DIREÇÃO CONCURSOS

Aula 01 – Web Services, UDDI, WSDL, SOAP, REST

Desenvolvimento de Sistemas para Analista Legislativo

Senado Federal

Prof. Hamilton Rodrigues

Sumário

Web Services

Introdução

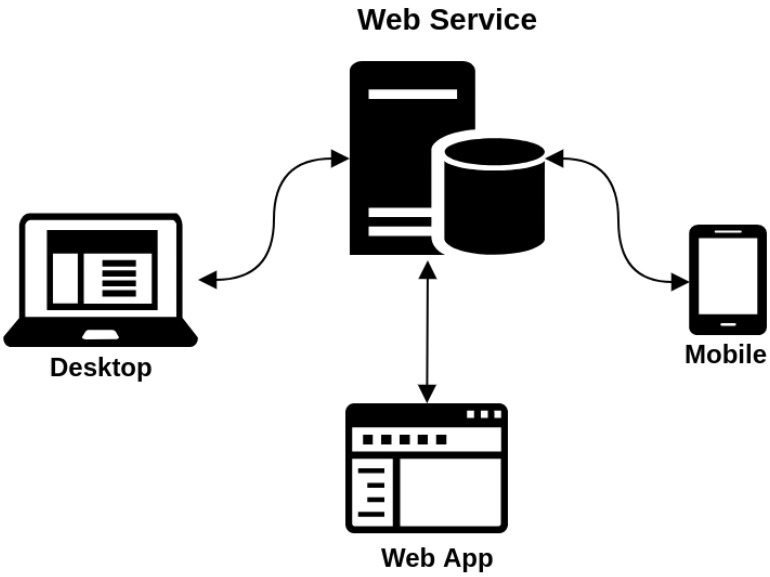
Web Services são uma forma padronizada de **integrar sistemas** onde uma aplicação oferece um serviço HTTP (o servidor) e a outra o consome (cliente). É um **método de comunicação** entre máquinas (computador-computador) em uma rede. Os Web Services foram criados com o objetivo de possibilitar a **interoperabilidade** entre os sistemas. Afinal, de que adianta termos vários sistemas "espalhados" em diversos lugares se eles não se comunicam?

Para facilitar o entendimento, vamos pensar em um caso concreto.

Vamos supor que temos 3 dispositivos diferentes (1 Desktop, 1 Web App e 1 Smartphone Mobile) onde executaremos aplicativos de informação meteorológica. Os aplicativos nesses dispositivos só têm as telas para mostrar as informações de tempo. Ou seja, só têm a "casca". Eles não possuem os **dados** da previsão meteorológica em si.

Os dados de previsão meteorológica estão centralizados em um banco de dados num servidor. A solução tecnológica ideal neste caso é que o servidor que possui os dados ofereça um **web service** com os dados de previsão meteorológica para que os clientes (Desktop, Web App e Smartphone Mobile) possam consumi-lo e disponibilizar a previsão de tempo nos seus aplicativos.

Na figura abaixo vemos o mecanismo de funcionamento desse um **web service**. Nela, o banco de dados que dispõe de informações abre um web service que por sua vez é consumido (invocado, chamado) pelos dispositivos clientes.



Essa é a ideia de um web service. Os dispositivos clientes (Desktop, Web App e Smartphone) têm seus respectivos aplicativos cada qual com as suas especificidades. Por exemplo, o app Desktop poderia ter sido implementado em C#.NET, o WebApp em Java e o Mobile em linguagem Swift iOS. Por sua vez, o web service poderia ter sido implementado em Java. A tecnologia web service permite a interoperabilidade entre todos esses sistemas heterogêneos. Para os sistemas (clientes e servidor) não interessa em qual tecnologia foi implementado o outro sistema com o qual ele irá se comunicar. O que interessa é a interface de comunicação, que é o **web service**.

Observe como o Cespe abordou isso na questão abaixo.

(CESPE – MEC – 2015)

Em um web service, a linguagem de implementação e a plataforma utilizada são relevantes para os clientes.

RESOLUÇÃO:

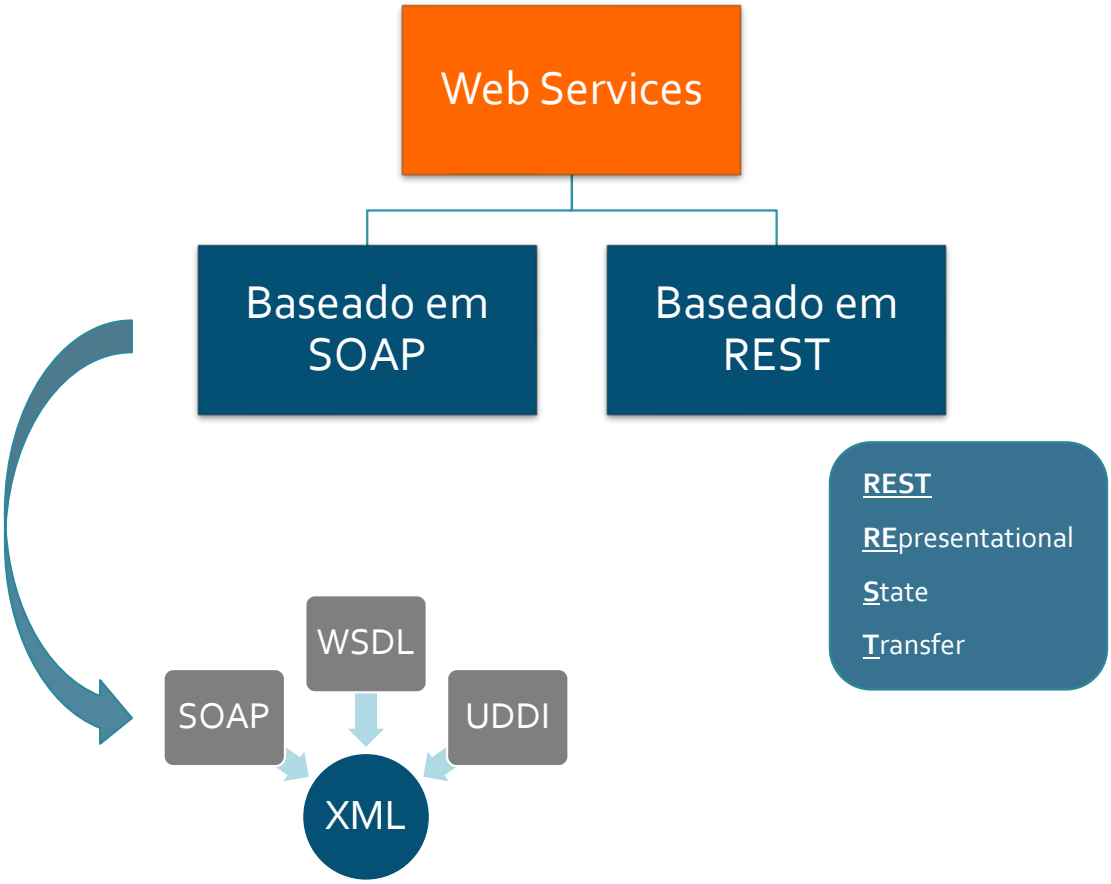
E aí, galera? Para o cliente que irá consumir o web service, interessa a linguagem de implementação e a plataforma utilizada no web service?

Não! O web service abstrai as especificidades internas do servidor e oferece uma interface agnóstica, que qualquer cliente autorizado poderia acessar sem ter conhecimento da implementação interna e plataforma utilizada pelo servidor (web service). Essa é uma das grandes vantagens dos web services.

Resposta: Errado

A tecnologia de webservice tem alguns conceitos importantes que serão discutidos a seguir.

Há basicamente 2 modelos de implementação de web service: os baseados em **SOAP** e os baseados em **REST**.



Modelo SOAP

Acrônimo que significa ***Simple Object Access Protocol***. É o protocolo de mensagens que especifica a forma de comunicação entre os web services e seus clientes. Seu propósito é prover extensibilidade, neutralidade e independência.

Com o uso do **SOAP** reduz-se o **acoplamento** entre os sistemas. Dentro de uma arquitetura distribuída, os servidores oferecem seus web services e os clientes os consomem, sem necessidade de conhecimento da estrutura interna dos servidores. Independentemente de qualquer modelo de programação ou outra implementação específica, o **SOAP** possibilita que dois processos se comuniquem, desconsiderando o hardware e a plataforma que eles estão sendo executados. Ele utiliza frequentemente o protocolo HTTP (*Hypertext Transfer Protocol*) para intercâmbio de mensagens em formato XML.

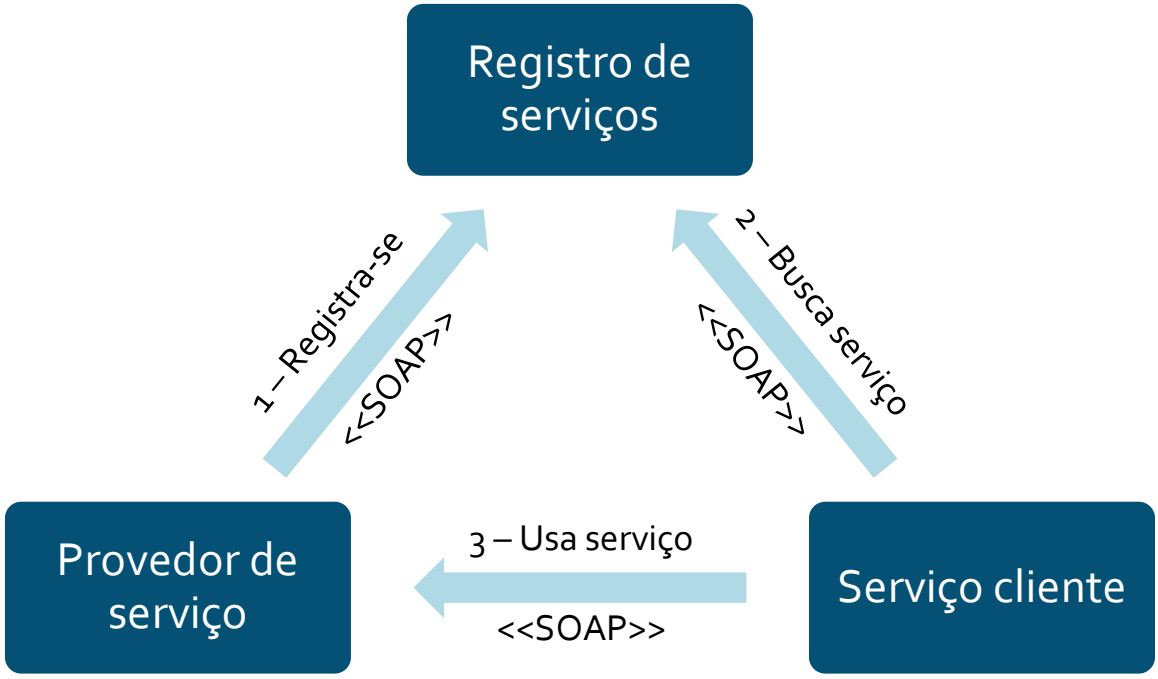
Um dos grandes benefícios do **SOAP** é que ele é aberto e foi adotado pela maioria das grandes empresas de hardware e software. Ele acabou se transformando em um padrão, de fato.

A arquitetura do protocolo SOAP tem 3 componentes principais. Os 2 componentes básicos são o **Cliente** e o **Provedor do serviço** (servidor). O papel do servidor (web service) é o de prover o serviço que é consumido pelo cliente. Além desses 2, temos um componente adicional que é o **Registro de serviços**.

Professor, e qual o papel do ***Registro de serviços****?* 🤔

O **Registro de serviços** funciona mais ou menos como uma "lista telefônica". Hoje em dia, com a internet, a lista telefônica está meio "demodê", mas você há de se lembrar que ela funciona como um registro de todos os números de telefone de empresas e pessoas de uma certa região.

No contexto dos web services, o **Registro de serviços** é um diretório com informações sobre todos os serviços disponíveis em uma rede, armazenados no padrão **UDDI**. Com o **Registro de serviços,** você não precisa saber exatamente o endereço IP e porta onde se localiza o web service. Basta consultar o que está disponível no diretório do **Registro de serviços**. Obviamente, para que o registro funcione, todo web service precisa se registrar previamente no diretório (registro de serviços).

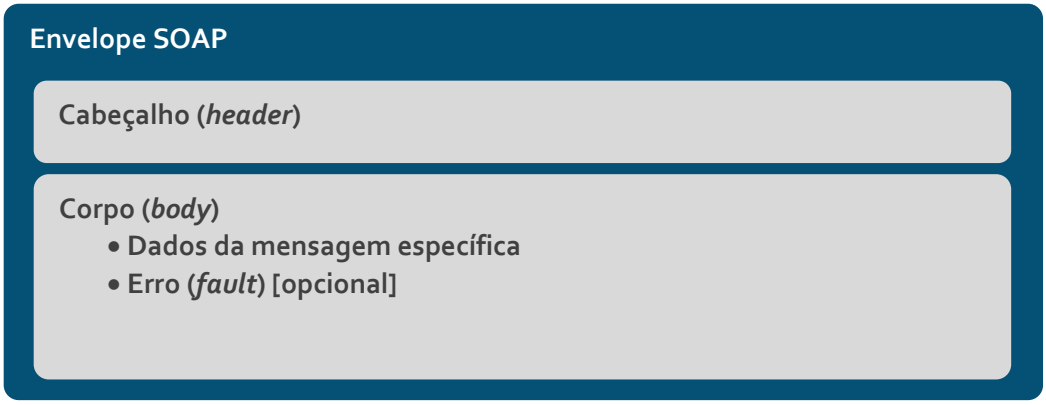


O esquema acima possibilita a visão da sequência em que o processo se dá. No **passo 1**, o **provedor de serviço** necessita se registrar no **Registro de serviços**. Se não, nunca será descoberto. O **provedor de serviço** registra todas as informações do web service por meio de um documento XML estruturado de acordo com o padrão **WSDL**.

No **passo 2**, o **Serviço Cliente** consulta o **Registro de serviços** para achar o serviço que deseja. Ao encontrar, ele obtém sua descrição (WSDL). No **passo 3**, o **Serviço Cliente** usa de fato o web service por meio de chamadas aos seus métodos.

Mensagem SOAP

A **mensagem SOAP** funciona como um pacote para transportar dados encapsulados em documentos no formato XML. Os componentes básicos da mensagem SOAP são: **envelope**, **cabeçalho**, **corpo** e **erro**.



O **envelope** é o container de toda a mensagem SOAP. Ele pode conter a declaração de *namespace*, que indica a versão de SOAP usada. Uma versão incorreta gera um erro do tipo *versionMismatch.* Contém também um atributo que define o estilo de codificação (*encoding style*). O "encoding style" é uma URI (Exemplo http://www.w3.org/2001/12/soap-encoding) que define como os dados são representados no documento XML.

O **cabeçalho** (***header***) possui informações de controle e processamento. Ele é opcional e carrega informações adicionais como, por exemplo, se a mensagem deve ser processada por um determinado nó intermediário. É importante lembrar que, ao trafegar pela rede, a mensagem normalmente passa por diversos pontos intermediários até alcançar o destino final. Se utilizado, o ***header*** deve ser o primeiro elemento do **envelope**.

O **corpo** (***body***) é a mensagem de fato (*payload*). Contém informação a ser transportada para o seu destino final.

O **erro** (***fault***) é um elemento opcional com informações e status de erro da mensagem.

Observe abaixo um exemplo de envelope SOAP.

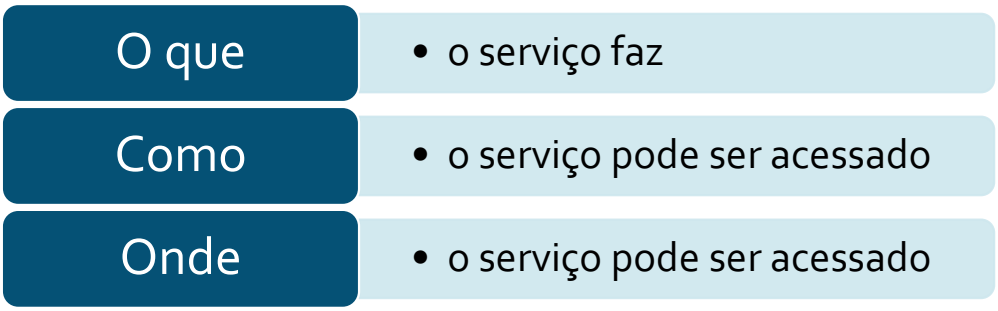
```
POST http://www.dominandoti.com.br/consultaCPF HTTP/1.1
Host: www.dominandoti.com.br
Content-Type: text/xml; charset="utf-8"
SOAPAction: "http://www.dominandoti.com.br/consultarCPF"
Content-Length: 314

<SOAP:Envelope xmlns:SOAP= "http://www.w3.org/2001/12/soap-envelope">
    <SOAP:Header>
    <!-- conteudo do cabecalho -->
    </SOAP:Header>
    <SOAP:Body xmlns:dti="http://www.dominandoti.com.br/tls/servicoCPF/wsdl">
        <dti:infoGestor>
          <dti:cpf>87598930104</dti:cpf>
        </dti:infoGestor>
    </SOAP:Body>
</SOAP:Envelope>
```

WSDL (Web service Description Language)

É uma **linguagem de descrição de web services** em formato XML. É por meio do **WSDL** que o servidor especifica os serviços que deseja expor na rede para quem quiser consultá-los.

Veja o exemplo abaixo de um webservice com um método chamado **consultaPrevisaoDoTempo** escrito em WSDL.

De uma forma resumida, podemos dizer que o WSDL define o **contrato** do serviço. Esse contrato estabelece respostas para 3 perguntas sobre o web service: **o que, como, onde**.



A seguir, um exemplo concreto de um documento XML de descrição de web service em linguagem WSDL.

```
<?xml version="1.0"?>
<definitions name="ServicoCPF"  targetNamespace="http://www.dominandoti.com.br/tls/servicoCPF/wsdl"
        xmlns="http://schemas.xmlsoap.org/wsdl/"
        xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
        xmlns:tns=" http://www.dominandoti.com.br/tls/servicoCPF/wsdl "
       xmlns:xsd="http://www.w3.org/2001/XMLSchema">

        <types>
        <xsd:element name="infoGestor">
                <xsd:complexType>
                        <xsd:sequence> <xsd:element  name="cpf" type="xsd:string" /> </xsd:sequence>
                </xsd:complexType>
        </xsd:element>
        </types>

       <message name="ConsultaCPFRequest">
                <part name= "info" element="infoGestor"/>
       </message>
       <message name="ConsultaCPFResponse">
                <part name="irregular" type="xsd:boolean"/>
       </message>
       <portType name="CPFPortType">
                <operation name="consultarCPF">
                        <input message="tns:ConsultaCPFRequest"/>
                        <output message="tns:ConsultaCPFResponse"/>
                </operation>
       </portType>

       <binding name="CPFBinding" type="tns:CPFPortType">
       <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="consultarCPF">
                <soap:operation soapAction="consultarCPF"/>
                <input> <soap:body use="literal"/> </input>
                <output> <soap:body use="literal"/> </output>
         </operation>
        </binding>

       <service name="CPFService">
                <port name="CPFPort" binding="tns:CPFBinding" >
                <soap:address location="http://www.dominandoti.com.br/consultaCPF" />
                </port>
       </service>
</definitions>
```

Definição de um type: o elemento **infoGestor** usado para passar o CPF a ser consultado como input do método **consultaCPF**

Formato da mensagem de requisição ao método **consultarCPF**

Formato da mensagem de resposta do método **consultarCPF**

Mensagem de requisição ao método **consultarCPF**

Mensagem de resposta do método **consultarCPF**

Operação (método) que o web service **CPFService** disponibiliza: **consultarCPF**.

Input (argumento) do método

Output (retorno) do método

Web service **CPFService**

Onde o web service **CPFService** se localiza

O exemplo acima de WSDL contém 1 único web service com 1 única operação. Esse XML poderia ser bem maior e mais complexo, conter diversos serviços, cada um com diversas operações, envolvendo diversos tipos diferentes, e por aí vai.

UDDI (*Universal Description Discovery and Integration*)

É um protocolo que foi criado para facilitar a arquitetura orientada a serviços via web services. O **UDDI** é um mecanismo que permite que os servidores registrem (ou publiquem) seus webservices em um **diretório** de forma que os clientes possam encontrá-los com facilidade.

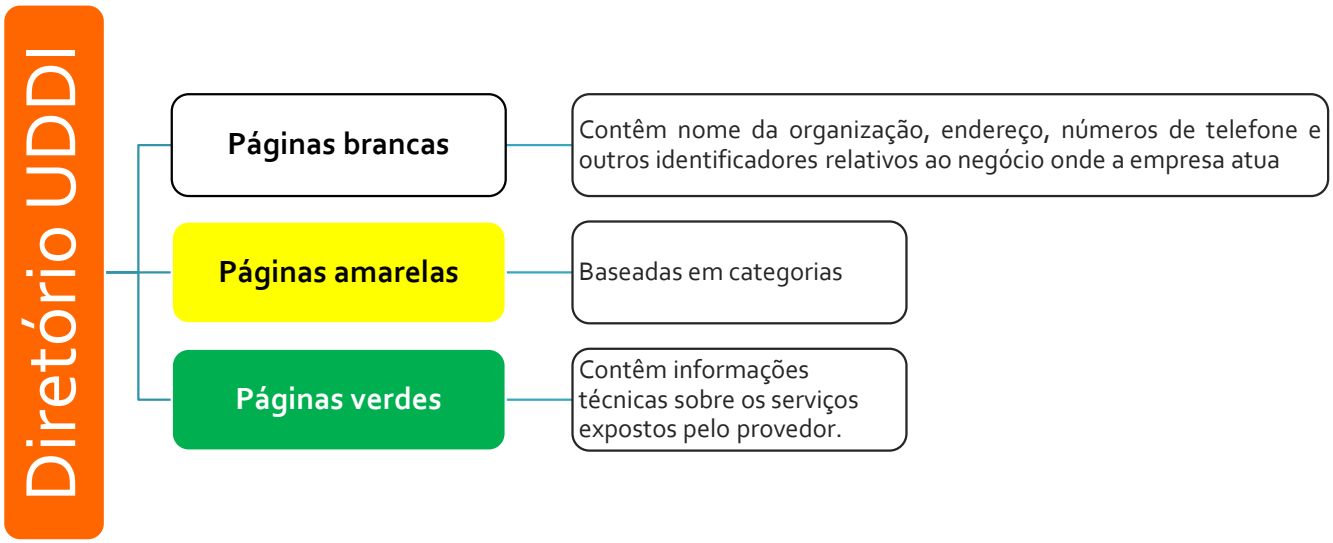
O UDDI é o protocolo que implementa a nossa "lista telefônica" que mencionamos anteriormente nesta aula como uma metáfora para o **Registro de serviços**.

Para clarificar como isso funciona, pense novamente no nosso exemplo de web service meteorológico. O fato de estar disponível um web service de consulta à previsão do tempo não significa necessariamente que os clientes saibam que ele exista. Com o **UDDI**, os clientes podem consultar os web services que estão disponíveis na rede por meio de um protocolo padronizado e assim poder invocá-los facilmente.

Registros UDDI são documentos XML que contêm essencialmente metadados sobre os serviços. Esses metadados são, principalmente, o nome dos web services, além dos nomes das operações de cada web service bem como os respectivos argumentos e retornos de cada operação.

Os Registros de serviços UDDI podem ser **públicos**, se forem disponibilizados na www (internet). Neste caso, qualquer um poderá acessar seu diretório e descobrir o que está registrado nele. Contudo, o mais comum é que os Registros de serviços UDDI sejam **privados**, isto é, visíveis somente no âmbito da intranet de uma organização.

Existem 3 categorias de informações dentro de um diretório UDDI: as chamadas páginas **brancas**, **amarelas** e **verdes**.



Pode parecer estranha essa classificação em páginas **brancas**, **amarelas** e **verdes** (a mim me parece!), mas isso já foi cobrado em concurso. Veja esta questão.

(Cespe - TRT-8 – 2016)

Os diretórios UDDI são catálogos de web services que descrevem o negócio e os serviços disponíveis por meio das páginas amarelas, as quais definem as principais características da companhia, e das páginas brancas, as quais detalham a interface para o serviço.

RESOLUÇÃO:

Na verdade, as páginas que definem as principais características da companhia são as brancas e não as amarelas.

E as páginas que detalham a interface para o serviço são as verdes, e não as brancas.

Resposta: Errado

Modelo REST

O modelo **REST** (***Representational State Transfer***) é uma outra abordagem, uma alternativa ao SOAP, que estudamos até aqui. O REST é mais que um protocolo. É um **estilo arquitetural**. Ele foi proposto em 2000 por um pesquisador da University of California, Irvine chamado Roy Fielding. Trata-se de um padrão aberto, não proprietário de nenhuma empresa. As principais propriedades do REST são as seguintes.

Comparativamente ao SOAP, as principais vantagens do REST são a sua **simplicidade** e **baixo *overhead* de comunicação**. Outra característica importante é que ele permite a **alta escalabilidade**. Alta escalabilidade é a capacidade de um sistema escalar, isto é, manter a boa performance mesmo quando o número de requisições aumenta muito.

Anteriormente, nesta aula, vimos o exemplo de um contrato padrão WSDL, um XML bem complexo. No SOAP, **cada web service tem seu próprio contrato WSDL**. O **REST** é mais simples. A curva de aprendizado para entender o funcionamento da arquitetura e implementar um serviço REST é menor.

Uniform contract

A tecnologia REST trabalha com o conceito de ***uniform contract***. Isso significa que não é necessário definir contratos individuais para cada web service. Os serviços compartilham um **contrato uniforme** e este contrato é definido por meio dos métodos HTTP. Para invocar um serviço REST, basta passar o **identificador de recurso** (*resource identifier*).

O **identificador de recurso** é o simples caminho URI (ou URL) onde o serviço se encontra. Por exemplo, imagine que você tem um banco de dados de questões de concurso e deseja oferecer um serviço REST com as questões. Um identificador de recurso possível seria http://www.meusite.com.br/questoes/id/123456. Esse é um dos motivos que faz o REST ser mais simples que o SOAP.

Outra característica do REST é o **baixo *overhead* de comunicação**. Lembre-se que os envelopes SOAP são documentos em formato XML. E o XML é cheio de marcações, ou seja, tem muita "gordura", muito texto extra para marcar e descrever os dados.

As comunicação com web services REST, por outro lado, é bem enxuta. A bem da verdade, nada impede que um web service REST se comunique por meio de mensagens XML. Mas, quase sempre, web services RESTful recebem e transmitem mensagens no formato **JSON**.

Mensagens JSON

O **JSON** (***JavaScript Object Notation***) é um formato auto descritivo mais leve que o XML.

Veja abaixo um exemplo de comparativo entre 2 documentos: um XML versus um JSON. Note como a mensagem JSON consome bem menos *bytes* para transmitir as mesmas informações. Acho que esse comparativo resume bem 2 vantagens do REST: **simplicidade** e **baixo *overhead***.

E essas vantagens decorrem do fato de o REST trabalhar com o formato JSON.

XML

```
<empinfo>
   <employees>
      <employee>
         <name>James Kirk</name>
         <age>40></age>
      </employee>
      <employee>
         <name>Jean-Luc Picard</name>
         <age>45</age>
      </employee>
      <employee>
         <name>Wesley Crusher</name>
         <age>27</age>
      </employee>
   </employees>
</empinfo>
```

JSON

```
{  "empinfo" :
      {
            "employees" :  [
            {
                  "name" : "James Kirk",
                  "age" : 40,
            },
            {
                  "name" : "Jean-Luc Picard",
                  "age" : 45,
            },
            {
                  "name" : "Wesley Crusher",
                  "age" : 27,
            }
                              ]
      }
}
```

Mas nem tudo são flores. O REST também tem desvantagens. Uma dessas desvantagens é que ele é **menos seguro** que o SOAP. O protocolo SOAP tem recursos avançados que incrementam a segurança da informação como criptografia, autenticações seguras, etc. O REST não tem tantos recursos de segurança.

Outra desvantagem é a inexistência no REST de **controle transacional**.

Professor, o que é controle transacional? 🤔

Boa pergunta! Controle transacional é quando você consegue estabelecer uma comunicação que obedeça a uma transação. Uma transação é um conjunto de operações que devem obrigatoriamente ser executadas em conjunto. Ou executa todas, ou não executa nenhuma.

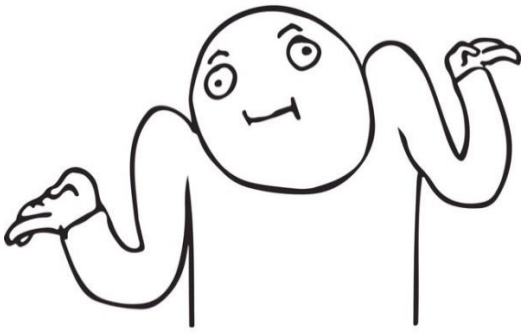
Quer ver um exemplo interessante de transação? Transferência de dinheiro entre bancos.

Esse tipo de transferência só tem 1 forma de dar certo: por meio de **controle de transação**. Neste caso, a transação contém 2 operações: 1 débito na conta de origem e 1 crédito na conta de destino. Se você vai transferir 100,00 da sua conta para a conta do seu amigo, primeiramente é feito um débito de 100,00 na sua conta e depois um crédito no mesmo valor na conta do seu amigo. Essas 2 operações (débito e crédito) têm necessariamente que fazer parte da **mesma transação**. Caso contrário, pode acontecer de o banco fazer o débito dos 100,00 da sua conta, após isso ocorrer um **erro** e o crédito dos 100,00 na conta de destino nunca ser realizado. Ou seja, o dinheiro sai da origem, mas não chega ao destino.

Com o **controle de transação** isso nunca ocorreria, porque, na ocorrência do **erro**, a transação desfaz a operação de débito e fica tudo como estava antes. Posteriormente, pode ser feita nova tentativa de transferência do valor, mas sempre dentro do controle de transação!

Voltando ao nosso assunto, que é web service **REST**, uma das desvantagens dele é o fato de ***não*** possuir uma forma confiável de fazer um **controle de transação**. Isso ocorre porque o REST é ***stateless***, isto é, não armazena informações de estado entre uma mensagem e outra. O REST é ***stateless*** de propósito. O fato de ser ***stateless*** dá a ele muita agilidade e baixo *overhead* de comunicação. Mas a desvantagem é a inexistência de um controle de transação confiável.

Não se pode ter tudo na vida.



A forma de trabalhar no REST é por meio de uma série de **métodos HTTP** que permitem criação, atualização, deleção e consulta de dados. Veja abaixo um tabela-resumo desses **métodos** (ou **operações**).

Métodos HTTP

Método	Descrição	Safe?	Idempotente?
GET	Recupera uma entidade em resposta à solicitação de um recurso.	Sim	Sim
POST	Permite a criação de novos recursos. Retorna uma entidade descrevendo o resultado da ação (novo recurso).	Não	Não
PUT	Atualiza uma entidade (cria a entidade caso ainda não exista).	Não	Sim
DELETE	Remove uma entidade.	Não	Sim
HEAD	Mecanismo usado por um cliente para verificar a existência de um recurso e, possivelmente, descobrir metadados sobre ele. Não recupera dados do recurso.	Sim	Sim
PATCH	Permite a realização de atualizações parciais.	Não	Não
OPTIONS	Verifica, no servidor, que outros verbos são aplicáveis a um recurso. Permite que desenvolvedores entendam como interagir com recursos.	Sim	Sim

Um ***safe method*** (**método seguro**) é aquele que **não** é capaz de alterar recursos. Portanto, ao utilizar **GET**, **HEAD** e **OPTIONS**, os recursos (dados) do serviço não são alterados. A vantagem dos ***safe methods*** é que podemos fazer **cache** dos seus recursos e retorná-los mais rapidamente no momento em que forem requisitados.

Um método **idempotente** é aquele que podemos chamar diversas vezes com a segurança de que todos os retornos serão iguais. Não importa se o método é chamado apenas 1 ou 10 vezes. O resultado deve ser o mesmo. Essencialmente, significa que o resultado de uma solicitação executada com sucesso é independente do número de vezes que ela é executada. No contexto dos serviços REST, os métodos **GET**, **PUT**, **DELETE**, **HEAD** e **OPTIONS** são **idempotentes**. Por outro lado, os métodos **POST** e **PATCH** **não** o são.

Você poderia se perguntar porque o método POST não é idempotente. Ora, a função do POST é justamente criar um recurso, um dado novo. Logo, a cada requisição POST, um novo recurso será criado, o que altera o estado da aplicação, fazendo com que seu retorno mude ao longo do tempo.

Resumindo o que vimos até agora sobre o padrão arquitetural REST, podemos compilar as seguintes **restrições** dessa tecnologia. Usamos aqui o termo "restrição" porque é o que é usado na literatura. Mas não entenda restrição como limitação. Veja essas restrições como **princípios orientadores** da tecnologia.

Restrições arquiteturais REST

Restrição	Descrição
Arquitetura cliente-servidor	O servidor é o web service. O cliente pode ser um aplicativo qualquer como um app de smartphone, um sistema Windows, Linux, Mac ou até mesmo outro web service.
Stateless	No lado do servidor, não armazena estado entre as requisições. Cada requisição do cliente contém toda informação necessária para seu atendimento por parte do serviço.
Cacheability	Possibilidade de fazer cache de alguns tipos de resposta de forma a retornar o resultado para o cliente de forma mais ágil.
Sistema em Camadas	Construção de diversos níveis de abstração incluindo, além do cliente e do serviço, camadas intermediárias como *proxy* e balanceadores de carga.
Código sob demanda	Servidores podem estender funcionalidades dos clientes ao transferir, além de dados, códigos executáveis.
Interface uniforme	Forma padronizada e unificada de os clientes acessarem os serviços (*uniform contract*).

Comparativo REST x SOAP

Depois de estudarmos os 2 modelos de implementação de web services, **REST** e **SOAP**, é interessante termos uma visão geral comparativa entre os 2 modelos. A tabela abaixo vai te ajudar a fixar os principais pontos.

REST	**SOAP**
+ simples	- simples
- *overhead* de comunicação	+ *overhead* de comunicação
Uniform contract	Cada serviço tem seu contrato WSDL específico
Stateless	*Stateful*
Sem controle de transação	Com controle de transação
- segurança	+ segurança

Questões comentadas pelo professor

1. (CESPE - 2019 - MPC-PA - Analista Ministerial – Tecnologia da Informação)

Quanto a web services, o REST

(a) possui uma única URL para identificar as diversas operações executadas por um web service.

(b) retorna os dados em somente um formato proprietário, chamado de RESTful.

(c) é uma alternativa ao padrão XML para representar informações.

(d) utiliza um formato baseado em execução de métodos para representar trocas de objetos em JavaScript.

(e) é um estilo arquitetônico para implementar web services.

RESOLUÇÃO:

(a) possui uma única URL para identificar as diversas operações executadas por um web service.

Errado. Acho que aqui o avaliador tentou confundir o concurseiro em relação ao conceito de *resource identifier*. De fato, com REST podemos ter diversas operações em 1 único *resource identifier*. Mas cada operação será em uma URL diferente.

Exemplo: URL1: http://www.meusite.com.br/clientes/consultaCPF/05988632640

URL2: http://www.meusite.com.br/clientes/consultaRG/345562

Observe que nosso *resource* clientes expôs 2 operações: consultaCPF e consultaRG. Entretanto, cada operação é executada por meio de uma URL diferente, e não em uma única URL, como foi afirmado nesta alternativa.

(b) retorna os dados em somente um formato proprietário, chamado de RESTful.

Errado. RESTful é a denominação de um web service que obedece à arquitetura REST. Serviços REST podem retornar dados em diversos formatos não-proprietários, como XML, JSON, YAML. Entretanto, o JSON é disparado o mais usado.

(c) é uma alternativa ao padrão XML para representar informações.

Errado. Seria correto dizer que o REST "é uma alternativa ao padrão SOAP para estruturar web services". Agora, dizer que é uma alternativa ao XML não faz muito sentido. Até porque, nada impede de um serviço REST transmitir mensagens XML, apesar da preferência pelo JSON.

(d) utiliza um formato baseado em execução de métodos para representar trocas de objetos em JavaScript.

A 1ª parte da afirmativa "utiliza um formato baseado em execução de métodos para representar trocas de objetos" está OK. O erro foi falar em JavaScript. Isso foi uma "casca de banana" porque o formato dos objetos é **JSON** cuja sigla significa **JavaScript Object Notation**. Apesar do JavaScript no nome, o JSON é somente uma notação, e não é obrigatório o uso da linguagem JavaScript para fazer um web service REST funcionar com JSON.

(e) é um estilo arquitetônico para implementar web services.

Certo.

Resposta: E

2. (IF-PE - 2019 - IF-PE - Técnico em Tecnologia da Informação - Desenvolvimento)

Os webservices REST (Transferência de Estado Representacional) são uma arquitetura de software para sistemas hypermedia. Os webservices estilo REST utilizam, para a realização de suas operações, os mesmos verbos do protocolo HTTP. As operações que simbolizam o CRUD (criar, ler, atualizar e deletar) são, respectivamente,

(a) POST, GET, PUT e DELETE.

(b) POST, HEAD, GET e DELETE.

(c) PUT, GET, HEAD e DELETE.

(d) PUT, TRACE, POST e DELETE.

(e) POST, GET, TRACE e DELETE.

RESOLUÇÃO:

CRUD é um jargão de TI para fazer referência a sistemas que fazem as operações básicas de manipulação de dados: **Create**, **Read**, **Update** e **Delete** (criar, ler, atualizar e deletar). No contexto de um banco de dados SQL, seria correspondente a INSERT, SELECT, UPDATE e DELETE.

No contexto do REST, também podemos fazer a seguinte correspondência entre as 4 operações CRUD e as operações HTTP.

Criar	POST
Ler	GET
Atualizar	PUT
Deletar	DELETE

Resposta: A

3. (CESPE - 2019 - TJ-AM - Assistente Judiciário - Programador)

```
1    import javax.jws.WebService;
2    import javax.jws.WebMethod;
3    import javax.jws.WebParam;
4
5    @WebService(
6     name = "WelcomeSOAP",
7     serviceName = "WelcomeSOAPService" )
8    public class WelcomeSOAP
9   {
10   @WebMethod( operationName = "welcome" )
11    public String welcome( @WebParam( name =
"name" ) String name )
12    {
13        return "Welcome to JAX-WS web services
with SOAP," + name + "!";
14     } \\ fim do método welcome
15     } \\ fim da classe WelcomeSOAP
```

Com relação a esse código do serviço web WelcomeSOAP, julgue o item que se segue.

O método welcome é marcado pela anotação @WebMethod para indicar que ele pode ser chamado remotamente; essa anotação usa o atributo operationName para especificar o nome do método que é exibido ao cliente do serviço web.

RESOLUÇÃO:

O código apresentado é uma forma interessante de implementar um web service em linguagem Java com o auxílio do **JAX-WS** (Java API for XML Web Services).

A vantagem da API JAX-WS é o uso intensivo de anotações (*annotations*), o que torna o código enxuto e intuitivo.

No código em questão, a anotação @WebService indica que a classe implementa um web service. Para este web service, foi implementado 1 método que pode ser chamado remotamente. Trata-se do método welcome da classe Java anotado com @WebMethod. O atributo operationName define que o nome público desse método para os clientes do web service será welcome também.

Tudo certo na afirmativa.

Resposta: Certo

4. (CESPE - 2019 - TJ-AM - Assistente Judiciário - Programador)

```
1    import javax.jws.WebService;
2    import javax.jws.WebMethod;
3    import javax.jws.WebParam;
4
5    @WebService(
6     name = "WelcomeSOAP",
7     serviceName = "WelcomeSOAPService" )
8    public class WelcomeSOAP
9   {
10   @WebMethod( operationName = "welcome" )
11    public String welcome( @WebParam( name =
"name" ) String name )
12    {
13        return "Welcome to JAX-WS web services
with SOAP," + name + "!";
14     } \\ fim do método welcome
15     } \\ fim da classe WelcomeSOAP
```

Com relação a esse código do serviço web WelcomeSOAP, julgue o item que se segue.

A anotação @WebService indica que a classe WelcomeSOAP implementa um serviço web; o atributo name especifica o nome da classe proxy que será gerada para o cliente, enquanto o atributo serviceName configura o nome de serviço.

RESOLUÇÃO:

Mesmo código da questão anterior. Uma implementação de serviço com JAX-WS.

O avaliador fez uma afirmativa certa. A anotação @WebService indica que a classe Java WelcomeSOAP implementa um web service. Ele diferenciou o atributo name = "WelcomeSOAP" que especifica o nome da classe proxy que será gerada para o cliente do atributo serviceName = "WelcomeSOAPService" que especifica o nome do serviço.

Resposta: Certo

5. (VUNESP - 2019 - Prefeitura de Valinhos - SP - Analista de Tecnologia da Informação – SAI)

Considere a seguinte chamada HTTP de um webservice para apagar o cadastro de um produto:

GET /produto/1234/apagar

Em relação à essa chamada, conclui-se que ela

(a) pode ser considerada RESTful, bastando apenas que o método seja alterado para DELETE.
(b) está incorreta, pois o URI não deve conter parâmetros da requisição, como o código do produto.
(c) somente estará correta se o corpo da requisição (request body) informar os dados do recurso a ser apagado.
(d) não funcionará, pois o método GET somente pode ser utilizado para obter dados e não para apagá-los.
(e) não corresponde a um webservice RESTful, pois promove alterações utilizando uma operação não padronizada sobre um recurso.

RESOLUÇÃO:

Esta é uma questão interessante. Observe a chamada que foi feita:

GET /produto/1234/apagar

Tem uma incongruência nessa chamada. O método HTTP **GET** é um método de **leitura**, isto é, obtenção de dados. Só que, pela URL, depreende-se que a operação desejada é apagar o produto cujo id = 1234. Apagar é uma operação de **escrita**, que não combina com o GET.

E aí, concurseiro?

A verdade é que essa chamada não necessariamente geraria um erro. O programador poderia implementar a operação apagar de forma que internamente fosse feito o DELETE. Nesse caso, a chamada funcionaria. **Entretanto**, lembre-se do seguinte: REST é um **padrão arquitetural**. Como padrão, ele dá orientações a respeito de como as coisas devem ser feitas. Usar um método HTTP de leitura como o GET para fazer uma deleção de recurso vai contra as recomendações do padrão.

A alternativa correta desta questão é a **E**. A chamada "não corresponde a um webservice RESTful, pois promove alterações utilizando uma operação não padronizada sobre um recurso."

Resposta: E

6. (IADES - 2019 - BRB - Analista de Tecnologia da Informação)

No contexto de microsserviços, trata-se de uma abstração da arquitetura da web. Resumidamente, consiste em princípios/regras/constraints que, quando seguidos, permitem a criação de um projeto com interfaces bem definidas, dessa forma, permitindo, por exemplo, que aplicações se comuniquem.

Disponível em: <https://becode.com.br/> . Acesso em: 8 ago. 2019, com adaptações.

Essa definição diz respeito a

(a) WSDL.

(b) SDK.

(c) JSON-RPC.

(d) XML-RPC.

(e) REST API.

RESOLUÇÃO:

Está falando do **REST API**. Muito mais que um protocolo, trata-se de um padrão arquitetural. Um conjunto de princípios/regras/constraints que permitem a criação de um projeto com interfaces bem definidas, dessa forma, permitindo, por exemplo, que aplicações se comuniquem.

O padrão REST, como bem falou este enunciado, é uma abstração da arquitetura web.

Resposta: E

7. (IDECAN - 2019 - IF-PB - Professor - Informática)

Sobre o estilo arquitetural REST (Representational State Transfer), é correto afirmar que

(a) a resposta das requisições é sempre feita no formato JSON, o que impossibilita a implementação de aplicativos móveis.

(b) faz uso de operações que mantêm informações sobre a sessão no lado recebedor, sendo assim chamada de *stateless*.

(c) algumas de suas restrições incluem arquitetura cliente-servidor, *statelessness*, sistema em camadas e interface uniforme.

(d) surgiu recentemente, no ano de 2010, como resultado dos esforços da Google.

(e) faz uso apenas dos verbos GET e POST, do HTTP..

RESOLUÇÃO:

(a) a resposta das requisições é sempre feita no formato JSON, o que impossibilita a implementação de aplicativos móveis.

Errado. Não é obrigatório o uso do formato JSON. O REST permite trabalhar com outros formatos como XML e YAML. O JSON é o mais usado e ele possibilita sim a implementação de aplicativos móveis. É muito comum termos apps móveis implementados como clientes de serviços web REST.

(b) faz uso de operações que mantêm informações sobre a sessão no lado recebedor, sendo assim chamada de *stateless*.

Errado. O REST é *stateless*, o que significa que ele **não** mantém informações sobre a sessão do lado do recebedor (web service).

(c) algumas de suas restrições incluem arquitetura cliente-servidor, *statelessness*, sistema em camadas e interface uniforme.

Certo!

(d) surgiu recentemente, no ano de 2010, como resultado dos esforços da Google.

Errado. O REST foi proposto em 2000 pelo pesquisador Roy Fielding, sem vínculo com o Google.

(e) faz uso apenas dos verbos GET e POST, do HTTP.

Errado. Além dos verbos (métodos) GET e POST, pode fazer uso também do PUT, DELETE, HEAD, PATCH e OPTIONS. Lembre-se da nossa tabela de resumo abaixo.

Método	Descrição	Safe?	Idempotente?
GET	Recupera uma entidade em resposta à solicitação de um recurso.	Sim	Sim
POST	Permite a criação de novos recursos. Retorna uma entidade descrevendo o resultado da ação (novo recurso).	Não	Não
PUT	Atualiza uma entidade (cria a entidade caso ainda não exista).	Não	Sim
DELETE	Remove uma entidade.	Não	Sim

HEAD	Mecanismo usado por um cliente para verificar a existência de um recurso e, possivelmente, descobrir metadados sobre ele. Não recupera dados do recurso.	Sim	Sim
PATCH	Permite a realização de atualizações parciais.	Não	Não
OPTIONS	Verifica, no servidor, que outros verbos são aplicáveis a um recurso. Permite que desenvolvedores entendam como interagir com recursos.	Sim	Sim

Resposta: C

8. (CESPE - 2019 - SLU-DF - Analista de Gestão de Resíduos Sólidos - Informática)

Acerca de REST e DHCP, julgue item que se segue.

Entre os princípios orientadores a serem seguidos na implantação de uma API RESTful Java inclui-se o *stateless*, em que cada solicitação do cliente para o servidor deve conter todas as informações necessárias, independentemente das informações armazenadas no servidor.

RESOLUÇÃO:

Certo. ***Stateless*** significa ausência de estado. O REST trabalha dessa forma. Não armazena estado do cliente entre as requisições. Não tem dados de sessão. Cada requisição ao servidor contém todas as informações necessárias para sua resposta.

Resposta: Certo

9. (CESPE - 2019 - SLU-DF - Analista de Gestão de Resíduos Sólidos - Informática)

Julgue o próximo item, a respeito de domain-driven design, design patterns, emergent design, enterprise content management e REST.

Em um web service REST que gerencie alguns tipos de serviço, os conflitos decorrentes de recursos que tenham identificadores iguais são automaticamente resolvidos no web service.

RESOLUÇÃO:

Errado. O web service REST não entra nesse mérito. Ele não tem como resolver conflitos decorrentes de recursos com identificadores iguais. Cada recurso tem que ter um identificador único.

Resposta: Errado

10. (CESPE - 2019 - SLU-DF - Analista de Gestão de Resíduos Sólidos - Informática)

Acerca de arquitetura de software, julgue o item a seguir.

Um web service pode assumir o papel de provedor de serviço e de consumidor de serviço.

RESOLUÇÃO:

Web service, como o próprio nome já diz, é um serviço. Portanto, a função precípua do web service é a de provedor de serviço.

E quem seria o consumidor do serviço, isto é, o cliente?

Qualquer sistema que seja capaz de se comunicar com o web service. Pode ser um app de smartphone, um robô na web, um sistema rodando em um Windows, Linux, Mac, etc.

O consumidor do serviço pode ser, inclusive, um outro web service!

Por isso se disse no enunciado desta questão que um web service pode assumir o papel tanto de provedor quanto de consumidor de serviço.

Resposta: Certo

11.(FCC - 2019 - SEFAZ-BA - Auditor Fiscal - Administração, Finanças e Controle Interno - Prova II)

Os web services são componentes de software na web que podem fornecer determinados serviços a aplicações criadas em diferentes linguagens. Podem usar o protocolo SOAP para transferência de mensagens em formato XML. Para descrever a estrutura destas mensagens geralmente utiliza-se

(a) REST.

(b) WSDL.

(c) CORBA.

(d) RESTFUL.

(e) HTML.

RESOLUÇÃO:

A estrutura das mensagens no protocolo SOAP é descrita segundo a notação **WSDL** (***Web Service Description Language***).

Resposta: B

12. (CCV-UFC - 2019 - UFC - Técnico de Tecnologia da Informação - Desenvolvimento de Sistemas)

Sobre Web services, assinale a alternativa correta.

(a) WSDL descreve os serviços disponibilizados à rede através de uma semântica JSON.

(b) XSLT é uma linguagem utilizada para transformar documentos JSON em documentos XML.

(c) As APIs de Web services que aderem às restrições de arquitetura REST são chamadas de APIs RESTful.

(d) SOAP é um protocolo para troca de informações estruturadas em uma plataforma descentralizada e distribuída baseadas em documentos JSON.

(e) *Representational State Transfer* (REST), é um estilo de arquitetura que define um conjunto de restrições e propriedades baseados em SOAP.

RESOLUÇÃO:

(a) WSDL descreve os serviços disponibilizados à rede através de uma semântica JSON.

Errado. WSDL descreve os serviços disponibilizados à rede através de uma semântica XML, e não JSON.

(b) XSLT é uma linguagem utilizada para transformar documentos JSON em documentos XML.

Errado. XSLT transforma XML em XML. Não envolve JSON.

(c) As APIs de Web services que aderem às restrições de arquitetura REST são chamadas de APIs RESTful.

Certo!

(d) SOAP é um protocolo para troca de informações estruturadas em uma plataforma descentralizada e distribuída baseadas em documentos JSON.

Errado. SOAP é baseado em documentos XML.

(e) *Representational State Transfer* (REST), é um estilo de arquitetura que define um conjunto de restrições e propriedades baseados em SOAP.

Errado. REST é baseado em JSON.

Resposta: C

13.(CESPE - 2019 - CGE - CE - Auditor de Controle Interno - Tecnologia da Informação)

Os web services evoluíram para integrar aplicativos e fornecer serviços, no entanto apresentam inúmeros riscos de ataques. O ataque que força a reinicialização do sistema, consumindo todos os recursos, é denominado

(a) XML injection.
(b) SQL injection.
(c) DoS attack
(d) cross-site scripting.
(e) invalid types.

RESOLUÇÃO:

Um web service deve estar acessível em uma rede poder ser consumido pelos seus clientes. A rede pode ser a própria internet ou uma intranet restrita.

Contudo, dentro da rede pode haver um sistema malicioso que acesse o web service com o intuito de atacar.

O tipo de ataque abordado nesta questão é o **DoS** (***Denial of Service***). Neste ataque, o código malicioso envia uma "avalanche" de requisições simultâneas ao serviço. Se o serviço não tiver inteligência para "brecar" essas requisições, ele vai tentar processar e responder a todas, o que pode ocasionar lentidão, travamento ou até mesmo a reinicialização do sistema.

Resposta: C

14. (COSEAC - 2019 - UFF - Técnico de Tecnologia da Informação)

Em relação aos web services, avalie se são verdadeiras (V) ou falsas (F) as afirmativas a seguir:

I Um web service tem uma estrutura interna fracamente acoplada.

II Um web service possui uma capacidade de criar soluções distribuídas e centralizadas.

III Um web service representa a materialização da ideia de uma serviço acessível em qualquer lugar do planeta.

As afirmativas I, II e III são, respectivamente:

(a) V, F e V.

(b) F, V e F.

(c) V, F e F.

(d) F, F e V.

(e) V, V e V.

RESOLUÇÃO:

I Um web service tem uma estrutura interna fracamente acoplada.

Verdadeiro. Essa é uma das vantagens da arquitetura do web service, o baixo acoplamento. Você pode trocar a estrutura interna do provedor de serviço e pode trocar o cliente. O importante é manter o contrato. E que todos respeitem esse contrato, tanto provedor de serviço quanto clientes.

II Um web service possui uma capacidade de criar soluções distribuídas e centralizadas.

Falso. Um web service possui uma capacidade de criar soluções distribuídas e **des**centralizadas

III Um web service representa a materialização da ideia de uma serviço acessível em qualquer lugar do planeta.

Gabarito oficial verdadeiro. A afirmativa está mal redigida, na minha opinião. O web service só será acessível em qualquer lugar do planeta se estiver público na internet. Se ficar restrito a uma intranet, isso não será possível.

Resposta: A

15.(UFSC - 2019 - UFSC - Técnico de Tecnologia da Informação)

Considere uma aplicação para Web do tipo webservice que implementa a abordagem RESTful. Assinale a alternativa que completa correta e respectivamente as seguintes frases.

O método _____ deve ser usado para alterar um dado no servidor. O método _____, por sua vez, deve ser usado para obter um dado do servidor. Finalmente, o método _____ deve ser usado para incluir um dado no servidor.

(a) PUT – PUT – GET

(b) POST – PUT – GET

(c) POST – GET – POST

(d) GET – POST – PUT

(e) PUT – GET – POST

RESOLUÇÃO:

O método **PUT** deve ser usado para alterar um dado no servidor. O método **GET**, por sua vez, deve ser usado para obter um dado do servidor. Finalmente, o método **POST** deve ser usado para incluir um dado no servidor.

Resposta: E

16. (CESPE - 2018 - STJ - Técnico Judiciário - Desenvolvimento de Sistemas)

Julgue o próximo item, relativo a model-view-controller (MVC), proxy reverso e representational state transfer (REST).

A REST define uma arquitetura cliente-servidor na qual o servidor não mantém contexto de cliente entre transações, ou seja, é *stateless* e toda transação contém as informações necessárias para satisfazer a solicitação.

RESOLUÇÃO:

Correto. Lembre-se sempre que REST é *stateless*, isto é, não armazena estado (contexto) de cliente.

Resposta: Certo

17.(FCC - 2018 - DPE-AM - Assistente Técnico de Defensoria - Programador)

De acordo com a arquitetura REST, um serviço Web RESTful

(a) deve manter um estado de cliente no servidor.

(b) não consegue tratar cada requisição de forma independente.

(c) suporta somente os métodos GET e POST.

(d) não funciona bem com os protocolos HTTP.

(e) não deve manter um estado de cliente no servidor.

RESOLUÇÃO:

(a) deve manter um estado de cliente no servidor.

Errado. Não deve, pois REST é *stateless*.

(b) não consegue tratar cada requisição de forma independente.

Errado. Consegue sim, porque ele é *stateless*.

(c) suporta somente os métodos GET e POST.

Errado. Além do GET e POST, suporta também PUT, DELETE, HEAD, PATCH e OPTIONS.

(d) não funciona bem com os protocolos HTTP.

Errado. Pelo contrário, o REST é todo baseado em HTTP.

(e) não deve manter um estado de cliente no servidor.

Certo! Não deve manter porque é *stateless*.

Resposta: E

18. (FCC - 2017 - TRF - 5ª REGIÃO - Analista Judiciário - Informática Desenvolvimento)

Se um serviço web baseado na arquitetura REST (RESTful) está localizado em http://www.trf5.jus.br/employee, quando o cliente fizer uma requisição a este serviço deverá

(a) usar a Web Services Description Language para descrever as regras de comunicação com o serviço.

(b) usar JAX-WS para sincronizar a comunicação com o serviço.

(c) estabelecer e manter essa conexão com o servidor até o final da troca de mensagens SOAP.

(d) usar um dos métodos HTTP como POST, GET, PUT ou DELETE.

(e) enviar um sinal beacon solicitando ao servidor uma porta para conexão.

RESOLUÇÃO:

(a) usar a Web Services Description Language para descrever as regras de comunicação com o serviço.

Errado. WSDL é para o modelo SOAP e não o REST.

(b) usar JAX-WS para sincronizar a comunicação com o serviço.

Errado. JAX-WS é uma API Java para implementação de web services no modelo SOAP e não REST.

(c) estabelecer e manter essa conexão com o servidor até o final da troca de mensagens SOAP.

Errado. O REST não é orientado a conexão. É orientado a métodos HTTP.

(d) usar um dos métodos HTTP como POST, GET, PUT ou DELETE.

Certo.

(e) enviar um sinal beacon solicitando ao servidor uma porta para conexão.

Errado. REST não é orientado a conexão.

Resposta: D

19. (PUC-PR - 2017 - TJ-MS - Técnico de Nível Superior - Analista de Sistemas)

Considerando os conceitos relacionados a Web services, analise as assertivas a seguir.

I. Web services são mecanismos utilizados na integração de sistemas.

II. WSDL é o protocolo utilizado durante uma requisição para transportar uma mensagem.

III. Interoperabilidade é a capacidade de um serviço ser descoberto facilmente.

IV. UDDI é um serviço de diretório para registrar e pesquisar serviços web.

V. XML é uma linguagem de marcação para troca de informações.

Está(ão) CORRETA(S) apenas a(s) assertiva(s):

(a) I, II, IV e V.

(b) III.

(c) I, IV e V.

(d) IV.

(e) I e V.

RESOLUÇÃO:

I. Web services são mecanismos utilizados na integração de sistemas.

Gabarito oficial verdadeiro. Não gostei do termo "mecanismo". Os web services são mais que um mecanismo. São um padrão arquitetural.

II. WSDL é o protocolo utilizado durante uma requisição para transportar uma mensagem.

Falso. WSDL não é protocolo. É linguagem de descrição de web service (Web Service Description Language).

III. Interoperabilidade é a capacidade de um serviço ser descoberto facilmente.

Falso. Interoperabilidade é a capacidade de um serviço de comunicar, se integrar com outros.

IV. UDDI é um serviço de diretório para registrar e pesquisar serviços web.

Verdadeiro.

V. XML é uma linguagem de marcação para troca de informações.

Verdadeiro.

Portanto, I, IV e V verdadeiras.

Resposta: C

20. (CESPE - 2017 - TRT - 7ª Região (CE) - Analista Judiciário - Tecnologia da Informação)

Assinale a opção que apresenta o método HTTP que deve ser usado para a busca de recursos por meio do web service RESTful.

(a) delete

(b) get

(c) put

(d) options

RESOLUÇÃO:

Método HTTP para busca de recursos é **GET**.

Resposta: B

21. (Gestão Concurso - 2018 - EMATER-MG - Analista de Sistemas I)

O serviço Web que define os componentes de uma especificação de serviço que podem ser usados para descobrir a existência de um serviço é conhecido por

(a) XML

(b) XSLT

(c) UDDI

(d) WSDL

RESOLUÇÃO:

O Serviço Web usado para descobrir a existência de um serviço na rede é o **UDDI** (***Universal Description Discovery and Integration***).

Em uma rede, podem estar disponíveis diversos webservices. Mas a forma de os potenciais clientes dos webservices descobrirem que eles existem é por meio do UDDI. Os clientes fazem uma requisição para o serviço UDDI e este responde com todos os webservices registrados.

Resposta: C

22. (FCC - 2018 - Prefeitura de São Luís - MA - Auditor Fiscal de Tributos I - Tecnologia da Informação (TI))

Web services são projetados para suportar interações interoperáveis entre máquinas em uma rede. Esta interoperabilidade é obtida por meio de um conjunto de padrões de tecnologia que, acoplados aos princípios de projeto orientado a serviço, formam um SOA fundamentado na tecnologia XML com base em 3 especificações.

São elas:

(a) Interface Message; Service e Binding.

(b) Web Service Description (WSD); Simple Object Access Protocol (SOAP) e Hypertext Transfer Protocol (HTTP).

(c) ServiceKey; BusinessEntity e BusinessKey.

(d) Web Services Definition Language (WSDL); Simple Object Access Protocol (SOAP) e Universal Description, Discovery, and Integration (UDDI).

(e) Web Services Definition Language (WSDL); Common Object Request Broker Architecture (CORBA) e Distributed Component Object Model (DCOM).

RESOLUÇÃO:

O tripé de tecnologias que viabilizam o funcionamento de uma arquitetura orientada a serviços com webservices é o seguinte:

- **WSDL** (**Web Services Definition Language**): especificação para descrição de um Web Service por meio de linguagem XML.
- **SOAP** (**Simple Object Access Protocol**): especificação protocolo de comunicação entre os servidores (Webservices) e seus respectivos clientes.
- **UDDI** (**Universal Description, Discovery, and Integration**): serviço que permite que os Web Services se registrem de forma a serem descobertos pelos potenciais clientes dentro de uma rede.

A alternativa que cita corretamente as 3 tecnologias é a **D**.

PS: O significado oficial da sigla WSDL é *Web Services* ***Description*** *Language* e não *Web Services* ***Definition*** *Language*. Vide site oficial da tecnologia, o W3c clicando aqui. Entretanto, os termos Description e Definition são muito próximos, o que não prejudica a interpretação da questão.

Resposta: D

23.(FGV - 2018 - Câmara de Salvador - BA - Analista de Tecnologia da Informação)

Usualmente, WebServices envolvem a utilização dos padrões XML, SOAP e WSDL.

A função de cada um deles é, respectivamente:

(a) descrever os parâmetros, disponibilizar o serviço, transferir as mensagens;

(b) transferir as mensagens, descrever o algoritmo, transportar as mensagens;

(c) rotular e formatar os dados, transferir as mensagens, descrever a disponibilidade do serviço;

(d) transferir as mensagens, descrever a disponibilidade do serviço, formatar os parâmetros;

(e) descrever as classes e suas interfaces, instanciar os objetos, descrever os algoritmos.

RESOLUÇÃO:

A correspondência correta é a seguinte:

Padrão	Função
XML	rotular e formatar os dados
SOAP	transferir as mensagens
WSDL	descrever a disponibilidade do serviço

Lembre-se: XML é uma linguagem de marcação. É ela que dá o formato dos dados.

O SOAP é o protocolo de troca de mensagens entre os Webservices e seus clientes.

O WSDL é a descrição do Webservice.

A alternativa que faz a correspondência correta é a letra C.

Resposta: C

24. (CESPE - 2018 - STJ - Técnico Judiciário - Suporte Técnico)

Julgue o item a seguir, acerca de arquiteturas de integração e web services.

Web service é uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes.

RESOLUÇÃO:

Correto. Web service é uma solução para comunicação entre máquina-máquina, ou seja, integração de sistemas.

Resposta: Certo

25.(CESPE - 2017 - TRE-BA - Técnico Judiciário – Programação de Sistemas)

No que se refere a web services, assinale a opção correta.

(a) As solicitações e respostas XML trafegam no protocolo HTTP, não sendo possível utilizá-las nos protocolos FTP e SMTP.

(b) Um dos componentes de um web service SOAP (simple object access protocol) é a UDDI (universal description, discovery and integration), a qual é um arquivo do tipo XML que descreve detalhadamente um web service, especificando como deve ser o formato de entrada e saída de cada operação.

(c) As duas formas de envio de mensagem para que um cliente possa efetuar solicitações a um web service são one-way messaging e request-response messaging.

(d) O WSDL (web services description language) é uma linguagem para o desenvolvimento de web services similar ao XML.

(e) Os web services permitem a integração de sistemas, todavia dependem da linguagem de programação sob a qual tenham sido desenvolvidos e do sistema operacional do computador onde esses sistemas forem executados.

RESOLUÇÃO:

(a) As solicitações e respostas XML trafegam no protocolo HTTP, não sendo possível utilizá-las nos protocolos FTP e SMTP.

Errado. Na maior parte das vezes as mensagens XML trafegam sob protocolo HTTP, mas não há nenhum impedimento para utilização de outros protocolos como FTP e SMTP.

(b) Um dos componentes de um web service SOAP (simple object access protocol) é a UDDI (universal description, discovery and integration), a qual é um arquivo do tipo XML que descreve detalhadamente um web service, especificando como deve ser o formato de entrada e saída de cada operação.

Errado. Essa descrição de "um arquivo do tipo XML que descreve detalhadamente um web service, especificando como deve ser o formato de entrada e saída de cada operação" se refere ao **WSDL**.

(c) As duas formas de envio de mensagem para que um cliente possa efetuar solicitações a um web service são one-way messaging e request-response messaging.

Correto. ***One-way messaging*** são mensagens unidirecionais. E o ***request-response*** é um protocolo em que o cliente faz uma requisição e o servidor responde.

(d) O WSDL (web services description language) é uma linguagem para o desenvolvimento de web services similar ao XML.

O **WSDL** não é uma linguagem. É uma especificação de um formato para descrever detalhadamente um webservice por meio da linguagem XML.

(e) Os web services permitem a integração de sistemas, todavia dependem da linguagem de programação sob a qual tenham sido desenvolvidos e do sistema operacional do computador onde esses sistemas forem executados.

Errado. O webservice é agnóstico em relação à linguagem de programação do sistema operacional. As especificações necessárias para um webservice funcionar (WSDL, SOAP, UDDI) são padrões abertos que independem das tecnologias subjacentes.

Resposta: C

26. (FAPEC - 2018 - UFMS - Analista de Tecnologia da Informação)

Considere as afirmações a seguir:

I - O Web Application Description Language (WADL) é um XML utilizado para descrever serviços RESTFul.

II - Web services baseados sobre a arquitetura REST são conhecidos como RESTFul Web services.

III - RESTFul Web services utilizam o HTTP para transportar o dado e o JSON para representar o dado.

Está(ão) correta(s):

(a) Apenas I.

(b) Apenas II.

(c) Apenas III.

(d) Apenas I e II.

(e) I, II e III.

RESOLUÇÃO:

Todas as afirmativas são verdadeiras.

Resposta: E

27.(FCM - 2018 - IFN-MG - Professor - Informática)

Sobre as afirmativas abaixo relacionadas aos conceitos de serviços Web

Considere os acrônimos:

•REST - Representational State Transfer;

•SOAP - Simple Object Access Protocol .

É correrto afirmar que o

(a) REST é uma arquitetura de rede, baseada no protocolo XML, que permite o serviço Web e o cliente se comunicar.

(b) SOAP é uma arquitetura de rede, baseada em mecanismos de solicitação e resposta tradicionais da Web, como solicitações GET e POST

(c) SOAP envia solicitação e resposta de seus serviços empacotados em envelopes, diferentemente do que ocorre nos serviços baseados em REST.

(d) SOAP é um protocolo independente de plataforma que faz chamadas de procedimentos remotos por meio de conexões FTP com poucas limitações.

(e) REST e o SOAP são um tipo de computação distribuída que permite a um aplicativo clientes ter acesso direto à memória principal de uma aplicação servidora.

RESOLUÇÃO:

Vamos analisar as alternativas individualmente.

(a) REST é uma arquitetura de rede, baseada no protocolo XML, que permite o serviço Web e o cliente se comunicar.

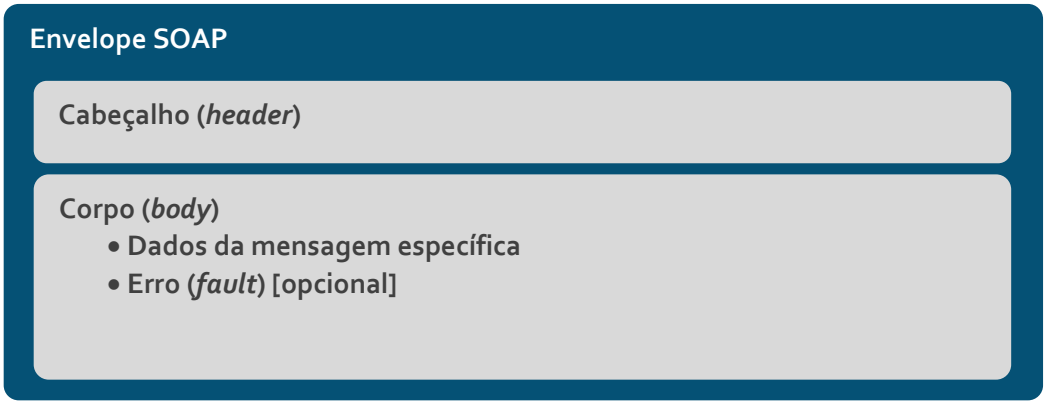
Errado. XML não é protocolo. É uma linguagem de marcação. XML = eXtensible Markup Language.

(b) SOAP é uma arquitetura de rede, baseada em mecanismos de solicitação e resposta tradicionais da Web, como solicitações GET e POST

Errado. O REST que é baseado em métodos HTTP tradicionais GET e POST. O SOAP se comunica por meio do RPC (Remote Procedure Call).

(c) SOAP envia solicitação e resposta de seus serviços empacotados em envelopes, diferentemente do que ocorre nos serviços baseados em REST.

Certo! O envelope SOAP é estruturado conforme esquema abaixo e é implementado no padrão WSDL.



(d) SOAP é um protocolo independente de plataforma que faz chamadas de procedimentos remotos por meio de conexões FTP com poucas limitações.

Errado. As chamadas de procedimentos remotos no SOAP ocorrem segundo o RPC (Remote Procedure Call) e não FTP (File Transfer Protocol).

(e) REST e o SOAP são um tipo de computação distribuída que permite a um aplicativo cliente ter acesso direto à memória principal de uma aplicação servidora.

Errado. REST e SOAP não acessam diretamente a memória da aplicação servidora. Na verdade, o que acessam é a interface do contrato do provedor do serviço.

Resposta: C

28. (INSTITUTO AOCP - 2018 - PRODEB - Analista de TIC II - Construção de Software)

Um serviço REST usualmente suporta mais de um formato para representação de seus recursos, sendo esta uma de suas características principais, já que facilita a inclusão de novos clientes e a interoperabilidade entre os projetos. Assinale a alternativa que apresenta somente formatos utilizados por um serviço REST.

(a) JAVA e JAVASCRIPT.

(b) JAVA, RUBY ON REALS e .NET.

(c) YAML e JSON.

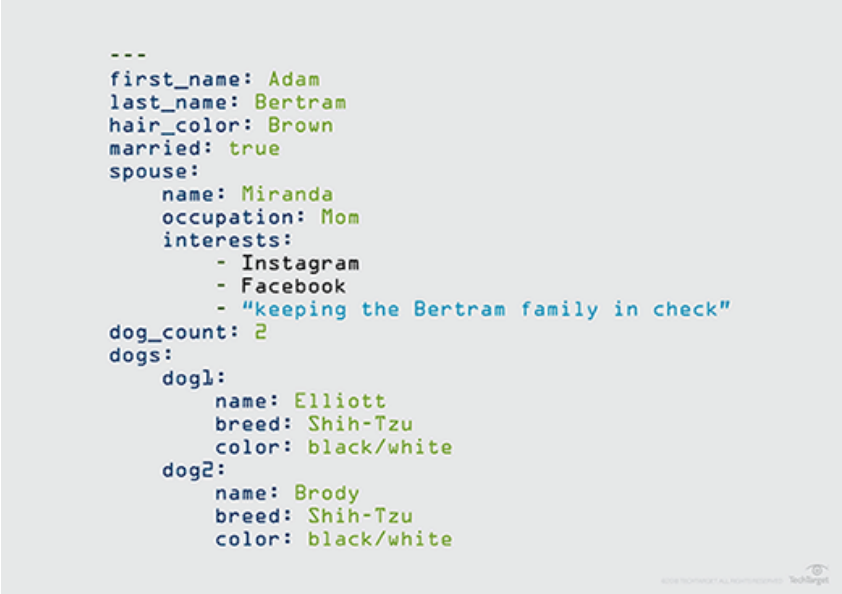
(d) XML e C++.

(e) JSON, CSS e SQL.

RESOLUÇÃO:

Serviços REST podem trabalhar com os formatos YAML e REST. Gabarito alternativa C. REST também permite trabalhar com o formato XML.

Pra quem não conhece, o formato YAML tem a seguinte aparência.



```yaml
---
first_name: Adam
last_name: Bertram
hair_color: Brown
married: true
spouse:
    name: Miranda
    occupation: Mom
    interests:
        - Instagram
        - Facebook
        - "keeping the Bertram family in check"
dog_count: 2
dogs:
    dog1:
        name: Elliott
        breed: Shih-Tzu
        color: black/white
    dog2:
        name: Brody
        breed: Shih-Tzu
        color: black/white
```

Java, JavaScript, Ruby, C++ são linguagens de programação.

SQL é linguagem de consulta.

Resposta: C

29. (CESGRANRIO - 2018 - LIQUIGÁS - Profissional Júnior - Analista de Sistemas)

Os Web Services são muito úteis para o desenvolvimento de sistemas de informação em empresas, pois

(a) utilizam bancos de dados "No SQL".

(b) possuem capacidade de aprendizado.

(c) suportam a interação entre sistemas de informação (interoperabilidade).

(d) viabilizam a interação entre os usuários de um sistemas de informação (web chat).

(e) permitem a interação de sistemas de informação com seus usuários, pois utilizam HTML como sua principal linguagem.

RESOLUÇÃO:

(a) utilizam bancos de dados "No SQL".

Errado. O web service é um contrato, uma interface de interoperabilidade entre sistemas. Pode até ser que, internamente, o serviço consulte um banco NoSQL, mas isso não faria parte do web service nem seria uma característica mandatória deles.

(b) possuem capacidade de aprendizado.

Não necessariamente. Pode ser que um método do web service internamente implemente um código de *machine learning*, por exemplo, o que daria capacidade de aprendizado ao serviço. Mas, assim como na letra A, isso não é uma característica mandatória dos web services.

(c) suportam a interação entre sistemas de informação (interoperabilidade).

Bingo! É isso aí. O web service serve para isso: integração de sistemas.

(d) viabilizam a interação entre os usuários de um sistemas de informação (web chat).

Errado. Os web services viabilizam a interação entre sistemas.

(e) permitem a interação de sistemas de informação com seus usuários, pois utilizam HTML como sua principal linguagem.

Errado. Os web services permitem a interação entre sistemas e outros sistemas, e não com usuários. Tampouco eles não costumam usar HTML como linguagem.

Resposta: C

30. (CCV-UFC - 2018 - UFC - Analista de Tecnologia da Informação)

Sobre REST e RESTful, o que é correto afirmar?

(a) REST significa Representational State Traffic.
(b) REST é um protocolo de transferência de dados.
(c) JAX-RS é a API Java para RESTful Web Services.
(d) WSDL é o padrão que deve ser utilizado com REST.
(e) REST permite a utilização de apenas texto e JSON como formato de dados.

RESOLUÇÃO:

(a) REST significa Representational State Traffic.
Errado. REST significa *Representational State Transfer*.

(b) REST é um protocolo de transferência de dados.
Errado. REST é um padrão arquitetural de sistema distribuído.

(c) JAX-RS é a API Java para RESTful Web Services.
Certo! Assim como temos a API Java JAX-WS para web services SOAP, temos também a JAX-RS para web services REST.

(d) WSDL é o padrão que deve ser utilizado com REST.
Errado. WSDL é o padrão que deve ser utilizado com SOAP.

(e) REST permite a utilização de apenas texto e JSON como formato de dados.
Errado. Além disso, REST permite diversos outros formatos como XML, YAML, CSV, etc.

Resposta: C

31. (CESPE - 2018 - STJ - Técnico Judiciário - Suporte Técnico)

Julgue o item a seguir, acerca de arquiteturas de integração e web services .

Os serviços Web RESTful utilizam o HTTP como um meio de comunicação entre cliente e servidor.

RESOLUÇÃO:

Perfeito. Serviços REST são todos baseados no HTTP. Aliás, o criador do REST (Roy Fielding) também participou da criação e evolução do HTTP. O REST faz uso dos métodos tradicionais do HTTP, como GET, PUT e POST, para funcionar.

Resposta: Certo

32. (CESPE - 2018 - CGM de João Pessoa - PB - Auditor Municipal de Controle Interno - Desenvolvimento de Sistemas)

Acerca de *service-oriented architecture*, web services, mensageria e CORBA (*common object request broker architecture*), julgue o item a seguir.

Web services permitem disponibilizar serviços de forma agnóstica quando a UDDI (*universal description, discovery and integration*) estabelece um formato padrão de mensagem que consiste em um documento XML capaz de hospedar dados RPC centrados em documentos, para que haja intercâmbio de dados de modelos síncronos (pedido e resposta) e assíncronos (orientados a processo).

RESOLUÇÃO:

Errado. Enunciado bem confuso este do Cespe.

Na verdade, o que estabelece o formato padrão de mensagem é o WSDL (*Web Service Description Language*).

UDDI é somente o mecanismo de diretório em que os serviços podem se registrar para posteriormente serem achados pelos clientes.

Resposta: Errado

33. (CESPE - 2017 - SEDF - Analista de Gestão Educacional - Tecnologia da Informação)

Em relação a web services, julgue o item seguinte.

Por oferecerem um framework de comunicação com base em contratos de serviços fisicamente desacoplados, os web services permitem que um contrato de serviços seja totalmente padronizado, independentemente de sua implementação.

RESOLUÇÃO:

Perfeito. Essa é a ideia de web service. Uma forma de estabelecimento de contrato padronizado. Com isso, os sistemas podem se comunicar de forma padronizada, sem se preocupar com a implementação interna de cada um.

Resposta: Certo

34. (CESPE - 2016 - TCE-PA - Auditor de Controle Externo - Área Informática - Administrador de Banco de Dados)

Julgue o item subsequente, relativo a SOA, web services e servidor web.

Os web services devem ser projetados para ser utilizados independentemente de paradigmas de programação.

RESOLUÇÃO:

Exato. Definindo um contrato padronizado, os web services podem ser utilizados independente de paradigma de programação do cliente e do provedor do serviço.

Resposta: Certo

35.(CESPE - 2016 - TCE-PA - Auditor de Controle Externo - Área Informática - Analista de Sistema)

Web service é um software que, armazenado em um computador, pode ser acessado por outro software em outro computador por meio de uma rede. A partir dessa afirmação, julgue o item subsequente.

Para que um web service funcione corretamente, os softwares cliente/servidor devem ser escritos na mesma linguagem.

RESOLUÇÃO:

Errado. A ideia do web service é justamente o contrário. A arquitetura do web service garante o desacoplamento entre o cliente e o servidor de forma que não há nenhuma necessidade de eles serem escritos na mesma linguagem.

Resposta: Errado

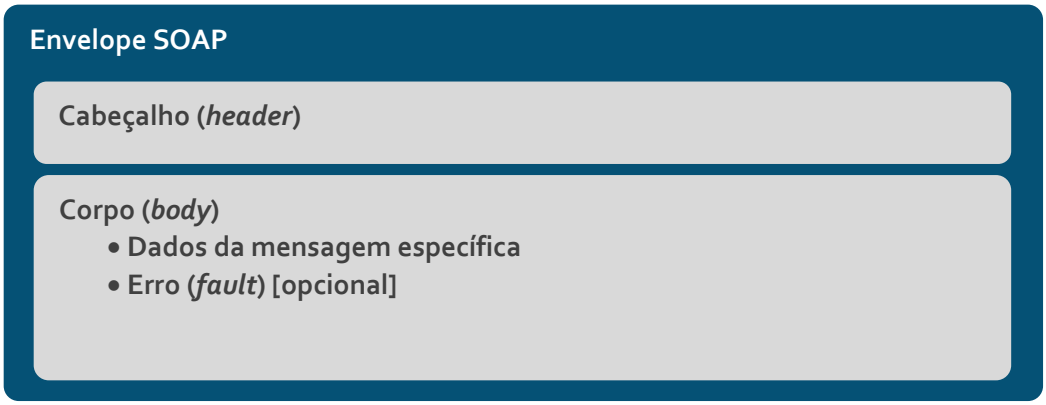
36. (CESPE - 2016 - TCE-PA - Auditor de Controle Externo - Área Informática - Analista de Sistema)

Web service é um software que, armazenado em um computador, pode ser acessado por outro software em outro computador por meio de uma rede. A partir dessa afirmação, julgue o item subsequente.

Ao se usar o protocolo SOAP (simple object access protocol), cada solicitação e cada resposta são colocadas em um envelope SOAP, nos momentos de invocação e retorno de um web service, respectivamente.

RESOLUÇÃO:

Exato. As mensagens SOAP são sempre envelopadas segundo o padrão da tecnologia. Abaixo, o esquema da mensagem SOAP para memorizar.



Resposta: Certo

37. (CESPE - 2016 - TRT - 8ª Região (PA e AP) - Técnico Judiciário - Tecnologia da Informação)

A interface de um webservice pode mudar ao longo do tempo sem comprometer a habilidade de interação do cliente com o serviço. A essa caraterística dá-se o nome de

(a) encapsulamento.

(b) acoplamento fraco.

(c) associação.

(d) publicação.

(e) manutenção de entidades.

RESOLUÇÃO:

Está falando de uma das grandes vantagens do web service, que é o **acoplamento fraco**, isto é, a possibilidade de mudar a interface sem comprometer a habilidade de interação do cliente com o serviço.

Resposta: B

38. (CESPE - 2016 - TRT - 8ª Região (PA e AP) - Técnico Judiciário - Tecnologia da Informação)

A respeito dos conceitos de web services e REST, assinale a opção correta.

(a) O método POST é utilizado na atualização de um recurso existente.

(b) Pode-se utilizar qualquer meio de transporte existente para o envio de uma requisição, incluindo HTTP, SMTP e TCP.

(c) O modelo REST impõe restrições ao formato da mensagem.

(d) Ao desenvolver uma aplicação, o recurso é transferido pela rede.

(e) As chamadas às URIs (uniform resource indicator) são feitas por meio de métodos HTTP, os quais indicam para o serviço a ação a ser realizada com o recurso.

RESOLUÇÃO:

(a) O método POST é utilizado na atualização de um recurso existente.

Errado. POST é pra criar um novo recurso. Para atualizar um recurso existente, usa-se PUT.

(b) Pode-se utilizar qualquer meio de transporte existente para o envio de uma requisição, incluindo HTTP, SMTP e TCP.

Errado. REST trabalha com o protocolo HTTP. SMTP é o protocolo de e-mail e TCP, juntamente com o IP, são os protocolos-base da internet.

(c) O modelo REST impõe restrições ao formato da mensagem.

Errado. REST é bem flexível quanto ao formato da mensagem. Pode-se usar JSON, XML, YAML, CSV, etc.

(d) Ao desenvolver uma aplicação, o recurso é transferido pela rede.

Errado. O que é transferido pelo rede é o dado. O recurso, isto é, a fonte dos dados, permanece no provedor do serviço.

(e) As chamadas às URIs (uniform resource indicator) são feitas por meio de métodos HTTP, os quais indicam para o serviço a ação a ser realizada com o recurso.

Certo! Exemplo de chamada a URI por meio de método HTTP:

GET http://www.direcaoconcursos.com.br/meuscursos/ds4327821

Resposta: E

39. (CESPE - 2016 - TRT - 8ª Região (PA e AP) - Técnico Judiciário - Tecnologia da Informação)

Acerca da arquitetura de microsserviços, assinale a opção correta.

(a) A arquitetura de microsserviços é um padrão para a criação de aplicações distribuídas, porém não possui alta escalabilidade.

(b) A comunicação entre os microsserviços é feita por meio de mecanismos padrões de tecnologia, como, por exemplo, o REST (representational state transfer).

(c) Microsserviços utilizam uma única base de dados lógica para a persistência de dados.

(d) Um requisito fundamental da arquitetura de microsserviços é o uso de versionamento de mudanças.

(e) Os microsserviços são componentes autônomos e de alto acoplamento, de modo que há a necessidade de se utilizar uma mesma linguagem na sua construção.

RESOLUÇÃO:

(a) A arquitetura de microsserviços é um padrão para a criação de aplicações distribuídas, porém não possui alta escalabilidade.

Errado. A arquitetura de microsserviços possui sim alta escalabilidade. Esse é um dos objetivos. Alta escalabilidade é a capacidade de um sistema escalar, isto é, manter a boa performance mesmo quando o número de requisições aumenta muito.

(b) A comunicação entre os microsserviços é feita por meio de mecanismos padrões de tecnologia, como, por exemplo, o REST (representational state transfer).

Certo!

(c) Microsserviços utilizam uma única base de dados lógica para a persistência de dados.

Errado. Não existe essa limitação. Podem ser utilizadas diversas bases de dados.

(d) Um requisito fundamental da arquitetura de microsserviços é o uso de versionamento de mudanças.

Errado. Esse não é um requisito da arquitetura de microsserviços.

(e) Os microsserviços são componentes autônomos e de alto acoplamento, de modo que há a necessidade de se utilizar uma mesma linguagem na sua construção.

Errado. São componentes de **baixo** acoplamento. Não há tampouco a necessidade de usar uma mesma linguagem na sua construção.

Resposta: B

40. (CESPE - 2016 - TRE-PI - Analista Judiciário - Análise de Sistemas)

Acerca de REST (representational state transfer), assinale a opção correta.

(a) Na implementação do REST, todos os recursos devem responder a todos os métodos.

(b) O método GET permite obter e alterar o estado atual de um recurso.

(c) O método EXPUNGE permite excluir um recurso.

(d) A arquitetura de comunicação entre aplicações baseia-se em um modelo rígido de recursos e localizações.

(e) O método MODIFY permite alterar um recurso.

RESOLUÇÃO:

(a) Na implementação do REST, todos os recursos devem responder a todos os métodos.

Errado. Não existe essa obrigatoriedade. Exemplo, pode haver um recurso que responda somente ao método GET. Outro pode responder somente a PUT e POST.

(b) O método GET permite obter e alterar o estado atual de um recurso.

Errado. GET é somente para obtenção de dados. Nunca alteração.

(c) O método EXPUNGE permite excluir um recurso.

Errado. Não existe esse método.

(d) A arquitetura de comunicação entre aplicações baseia-se em um modelo rígido de recursos e localizações.

Gabarito oficial certo. Eu particularmente não gostei do termo "rígido". Talvez o Cespe tenha usado esse termo para destacar que é necessário respeitar o modelo de localizações de recursos via URI. Exemplo:

GET http://www.meusite.com.br/meurecurso/id/123

(e) O método MODIFY permite alterar um recurso.

Errado. Não existe esse método.

Só um adendo. As alternativas C e E mencionaram métodos HTTP que não existem. Memorize nossa tabela de métodos HTTP para você não ter dúvida nesse tipo de questão.

Método	Descrição	Safe?	Idempotente?
GET	Recupera uma entidade em resposta à solicitação de um recurso.	Sim	Sim
POST	Permite a criação de novos recursos. Retorna uma entidade descrevendo o resultado da ação (novo recurso).	Não	Não
PUT	Atualiza uma entidade (cria a entidade caso ainda não exista).	Não	Sim
DELETE	Remove uma entidade.	Não	Sim
HEAD	Mecanismo usado por um cliente para verificar a existência de um recurso e, possivelmente, descobrir metadados sobre ele. Não recupera dados do recurso.	Sim	Sim
PATCH	Permite a realização de atualizações parciais.	Não	Não
OPTIONS	Verifica, no servidor, que outros verbos são aplicáveis a um recurso. Permite que desenvolvedores entendam como interagir com recursos.	Sim	Sim

Resposta: D

41. (CESPE - 2015 - MEC - Analista de Sistemas)

Acerca da utilização dos protocolos SOAP e REST, julgue o item seguinte.

Os métodos PUT e DELETE no protocolo REST não são considerados como idempotentes.

RESOLUÇÃO:

Errado. PUT e DELETE são ambos idempotentes.

Um método **idempotente** é aquele que podemos chamar diversas vezes com a segurança de que todos os retornos serão iguais. Não importa se o método é chamado apenas 1 ou 10 vezes. O resultado deve ser o mesmo.

O PUT é utilizado para atualizar recursos. Vamos supor que o web service possua um recurso Pessoa cujo nome é João. Se você chamar o PUT 1000 vezes pedindo para alterar o nome para Lucas, o resultado vai ser sempre o mesmo. O nome será alterado para Lucas.

O DELETE é para deletar um recurso do web service. Se o web service tem um serviço de cadastro de produtos e você chamada o DELETE 1000 vezes para deletar o produto cujo id é 123, esse produto será deletado na 1ª chamada e, a partir daí, nas 999 chamadas seguintes o recurso permanecerá íntegro, sem deletar mais nada. Ele não vai deletar outros produtos além daquele cujo id é 123.

Método	Idempotente?
GET	Sim
POST	Não
PUT	Sim
DELETE	Sim
HEAD	Sim
PATCH	Não
OPTIONS	Sim

Resposta: Errado

42. (CESPE - 2015 - MEC - Arquiteto de Sistemas)

Com relação a tecnologias Web, julgue o item a seguir.

Em uma web service, a linguagem de implementação e a plataforma utilizada são relevantes para os clientes.

RESOLUÇÃO:

Errado. É o contrário. Em um web service, a linguagem de implementação e a plataforma utilizada ***não*** são relevantes para os clientes. Para os clientes, basta respeitar o contrato estabelecido pelo serviço.

Resposta: Errado

43. (CESPE - 2015 - MEC - Arquiteto de Sistemas)

Com relação a tecnologias Web, julgue o item a seguir.

As principais características do REST (representationl state transfer) são interface uniforme, stateless e cache.

RESOLUÇÃO:

Certo. A interface uniforme é o conceito de *uniform contract* que estudamos. Com *uniform contract* + *resource identifier* (URI que identifica o recurso) configuramos um serviço REST.

Adicionalmente, outras características do REST são o fato de ser *stateless* (sem estado) e a possibilidade de fazer cache para alguns tipos de requisições de leitura, como por exemplo o método HTTP GET.

Resposta: Certo

44. (CESPE - 2015 - MEC - Gerente de Projetos)

No que se refere a padrões SOA (service-oriented architecture) e a REST (representational state transfer), julgue o item subsecutivo.

Entre as restrições da REST está a interface uniforme, a qual requer que um serviço ofereça várias operações e aguarde a solicitação dessas operações pelo servidor.

RESOLUÇÃO:

A 1ª parte da afirmativa está ok. Uma das restrições do REST, de fato, é a interface uniforme. O problema está no final. O correto seria:

"Entre as restrições da REST está a interface uniforme, a qual requer que um serviço ofereça várias operações e aguarde a solicitação dessas operações pelo cliente. "

Resposta: Errado

45. (CESPE - 2015 - TJ-DFT - Analista Judiciário - Analista de Sistemas)

A Arquitetura Orientada a Serviços (SOA), no cenário dos modelos arquiteturais modernos, enfatiza o reúso como elemento chave para a maximização dos resultados em tecnologia da informação. A catalogação e a gerência dos ativos de software da organização na condição de serviços de aplicativos são os pilares em meio aos quais devem se sustentar essa nova arquitetura. A esse respeito, julgue o próximo item.

Em um Web Service RESTful, cada método é identificado por uma URL única. Assim, quando o servidor recebe uma solicitação, ele identifica de forma inequívoca a operação que será executada.

RESOLUÇÃO:

Exatamente. A URL única é o *resource identifier*. Com ele, o servidor consegue identificar de forma unívoca, isto é, sem dúvida, a operação que deve ser executada.

Resposta: Certo

46. (CESPE - 2015 - TCE-RN - Inspetor - Tecnologia da Informação - Cargo 5)

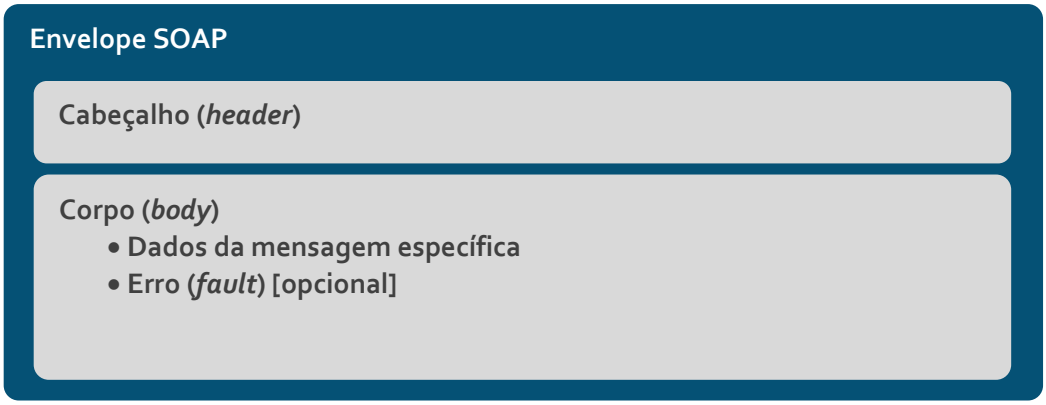
A respeito de arquitetura para desenvolvimento de sistemas, de programação orientada a aspectos (POA) e de banco de dados distribuídos, julgue o item subsecutivo.

Em web service, o objetivo do cabeçalho do envelope SOAP (SOAP header block) é o de fazer cumprir as regras que governam a troca de mensagens SOAP, por meio de dados trocados em formato XML.

RESOLUÇÃO:

Na verdade, no REST, o que garante o cumprimento das regras que governam a troca de mensagens é o próprio SOAP (*Simple Object Access* ***Protocol***). Grifei a palavra *protocol* para evidenciar que a função de governar a troca de mensagens é do próprio protocolo.

O **cabeçalho** (header) possui informações de controle e processamento. Ele é opcional e carrega informações adicionais como, por exemplo, se a mensagem deve ser processada por um determinado nó intermediário.



Resposta: Errado

47. (CESPE - 2015 - TRE-MT - Analista Judiciário - Análise de Sistemas)

Acerca de REST (representational state transfer), assinale a opção correta.

(a) O protocolo REST utiliza SOAP e XML.

(b) REST utiliza recurso não identificável baseado em PUT e GET.

(c) REST pode ser utilizado para implementar WebServices de baixo overhead.

(d) Embora opcionalmente, um recurso REST pode conter uma URI.

(e) REST consiste em um estilo de desenvolvimento baseado em complexa interação cliente/servidor.

RESOLUÇÃO:

(a) O protocolo REST utiliza SOAP e XML.

Errado. REST não utiliza SOAP. Ele é uma alternativa ao SOAP.

(b) REST utiliza recurso não identificável baseado em PUT e GET.

Errado. REST utiliza recurso identificável baseado em URI (ou URL).

(c) REST pode ser utilizado para implementar WebServices de baixo overhead.

Certo! Uma das vantagens do REST é o baixo overhead. As mensagens JSON são mais leves que os XML além de outras características que fazem o REST ter menor overhead.

(d) Embora opcionalmente, um recurso REST pode conter uma URI.

Errado. O URI é obrigatório.

(e) REST consiste em um estilo de desenvolvimento baseado em complexa interação cliente/servidor.

Errado. A interação cliente/servidor no REST é simples.

Resposta: C

48. (FCC - 2019 - TJ-MA - Analista Judiciário - Analista de Sistemas - Desenvolvimento)

O SOAP é um protocolo que permite que aplicações troquem informações no formato XML via HTTP. Uma mensagem SOAP

(a) deve conter um elemento Header com informações de cabeçalho.

(b) pode ou não conter um elemento Envelope para identificar a mensagem como SOAP.

(c) não deve conter uma referência a Document Type Definition – DTD.

(d) pode ou não conter um elemento Body com requisições e respostas.

(e) deve conter um elemento Fault para tratar erros e mostrar informações de status.

RESOLUÇÃO:

(a) deve conter um elemento Header com informações de cabeçalho.

Errado. O header é opcional, e não obrigatório.

(b) pode ou não conter um elemento Envelope para identificar a mensagem como SOAP.

Errado. O Envelope é obrigatório, e não opcional.

(c) não deve conter uma referência a Document Type Definition – DTD.

Certo. A mensagem SOAP utiliza os *namespaces* do WSDL. Não deve conter referência a DTD.

(d) pode ou não conter um elemento Body com requisições e respostas.

Errado. O *body* é obrigatório.

(e) deve conter um elemento Fault para tratar erros e mostrar informações de status.

Errado. O *fault* é opcional.

Resposta: C

49. (CESPE - 2019 - TJ-AM - Analista Judiciário - Analista de Sistemas)

Com relação à interoperabilidade entre sistemas, julgue o item seguinte.

Diferentemente do que ocorre em serviços web embasados em SOAP, as solicitações e as respostas dos serviços REST não são empacotadas em envelopes.

RESOLUÇÃO:

Certo. As mensagens SOAP devem obedecer a especificação WSDL, o que faz com que sempre sejam empacotadas em envelopes.

Já no REST não há essa obrigatoriedade de estar tudo dentro do envelope. Uma solicitação ou resposta REST pode ser um arquivo JSON ou YAML, os quais não são envelopes.

Exemplo de JSON

JSON

```
{  "empinfo" :
    {
        "employees" :  [
        {
            "name" : "James Kirk",
            "age" : 40,
        },
        {
            "name" : "Jean-Luc Picard",
            "age" : 45,
        },
        {
            "name" : "Wesley Crusher",
            "age" : 27,
        }
                    ]
    }
}
```

Exemplo de YAML

```
---
first_name: Adam
last_name: Bertram
hair_color: Brown
married: true
spouse:
    name: Miranda
    occupation: Mom
    interests:
        - Instagram
        - Facebook
        - "keeping the Bertram family in check"
dog_count: 2
dogs:
    dog1:
        name: Elliott
        breed: Shih-Tzu
        color: black/white
    dog2:
        name: Brody
        breed: Shih-Tzu
        color: black/white
```

Resposta: Certo

50. (FCC - 2019 - TRF - 4ª REGIÃO - Analista Judiciário - Sistemas de Tecnologia da Informação)

Considere as definições descritas abaixo:

(I) Define os componentes de uma especificação de serviço que pode ser usada para descobrir sua existência. Esses componentes incluem informações sobre o provedor de serviço, os serviços fornecidos, o local da descrição da interface de serviço e informações sobre os relacionamentos de negócios.

(II) Padrão de trocas de mensagens que oferece suporte à comunicação entre os serviços. Define os componentes essenciais e opcionais das mensagens passadas entre serviços.

(III) Linguagem de definição de serviço Web, que é um padrão para a definição de interface de serviço. Define como as operações de serviço (nomes de operação, parâmetros e seus tipos) e associações de serviço devem ser definidas.

As descrições I, II e III correspondem, correta e respectivamente, a

(a) WSDL, UDDI e SOAP.

(b) SOAP, WSDL e UDDI.

(c) SOAP, UDDI e WSDL.

(d) UDDI, WSDL e SOAP.

(e) UDDI, SOAP e WSDL.

RESOLUÇÃO:

(I) Define os componentes de uma especificação de serviço que pode ser usada para descobrir sua existência. Esses componentes incluem informações sobre o provedor de serviço, os serviços fornecidos, o local da descrição da interface de serviço e informações sobre os relacionamentos de negócios.

Está falando do **UDDI**. O diretório para registro e descoberta de serviços no SOAP.

(II) Padrão de trocas de mensagens que oferece suporte à comunicação entre os serviços. Define os componentes essenciais e opcionais das mensagens passadas entre serviços.

Está falando do próprio protocolo **SOAP**.

(III) Linguagem de definição de serviço Web, que é um padrão para a definição de interface de serviço. Define como as operações de serviço (nomes de operação, parâmetros e seus tipos) e associações de serviço devem ser definidas.

Está falando do **WSDL** (Web Service Description Language).

Resposta: E

Lista de questões comentadas

1. (CESPE - 2019 - MPC-PA - Analista Ministerial – Tecnologia da Informação)

Quanto a web services, o REST

(a) possui uma única URL para identificar as diversas operações executadas por um web service.

(b) retorna os dados em somente um formato proprietário, chamado de RESTful.

(c) é uma alternativa ao padrão XML para representar informações.

(d) utiliza um formato baseado em execução de métodos para representar trocas de objetos em JavaScript.

(e) é um estilo arquitetônico para implementar web services.

---

2. (IF-PE - 2019 - IF-PE - Técnico em Tecnologia da Informação - Desenvolvimento)

Os webservices REST (Transferência de Estado Representacional) são uma arquitetura de software para sistemas hypermedia. Os webservices estilo REST utilizam, para a realização de suas operações, os mesmos verbos do protocolo HTTP. As operações que simbolizam o CRUD (criar, ler, atualizar e deletar) são, respectivamente,

(a) POST, GET, PUT e DELETE.

(b) POST, HEAD, GET e DELETE.

(c) PUT, GET, HEAD e DELETE.

(d) PUT, TRACE, POST e DELETE.

(e) POST, GET, TRACE e DELETE.

---

3. (CESPE - 2019 - TJ-AM - Assistente Judiciário - Programador)

```
1    import javax.jws.WebService;
2    import javax.jws.WebMethod;
3    import javax.jws.WebParam;
4
5    @WebService(
6     name = "WelcomeSOAP",
7     serviceName = "WelcomeSOAPService" )
8    public class WelcomeSOAP
9   {
10   @WebMethod( operationName = "welcome" )
11    public String welcome( @WebParam( name =
"name" ) String name )
12    {
13        return "Welcome to JAX-WS web services
with SOAP," + name + "!";
14     } \\ fim do método welcome
15     } \\ fim da classe WelcomeSOAP
```

Com relação a esse código do serviço web WelcomeSOAP, julgue o item que se segue.

O método welcome é marcado pela anotação @WebMethod para indicar que ele pode ser chamado remotamente; essa anotação usa o atributo operationName para especificar o nome do método que é exibido ao cliente do serviço web.

4. (CESPE - 2019 - TJ-AM - Assistente Judiciário - Programador)

```
1    import javax.jws.WebService;
2    import javax.jws.WebMethod;
3    import javax.jws.WebParam;
4
5    @WebService(
6     name = "WelcomeSOAP",
7     serviceName = "WelcomeSOAPService" )
8    public class WelcomeSOAP
9   {
10   @WebMethod( operationName = "welcome" )
11    public String welcome( @WebParam( name =
"name" ) String name )
12    {
13        return "Welcome to JAX-WS web services
with SOAP," + name + "!";
14     } \\ fim do método welcome
15     } \\ fim da classe WelcomeSOAP
```

Com relação a esse código do serviço web WelcomeSOAP, julgue o item que se segue.

A anotação @WebService indica que a classe WelcomeSOAP implementa um serviço web; o atributo name especifica o nome da classe proxy que será gerada para o cliente, enquanto o atributo serviceName configura o nome de serviço.

---

5. (VUNESP - 2019 - Prefeitura de Valinhos - SP - Analista de Tecnologia da Informação – SAI)

Considere a seguinte chamada HTTP de um webservice para apagar o cadastro de um produto:

GET /produto/1234/apagar

Em relação à essa chamada, conclui-se que ela

(a) pode ser considerada RESTful, bastando apenas que o método seja alterado para DELETE.
(b) está incorreta, pois o URI não deve conter parâmetros da requisição, como o código do produto.
(c) somente estará correta se o corpo da requisição (request body) informar os dados do recurso a ser apagado.
(d) não funcionará, pois o método GET somente pode ser utilizado para obter dados e não para apagá-los.
(e) não corresponde a um webservice RESTful, pois promove alterações utilizando uma operação não padronizada sobre um recurso.

---

6. (IADES - 2019 - BRB - Analista de Tecnologia da Informação)

No contexto de microsserviços, trata-se de uma abstração da arquitetura da web. Resumidamente, consiste em princípios/regras/constraints que, quando seguidos, permitem a criação de um projeto com interfaces bem definidas, dessa forma, permitindo, por exemplo, que aplicações se comuniquem.

Disponível em: <https://becode.com.br/> . Acesso em: 8 ago. 2019, com adaptações.

Essa definição diz respeito a

(a) WSDL.

(b) SDK.

(c) JSON-RPC.

(d) XML-RPC.

(e) REST API.

---

7. (IDECAN - 2019 - IF-PB - Professor - Informática)

Sobre o estilo arquitetural REST (Representational State Transfer), é correto afirmar que

(a) a resposta das requisições é sempre feita no formato JSON, o que impossibilita a implementação de aplicativos móveis.

(b) faz uso de operações que mantêm informações sobre a sessão no lado recebedor, sendo assim chamada de *stateless*.

(c) algumas de suas restrições incluem arquitetura cliente-servidor, *statelessness*, sistema em camadas e interface uniforme.

(d) surgiu recentemente, no ano de 2010, como resultado dos esforços da Google.

(e) faz uso apenas dos verbos GET e POST, do HTTP.

---

8. (CESPE - 2019 - SLU-DF - Analista de Gestão de Resíduos Sólidos - Informática)

Acerca de REST e DHCP, julgue item que se segue.

Entre os princípios orientadores a serem seguidos na implantação de uma API RESTful Java inclui-se o *stateless*, em que cada solicitação do cliente para o servidor deve conter todas as informações necessárias, independentemente das informações armazenadas no servidor.

---

9. (CESPE - 2019 - SLU-DF - Analista de Gestão de Resíduos Sólidos - Informática)

Julgue o próximo item, a respeito de domain-driven design, design patterns, emergent design, enterprise content management e REST.

Em um web service REST que gerencie alguns tipos de serviço, os conflitos decorrentes de recursos que tenham identificadores iguais são automaticamente resolvidos no web service.

---

10. (CESPE - 2019 - SLU-DF - Analista de Gestão de Resíduos Sólidos - Informática)

Acerca de arquitetura de software, julgue o item a seguir.

Um web service pode assumir o papel de provedor de serviço e de consumidor de serviço.

11.(FCC - 2019 - SEFAZ-BA - Auditor Fiscal - Administração, Finanças e Controle Interno - Prova II)

Os web services são componentes de software na web que podem fornecer determinados serviços a aplicações criadas em diferentes linguagens. Podem usar o protocolo SOAP para transferência de mensagens em formato XML. Para descrever a estrutura destas mensagens geralmente utiliza-se

(a) REST.

(b) WSDL.

(c) CORBA.

(d) RESTFUL.

(e) HTML.

---

12. (CCV-UFC - 2019 - UFC - Técnico de Tecnologia da Informação - Desenvolvimento de Sistemas)

Sobre Web services, assinale a alternativa correta.

(a) WSDL descreve os serviços disponibilizados à rede através de uma semântica JSON.

(b) XSLT é uma linguagem utilizada para transformar documentos JSON em documentos XML.

(c) As APIs de Web services que aderem às restrições de arquitetura REST são chamadas de APIs RESTful.

(d) SOAP é um protocolo para troca de informações estruturadas em uma plataforma descentralizada e distribuída baseadas em documentos JSON.

(e) *Representational State Transfer* (REST), é um estilo de arquitetura que define um conjunto de restrições e propriedades baseados em SOAP.

---

13. (CESPE - 2019 - CGE - CE - Auditor de Controle Interno - Tecnologia da Informação)

Os web services evoluíram para integrar aplicativos e fornecer serviços, no entanto apresentam inúmeros riscos de ataques. O ataque que força a reinicialização do sistema, consumindo todos os recursos, é denominado

(a) XML injection.

(b) SQL injection.

(c) DoS attack

(d) cross-site scripting.

(e) invalid types.

---

14. (COSEAC - 2019 - UFF - Técnico de Tecnologia da Informação)

Em relação aos web services, avalie se são verdadeiras (V) ou falsas (F) as afirmativas a seguir:

I Um web service tem uma estrutura interna fracamente acoplada.

II Um web service possui uma capacidade de criar soluções distribuídas e centralizadas.

III Um web service representa a materialização da ideia de uma serviço acessível em qualquer lugar do planeta.

As afirmativas I, II e III são, respectivamente:

(a) V, F e V.

(b) F, V e F.

(c) V, F e F.

(d) F, F e V.

(e) V, V e V.

15.(UFSC - 2019 - UFSC - Técnico de Tecnologia da Informação)

Considere uma aplicação para Web do tipo webservice que implementa a abordagem RESTful. Assinale a alternativa que completa correta e respectivamente as seguintes frases.

O método _____ deve ser usado para alterar um dado no servidor. O método _____, por sua vez, deve ser usado para obter um dado do servidor. Finalmente, o método _____ deve ser usado para incluir um dado no servidor.

(a) PUT – PUT – GET

(b) POST – PUT – GET

(c) POST – GET – POST

(d) GET – POST – PUT

(e) PUT – GET – POST

---

16. (CESPE - 2018 - STJ - Técnico Judiciário - Desenvolvimento de Sistemas)

Julgue o próximo item, relativo a model-view-controller (MVC), proxy reverso e representational state transfer (REST).

A REST define uma arquitetura cliente-servidor na qual o servidor não mantém contexto de cliente entre transações, ou seja, é *stateless* e toda transação contém as informações necessárias para satisfazer a solicitação.

17.(FCC - 2018 - DPE-AM - Assistente Técnico de Defensoria - Programador)

De acordo com a arquitetura REST, um serviço Web RESTful

(a) deve manter um estado de cliente no servidor.

(b) não consegue tratar cada requisição de forma independente.

(c) suporta somente os métodos GET e POST.

(d) não funciona bem com os protocolos HTTP.

(e) não deve manter um estado de cliente no servidor.

---

18. (FCC - 2017 - TRF - 5ª REGIÃO - Analista Judiciário - Informática Desenvolvimento)

Se um serviço web baseado na arquitetura REST (RESTful) está localizado em http://www.trf5.jus.br/employee, quando o cliente fizer uma requisição a este serviço deverá

(a) usar a Web Services Description Language para descrever as regras de comunicação com o serviço.

(b) usar JAX-WS para sincronizar a comunicação com o serviço.

(c) estabelecer e manter essa conexão com o servidor até o final da troca de mensagens SOAP.

(d) usar um dos métodos HTTP como POST, GET, PUT ou DELETE.

(e) enviar um sinal beacon solicitando ao servidor uma porta para conexão.

---

19. (PUC-PR - 2017 - TJ-MS - Técnico de Nível Superior - Analista de Sistemas)

Considerando os conceitos relacionados a Web services, analise as assertivas a seguir.

I. Web services são mecanismos utilizados na integração de sistemas.

II. WSDL é o protocolo utilizado durante uma requisição para transportar uma mensagem.

III. Interoperabilidade é a capacidade de um serviço ser descoberto facilmente.

IV. UDDI é um serviço de diretório para registrar e pesquisar serviços web.

V. XML é uma linguagem de marcação para troca de informações.

Está(ão) CORRETA(S) apenas a(s) assertiva(s):

(a) I, II, IV e V.

(b) III.

(c) I, IV e V.

(d) IV.

(e) I e V.

---

20. (CESPE - 2017 - TRT - 7ª Região (CE) - Analista Judiciário - Tecnologia da Informação)

Assinale a opção que apresenta o método HTTP que deve ser usado para a busca de recursos por meio do web service RESTful.

(a) delete

(b) get

(c) put

(d) options

---

21. (Gestão Concurso - 2018 - EMATER-MG - Analista de Sistemas I)

O serviço Web que define os componentes de uma especificação de serviço que podem ser usados para descobrir a existência de um serviço é conhecido por

(a) XML

(b) XSLT

(c) UDDI

(d) WSDL

---

22. (FCC - 2018 - Prefeitura de São Luís - MA - Auditor Fiscal de Tributos I - Tecnologia da Informação (TI))

Web services são projetados para suportar interações interoperáveis entre máquinas em uma rede. Esta interoperabilidade é obtida por meio de um conjunto de padrões de tecnologia que, acoplados aos princípios de projeto orientado a serviço, formam um SOA fundamentado na tecnologia XML com base em 3 especificações.

São elas:

(a) Interface Message; Service e Binding.

(b) Web Service Description (WSD); Simple Object Access Protocol (SOAP) e Hypertext Transfer Protocol (HTTP).

(c) ServiceKey; BusinessEntity e BusinessKey.

(d) Web Services Definition Language (WSDL); Simple Object Access Protocol (SOAP) e Universal Description, Discovery, and Integration (UDDI).

(e) Web Services Definition Language (WSDL); Common Object Request Broker Architecture (CORBA) e Distributed Component Object Model (DCOM).

---

23. (FGV - 2018 - Câmara de Salvador - BA - Analista de Tecnologia da Informação)

Usualmente, WebServices envolvem a utilização dos padrões XML, SOAP e WSDL.

A função de cada um deles é, respectivamente:

(a) descrever os parâmetros, disponibilizar o serviço, transferir as mensagens;

(b) transferir as mensagens, descrever o algoritmo, transportar as mensagens;

(c) rotular e formatar os dados, transferir as mensagens, descrever a disponibilidade do serviço;

(d) transferir as mensagens, descrever a disponibilidade do serviço, formatar os parâmetros;

(e) descrever as classes e suas interfaces, instanciar os objetos, descrever os algoritmos.

---

24. (CESPE - 2018 - STJ - Técnico Judiciário - Suporte Técnico)

Julgue o item a seguir, acerca de arquiteturas de integração e web services.

Web service é uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes.

---

25. (CESPE - 2017 - TRE-BA - Técnico Judiciário – Programação de Sistemas)

No que se refere a web services, assinale a opção correta.

(a) As solicitações e respostas XML trafegam no protocolo HTTP, não sendo possível utilizá-las nos protocolos FTP e SMTP.

(b) Um dos componentes de um web service SOAP (simple object access protocol) é a UDDI (universal description, discovery and integration), a qual é um arquivo do tipo XML que descreve detalhadamente um web service, especificando como deve ser o formato de entrada e saída de cada operação.

(c) As duas formas de envio de mensagem para que um cliente possa efetuar solicitações a um web service são one-way messaging e request-response messaging.

(d) O WSDL (web services description language) é uma linguagem para o desenvolvimento de web services similar ao XML.

(e) Os web services permitem a integração de sistemas, todavia dependem da linguagem de programação sob a qual tenham sido desenvolvidos e do sistema operacional do computador onde esses sistemas forem executados.

---

26. (FAPEC - 2018 - UFMS - Analista de Tecnologia da Informação)

Considere as afirmações a seguir:

I - O Web Application Description Language (WADL) é um XML utilizado para descrever serviços RESTFul.

II - Web services baseados sobre a arquitetura REST são conhecidos como RESTFul Web services.

III - RESTFul Web services utilizam o HTTP para transportar o dado e o JSON para representar o dado.

Está(ão) correta(s):

(a) Apenas I.

(b) Apenas II.

(c) Apenas III.

(d) Apenas I e II.

(e) I, II e III.

27.(FCM - 2018 - IFN-MG - Professor - Informática)

Sobre as afirmativas abaixo relacionadas aos conceitos de serviços Web

Considere os acrônimos:

•REST - Representational State Transfer;

•SOAP - Simple Object Access Protocol .

É correrto afirmar que o

(a) REST é uma arquitetura de rede, baseada no protocolo XML, que permite o serviço Web e o cliente se comunicar.

(b) SOAP é uma arquitetura de rede, baseada em mecanismos de solicitação e resposta tradicionais da Web, como solicitações GET e POST

(c) SOAP envia solicitação e resposta de seus serviços empacotados em envelopes, diferentemente do que ocorre nos serviços baseados em REST.

(d) SOAP é um protocolo independente de plataforma que faz chamadas de procedimentos remotos por meio de conexões FTP com poucas limitações.

(e) REST e o SOAP são um tipo de computação distribuída que permite a um aplicativo clientes ter acesso direto à memória principal de uma aplicação servidora.

---

28. (INSTITUTO AOCP - 2018 - PRODEB - Analista de TIC II - Construção de Software)

Um serviço REST usualmente suporta mais de um formato para representação de seus recursos, sendo esta uma de suas características principais, já que facilita a inclusão de novos clientes e a interoperabilidade entre os projetos. Assinale a alternativa que apresenta somente formatos utilizados por um serviço REST.

(a) JAVA e JAVASCRIPT.

(b) JAVA, RUBY ON REALS e .NET.

(c) YAML e JSON.

(d) XML e C++.

(e) JSON, CSS e SQL.

---

29. (CESGRANRIO - 2018 - LIQUIGÁS - Profissional Júnior - Analista de Sistemas)

Os Web Services são muito úteis para o desenvolvimento de sistemas de informação em empresas, pois

(a) utilizam bancos de dados "No SQL".

(b) possuem capacidade de aprendizado.

(c) suportam a interação entre sistemas de informação (interoperabilidade).

(d) viabilizam a interação entre os usuários de um sistemas de informação (web chat).

(e) permitem a interação de sistemas de informação com seus usuários, pois utilizam HTML como sua principal linguagem.

---

30. (CCV-UFC - 2018 - UFC - Analista de Tecnologia da Informação)

Sobre REST e RESTful, o que é correto afirmar?

(a) REST significa Representational State Traffic.
(b) REST é um protocolo de transferência de dados.
(c) JAX-RS é a API Java para RESTful Web Services.
(d) WSDL é o padrão que deve ser utilizado com REST.
(e) REST permite a utilização de apenas texto e JSON como formato de dados.

31.(CESPE - 2018 - STJ - Técnico Judiciário - Suporte Técnico)

Julgue o item a seguir, acerca de arquiteturas de integração e web services .

Os serviços Web RESTful utilizam o HTTP como um meio de comunicação entre cliente e servidor.

32.(CESPE - 2018 - CGM de João Pessoa - PB - Auditor Municipal de Controle Interno - Desenvolvimento de Sistemas)

Acerca de *service-oriented architecture*, web services, mensageria e CORBA (*common object request broker architecture*), julgue o item a seguir.

Web services permitem disponibilizar serviços de forma agnóstica quando a UDDI (*universal description, discovery and integration*) estabelece um formato padrão de mensagem que consiste em um documento XML capaz de hospedar dados RPC centrados em documentos, para que haja intercâmbio de dados de modelos síncronos (pedido e resposta) e assíncronos (orientados a processo).

33.(CESPE - 2017 - SEDF - Analista de Gestão Educacional - Tecnologia da Informação)

Em relação a web services, julgue o item seguinte.

Por oferecerem um framework de comunicação com base em contratos de serviços fisicamente desacoplados, os web services permitem que um contrato de serviços seja totalmente padronizado, independentemente de sua implementação.

34. (CESPE - 2016 - TCE-PA - Auditor de Controle Externo - Área Informática - Administrador de Banco de Dados)

Julgue o item subsequente, relativo a SOA, web services e servidor web.

Os web services devem ser projetados para ser utilizados independentemente de paradigmas de programação.

35.(CESPE - 2016 - TCE-PA - Auditor de Controle Externo - Área Informática - Analista de Sistema)

Web service é um software que, armazenado em um computador, pode ser acessado por outro software em outro computador por meio de uma rede. A partir dessa afirmação, julgue o item subsequente.

Para que um web service funcione corretamente, os softwares cliente/servidor devem ser escritos na mesma linguagem.

---

36. (CESPE - 2016 - TCE-PA - Auditor de Controle Externo - Área Informática - Analista de Sistema)

Web service é um software que, armazenado em um computador, pode ser acessado por outro software em outro computador por meio de uma rede. A partir dessa afirmação, julgue o item subsequente.

Ao se usar o protocolo SOAP (simple object access protocol), cada solicitação e cada resposta são colocadas em um envelope SOAP, nos momentos de invocação e retorno de um web service, respectivamente.

37.(CESPE - 2016 - TRT - 8ª Região (PA e AP) - Técnico Judiciário - Tecnologia da Informação)

A interface de um webservice pode mudar ao longo do tempo sem comprometer a habilidade de interação do cliente com o serviço. A essa caraterística dá-se o nome de

(a) encapsulamento.

(b) acoplamento fraco.

(c) associação.

(d) publicação.

(e) manutenção de entidades.

---

38. (CESPE - 2016 - TRT - 8ª Região (PA e AP) - Técnico Judiciário - Tecnologia da Informação)

A respeito dos conceitos de web services e REST, assinale a opção correta.

(a) O método POST é utilizado na atualização de um recurso existente.

(b) Pode-se utilizar qualquer meio de transporte existente para o envio de uma requisição, incluindo HTTP, SMTP e TCP.

(c) O modelo REST impõe restrições ao formato da mensagem.

(d) Ao desenvolver uma aplicação, o recurso é transferido pela rede.

(e) As chamadas às URIs (uniform resource indicator) são feitas por meio de métodos HTTP, os quais indicam para o serviço a ação a ser realizada com o recurso.

---

39. (CESPE - 2016 - TRT - 8ª Região (PA e AP) - Técnico Judiciário - Tecnologia da Informação)

Acerca da arquitetura de microsserviços, assinale a opção correta.

(a) A arquitetura de microsserviços é um padrão para a criação de aplicações distribuídas, porém não possui alta escalabilidade.

(b) A comunicação entre os microsserviços é feita por meio de mecanismos padrões de tecnologia, como, por exemplo, o REST (representational state transfer).

(c) Microsserviços utilizam uma única base de dados lógica para a persistência de dados.

(d) Um requisito fundamental da arquitetura de microsserviços é o uso de versionamento de mudanças.

(e) Os microsserviços são componentes autônomos e de alto acoplamento, de modo que há a necessidade de se utilizar uma mesma linguagem na sua construção.

---

40. (CESPE - 2016 - TRE-PI - Analista Judiciário - Análise de Sistemas)

Acerca de REST (representational state transfer), assinale a opção correta.

(a) Na implementação do REST, todos os recursos devem responder a todos os métodos.

(b) O método GET permite obter e alterar o estado atual de um recurso.

(c) O método EXPUNGE permite excluir um recurso.

(d) A arquitetura de comunicação entre aplicações baseia-se em um modelo rígido de recursos e localizações.

(e) O método MODIFY permite alterar um recurso.

41. (CESPE - 2015 - MEC - Analista de Sistemas)

Acerca da utilização dos protocolos SOAP e REST, julgue o item seguinte.

Os métodos PUT e DELETE no protocolo REST não são considerados como idempotentes.

42. (CESPE - 2015 - MEC - Arquiteto de Sistemas)

Com relação a tecnologias Web, julgue o item a seguir.

Em uma web service, a linguagem de implementação e a plataforma utilizada são relevantes para os clientes.

43. (CESPE - 2015 - MEC - Arquiteto de Sistemas)

Com relação a tecnologias Web, julgue o item a seguir.

As principais características do REST (representationl state transfer) são interface uniforme, stateless e cache.

44. (CESPE - 2015 - MEC - Gerente de Projetos)

No que se refere a padrões SOA (service-oriented architecture) e a REST (representational state transfer), julgue o item subsecutivo.

Entre as restrições da REST está a interface uniforme, a qual requer que um serviço ofereça várias operações e aguarde a solicitação dessas operações pelo servidor.

45. (CESPE - 2015 - TJ-DFT - Analista Judiciário - Analista de Sistemas)

A Arquitetura Orientada a Serviços (SOA), no cenário dos modelos arquiteturais modernos, enfatiza o reúso como elemento chave para a maximização dos resultados em tecnologia da informação. A catalogação e a gerência dos ativos de software da organização na condição de serviços de aplicativos são os pilares em meio aos quais devem se sustentar essa nova arquitetura. A esse respeito, julgue o próximo item.

Em um Web Service RESTful, cada método é identificado por uma URL única. Assim, quando o servidor recebe uma solicitação, ele identifica de forma inequívoca a operação que será executada.

---

46. (CESPE - 2015 - TCE-RN - Inspetor - Tecnologia da Informação - Cargo 5)

A respeito de arquitetura para desenvolvimento de sistemas, de programação orientada a aspectos (POA) e de banco de dados distribuídos, julgue o item subsecutivo.

Em web service, o objetivo do cabeçalho do envelope SOAP (SOAP header block) é o de fazer cumprir as regras que governam a troca de mensagens SOAP, por meio de dados trocados em formato XML.

---

47. (CESPE - 2015 - TRE-MT - Analista Judiciário - Análise de Sistemas)

Acerca de REST (representational state transfer), assinale a opção correta.

(a) O protocolo REST utiliza SOAP e XML.

(b) REST utiliza recurso não identificável baseado em PUT e GET.

(c) REST pode ser utilizado para implementar WebServices de baixo overhead.

(d) Embora opcionalmente, um recurso REST pode conter uma URI.

(e) REST consiste em um estilo de desenvolvimento baseado em complexa interação cliente/servidor.

---

48. (FCC - 2019 - TJ-MA - Analista Judiciário - Analista de Sistemas - Desenvolvimento)

O SOAP é um protocolo que permite que aplicações troquem informações no formato XML via HTTP. Uma mensagem SOAP

(a) deve conter um elemento Header com informações de cabeçalho.

(b) pode ou não conter um elemento Envelope para identificar a mensagem como SOAP.

(c) não deve conter uma referência a Document Type Definition – DTD.

(d) pode ou não conter um elemento Body com requisições e respostas.

(e) deve conter um elemento Fault para tratar erros e mostrar informações de status.

---

49. (CESPE - 2019 - TJ-AM - Analista Judiciário - Analista de Sistemas)

Com relação à interoperabilidade entre sistemas, julgue o item seguinte.

Diferentemente do que ocorre em serviços web embasados em SOAP, as solicitações e as respostas dos serviços REST não são empacotadas em envelopes.

---

50. (FCC - 2019 - TRF - 4ª REGIÃO - Analista Judiciário - Sistemas de Tecnologia da Informação)

Considere as definições descritas abaixo:

(I) Define os componentes de uma especificação de serviço que pode ser usada para descobrir sua existência. Esses componentes incluem informações sobre o provedor de serviço, os serviços fornecidos, o local da descrição da interface de serviço e informações sobre os relacionamentos de negócios.

(II) Padrão de trocas de mensagens que oferece suporte à comunicação entre os serviços. Define os componentes essenciais e opcionais das mensagens passadas entre serviços.

(III) Linguagem de definição de serviço Web, que é um padrão para a definição de interface de serviço. Define como as operações de serviço (nomes de operação, parâmetros e seus tipos) e associações de serviço devem ser definidas.

As descrições I, II e III correspondem, correta e respectivamente, a

(a) WSDL, UDDI e SOAP.

(b) SOAP, WSDL e UDDI.

(c) SOAP, UDDI e WSDL.

(d) UDDI, WSDL e SOAP.

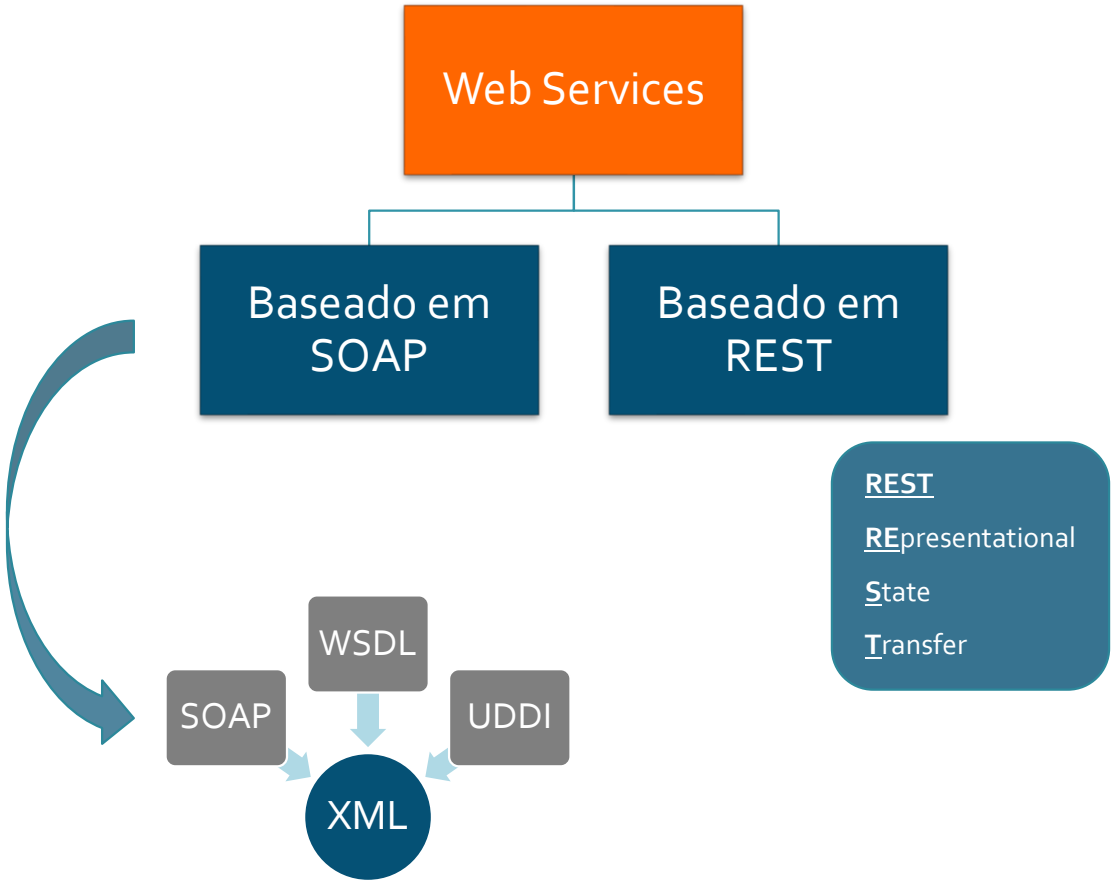
(e) UDDI, SOAP e WSDL.

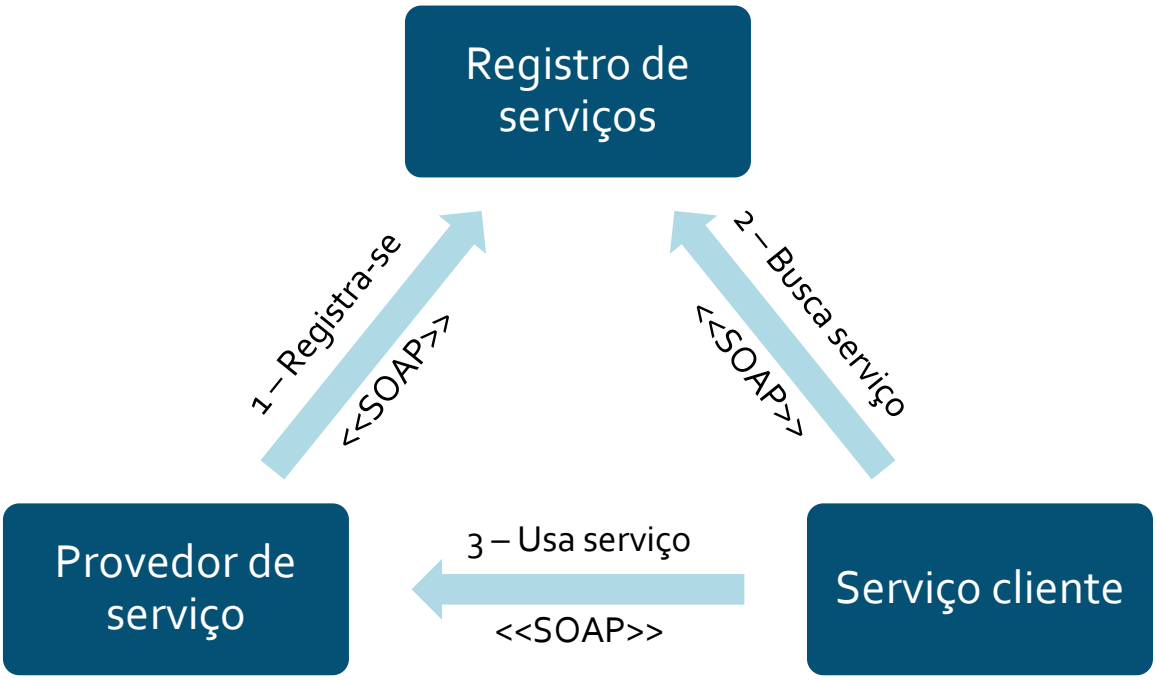
Gabarito

1. E
2. A
3. Certo
4. Certo
5. E
6. E
7. C
8. Certo
9. Errado
10. Certo
11. B
12. C
13. C
14. A
15. E
16. Certo
17. E
18. D
19. C
20. B
21. C
22. D
23. C
24. Certo
25. C
26. E
27. C
28. C
29. C
30. C
31. Certo
32. Errado
33. Certo
34. Certo
35. Errado
36. Certo
37. B
38. E
39. B
40. D
41. Errado
42. Errado
43. Certo
44. Errado
45. Certo
46. Errado
47. C
48. C
49. Certo
50. E

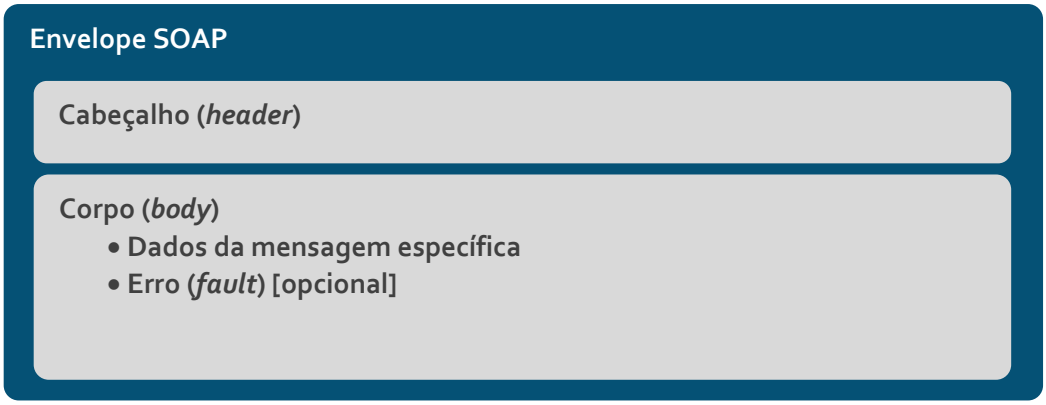
Resumo direcionado

Modelos de web services





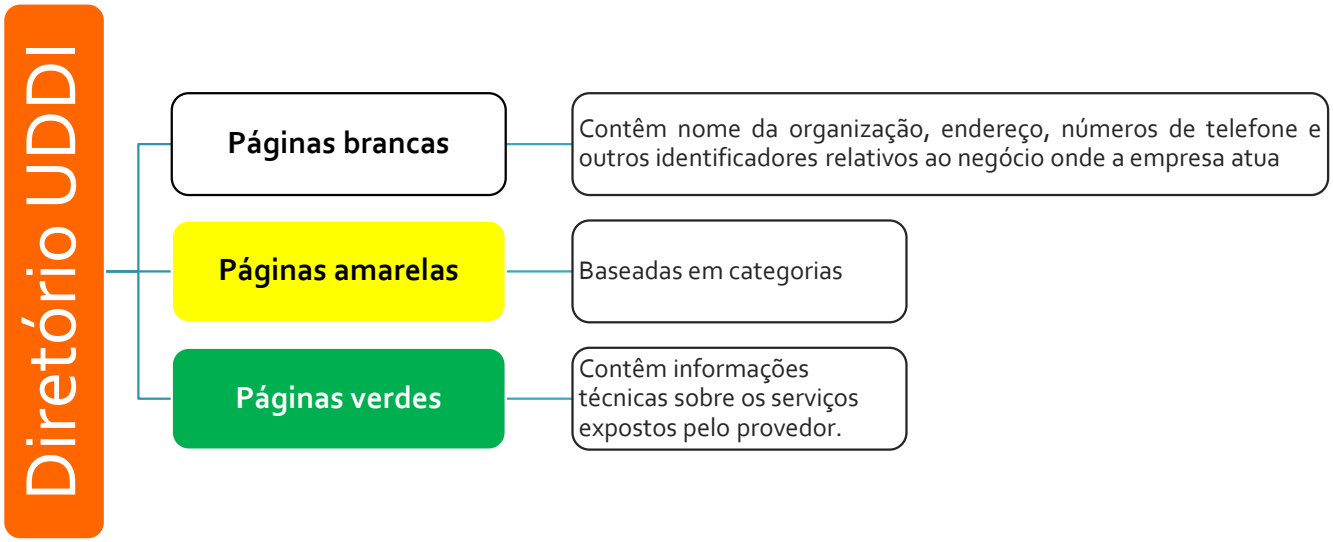
Mensagem SOAP



WSDL

O que	• o serviço faz
Como	• o serviço pode ser acessado
Onde	• o serviço pode ser acessado

UDDI


Diretório UDDI
Páginas brancas
Contêm nome da organização, endereço, números de telefone e outros identificadores relativos ao negócio onde a empresa atua
Páginas amarelas
Baseadas em categorias
Páginas verdes
Contêm informações técnicas sobre os serviços expostos pelo provedor.

Propriedades do REST

REST

Métodos HTTP

Método	Descrição	Safe?	Idempotente?
GET	Recupera uma entidade em resposta à solicitação de um recurso.	Sim	Sim
POST	Permite a criação de novos recursos. Retorna uma entidade descrevendo o resultado da ação (novo recurso).	Não	Não
PUT	Atualiza uma entidade (cria a entidade caso ainda não exista).	Não	Sim
DELETE	Remove uma entidade.	Não	Sim
HEAD	Mecanismo usado por um cliente para verificar a existência de um recurso e, possivelmente, descobrir metadados sobre ele. Não recupera dados do recurso.	Sim	Sim
PATCH	Permite a realização de atualizações parciais.	Não	Não
OPTIONS	Verifica, no servidor, que outros verbos são aplicáveis a um recurso. Permite que desenvolvedores entendam como interagir com recursos.	Sim	Sim

Restrições arquiteturais REST

Restrição	Descrição
Arquitetura cliente-servidor	O servidor é o web service. O cliente pode ser um aplicativo qualquer como um app de smartphone, um sistema Windows, Linux, Mac ou até mesmo outro web service.
Stateless	No lado do servidor, não armazena estado entre as requisições. Cada requisição do cliente contém toda informação necessária para seu atendimento por parte do serviço.
Cacheability	Possibilidade de fazer cache de alguns tipos de resposta de forma a retornar o resultado para o cliente de forma mais ágil.
Sistema em Camadas	Construção de diversos níveis de abstração incluindo, além do cliente e do serviço, camadas intermediárias como *proxy* e balanceadores de carga.
Código sob demanda	Servidores podem estender funcionalidades dos clientes ao transferir, além de dados, códigos executáveis.
Interface uniforme	Forma padronizada e unificada de os clientes acessarem os serviços (*uniform contract*).

Comparativo REST x SOAP

REST	SOAP
+ simples	- simples
- *overhead* de comunicação	+ *overhead* de comunicação
Uniform contract	Cada serviço tem seu contrato WSDL específico
Stateless	*Stateful*
Sem controle de transação	Com controle de transação
- segurança	+ segurança