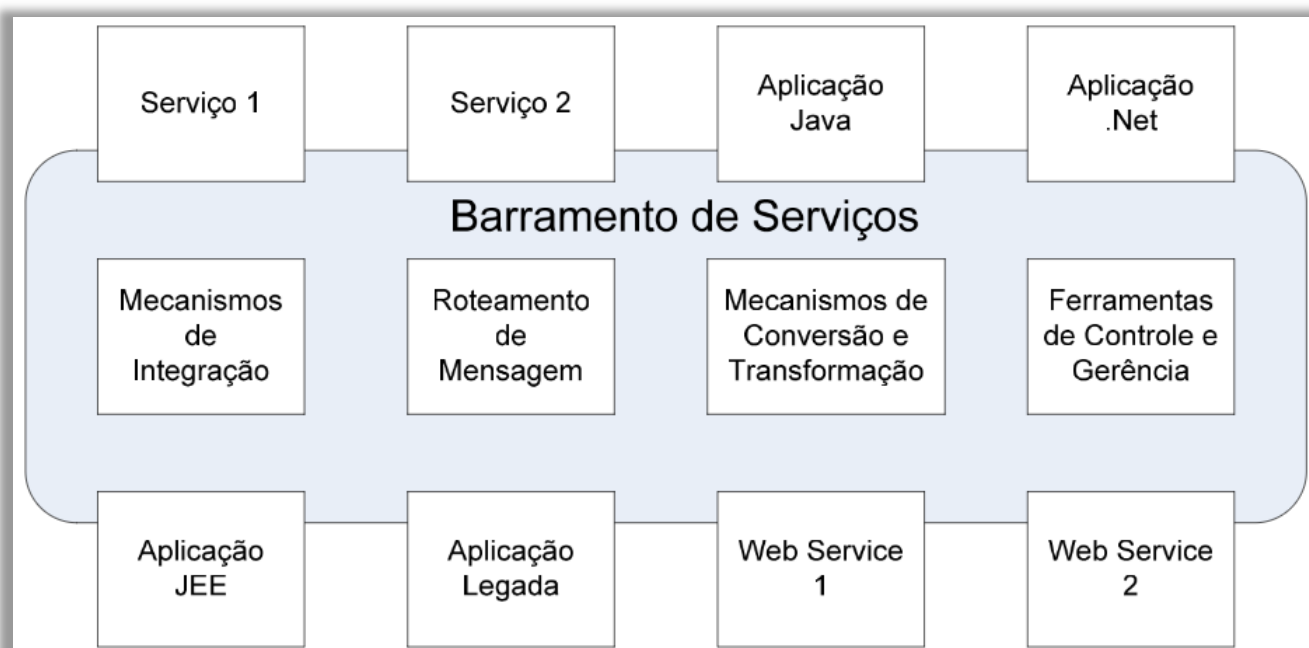


responsável por quatro tarefas totalmente diferentes. É uma péssima ideia - gera alto acoplamento! Um serviço nunca será 100% desacoplado, mas o desafio é achar o grau de desacoplamento que torne o serviço flexível para compor outros serviços com custo não tão elevado.

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 dos atrasos). **Se o tráfego estiver intenso, o motorista vai tomar uma rota alternativa para chegar ao destino. 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 e 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. *Para quê?*

Bem... 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!** Para finalizar... não confundam ao ler rápido em uma prova ESB com USB!

## 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, manter, não há nenhum problema – desde que eu mantenha os padrões de entrada e saída definidos no contrato. Dessa forma, o consumidor não é afetado! Vejamos...

**Se eu contrato um serviço dos correios para entregar apostilas para os meus alunos, eu não quero saber como eles organizarão a logística de entrega!** Eles podem mudar o que quiserem, desde que a apostila chegue para os alunos. Serviços devem ser consumidos independente da linguagem de programação, arquitetura, etc. Serviços são projetados para limitar informações no contrato de serviços ao que é realmente necessário para que eles sejam funcionalmente úteis.

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. *Fechado?*

## Reusabilidade de Serviços (*Service Reusability*)

**É de se esperar que um serviço seja reutilizável, isto é, 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 reuso 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 da 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 (autonomia de serviço) pode trazer alguns benefícios claros como, por exemplo, 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 conseqüentemente 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; e 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 ocorre quando um serviço não precisa reter nenhum dado do estado/conexão para que outro serviço processe a sua lógica de negócio. Podemos citar, como exemplo, a utilização do protocolo HTTP. *Vocês sabem como ele funciona?*

**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, logo é recomendada a **construção de serviços *stateless* (sem estado)**. 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. Outro nome para esse mesmo princípio é: princípio da alocação de recursos.

## 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 transmitam 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.** Vamos ver agora a composição de serviços...

## Composição de Serviços (*Service Composability*)

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

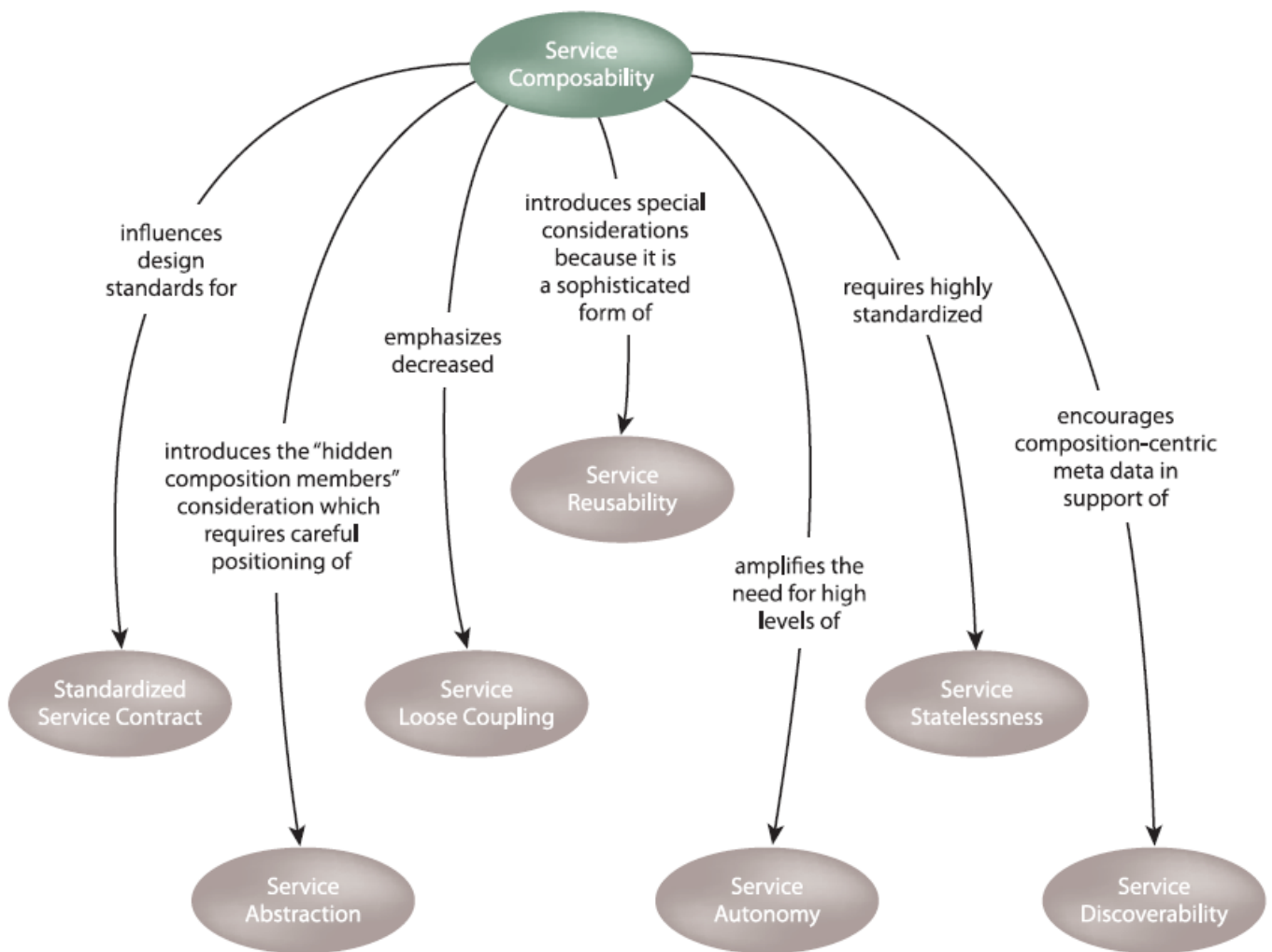
Eles também oferecem outro serviço que – dado um CEP – retorna os tipos de frete e seus valores. Esses serviços isoladamente resolvem problemas específicos. Quando compostos, esses resolvem





outros tipos de problemas. 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 para englobar e atender um problema maior. **Estas características se relacionam de forma interdependente.** Por exemplo: 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*.



## Composição De Serviços

INCIDÊNCIA EM PROVA: MÉDIA

**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 no esquema da imagem a seguir. Um serviço maior é composto por outros serviços menores...



**A composição de serviços potencializa os diversos benefícios oferecidos pela arquitetura orientada a serviços.** Como assim, Diego? 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 de Serviços

A orquestração de serviços representa o processo em que diversos recursos dentro de uma organização podem ser coordenados para executar um conjunto de lógicas de processo de negócio.

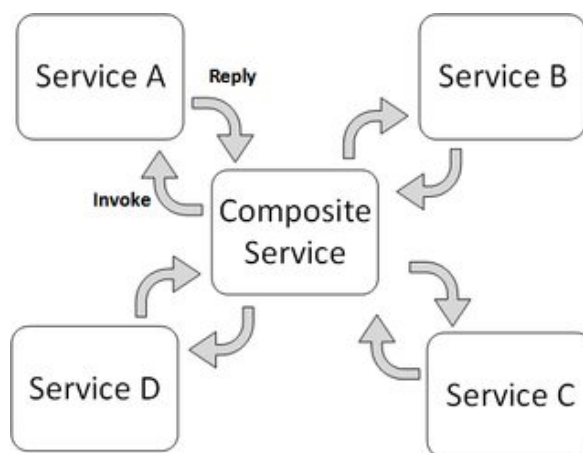


Essa tecnologia é mais comumente associada a uma plataforma central de gerenciamento de atividades que fornecem benefícios como transformação de dados e gerenciamento de transações.



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 seguinte. **Além disso, a lógica de processo de negócio – na orquestração – pertence à organização, isto é, trata-se de uma lógica privada e intra-organização – em contraste com a coreografia (que veremos a seguir).**



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.**

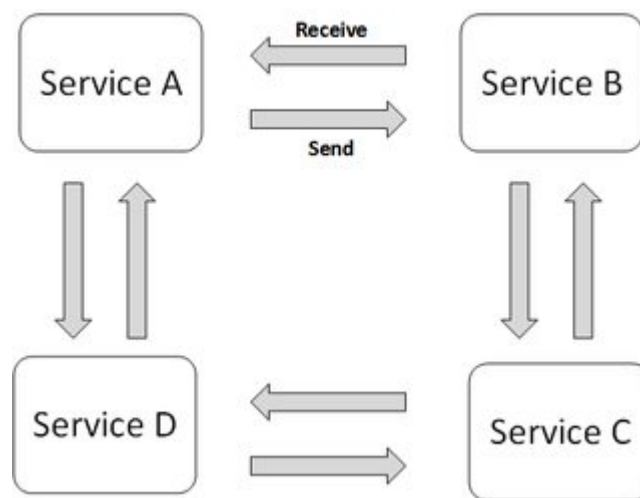
## Coreografia de Serviços

A coreografia de serviços, em contraste com a orquestração, não possui 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 (como um maestro), fazendo verificações de pré-condições e pós-condições. **Já, na coreografia, todos auxiliam no fluxo das operações (como se seguissem uma coreografia).**

**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 busca incluir uma grande variedade de tecnologias para componentes de serviços e para os métodos de acesso que são utilizados para conectá-los. **Para componentes, isso 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.

Essas tecnologias são usadas em geral (Ex: Web Services, sistemas de mensagens e Remote Procedure Call). 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).



## Manifesto SOA

INCIDÊNCIA EM PROVA: BAIXÍSSIMA

Assim como existe o Manifesto Ágil, que é um documento que formaliza princípios que fundamentam o desenvolvimento ágil de software, existe também o Manifesto SOA. Vejamos:

**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:**



**ISSO É, MESMO VALORIZANDO OS ITENS À DIREITA, VALORIZAMOS MAIS OS ITENS À ESQUERDA.**

### PRINCÍPIOS ORIENTADORES

Respeitar a estrutura social e de poder da organização.

Reconhecer que SOA, em última instância, requer mudanças em múltiplos níveis.

O escopo da adoção de SOA pode variar.

Manter os esforços gerenciáveis e dentro de limites significativos.

Produtos e padrões, por si só, não proverão uma SOA nem aplicarão os paradigmas de orientação a serviço por você.

SOA pode ser realizada através de uma variedade de tecnologias e padrões.

Estabelecer um conjunto uniforme de padrões e políticas corporativas embasado em padrões da indústria, de facto, e da comunidade.

Buscar uniformidade no exterior e permitir diversidade no interior.

Identificar serviços através da colaboração entre partes interessadas no negócio e na tecnologia.



Maximizar o uso de serviços considerando o escopo de utilização atual e futuro.

Verificar que os serviços satisfaçam os requisitos e objetivos de negócio.

Evoluir os serviços e sua organização em resposta ao uso real.

Separar os diferentes aspectos de um sistema que mudam com diferentes frequências.

Reduzir dependências implícitas e publicar todas as dependências externas para aumentar a robustez e diminuir o impacto de mudanças.

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](http://www.soa-manifesto.com/annotated_portuguese.htm)



## Versionamento de Serviços

INCIDÊNCIA EM PROVA: BAIXÍSSIMA

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<sup>1</sup> 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 um conjunto de políticas relacionadas à gestão dos serviços.** Boas práticas envolvem ter um repositório único na organização para manter a dependência e rastreabilidade dos artefatos de um serviço numa forma controlada, além de definir o tempo de guarda das versões.

**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.

---

<sup>1</sup> Um programa SOA é a aplicação de SOA em nível organizacional com um fim bem definido.





Toda organização que tenha um portfólio considerável de serviços deve disponibilizar um mecanismo eficiente de consulta. **Assim, toda vez que consumidores estiverem interessados em serviços, seu mecanismo de registro poderá ajudá-los a obter respostas facilmente.** Quando mudanças são eminentes, ter um mecanismo que possibilite a acomodação de mudanças apenas mais por configuração e menos por implementação também é 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 mudanças de forma simples e rápida. Neste caso, o uso de um bom barramento de serviços (ESB) 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: 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.



## Padrões SOA

INCIDÊNCIA EM PROVA: BAIXÍSSIMA

**Padrões SOA são princípios propostos para desenvolver uma solução lógica de serviços dentro de SOA!** 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.

SOA desenvolvido sem orientações concretas provavelmente falhará. Para garantir que o movimento em direção a orientação a serviços seja uma mudança positiva, é útil adotar um conjunto de padrões de projeto. **Eles 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, além de ajudar a atingir 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 mercado 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.



## Sistemas de Governança

INCIDÊNCIA EM PROVA: BAIXÍSSIMA

**O conceito de governança – em um contexto genérico – 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 em um contexto específico de tecnologia da informação, o conceito de governança pode ser definido como:

*"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, reduzir os custos, suportar as melhores decisões e conseqüentemente alinhar TI aos negócios".*

Eu 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; entre outros. *E o que seria a Governança SOA?*

**Trata-se de 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. A 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. Os princípios são: 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](https://www.opengroup.org/soa/source-book/gov/index.htm)



## Mensageria e CORBA

INCIDÊNCIA EM PROVA: BAIXÍSSIMA

**Antes de detalhar do que se trata os conceitos de Mensageria e CORBA, é interessante discutir 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 – à época – 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 também.

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, isto é, 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, isto é, 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 vantagem disso? Reduzir o grau de acoplamento (isto é, 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. Em suma, podemos dizer que a mensageria é um padrão de comunicação assíncrona utilizado em sistemas distribuídos para troca de mensagens entre diferentes componentes ou serviços.

É uma abordagem que permite o envio, recebimento e processamento de mensagens entre aplicações de forma independente e escalável. No contexto da mensageria, os sistemas são compostos por diferentes partes chamadas de produtores de mensagens e consumidores de mensagens. Os produtores de mensagens são responsáveis por gerar e enviar as mensagens para uma fila (ou tópico) de mensagens.

Os consumidores de mensagens, por sua vez, recebem essas mensagens da fila e as processam de acordo com sua lógica de negócio. Vejamos seus principais benefícios:

BENEFÍCIO	DESCRIÇÃO
ASSINCRONICIDADE	As mensagens podem ser enviadas e processadas de forma assíncrona, ou seja, o produtor de mensagens não precisa esperar pela resposta imediata do consumidor. Isso permite uma maior flexibilidade e escalabilidade no sistema, já que os componentes podem trabalhar de forma independente e não precisam estar ativos simultaneamente.
ACOPLAMENTO FRACO	A comunicação por mensagens permite um acoplamento fraco entre os componentes do sistema. Os produtores e consumidores não precisam conhecer detalhes de implementação uns dos outros, pois a comunicação é feita por meio das mensagens. Isso facilita a manutenção e evolução dos sistemas, uma vez que os componentes podem ser modificados sem afetar diretamente os outros.
ESCALABILIDADE	O uso de filas de mensagens permite uma maior escalabilidade horizontal do sistema. Mais produtores e consumidores podem ser adicionados conforme necessário, e as filas podem ser dimensionadas para suportar grandes volumes de mensagens, garantindo um processamento eficiente e distribuído.
TOLERÂNCIA A FALHAS	A mensageria permite a implementação de sistemas resilientes, capazes de lidar com falhas de rede, indisponibilidade temporária de componentes e falhas pontuais. As mensagens podem ser armazenadas em filas duráveis, garantindo que nenhuma mensagem seja perdida e possa ser processada posteriormente.

A mensageria é amplamente utilizada em cenários como integração de sistemas, processamento de eventos em tempo real, comunicação entre microsserviços, sistemas distribuídos em nuvem e muitas outras aplicações onde a troca assíncrona de mensagens é necessária. **Já o CORBA<sup>1</sup> é uma**

<sup>1</sup> CORBA = Common Object Request Broker Architecture



**arquitetura padrão para estabelecer e simplificar a troca de dados entre sistemas distribuídos heterogêneos por meio de uma estrutura comum para gerenciamento de objetos distribuídos.**

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, entre outras. Vamos resumir o principal sobre o CORBA...

Trata-se de uma **arquitetura de software baseada em objetos** que foi desenvolvida pela Object Management Group (OMG) como um padrão para permitir a comunicação e interoperabilidade entre componentes de software distribuídos em uma rede. É uma tecnologia que facilita a construção de sistemas distribuídos heterogêneos, nos quais os componentes podem ser desenvolvidos em diferentes linguagens de programação e executar em diferentes plataformas.

O CORBA utiliza um modelo cliente/servidor, no qual os objetos distribuídos podem ser acessados e invocados por meio de solicitações de serviços (chamadas de métodos). Seus componentes são:

PRINCIPAIS COMPONENTES	DESCRIÇÃO
OBJETOS	São componentes de software que encapsulam a lógica de negócio e oferecem serviços e operações para serem invocados remotamente. Os objetos CORBA são implementados usando uma linguagem de programação suportada pelo CORBA, como C++, Java ou Python, e são registrados em um Serviço de Nomes (Naming Service).
OBJECT REQUEST BROKER (ORB)	É uma infraestrutura de software responsável por facilitar a comunicação entre os objetos CORBA. O ORB atua como intermediário entre os clientes que solicitam serviços e os objetos distribuídos que oferecem esses serviços. Ele lida com a localização do objeto correto, a serialização dos parâmetros da solicitação, o envio da solicitação ao objeto e a entrega da resposta ao cliente.
INTERFACE DEFINITION LANGUAGE (IDL)	É uma linguagem neutra de definição de interfaces utilizada para especificar a estrutura e as operações dos objetos CORBA. A IDL permite que desenvolvedores descrevam a



interface de um objeto CORBA de forma independente da linguagem de programação usada para implementá-lo. A partir da IDL, é possível gerar código-fonte em diferentes linguagens para a implementação do objeto e a criação dos stubs e skeletons necessários para a comunicação CORBA.

Com o CORBA, os objetos distribuídos podem ser acessados de forma transparente, como se fossem objetos locais, independentemente de onde estejam executando. **Isso significa que um objeto CORBA pode ser invocado por um cliente em uma máquina diferente, em uma plataforma diferente e até mesmo usando uma linguagem de programação diferente.** O CORBA oferece diversos benefícios...

Entre eles, podemos citar a reutilização de componentes, interoperabilidade entre sistemas heterogêneos, extensibilidade, escalabilidade e flexibilidade. No entanto, o CORBA também possui complexidade adicional devido à necessidade de definir interfaces usando a IDL e configurar e gerenciar o ORB. **Embora o CORBA tenha sido amplamente utilizado no passado, sua adoção diminuiu com o tempo em favor de tecnologias mais recentes, como WS e SOA.**

No entanto, o CORBA ainda é usado em alguns domínios específicos que exigem interoperabilidade entre sistemas legados e novos ou onde a integração de sistemas distribuídos complexos é necessária. Por fim, vamos falar rapidamente sobre **stream**: trata-se de uma sequência contínua de dados que é transmitida, processada e consumida em tempo real. Ela é composta por um fluxo de elementos, como bytes, caracteres, números, eventos ou até mesmo objetos mais complexos.

**Uma stream é conceitualmente semelhante a um fluxo de água contínuo, onde a água flui constantemente.** Da mesma forma, uma stream de dados é uma sequência contínua de informações que podem ser lidas, gravadas ou manipuladas de forma incremental, em vez de serem acessadas em sua totalidade de uma vez. Elas são amplamente utilizadas em sistemas de processamento de dados em tempo real. *Eles estão na moda atualmente, concordam?*

Eu falo de transmissões de vídeo, áudio, mensagens instantâneas, feeds de redes sociais e muitos outros casos de uso. Elas permitem que os dados sejam transmitidos e processados de forma contínua, conforme são gerados, em vez de esperar pela disponibilidade completa dos dados. As streams podem ter várias fontes e destinos, dependendo do contexto. **As operações em geral envolvem a leitura, gravação, filtragem, transformação e agregação dos dados em tempo real.**

As streams podem ser processadas de forma sequencial, onde cada elemento é processado um de cada vez, ou de forma paralela, onde os elementos podem ser processados simultaneamente em diferentes threads ou nós de um sistema distribuído. Uma vantagem é a capacidade de lidar com grandes volumes de dados de forma eficiente, uma vez que os dados são processados à medida que chegam, em vez de armazenar tudo em memória ou em um local de armazenamento intermediário.

Além disso, as streams **permitem a construção de sistemas assíncronos e resilientes**, onde os produtores e consumidores de dados podem operar em diferentes velocidades e se adaptar a flutuações de carga, atrasos de rede ou falhas pontuais, garantindo a continuidade do fluxo de



dados. Galera, esses assuntos não caem muito em prova, mas não são assuntos complexos, então vale a leitura.





## QUESTÕES COMENTADAS – CESPE

1. (CESPE / MPE-TO – 2024) SOA (Service Oriented Architecture) é uma arquitetura empregada para reutilizar serviços em sistemas diferentes ou combinar vários serviços independentes para realizar tarefas complexas.

### Comentários:

SOA é uma abordagem de design de software que envolve a organização e a utilização de serviços distribuídos que podem ser reusados para suportar requisitos de negócios aplicáveis. Esta arquitetura permite que serviços sejam combinados ou reutilizados em diferentes aplicações ou contextos, facilitando a integração e a flexibilidade. SOA é particularmente valorizada por sua capacidade de permitir que sistemas diferentes colaborem e executem tarefas complexas por meio da comunicação entre serviços independentes, o que aumenta a eficiência operacional e reduz a redundância de desenvolvimento.

**Gabarito:** Correto

2. (CESPE / BNB – 2022) Implementações de SOA por meio de um barramento de serviços implicam a criação de um ponto único de falha, uma vez que clientes e serviços da organização não poderão se comunicar se o barramento de serviços se tornar inoperante.

### Comentários:

A criação de um barramento de serviços para implementar uma arquitetura de SOA significa que todos os serviços e clientes da organização se comunicarão através desse barramento. Isso significa que, se o barramento de serviços se tornar inoperante, todos os serviços e clientes da organização não poderão se comunicar. Isso cria um único ponto de falha, pois todos os serviços e clientes da organização dependem do barramento de serviços para se comunicar. Se o barramento de serviços não estiver funcionando, todos os serviços e clientes da organização serão afetados.

**Gabarito:** Correto

3. (CESPE / FUNPRESP-EXE - 2022) Em uma arquitetura orientada a serviços, o registro central é o elemento que disponibiliza os serviços, enquanto que o provedor age como concentrador de serviços.

### Comentários:

Na verdade, é o inverso o provedor de serviços disponibiliza os serviços e o registro central concentra os serviços.



**Gabarito:** Errado

---

4. (CESPE / FUNPRES-EXE - 2022) ESB (Enterprise Service Bus) é um barramento sobre o qual os serviços são oferecidos aos clientes, e que provê interoperabilidade entre sistemas heterogêneos.

**Comentários:**

ESB é um barramento de serviços que fornece uma forma de mover mensagens entre serviços. Além disso, 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.

**Gabarito:** Correto

---

5. (CESPE / FUNPRES-EXE - 2022) A BPEL (Bussiness Process Execution Language) é utilizada para implementar uma composição de serviços do tipo coordenação.

**Comentários:**

Na verdade, a BPEL 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.

**Gabarito:** Errado

---

6. (CESPE / Petrobrás - 2022) Com a SOA é mais fácil escalar ou ampliar o uso de uma funcionalidade para plataformas ou ambientes novos, o que resulta em maior agilidade e eficiência no desenvolvimento, por outro lado, em geral, acaba por promover aumento nos custos da solução.

**Comentários:**

SOA promove aumento nos custos de uma solução? Na verdade, é o contrário, SOA traz mais efetividade nos custos. Isso é explícito no Manifesto SOA. O restante do item está correto.

**Gabarito:** Errado

---

7. (CESPE / STM – 2018) Em SOA, orquestração é a forma de arranjar serviços diferentes para serem executados em uma ordem preestabelecida.

**Comentários:**



A orquestração e a coreografia arranjam (no sentido de arranjo) serviços diferentes para serem executados em uma ordem preestabelecida. *Qual a diferença, então?* A orquestração possui um orquestrador único que define a ordem de realização dos serviços, fazendo com que os mesmos não necessitem conhecer uns aos outros. Na coreografia, é necessário que os serviços saibam da existência uns dos outros, para, então, serem executados em uma ordem preestabelecida.

**Gabarito:** Correto

8. (CESPE / TCE-PA – 2016) Os quatro atores necessários à composição do ciclo de vida de uma solução SOA incluem o arquiteto SOA, que é o responsável por mapear os processos de negócio.

**Comentários:**

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.

A questão se refere ao papel de Consultor e, não, ao Arquiteto SOA.

**Gabarito:** Errado

9. (CESPE / TCE-PR – 2016 – Letra A) Os componentes de um web service não devem ser reutilizados, para que tenham uma evolução independente, característica fundamental de soluções SOA.

**Comentários:**

É claro que devem ser reutilizados! Trata-se, inclusive, de um dos princípios: reusabilidade de serviços (*service reusability*).

**Gabarito:** Errado

10. (CESPE / TCE-PR – 2016 – Letra B) Um barramento de serviços ESB (Enterprise Service BUS) é a implementação física de infraestrutura com base em web services.

**Comentários:**

Trata-se de uma implementação conceitual! O barramento provê uma camada de abstração acima de um sistema de mensageria que permite a integração entre os aplicativos.



**Gabarito:** Errado

---

**11. (CESPE / TCE-PR – 2016 – Letra C)** Conforme as soluções SOA, cabe ao cliente monitorar o desempenho da aplicação, uma vez que ele é o maior interessado em utilizar os serviços fornecidos.

**Comentários:**

O monitoramento de aplicações visa garantir o desempenho adequado dos sistemas. Ao medir a experiência do usuário, as empresas conseguem enxergar como uma aplicação se comporta na máquina do funcionário, permitindo que os profissionais de TI atuem de forma pró-ativa para solucionar problemas. Portanto, é claro que não cabe ao cliente – cabe à equipe de sustentação.

**Gabarito:** Errado

---

**12. (CESPE / TCE-PR – 2016 – Letra E)** Inicialmente, o consultor de negócios, o arquiteto SOA, o provedor de serviço e o consumidor de serviço são suficientes para a composição do ciclo de vida de soluções SOA.

**Comentários:**

Perfeito! Alguns papéis participam do ciclo de vida de soluções, como consultor de negócios, arquiteto, provedor e consumidor.

**Gabarito:** Correto

---

**13. (CESPE / TCE-PR – 2016)** A respeito de SOA e CORBA, assinale a opção correta.

- a) No modelo operacional triangular, o registro do serviço determina o comportamento de quem disponibiliza o serviço, ou seja, do dono do serviço, que é responsável, entre outros aspectos, pela infraestrutura de acesso ao serviço.
- b) A utilização de SOA em uma organização permite a descoberta de novos processos de negócio para uma posterior associação com serviços de TI.
- c) Entre os possíveis mapeamentos de negócio e de TI, a abordagem de baixo para cima (bottom-up) determina que a organização identifique, primeiramente, os processos que considerar prioritários.
- d) Assim como o DCOM, o CORBA é executado apenas em ambiente Windows.



e) O modelo de referência da OMG (Object Management Group) para CORBA define a interface de aplicação, isto é, o conjunto de dados públicos do objeto que possibilita a comunicação por meio de chamadas aos métodos desse objeto com os parâmetros apropriados.

**Comentários:**

(a) Errado, o dono do serviço é o provedor e, não, o registro; (b) Errado, eu não sei por que a banca considerou como errado – para mim, está correto; (c) Errado, a questão trata da abordagem top-down e, não, bottom-up; (d) Errado, CORBA não se limita apenas ao ambiente Windows; (e) Correto, o modelo de referência é composto por: Objetos de Serviço; Facilidades Comuns; Interfaces de Domínio; Interface de Aplicação; e Objetos de Aplicação. A Interface de Aplicação é o conjunto de dados públicos do objeto que possibilita a comunicação por meio de chamadas aos métodos desse objeto com os parâmetros apropriados.

**Gabarito:** Letra E

---

**14. (CESPE / TCE-PR – 2016 – Letra B)** Um barramento de serviços ESB (Enterprise Service BUS) é a implementação física de infraestrutura com base em web services.

**Comentários:**

Não se trata de uma implementação física, mas conceitual – é uma abstração, um modelo conceitual da infraestrutura.

**Gabarito:** Errado

---

**15. (CESPE / MPOG-ATI – 2015)** Em arquiteturas orientadas a serviço, um serviço deve ser implementado de forma a garantir um fraco acoplamento.

**Comentários:**

Perfeito! A questão trata do princípio o fraco/baixo acoplamento.

**Gabarito:** Correto

---

**16. (CESPE / TRE-MT – 2015)** Assinale a opção correta, relativa a SOA (service-oriented architecture).

a) Um dos objetivos da orientação a serviços é estabelecer maior interoperabilidade intrínseca, a fim de reduzir a necessidade de integração.

b) A SOA visa diminuir a perspectiva federada de serviços, de modo a compartilhar dados entre diferentes plataformas computacionais.



- c) Em se tratando de uma solução orientada a serviços, as unidades de lógica encapsulam funcionalidades específicas a um processo de negócio.
- d) Por fornecerem um intervalo de funcionalidades genéricas, os serviços não agnósticos não devem ser valorizados na SOA.
- e) Serviços não agnósticos adaptam-se inúmeras vezes a diferentes processos de negócio como parte de diferentes soluções orientadas a serviços.

#### Comentários:

(a) Correto, um dos objetivos realmente é estabelecer uma maior interoperabilidade; (b) Errado, ela visa aumentar a perspectiva federada de serviços; (c) Errado, elas encapsulam funcionalidades não específicas a nenhum aplicativo ou processo de negócio; (d) Errado, são serviços agnósticos que fornecem um intervalo de funcionalidades genéricas; (e) Errado, são serviços agnósticos que podem ser adaptados inúmeras vezes.

**Gabarito:** Letra A

---

**17. (CESPE / TCE-RN – 2015)** A arquitetura CORBA permite realizar a intercomunicação entre computadores de arquiteturas e portes distintos por meio do protocolo-padrão EIGRP, versão melhorada do IGRP, que permite compor, de forma síncrona ou assíncrona, objetos, dados e unidades individuais.

#### Comentários:

Na verdade, ele utiliza o Protocolo IIOP (Internet Inter-ORB Protocol) e, não, EIGRP.

**Gabarito:** Errado

---

**18. (CESPE / MEC – 2015)** Enterprise Service Bus é um barramento para integração de serviços coletivos de bancos de dados distribuídos sobre arquitetura IP (Internet Protocol) para independência de camada de serviços.

#### Comentários:

Na verdade, trata de serviços distribuídos e, não, bancos de dados distribuídos; ademais, não é necessariamente sobre a Arquitetura IP – ele pode funcionar com diversas tecnologias.

**Gabarito:** Errado

---



**19. (CESPE / TCU – 2015)** Um projeto de software orientado pela governança SOA deve estar alinhado não só com a governança de TI, mas também com a governança da arquitetura empresarial.

**Comentários:**

É até um pouco óbvio: se a Governança SOA estende a Governança de TI, que está alinhada à governança da arquitetura empresarial, então ela também estará alinhada à governança da arquitetura empresarial.

**Gabarito:** Correto

---

**20. (CESPE / ANTAQ – 2014)** No uso de SOA, a troca de dados requer protocolos intermediários, os quais poderão representar uma perda de desempenho das aplicações.

**Comentários:**

Vamos parar um pouco e tentar refletir sobre esse item. Observe que ele diz: “*No uso de uma arquitetura orientada a serviços (...)*”, portanto estamos falando de uma implementação da arquitetura (WS-\*, REST, etc) e, não, da arquitetura em si. Nesse sentido, a troca de dados realmente requer protocolos intermediários para fazer a comunicação entre os serviços (Ex: SOAP, HTTP). Se eu uso qualquer meio intermediário, por mais leve que ele seja, gerará um *overhead*.

Podemos dizer – sim – que esses protocolos intermediários poderão representar uma perda de desempenho. Exemplo: WS-\* gera mais overhead que REST – apesar de ambas gerarem overhead.

**Gabarito:** Correto

---

**21. (CESPE / ANATEL – 2014)** Os serviços compostos podem apresentar limitações de segurança, especialmente pelo fato de permitirem que os serviços básicos que os compõem sejam chamados individualmente; não havendo mecanismos que permitam que os serviços básicos sejam chamados apenas pelos serviços de mais alto nível.

**Comentários:**

De fato, serviços compostos podem apresentar limitações de segurança. No entanto, há mecanismos que permitam que os serviços básicos sejam chamados apenas pelos serviços de mais alto nível. Na orquestração, por exemplo, é o coordenador que comanda as chamadas aos serviços.

**Gabarito:** Errado

---

**22. (CESPE / SUFRAMA – 2014)** O desenvolvimento de aplicações em ambientes corporativos que utilizam SOA permite alto acoplamento e grande redundância de funcionalidades.



### Comentários:

Na verdade, a arquitetura orientada a serviços permite o baixo acoplamento e baixa redundância.

**Gabarito:** Errado

**23. (CESPE / SUFRAMA – 2014)** Entre os princípios básicos da SOA estão os serviços que abstraem a lógica, que compartilham um contrato formal, que evitam alocação de recursos por longos períodos e que são autônomos e reutilizáveis.

### Comentários:

Perfeito! A questão trata dos quatro princípios: Contrato de Serviço Padronizado (*Service Contract*); Reusabilidade de Serviços (*Service Reusability*); Autonomia de Serviços (*Service Autonomy*); e Independência de Estados de Serviços (*Service Statelessness*);

**Gabarito:** Correto

**24. (CESPE / SUFRAMA – 2014)** No CORBA, o ORB (Object Request Broker) é o responsável por encontrar a implementação de objeto para o pedido feito pelo cliente, preparar a implementação de objeto para receber o pedido e transmitir os dados do pedido.

### Comentários:

O ORB (Object Request Broker) realmente é responsável por encontrar a implementação de objeto para o pedido feito pelo cliente, preparar a implementação de objeto para receber o pedido e transmitir os dados do pedido. Ele é um módulo intermediário entre cliente/objeto, sendo responsável por aceitar requisições, enviá-las ao objeto correto e entregar a resposta ao cliente.

**Gabarito:** Correto

**25. (CESPE / ANTT – 2013)** A SOA pode ser definida como um tipo de arquitetura que utiliza serviços como blocos de construção para facilitar a integração em ambientes corporativos e a reutilização de componentes por meio do baixo acoplamento.

### Comentários:

#### O QUE SOA É...

Uma forma de aproximar a linguagem do negócio e da tecnologia da informação, facilitando a integração de ambientes corporativos por meio de serviços.

Perfeito! O baixo acoplamento é um dos princípios da arquitetura orientada a serviços.





**Gabarito:** Correto

**26.(CESPE / STF – 2013)** A arquitetura orientada a serviços é utilizada para interoperabilidade de sistemas heterogêneos por meio de conjunto de serviços fracamente acoplados. A orientação a serviços utiliza protocolos padrão e interfaces convencionais para facilitar o acesso à lógica de negócios e às informações entre serviços distintos.

**Comentários:**

#### 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.

A arquitetura orientada a serviços é realmente utilizada para interoperabilidade de sistemas heterogêneos. Eles, de fato, fazem isso por meio de serviços fracamente acoplados. Ela realmente utiliza protocolos padrões e interfaces convencionais.

**Gabarito:** Correto

**27.(CESPE / MPOG – 2013)** O SOA garante serviços fortemente acoplados, fracamente coesos e com alta possibilidade de reutilização.

**Comentários:**

Pela milionésima vez, a arquitetura orientada a serviços garante serviços fracamente acoplados e altamente coesos (de fato, com alta possibilidade de reutilização).

**Gabarito:** Errado

**28.(CESPE / MPOG – 2013)** O SOA promove a integração entre o negócio e a tecnologia da informação por meio de serviços, que são o principal componente dessa arquitetura.

**Comentários:**

#### O QUE SOA É...

Uma forma de aproximar a linguagem do negócio e da tecnologia da informação, facilitando a integração de ambientes corporativos por meio de serviços.

Perfeito! SOA realmente promove a integração entre o negócio e a tecnologia da informação por meio de serviços, que são o principal componente dessa arquitetura.

**Gabarito:** Correto



**29. (CESPE / SERPRO – 2013)** Por ser dependente de tecnologia, o ambiente de SOA tem de ser implementado em protocolos específicos.

**Comentários:**

Na verdade, trata-se de uma arquitetura independente de tecnologia.

**Gabarito:** Errado

---

**30. (CESPE / SERPRO – 2013)** No nível do aplicativo, os serviços fornecidos pela SOA existem como softwares fisicamente dependentes que dão suporte à obtenção dos objetivos estratégicos associados a computação orientada a serviços.

**Comentários:**

Na verdade, os serviços fornecidos existem como softwares fisicamente independentes.

**Gabarito:** Errado

---

**31. (CESPE / SERPRO – 2013)** Em um ambiente de SOA, recursos em uma rede são disponibilizados como serviços dependentes entre si, que só podem ser acessados com o conhecimento de sua

- implementação interna.

**Comentários:**

Na verdade, os serviços são independentes e não é necessário conhecer a implementação interna.

**Gabarito:** Errado

---

**32. (CESPE / ANTT – 2013)** No padrão CORBA, a IDL é uma linguagem utilizada para implementar o conteúdo de um objeto CORBA.

**Comentários:**

O IDL (Interface Definition Language) descreve interfaces das implementações dos objetos na rede; ele não implementa o conteúdo de um objeto CORBA!

**Gabarito:** Errado

---

**33. (CESPE / SERPRO – 2013)** Na arquitetura orientada a serviços, são utilizados serviços web objetos da arquitetura CORBA (common object request broker architecture), na qual são definidas as interfaces de comunicação entre as extremidades da rede de componentes.



### Comentários:

Essa questão é um pouco confusa! Não compreendo o que o examinador quis dizer com "(...) *serviços web objetos*", mas vamos tentar solucioná-la. Vocês sabem que SOA é independente de tecnologia, logo eu posso implementar SOA utilizando Web Services ou CORBA! A questão fala que, na arquitetura orientada a serviços, são utilizados serviços web objetos da arquitetura CORBA.

Para mim, isso não faz sentido – você pode utilizar Serviços Web ou pode utilizar objetos CORBA, mas a questão trata ambos como uma coisa só! Além disso, as interfaces de comunicação entre as extremidades da rede de computadores são definidas no UDDI (no caso de Web Services) ou no IDL (no caso de CORBA).

**Gabarito:** Errado

---

**34. (CESPE / MPOG – 2013)** Os padrões CORBA auxiliam a comunicação lógica entre objetos em arquiteturas de objetos distribuídos mesmo onde objetos implementados possuam diferentes linguagens ou plataformas.

### Comentários:

CORBA é realmente independente de tecnologias específicas (linguagens, plataformas, etc).

**Gabarito:** Correto

---

**35. (CESPE / ANAC – 2012)** Ao utilizar-se a arquitetura orientada a serviços (SOA), segue-se um conceito de arquitetura corporativa, situação em que os códigos são gerados para toda a empresa e são reutilizados de maneira eficiente e por várias aplicações.

### Comentários:

Perfeito! A questão trata da interoperabilidade intrínseca da arquitetura orientada a serviços.

**Gabarito:** Correto

---

**36. (CESPE / MEC – 2011)** A arquitetura orientada a serviço constitui um modelo arquitetônico que visa aumentar a agilidade e melhorar a relação custo/benefício de uma organização com referência à implantação de sistemas interoperáveis. Esse modelo tem como princípio a disponibilização de unidades lógicas de solução, em que a orientação a serviços tem sido aplicada de forma significativa.

### Comentários:



De fato, trata-se de uma arquitetura; aumenta a agilidade para atender novas demandas, visto que preconiza uma arquitetura fracamente acoplada; por conta disso, também melhor o custo/benefício da implantação de sistemas interoperáveis; unidades lógicas nada mais são que os serviços disponibilizados de forma padronizada e homogênea com fraco acoplamento e alta coesão.

**Gabarito:** Correto

---

**37. (CESPE / MEC – 2011)** A arquitetura SOA, orientada para a criação de componentes fracamente acoplados, é muito utilizada para componentes que não tenham interface bem definida ou cujos detalhes de implementação não sejam claros.

**Comentários:**

Opa... a interface deve ser muito bem definida!

**Gabarito:** Errado

---

**38. (CESPE / INMETRO – 2009)** A SOA estabelece que uma aplicação é construída por meio dos seguintes serviços: consumidor do serviço, fornecedor do serviço, localizador do serviço e publicador do serviço.

**Comentários:**

No modelo mais tradicional, são três papéis: consumidor ou solicitante de serviço, fornecedor ou prestador de serviço e registro de serviços.

**Gabarito:** Errado

---

**39. (CESPE / SECONT-ES – 2009)** SOA é uma arquitetura orientada a serviços, utilizada para interoperabilidade de sistemas por meio de conjunto de interfaces de serviços fracamente acoplados, em que um serviço pode ser descrito como uma representação lógica de uma atividade de negócio que tem um resultado específico, como, por exemplo, um relatório resultante de um data mining.

**Comentários:**

Perfeito! SOA é 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. Ele, de fato, pode descrever um serviço como uma representação lógica de uma atividade de negócio que tem um resultado específico.



**Gabarito:** Correto

---

**40. (CESPE / CEHAP-PB – 2009 – Letra C)** No padrão CORBA, o ORB localiza o objeto que fornece o serviço, prepara-o para uma solicitação, envia o serviço solicitado e retorna o resultado ao solicitante.

**Comentários:**

ORB realmente localiza o objeto que fornece o serviço, prepara-o para uma solicitação, envia o serviço solicitado e retorna o resultado ao solicitante. Ele é um módulo intermediário entre cliente/objeto, sendo responsável por aceitar requisições, enviá-las ao objeto correto e entregar a resposta ao cliente.

**Gabarito:** Correto

---

**41. (CESPE / STJ – 2005)** A arquitetura orientada a serviços (SOA) é uma forma de desenvolvimento monolítica em que os componentes de sistemas são serviços autônomos baseados em XML.

**Comentários:**

**O QUE SOA É...**

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

Não se trata de uma forma de desenvolvimento monolítica e, sim, distribuída! Além disso, os serviços não são necessariamente baseados em XML.

**Gabarito:** Errado

---



## QUESTÕES COMENTADAS – FCC

1. (FCC / TRF4 – 2014) 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.

**Gabarito:** Letra C

2. (FCC / AL-RN – 2013) Uma Arquitetura Orientada a Serviços (SOA) é uma forma de arquitetura de sistemas distribuídos que é tipicamente caracterizada 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 provedores 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:** Letra A

3. (FCC / DPE-SP – 2013) 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. No entanto, 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:** Letra B

4. (FCC / TRT11 – 2013) 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, trata-se da 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:** Letra D

5. (FCC / TRT-PE – 2012)

*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.	<b>Não é uma ferramenta.</b>	

(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:** Letra D

6. (FCC / TJ-PE – 2012) 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:

(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); (c) Correto. Uma boa prática é definir o tempo de guarda das versões de um serviço em funcionamento; (d) Correto. Como são projetos criados e realizados em paralelo e que evoluem como serviços, existem também versões (incluindo operações e implementações); (e) Correto. A autonomia, como nós vimos em aula, é inclusive um dos princípios de design do SOA.

**Gabarito:** Letra A

#### 7. (FCC / TJ-PE – 2012) 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:

(I) Correto; (II) Correto; (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 (Ex: EAI).

**Gabarito:** Letra B

**8. (FCC / TJ-PE – 2012)** 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:



(a) Correto; (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. Ele deve prestar o serviço de conectividade; (d) Errado. ESB não implementa SOA; (e) Correto, ele realmente define um protocolo e os fornecedores.

**Gabarito:** Letra D

**9. (FCC / TRT-AM – 2011)** 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 próprio 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, isto é, dependentes entre si.

**Gabarito:** Letra A

**10. (FCC / TRT-RS – 2011)** 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:

(I) Correto. 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; (II) Errado. Na verdade, temos baixo grau de acoplamento; (III) Errado. Na verdade, o processamento é distribuído.

**Gabarito:** Letra A

**11. (FCC / TRT-RS – 2011)** 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:



Todos os itens estão corretos, exceto o segundo: serviço é um componente fracamente acoplado.

**Gabarito:** Letra B

**12. (FCC / TRE-PE – 2011)** 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:

A questão trata do CORBA! 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).

**Gabarito:** Letra D

**13. (FCC / DPE-SP – 2010)** 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.	

A arquitetura orientada a serviços não é uma tecnologia.



14. (FCC / SEFAZ-SP – 2009) 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.	<b>Não é um produto.</b>	Não é um web service.
Não é um projeto de TI.	Não é um software.	Não é um framework.
<b>Não é uma metodologia.</b>	Não é solução de negócio.	Não é um middleware.
Não é um serviço.	Não é uma ferramenta.	

(I) Errado. Não se trata de um produto – pelo menos acertou na parte do *loosely coupled*; (II) Errado. Não se trata de uma metodologia ou meta. Ademais não utiliza padrões e protocolos específicos; (III) Correto. Realmente não requerem renovação completa de tecnologia e processos de negócio – é uma abordagem conceitual; (IV) Correto. Trata-se de uma abordagem arquitetura corporativa.

15. (FCC / METRÔ-SP – 2008) 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:

(a) Errado, ele reduz o acoplamento; (b) Errado, ele não implementa SOA; (c) Errado, não é obrigatório utilizar Web Services; (d) Correto, divide-se funções grandes em pequenas e distribui em forma de serviços; (e) Errado, não existe essa relação.

**Gabarito:** Letra D

---





## QUESTÕES COMENTADAS – FGV

1. (FGV / DPE-RJ – 2014) A arquitetura orientada a serviço (SOA) estabelece um princípio de arquitetura para aplicações distribuídas que fornecem serviços que atendem uma função específica do negócio. 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) Correto, de acordo com o princípio da abstração; (e) Errado, o princípio do baixo acoplamento preconiza justamente o inverso.

**Gabarito:** Letra D

2. (FGV / MEC – 2009) 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, isto é, serviços podem funcionar em ambientes completamente diferentes; Reúso Caixa-Preta, isto é, podem ser reutilizá-los sem precisar entender seu funcionamento interno, apenas suas interfaces e funcionalidades; Composição, isto é, podem se integrar com diversos outros componentes; distribuição, isto é, 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:** Letra B

3. (FGV / Senado Federal – 2008) 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.	



(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, o serviço é identificado por uma URI para que possa ser acessado.

**Gabarito:** Letra B

---



## QUESTÕES COMENTADAS – DIVERSAS BANCAS

1. (CESGRANRIO / IPEA – 2024) O Ipea resolveu fazer uso de um método de desenvolvimento de web services que usa componentes de software, chamados de serviços, para criar aplicações associadas à obtenção de dados de saneamento básico, de modo a melhorar a parte dinâmica de seus sites.

Considerando-se essa decisão e o requisito de que as respostas aos requerimentos dos usuários externos devem ser resolvidas mais rapidamente, o método de desenvolvimento indicado é o?

- a) EDA
- b) SOA
- c) WSA
- d) Adapter
- e) Singleton

### Comentários:

(a) Errado. EDA (Event-Driven Architecture) é uma arquitetura focada em eventos e reações a eventos, não diretamente na criação de web services;

(b) Correto. SOA (Service-Oriented Architecture) é um método de desenvolvimento que utiliza componentes de software chamados serviços para criar aplicações. Essa abordagem é adequada para melhorar a dinâmica dos sites e responder rapidamente às solicitações dos usuários;

(c) Errado. WSA (Web Services Architecture) é uma arquitetura geral para web services, mas não é um método específico de desenvolvimento como o SOA;

(d) Errado. Adapter é um padrão de design que permite que interfaces incompatíveis trabalhem juntas, não um método de desenvolvimento de web services;

(e) Errado. Singleton é um padrão de design que garante que uma classe tenha apenas uma instância, não um método de desenvolvimento de web services.

**Gabarito:** Letra B

2. (CESGRANRIO / FINEP – 2011) 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:

Não se trata de um middleware e, sim, de uma arquitetura para middleware – por essa razão, a questão foi anulada.

**Gabarito:** Anulada

3. (CESGRANRIO / IBGE – 2010) 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:

Um dos princípios básicos é o baixo acoplamento e, não, alto acoplamento.

**Gabarito:** Letra A

4. (CESGRANRIO / Casa da Moeda – 2009) 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:

(a) Correto, trata-se do princípio do baixo acoplamento; (b) Errado, fere o princípio da independência de estados; (c) Errado, não existe essa característica; (d) Errado, fere o princípio da abstração de serviços; (e) Errado, fere o princípio do contrato de serviços.

**Gabarito:** Letra A

5. (FIP / CMSJC – 2009) 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 recursos preconiza justamente o inverso; (c) Correto. De acordo com o que 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.



6. (CESGRANRIO / PETROBRÁS – 2008) 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 dividida 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:

O princípio da independência de estados (também chamado de manutenção de estados) preconiza que se evite reter o estado entre os acessos de clientes distintos.

7. (CESGRANRIO / PETROBRÁS – 2008) 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:



A questão trata do IDL (Interface Definition Language). Trata-se de 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.

**Gabarito:** Letra E

---





## LISTA DE QUESTÕES – CESPE

1. **(CESPE / MPE-TO – 2024)** SOA (Service Oriented Architecture) é uma arquitetura empregada para reutilizar serviços em sistemas diferentes ou combinar vários serviços independentes para realizar tarefas complexas.
2. **(CESPE / BNB – 2022)** Implementações de SOA por meio de um barramento de serviços implicam a criação de um ponto único de falha, uma vez que clientes e serviços da organização não poderão se comunicar se o barramento de serviços se tornar inoperante.
3. **(CESPE / FUNPRES-EXE - 2022)** Em uma arquitetura orientada a serviços, o registro central é o elemento que disponibiliza os serviços, enquanto que o provedor age como concentrador de serviços.
4. **(CESPE / FUNPRES-EXE - 2022)** ESB (Enterprise Service Bus) é um barramento sobre o qual os serviços são oferecidos aos clientes, e que provê interoperabilidade entre sistemas heterogêneos.
5. **(CESPE / FUNPRES-EXE - 2022)** A BPEL (Business Process Execution Language) é utilizada para implementar uma composição de serviços do tipo coordenação.
6. **(CESPE / Petrobrás - 2022)** Com a SOA é mais fácil escalar ou ampliar o uso de uma funcionalidade para plataformas ou ambientes novos, o que resulta em maior agilidade e eficiência no desenvolvimento, por outro lado, em geral, acaba por promover aumento nos custos da solução.
7. **(CESPE / STM – 2018)** Em SOA, orquestração é a forma de arranjar serviços diferentes para serem executados em uma ordem preestabelecida.
8. **(CESPE / TCE-PA – 2016)** Os quatro atores necessários à composição do ciclo de vida de uma solução SOA incluem o arquiteto SOA, que é o responsável por mapear os processos de negócio.
9. **(CESPE / TCE-PR – 2016 – Letra A)** Os componentes de um web service não devem ser reutilizados, para que tenham uma evolução independente, característica fundamental de soluções SOA.
10. **(CESPE / TCE-PR – 2016 – Letra B)** Um barramento de serviços ESB (Enterprise Service BUS) é a implementação física de infraestrutura com base em web services.
11. **(CESPE / TCE-PR – 2016 – Letra C)** Conforme as soluções SOA, cabe ao cliente monitorar o desempenho da aplicação, uma vez que ele é o maior interessado em utilizar os serviços fornecidos.



- 12. (CESPE / TCE-PR – 2016 – Letra E)** Inicialmente, o consultor de negócios, o arquiteto SOA, o provedor de serviço e o consumidor de serviço são suficientes para a composição do ciclo de vida de soluções SOA.
- 13. (CESPE / TCE-PR – 2016)** A respeito de SOA e CORBA, assinale a opção correta.
- a) No modelo operacional triangular, o registro do serviço determina o comportamento de quem disponibiliza o serviço, ou seja, do dono do serviço, que é responsável, entre outros aspectos, pela infraestrutura de acesso ao serviço.
  - b) A utilização de SOA em uma organização permite a descoberta de novos processos de negócio para uma posterior associação com serviços de TI.
  - c) Entre os possíveis mapeamentos de negócio e de TI, a abordagem de baixo para cima (bottom-up) determina que a organização identifique, primeiramente, os processos que considerar prioritários.
  - d) Assim como o DCOM, o CORBA é executado apenas em ambiente Windows.
  - e) O modelo de referência da OMG (Object Management Group) para CORBA define a interface de aplicação, isto é, o conjunto de dados públicos do objeto que possibilita a comunicação por meio de chamadas aos métodos desse objeto com os parâmetros apropriados.
- 14. (CESPE / TCE-PR – 2016 – Letra B)** Um barramento de serviços ESB (Enterprise Service BUS) é a implementação física de infraestrutura com base em web services.
- 15. (CESPE / MPOG-ATI – 2015)** Em arquiteturas orientadas a serviço, um serviço deve ser implementado de forma a garantir um fraco acoplamento.
- 16. (CESPE / TRE-MT – 2015)** Assinale a opção correta, relativa a SOA (service-oriented architecture).
- a) Um dos objetivos da orientação a serviços é estabelecer maior interoperabilidade intrínseca, a fim de reduzir a necessidade de integração.
  - b) A SOA visa diminuir a perspectiva federada de serviços, de modo a compartilhar dados entre diferentes plataformas computacionais.
  - c) Em se tratando de uma solução orientada a serviços, as unidades de lógica encapsulam funcionalidades específicas a um processo de negócio.
  - d) Por fornecerem um intervalo de funcionalidades genéricas, os serviços não agnósticos não devem ser valorizados na SOA.



e) Serviços não agnósticos adaptam-se inúmeras vezes a diferentes processos de negócio como parte de diferentes soluções orientadas a serviços.

17. (CESPE / TCE-RN – 2015) A arquitetura CORBA permite realizar a intercomunicação entre computadores de arquiteturas e portes distintos por meio do protocolo-padrão EIGRP, versão melhorada do IGRP, que permite compor, de forma síncrona ou assíncrona, objetos, dados e unidades individuais.
18. (CESPE / MEC – 2015) Enterprise Service Bus é um barramento para integração de serviços coletivos de bancos de dados distribuídos sobre arquitetura IP (Internet Protocol) para independência de camada de serviços.
19. (CESPE / TCU – 2015) Um projeto de software orientado pela governança SOA deve estar alinhado não só com a governança de TI, mas também com a governança da arquitetura empresarial.
20. (CESPE / ANTAQ – 2014) No uso de SOA, a troca de dados requer protocolos intermediários, os quais poderão representar uma perda de desempenho das aplicações.
21. (CESPE / ANATEL – 2014) Os serviços compostos podem apresentar limitações de segurança, especialmente pelo fato de permitirem que os serviços básicos que os compõem sejam chamados individualmente; não havendo mecanismos que permitam que os serviços básicos sejam chamados apenas pelos serviços de mais alto nível.
22. (CESPE / SUFRAMA – 2014) O desenvolvimento de aplicações em ambientes corporativos que utilizam SOA permite alto acoplamento e grande redundância de funcionalidades.
23. (CESPE / SUFRAMA – 2014) Entre os princípios básicos da SOA estão os serviços que abstraem a lógica, que compartilham um contrato formal, que evitam alocação de recursos por longos períodos e que são autônomos e reutilizáveis.
24. (CESPE / SUFRAMA – 2014) No CORBA, o ORB (Object Request Broker) é o responsável por encontrar a implementação de objeto para o pedido feito pelo cliente, preparar a implementação de objeto para receber o pedido e transmitir os dados do pedido.
25. (CESPE / ANTT – 2013) A SOA pode ser definida como um tipo de arquitetura que utiliza serviços como blocos de construção para facilitar a integração em ambientes corporativos e a reutilização de componentes por meio do baixo acoplamento.
26. (CESPE / STF – 2013) A arquitetura orientada a serviços é utilizada para interoperabilidade de sistemas heterogêneos por meio de conjunto de serviços fracamente acoplados. A orientação a serviços utiliza protocolos padrão e interfaces convencionais para facilitar o acesso à lógica de negócios e às informações entre serviços distintos.



27. (CESPE / MPOG – 2013) O SOA garante serviços fortemente acoplados, fracamente coesos e com alta possibilidade de reutilização.
28. (CESPE / MPOG – 2013) O SOA promove a integração entre o negócio e a tecnologia da informação por meio de serviços, que são o principal componente dessa arquitetura.
29. (CESPE / SERPRO – 2013) Por ser dependente de tecnologia, o ambiente de SOA tem de ser implementado em protocolos específicos.
30. (CESPE / SERPRO – 2013) No nível do aplicativo, os serviços fornecidos pela SOA existem como softwares fisicamente dependentes que dão suporte à obtenção dos objetivos estratégicos associados a computação orientada a serviços.
31. (CESPE / SERPRO – 2013) Em um ambiente de SOA, recursos em uma rede são disponibilizados como serviços dependentes entre si, que só podem ser acessados com o conhecimento de sua implementação interna.
32. (CESPE / ANTT – 2013) No padrão CORBA, a IDL é uma linguagem utilizada para implementar o conteúdo de um objeto CORBA.
33. (CESPE / SERPRO – 2013) Na arquitetura orientada a serviços, são utilizados serviços web objetos da arquitetura CORBA (common object request broker architecture), na qual são definidas as interfaces de comunicação entre as extremidades da rede de componentes.
34. (CESPE / MPOG – 2013) Os padrões CORBA auxiliam a comunicação lógica entre objetos em arquiteturas de objetos distribuídos mesmo onde objetos implementados possuam diferentes linguagens ou plataformas.
35. (CESPE / ANAC – 2012) Ao utilizar-se a arquitetura orientada a serviços (SOA), segue-se um conceito de arquitetura corporativa, situação em que os códigos são gerados para toda a empresa e são reutilizados de maneira eficiente e por várias aplicações.
36. (CESPE / MEC – 2011) A arquitetura orientada a serviço constitui um modelo arquitetônico que visa aumentar a agilidade e melhorar a relação custo/benefício de uma organização com referência à implantação de sistemas interoperáveis. Esse modelo tem como princípio a disponibilização de unidades lógicas de solução, em que a orientação a serviços tem sido aplicada de forma significativa.
37. (CESPE / MEC – 2011) A arquitetura SOA, orientada para a criação de componentes fracamente acoplados, é muito utilizada para componentes que não tenham interface bem definida ou cujos detalhes de implementação não sejam claros.



- 38. (CESPE / INMETRO – 2009)** A SOA estabelece que uma aplicação é construída por meio dos seguintes serviços: consumidor do serviço, fornecedor do serviço, localizador do serviço e publicador do serviço.
- 39. (CESPE / SECONT-ES – 2009)** SOA é uma arquitetura orientada a serviços, utilizada para interoperabilidade de sistemas por meio de conjunto de interfaces de serviços fracamente acoplados, em que um serviço pode ser descrito como uma representação lógica de uma atividade de negócio que tem um resultado específico, como, por exemplo, um relatório resultante de um data mining.
- 40. (CESPE / CEHAP-PB – 2009 – Letra C)** No padrão CORBA, o ORB localiza o objeto que fornece o serviço, prepara-o para uma solicitação, envia o serviço solicitado e retorna o resultado ao solicitante.
- 41. (CESPE / STJ – 2005)** A arquitetura orientada a serviços (SOA) é uma forma de desenvolvimento monolítica em que os componentes de sistemas são serviços autônomos baseados em XML.



## GABARITO

- |     |         |     |        |
|-----|---------|-----|--------|
| 1.  | CORRETO | 41. | ERRADO |
| 2.  | CORRETO |     |        |
| 3.  | ERRADO  |     |        |
| 4.  | CORRETO |     |        |
| 5.  | ERRADO  |     |        |
| 6.  | ERRADO  |     |        |
| 7.  | CORRETO |     |        |
| 8.  | ERRADO  |     |        |
| 9.  | ERRADO  |     |        |
| 10. | ERRADO  |     |        |
| 11. | ERRADO  |     |        |
| 12. | CORRETO |     |        |
| 13. | LETRA E |     |        |
| 14. | ERRADO  |     |        |
| 15. | CORRETO |     |        |
| 16. | LETRA A |     |        |
| 17. | ERRADO  |     |        |
| 18. | ERRADO  |     |        |
| 19. | CORRETO |     |        |
| 20. | CORRETO |     |        |
| 21. | ERRADO  |     |        |
| 22. | ERRADO  |     |        |
| 23. | CORRETO |     |        |
| 24. | CORRETO |     |        |
| 25. | CORRETO |     |        |
| 26. | CORRETO |     |        |
| 27. | ERRADO  |     |        |
| 28. | CORRETO |     |        |
| 29. | ERRADO  |     |        |
| 30. | ERRADO  |     |        |
| 31. | ERRADO  |     |        |
| 32. | ERRADO  |     |        |
| 33. | ERRADO  |     |        |
| 34. | CORRETO |     |        |
| 35. | CORRETO |     |        |
| 36. | CORRETO |     |        |
| 37. | ERRADO  |     |        |
| 38. | ERRADO  |     |        |
| 39. | CORRETO |     |        |
| 40. | CORRETO |     |        |



## LISTA DE QUESTÕES – FCC

1. (FCC / TRF4 – 2014) 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.

2. (FCC / AL-RN – 2013) Uma Arquitetura Orientada a Serviços (SOA) é uma forma de arquitetura de sistemas distribuídos que é tipicamente caracterizada 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 provedores 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.

3. (FCC / DPE-SP – 2013) 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.

4. (FCC / TRT11 – 2013) 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.

5. (FCC / TRT-PE – 2012)

*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.

**6. (FCC / TJ-PE – 2012) 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.

**7. (FCC / TJ-PE – 2012) 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.

8. (FCC / TJ-PE – 2012) 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.

9. (FCC / TRT-AM – 2011) 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.

**10. (FCC / TRT-RS – 2011) 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.

**11. (FCC / TRT-RS – 2011) 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.

**12. (FCC / TRE-PE – 2011)** 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.

**13. (FCC / DPE-SP – 2010)** 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.

**14. (FCC / SEFAZ-SP – 2009)** 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.

**15. (FCC / METRÔ-SP – 2008) 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.



## GABARITO

- |    |         |     |         |     |         |
|----|---------|-----|---------|-----|---------|
| 1. | LETRA C | 6.  | LETRA A | 11. | LETRA B |
| 2. | LETRA A | 7.  | LETRA B | 12. | LETRA D |
| 3. | LETRA B | 8.  | LETRA D | 13. | LETRA C |
| 4. | LETRA D | 9.  | LETRA A | 14. | LETRA C |
| 5. | LETRA D | 10. | LETRA A | 15. | LETRA D |





## LISTA DE QUESTÕES – FGV

1. **(FGV / DPE-RJ – 2014)** A arquitetura orientada a serviço (SOA) estabelece um princípio de arquitetura para aplicações distribuídas que fornecem serviços que atendem uma função específica do negócio. 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.
  
2. **(FGV / MEC – 2009)** 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.
  
3. **(FGV / Senado Federal – 2008)** 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.



## GABARITO

1. LETRA D

2. LETRA B

3. LETRA B





## LISTA DE QUESTÕES – DIVERSAS BANCAS

1. **(CESGRANRIO / IPEA – 2024)** O Ipea resolveu fazer uso de um método de desenvolvimento de web services que usa componentes de software, chamados de serviços, para criar aplicações associadas à obtenção de dados de saneamento básico, de modo a melhorar a parte dinâmica de seus sites.

Considerando-se essa decisão e o requisito de que as respostas aos requerimentos dos usuários externos devem ser resolvidas mais rapidamente, o método de desenvolvimento indicado é o?

- a) EDA
- b) SOA
- c) WSA
- d) Adapter
- e) Singleton

2. **(CESGRANRIO / FINEP – 2011)** 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)

3. **(CESGRANRIO / IBGE – 2010)** 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.

**4. (CESGRANRIO / Casa da Moeda – 2009)** 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 reuso dos mesmos.

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

**5. (FIP / CMSJC – 2009)** 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.

**6. (CESGRANRIO / PETROBRÁS – 2008)** 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 dividida 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.

7. (CESGRANRIO / PETROBRÁS – 2008) 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



## GABARITO

- |    |         |    |         |    |         |
|----|---------|----|---------|----|---------|
| 1. | LETRA B | 4. | LETRA A | 7. | LETRA E |
| 2. | ANULADA | 5. | LETRA C |    |         |
| 3. | LETRA A | 6. | LETRA D |    |         |







# 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.