

Pontifícia Universidade Católica do Rio Grande do Sul  
Faculdade de Informática  
Programa de Pós-Graduação em Ciência da Computação



# Web Services

– Conceitual

Juliano Moraes,  
Marcus Breda,  
Paulo Gil,  
Rafael Medaglia

PROGRAMA DE PÓS-GRADUAÇÃO  
**PPGCC**  
EM CIÊNCIA DA COMPUTAÇÃO

---

---

---

---


---

---

---

---

## Tópicos



- Introdução
- CONTEXTO HISTÓRICO
- WEB SERVICES
- Conclusões

---

---

---

---


---

---

---

---

## Introdução (1/3)



- Web Services disponibilizam uma maneira de diferentes tipos de aplicação, possivelmente rodando em diversas plataformas e sistemas operacionais interagem.
- Originalmente foi criado como uma interface padronizada baseada em tecnologias já conhecidas, basicamente a especificação XML e o protocolo HTTP.

---

---

---

---

---

---

---

---

## Introdução (2/3)



- Essa padronização foi desenvolvida pela W3C, um consórcio de empresas de tecnologia que tem como o objetivo criar padrões comuns para conteúdo da Web, apoiada por grandes empresas como a Microsoft, IBM, HP entre outras.
- A definição de Web Services de acordo com a W3C diz que é um sistema de software responsável por proporcionar a interação entre duas máquinas através de uma rede

---

---

---

---

---

---

---

---

## Introdução (3/3)



- Para possibilitar essa interação uma interface descrita em um formato específico, WSDL (Web Services Description Language), permite que sistemas interajam com um Web Service usando essa interface e enviando mensagens SOAP (Simple Object Access Protocol) ou utilizando outros protocolos. As mensagens SOAP basicamente são documentos XML serializados seguindo o padrão W3C enviados em cima de um protocolo de rede.

---

---

---

---

---

---

---

---

## Tópicos



- Introdução
- CONTEXTO HISTÓRICO
- WEB SERVICES
- Conclusões

---

---

---

---

---

---

---

---

## Origens



- Por volta de 1999, surgiu a necessidade de se padronizar a comunicações entre diferente plataformas (PC, Mainframe, Mac, Windows, Linux entre outros) e linguagens de programação (PHP, C#, Java, etc).
- Diversos padrões foram propostos, DCOM e CORBA mas nenhum obteve êxito suficiente, tanto pela independência de plataforma como pelo modelo proposto.

---

---

---

---

---

---

---

---

## Origens



Finalmente surgiram os Web Services para a comunicação entre as diferentes plataformas, inclusive atualmente muitos sistemas legado estão sendo integrados aos novos sistemas através de Web Services.

---

---

---

---

---

---

---

---

## Tópicos



- Introdução
- CONTEXTO HISTÓRICO
- WEB SERVICES
- Conclusões

---

---

---

---

---

---

---

---

## Web Services

- Um Web Service é uma noção abstrata que deve ser implementada por um agente concreto. O agente por sua vez é um pedaço de software ou hardware que envia e recebe mensagens.
- Podemos entender que o Web Service é uma interface que precisa ser implementada por uma aplicação independente de linguagem e de sistema operacional para viabilizar a troca de informações entre aplicações.

---

---

---

---

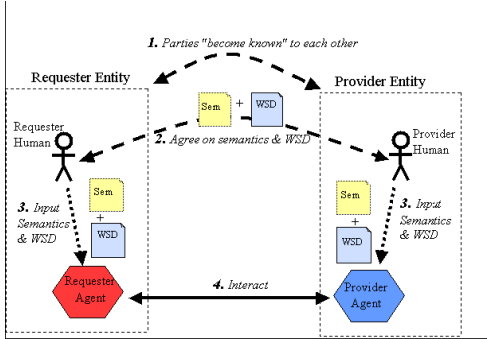
---

---

---

---

## Diagrama de Alto nível




---

---

---

---

---

---

---

---

## Frameworks

Nome	Plataforma	Tipo	Utilizações	Protocolos
Apache Axis	Java	Cliente/ Servidor	WS-ReliableMessaging, WS-Coordination, WS-Security, WS-AtomicTransaction WS-Addressing	SOAP, WSDL
Java Web Services Development Pack	Java	Cliente/ Servidor	WS-Addressing	SOAP, WSDL
Windows Communication Foundation	.NET	Cliente/ Servidor	WS-Addressing, WS-ReliableMessaging, WS-Security	SOAP, WSDL
gSOAP	C++	Cliente/ Servidor	WS-Addressing, WS-Discovery, WS-Enumeration, WS-Security	SOAP, XMLRPC, WSDL

---

---

---

---

---

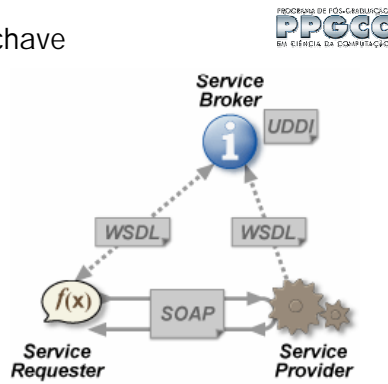
---

---

---

## Protocolos chave

- SOAP
- UDDI
- WSDL



---

---

---

---

---

---

---

---

## WSDL

- O WSDL é uma linguagem para descrever Web Services, é como um índice dos métodos disponíveis em um certo Web Service. Qualquer aplicação pode requisitar esse índice para saber como deve enviar as suas requisições. Independente de tecnologias bastando essas terem acordado quanto a descrição do serviço.

---

---

---

---

---

---

---

---

## WSDL - componentes

- Tipos de dados: Denominados de tipos <types>.
- Parâmetros de entrada e saída de um serviço: Denominados de mensagem <message>.
- O relacionamento entre parâmetros de entrada e saída: Assinatura do método, denominada de operações <operation>.
- Agrupamento lógico de operações <portType>.
- O protocolo a ser usado para acessar os métodos de um objeto: Denominado de vínculo, define o protocolo a ser usado para acessar os métodos de um objeto (SOAP, HTTP ou MIME).
- Endereço do serviço. Além dos componentes acima, define um serviço.

---

---

---

---

---

---

---

---

## WSDL - componentes



- Os elementos <portType> possuem um conjunto de operações <operation> que são classificadas em quatro tipos:
- Operação unidirecional: define uma mensagem enviada de um cliente para um serviço (sem resposta)
- Solicitação-Resposta: o cliente envia uma solicitação a um serviço e recebe como resultado uma mensagem de resposta (ex RPC)
- Pedido-Resposta: define uma mensagem enviada de um serviço para o cliente, resultando em uma mensagem enviada do cliente de volta para o serviço.
- Notificação: Quando o serviço envia uma mensagem para o cliente.

---

---

---

---

---

---

---

---

## WSDL – Exemplo 1/4



```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<definitions xmlns:SOAP:ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP:
  ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="urn:server.hello"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="urn:server.hello" >
<types>
<xsd:schema targetNamespace="urn:server.hello" >
<xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/"
/>
<xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
</xsd:schema>
</types>
```

---

---

---

---

---

---

---

---

## WSDL – Exemplo 2/4



```
<message name="helloRequest" >
<part name="name" type="xsd:string" />
</message>
<message name="helloResponse" >
<part name="return" type="xsd:string" />
</message>
<portType name="server.helloPortType" >
<operation name="hello" >
<documentation>Retorna o nome</documentation>
<input message="tns:helloRequest" />
<output message="tns:helloResponse" />
</operation>
</portType>
```

---

---

---

---

---

---

---

---

## WSDL – Exemplo 3/4



```
<binding name="server.helloBinding" type="tns:server.helloPortType" />
<soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
<operation name="hello">
  <soap:operation soapAction="urn:server.hello#hello" style="rpc" />
</input>
  <soap:body use="encoded" namespace="urn:server.hello"
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
  <soap:body use="encoded" namespace="urn:server.hello"
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
</binding>
```

---

---

---

---

---

---

---

---

---

---

## WSDL – Exemplo 4/4



```
<service name="server.hello">
  <port name="server.helloPort" binding="tns:server.helloBinding">
    <soap:address location="http://localhost/nuSOAP/server2.php" />
  </port>
</service>
</definitions>
```

---

---

---

---

---

---

---

---

---

---

## SOAP



(Simple Object Access Protocol)

- SOAP é um protocolo para intercâmbio de mensagens entre programas de computador.

Concepção:

- Simplicidade,
- independente do fornecedor,
- independente da linguagem,
- independente do modelo de objectos,
- independente do transporte.

---

---

---

---

---

---

---

---

---

---

## SOAP - Exemplo



- Mensagem SOAP enviada pelo cliente solicitando os detalhes do produto com ID=827635

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetails
xmlns="http://warehouse.example.com/ws">
      <productID>827635</productID>
    </getProductDetails>
  </soap:Body>
</soap:Envelope>
```

---

---

---

---

---

---

---

---

## SOAP - Exemplo



- Essa é a possível resposta:

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetailsResponse
xmlns="http://warehouse.example.com/ws">
      <getProductDetailsResult>
        <productName>Toptimate 3-Piece Set</productName>
        <productID>827635</productID>
        <description>3-Piece luggage set. Black
Polyester.</description>
        <price currency="NIS">96.50</price>
        <inStock>true</inStock>
      </getProductDetailsResult>
    </getProductDetailsResponse>
  </soap:Body>
</soap:Envelope>
```

---

---

---

---

---

---

---

---

## UDDI



(Universal Descriptor, Discovery and Integration)

- Um serviço de registro UDDI é um Web Service que gerencia informação sobre provedores, implementações e metadados de serviços. Provedores de serviços podem utilizar UDDI para publicar os serviços que eles oferecem. Usuários de serviços podem usar UDDI para descobrir serviços que lhes interessem e obter os metadados necessários para utilizar esses serviços.

---

---

---

---

---

---

---

---



## Tópicos



- Introdução
- CONTEXTO HISTÓRICO
- WEB SERVICES
- Conclusões

---

---

---

---

---

---

---

---

## Conclusões



- A principal vantagem do uso de Web Services deve-se a sua Arquitetura Orientada a Serviços (SOA).
- Em uma SOA a ligação consiste na troca de mensagens que permitem estabelecer vários controles possibilitando restrições para o envio/recebimento de informações e separando o código da assinatura do serviço.
- Outra grande vantagem dos Web Services é a facilidade e flexibilidade de implementação entre diferentes tecnologias.

---

---

---

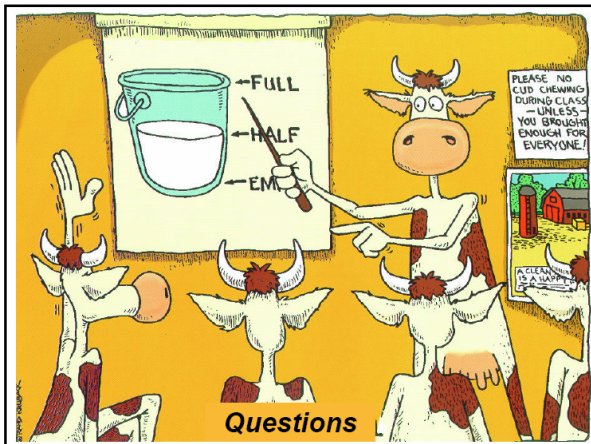
---

---

---

---

---



---

---

---

---

---

---

---

---