



Sistemas Distribuídos para Web

Web Services

Curso Superior: Sistemas para Internet
Prof. Ricardo Alexandre Neves

SDW

Aula 11

Agenda

INTRODUÇÃO

SERVICE ORIENTED ARCHITECTURE

DEFINIÇÃO

CARACTERÍSTICAS

ARQUITETURA WEB SERVICES

XML – XSD

WSDL

Introdução

- * Plataformas middleware distribuídas:
 - * RPC;
 - * CORBA;
 - * DCOM;
 - * JAVA RMI.
- * Limitações em ambiente WEB
 - * Não utilização de tecnologias abertas;
 - * CORBA: Protocolo IIOP (firewall – tráfego);
 - * JAVA RMI: Limitada a linguagem Java;
 - * DCOM: Protocolos proprietários.

Introdução

- * Arquitetura Web Services:
 - * Formada por um conjunto de protocolos normatizados e abertos;
 - * Maior vantagem competitiva em relação às tecnologias concorrentes;
 - * Descritos, publicados, pesquisados, encontrados pela Internet;
 - * Independente de plataforma;

Introdução

- * Protocolos abertos utilizados:
 - * XML (Extensible Markup Language);
 - * SOAP (Simple Object Access Protocol);
 - * UDDI (Universal Description, Discovery and Integration);
 - * WSDL (Web Service Definition Language); e o
 - * HTTP (Hypertext Transfer Protocol).

Introdução

- * Principais diferenças entre o CORBA, JAVA RMI e os Web Services:

	CORBA	JAVA RMI	WEB SERVICE
Mecanismo de invocação	CORBA RMI	Java RMI	JAX-RPC
Formato dos dados	CDR	Serialização	XML
Protocolo de transporte	IIOP	JRMP	HTTP, SMTP, FTP, ...
Descrição da interface	CORBA IDL	Java Interface	WSDL
Mecanismo de descoberta	<i>Naming Service</i>	Java RMI <i>Registry</i>	UDDI

Tabela 8.1 – Diferenças entre o CORBA, o Java RMI e os Web Services

Service Oriented Architecture (SOA)

- * Introduzido pela Sun Microsystems – final década de 90;
- * Aproxima e facilita o desenvolvimento e composição de serviços modulares;
- * Integração e reutilização - criação de aplicações distribuídas.

SOA - Definição

- * Padronização – W3C (The World Wide Web Consortium);
- * Conjunto de componentes que podem ser invocados remotamente e cujas interfaces podem ser publicadas e pesquisadas;
- * Os componentes estão disponíveis como serviços independentes (modo standard);

SOA - Definição

- * Arquitetura de software que define o uso de um conjunto de serviços para suportar requisitos e usuários de um processo de negócio;
- * Não está associada a nenhuma tecnologia específica;
- * Pode ser implementada com vários standards
→ incluindo RPC e CORBA

SOA - Características

- * Arquitetura que proporciona a criação de aplicações combinando um conjunto de serviços;
- * Serviços operam entre si por meio de uma definição formal;
- * Independente de plataforma ou linguagem;
- * Semelhante ao CORBA (interoperação);
- * Mais adequado ao ambiente Web.

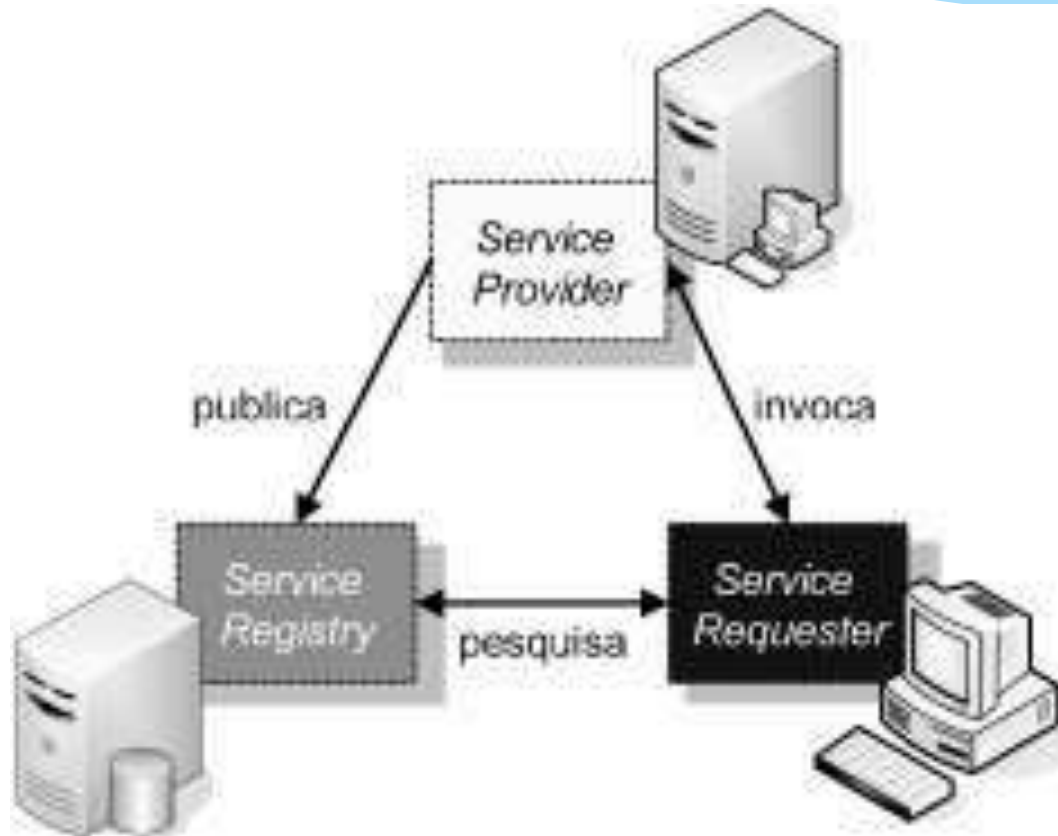
SOA - Características

- * Escalável: Funciona numa variedade de situações;
- * Fracamente acoplada: Clientes e servidores devem ser independentes;
- * Interoperabilidade: Comunicação independente de plataforma ou máquina;
- * Descoberta: Serviços descritos para serem descobertos dinamicamente – mecanismos de pesquisa;
- * Padrões: Protocolos de interação são normalizados, abertos e standards;
- * Interoperabilidade entre aplicações independentes.

SOA – Arquitetura Conceitual

- * Funcionalidade dos serviços – Interfaces normalizadas;
- * Os detalhes de implementação são ocultos a seus utilizadores;
- * Constituída por 3 entidades:
 - * Service Requester;
 - * Service Register;
 - * Service Provider.
- * Constituída por 3 operações:
 - * Publicação;
 - * Pesquisa;
 - * Invocação.

SOA – Arquitetura Conceitual



SOA – Arquitetura Conceitual

- * Service Requester: Invoca um serviço. Encontra a descrição de um serviço publicada no Service Registry;
- * Service Registry: Publicar os serviços. Permitir que os service requestors pesquisem em seu catálogo (informação sobre serviços);
- * Service Provider: Entidade responsável por construir/desenvolver um serviço, criar uma descrição e receber mensagens de invocação dos service requestors.

SOA com Web Services

- * O uso de Web Services não implica, forçosamente a implementação de uma solução SOA;
- * É possível desenvolver um sistema baseado em SOA sem recorrer obrigatoriamente aos W.S.;
- * Desenvolvimento de uma arquitetura SOA baseada em Web Services -> Modularidade;
- * Modularidade: Grau em que os componentes do sistema podem ser separados ou combinados;
- * Arquitetura SOA: Aplicação (peças de lego):
 - * Configuráveis e compatíveis que executam determinados serviços;
 - * Interação com outras “peças” por meio da rede com protocolos e normas abertas;
 - * Web Services (peças) podem ser escritas em qualquer linguagem e executadas em qualquer tipo de computador e local.

Arquitetura dos Web Services

- * Fornecedores de serviços -> Responsáveis por criar uma descrição do serviço (usando WSDL);
- * Publicá-la num servidor de registro (UDDI);
- * Receber mensagens de pedidos (formatadas com o SOAP) vindas de um ou mais clientes ;
- * Transportadas num formato e por um protocolo normalizado e aberto (XML e o HTTP).

Arquitetura dos Web Services

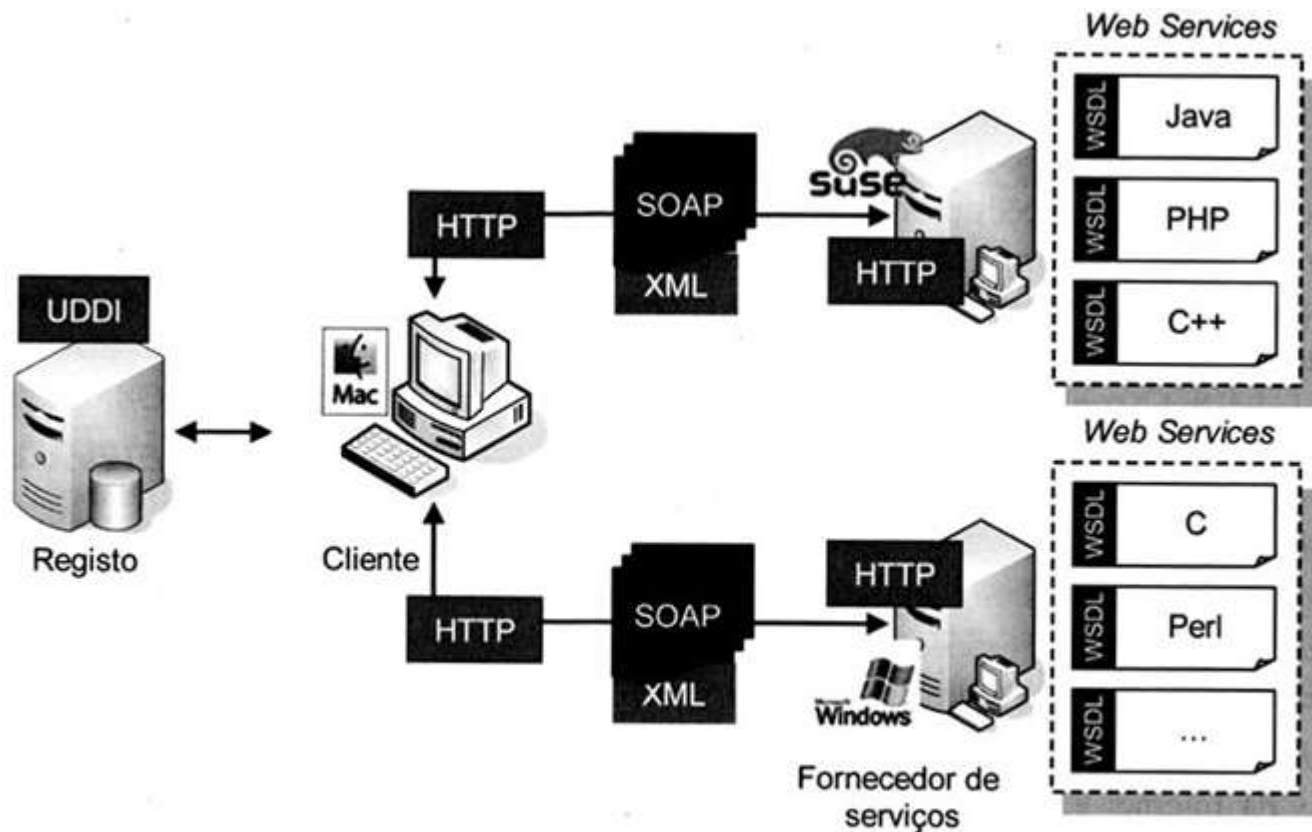


Figura 8.4 – Independência da plataforma e da linguagem de programação

Ciclo de Vida

- * 4 fases distintas:
 - * Descrição: Processo que o Web Service descreve sua interface para que os clientes possam solicitar suas funcionalidades em WSDL;
 - * Publicação: Processo opcional – Fornecedor publica a existência dos Web Services efetuando registro em um repositório UDDI;
 - * Pesquisa: Processo em que o cliente descobre os Web Services pretendidos;
 - * Invocação: Processo em que o cliente pede com que um serviço seja executado (envio e recebimento de mensagens SOAP).

Ciclo de Vida



Figura 8.5 – Ciclo de vida de um *Web Service*

Arquitetura



Figura 8.6 – Arquitectura dos *Web Services*

Arquitetura

- * Camada de Invocação:

- * Subcamada de transporte (mensagens clientes e Web Services)

- * Protocolos: HTTP, SMTP E O FTP;
 - * Paradigma Pedido/Resposta;
 - * Cliente estabelece ligação com o servidor e envia pedido;
 - * Servidor analisa e envia resposta ao cliente;

- * Subcamada de codificação:

- * Base linguagem XML;
 - * Método padrão de codificação e formatação de dados;
 - * Permite transferência de informação entre plataformas ;

- * Subcamada de Troca de Mensagens:

- * Suportada pelo protocolo SOAP;
 - * Define o formato das mensagens usadas durante clientes e serviços.

Arquitetura

- * Camada de Descrição:

- * Solução baseada em XML para definir as interfaces dos serviços, dados, tipos de mensagens, operações, modelos de interação e protocolos;
- * Definições descritas de acordo com a norma WSDL;

- * Camada de Publicação e Pesquisa:

- * Fornece conjunto de funcionalidades: publicação e a pesquisa de Web Services;
- * UDDI – Componente fundamental (mecanismo de registro e descoberta de serviços na Web);
- * Sistema de registro instanciado como um conjunto de documentos XML que contém uma descrição de entidades de negócio e dos Web Services.

Arquitetura

CAMADA	TECNOLOGIA	DESCRIÇÃO
PUBLICAÇÃO E PESQUISA	UDDI	Os <i>Web Services</i> são publicados num registo UDDI para que as aplicações clientes possam pesquisar serviços.
DESCRIÇÃO DOS SERVIÇOS	WSDL	Os <i>Web Services</i> usam uma linguagem de descrição chamada WSDL. Uma aplicação cliente necessita de obter o WSDL de um serviço antes de o invocar.
TROCA DE MENSAGENS	SOAP	Uma mensagem SOAP é um documento XML que encapsula os dados trocados entre um cliente e um serviço.
CODIFICAÇÃO DOS DADOS	XML	É a linguagem usada para descrever os protocolos associados aos <i>Web Services</i> .
TRANSPORTE	HTTP	É o protocolo de comunicação usado para transportar os dados entre clientes e serviços.

Tabela 8.2 – Resumo das normas e protocolos associadas aos *Web Services*

XML

* Exemplos de códigos XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<transaccacao id="17" banco="Caixa Geral de Depósitos" tipo="Visa">
  <nib>00298 456372348965275 9</nib>
  <operacao tipo="Pagamento">
    Carregamento do Telemóvel
  </operacao>
  <instituicao>Telecomunicações Móveis Nacionais (TMN)</instituicao>
  <montante moeda="euro">25</montante>
  <saldo moeda="euro">347</saldo>
  <notas>
    <nota>Caixa Multibanco</nota>
    <nota>Universidade da Madeira</nota>
    <nota>Caminho da Penteada</nota>
  </notas>
</transaccacao>
```


XSD: Definição

- * SOAP e WSDL suportam tipos de dados baseados em XSD (XML Schema Definition);
- * Tal especificação define normas para a descrição de tipos primitivos de dados, assim como estruturas complexas;
- * XSD é a linguagem utilizada para definir estrutura e as restrições de novos elementos e atributos;
- * Os novos elementos são definidos através das tags:
 - * `<simpleType>` e `<complexType>`

XSD: Exemplo

```
<types>
  <xsd:schema targetNamespace="http://abc.pt/types"
              xmlns:xsd="http://www.w3.org/2001/schema">
    <xsd:element name="Morada" type="Morada"/>
    <xsd:simpleType name="Distancia">
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="centimetro" />
        <xsd:enumeration value="metro" />
        <xsd:enumeration value="kilometro" />
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="Morada" >
      <xsd:sequence>
        <xsd:element name="nome" type="xsd:string"/>
        <xsd:element name="rua" type="xsd:string"/>
        <xsd:element name="nº" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </types>
```

WSDL

(Web Service Description Language)

- * Documento descrito em XML que descreve as funcionalidades de um Web Service, os tipos de dados e os protocolos;
 - * Envio e recebimento de mensagens;
- * As operações e as mensagens são descritas de forma abstrata;
- * São associadas a um protocolo de rede e formato da mensagem;
- * Objetivo semelhante ao CORBA IDL (AMBOS: definir interfaces de serviços ou objetos remotos e tipos de dados);

WSDL

(Web Service Description Language)

- * Permite a separação da descrição das funcionalidades, dos detalhes concretos de sua implementação;
- * A definição da interface fornece:
 - * Assinatura para todas as operações disponíveis (incluindo nome da operação, parâmetros de entrada e de saída e exceções);
- * Versão atual da WSDL segundo o W3C:
 - http://www.w3.org/standards/techs/wsdl#w3c_all (status)
 - <http://www.w3.org/TR/wsdl> (Documentação versão 1.1)

WSDL - Especificação

- * WSDL inclui a especificação de 3 componentes principais:
 - * Operações;
 - * Mensagens e pré-condições; e
 - * Efeitos.
- * Uma operação é semelhante a um método em Java;
- * As mensagens são parâmetros de entrada e saída das operações:
 - * São definidas utilizando o XSD;
- * Cada operação tem um conjunto de pré-condições e efeitos;
- * As pré-condições são geralmente condições lógicas que têm que ser avaliadas e se verdadeira a operação é executada.
- * Os efeitos são alterações do mundo real que podem ocorrer após a execução da operação.

WSDL - Especificação

- * Elementos que fazem parte da definição dos serviços em WSDL.

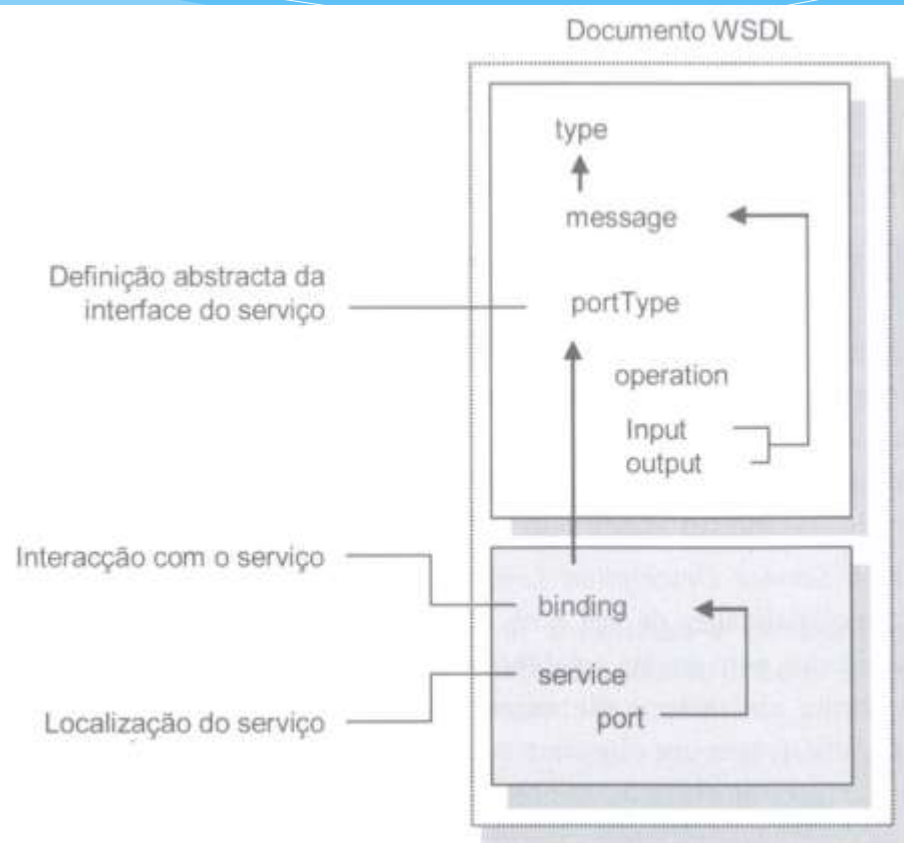


Figura 8.7 – Esquema da definição de um documento WSDL

WSDL - Especificação

- * Perspectiva mais detalhada de cada componente presente em um documento WSDL.

```
<definitions>
  <import>*
  <types>
    <schema></schema>*
  </types>
  <message>*
    <part></part>*
  </message>
  <portType>*
    <operation>*
```

types: Definição dos tipo de dados usados pelo *Web Service*

message: Definição abstracta das mensagens que são recebidas e enviadas

portType: Um conjunto de operações abstractas suportadas pelo *Web Service*

WSDL - Especificação

- * Perspectiva mais detalhada de cada componente presente em um documento WSDL (continuação)

```
<input></input>
<output></output>
<fault></fault>*
</operation>
</portType>
<binding>*
  <operation>*
    <input></input>
    <output></output>
  </operation>
</binding>
<service>*
  <port></port>*
</service>
</definitions>
```

operation: Descrição abstracta de uma operação suportada pelo serviço

binding: Especificação dos protocolos usados para transportar e formatar os dados usados pelas operações de um portType

service: O serviço disponibilizado aos clientes

port: Definição de um endereço único para um *binding* previamente definido

WSDL - Elementos

- * Elemento `<definitions>`: Reserva uma secção para declarar namespaces;
 - * Namespaces são indispensáveis;
 - * Tags XML são definidas pelos utilizadores (distribuídas pelo mundo);
 - * Escolha do mesmo nome por utilizadores diferentes para significados diferentes;
 - * Namespace utilizada para resolver o problema;
 - * É referenciado por uma URL;
 - * Declarado no início do documento WSDL.

WSDL - Elementos

- * Exemplo <definitions>

```
<definitions
  targetNamespace="myNS"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soapenc="http://schemas.xmlsoap.org/encoding/"
  xmlns:tns="myNS">
```

WSDL - Elementos

- * Elemento `<import>`: Permite separar a definição de um serviço em documentos independentes e incluí-los no documento principal;
 - * Ambiente modular;
 - * Reutilização;
- * Exemplo: Importar o namespace de outro arquivo, mas não o arquivo em si.
- * Elemento `<types>`: Fornece definições dos tipos de dados utilizados para descrever as mensagens trocadas.
 - * Elemento que define os tipos de dados utilizados pelo elemento `<message>`.

WSDL - Elementos

- * Elemento <message>: Definição abstrata das mensagens a serem transmitidas do cliente para o Web Service e vice-versa;
 - * Composta por várias partes lógicas, cada uma associada com uma definição integrada em um sistema de tipos;
 - * Esses elementos referenciam os tipos definidos na secção <types>;
 - * Uma mensagem consiste em um ou mais subelementos <part>.

WSDL - Elementos

- * Exemplo de <import>:

```
<import namespace="http://exemplo.pt/myservico"  
        Location="http://exemplo.pt/myservico/meusdados.xsd" />
```

- * Exemplo de <message>:

```
<message name="email">  
    <part name="msg" type="xsd:string" />  
    <part name="endereco" type="Morada" />  
</message>
```

WSDL - Elementos

- * Elemento <portType>: Especifica a interface que agrupa um conjunto de operações abstratas disponibilizadas por um Web Service;
 - * Tag com funcionamento semelhante ao conceito de interfaces do CORBA;
- * Elemento <operation>: Representa uma descrição abstrata de uma operação suportada por um Web Service;
 - * Comparado a um método em Java;
 - * Identificada pelo atributo “name”;
 - * Define uma mensagem de entrada pelo subelemento <input>;
 - * Define uma mensagem de saída pelo subelemento <output>;
 - * Esses dois elementos <input> e <output> referenciam elementos do tipo <message> definidos no mesmo documento WSDL.

WSDL - Elementos

- * Exemplo de <portType>:

```
<portType name="myPortTypeID29">  
  <operation name="obtemEndereco" .... />  
  </operation>  
  <operation name="obtemNome" .... />  
  </operation>  
</portType>
```

WSDL - Elementos

- * O WSDL tem 4 primitivas de invocação de operações:
 - * One-way operation;
 - * Request-response operation;
 - * Solicite-response operation; e
 - * Notification operation.

TIPO	DEFINIÇÃO
<i>One-way</i>	A operação recebe uma mensagem mas não retorna nenhuma mensagem.
<i>Request-response</i>	A operação recebe uma mensagem e devolve uma resposta.
<i>Solicit-response</i>	A operação envia uma mensagem e irá esperar por uma resposta.
<i>Notification</i>	A operação envia uma mensagem mas não esperará por uma resposta.

Tabela 8.3 – As primitivas de invocação de operações

WSDL - Elementos

* Exemplo de <operation>:

```
<message name= "Identificacao">  
  <part name= "nome" type= "xs: string"/>  
  <part name= "morada" type= "xs: string" />  
</message>  
  
<portType name="registro">  
  <operation name=" atualizar">  
    <input name="ID" message= "Identificacao"/>  
  </operation>  
</portType>
```

WSDL - Exercícios

- * Entre no site abaixo e leia o material:

<http://www.ibm.com/developerworks/br/library/j-jws20/>