


---

## Web Services

Roberto Gómez Cárdenas  
rogomez@itesm.mx  
<http://homepage.cem.itesm.mx/rogomez>

Lámina 1 Roberto Gómez C.




---

## Web Services

- Servicios web.
- Interfaz red a una aplicación basada en tecnologías internet estandarizadas.
- Aplicaciones auto-contenidas, auto-descritas que pueden ser publicadas, localizadas e invocadas a través de la Web.
- Una vez desarrolladas, otras aplicaciones (y otros servicios Web) pueden descubrirlas e invocar el servicio dado.
- Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios web.

Lámina 2 Roberto Gómez C.



## Factores que influyeron para su surgimiento

---

- Computación distribuida: RPC, CORBA, RMI, DCOM
  - Sistemas fuertemente acoplados
- Integración de aplicaciones EAI
  - Enterprise Application Integration
  - Uso de software y principios de arquitectura de sistemas para integrar un conjunto de aplicaciones
- Aparición de XML
- Necesidad de intercambios B2B
- Comercio electrónico y burbuja internet
- Microsoft vs Java
  - Compatibilidad

Lámina 3
Roberto Gómez C.



## Un poco de historia

---



**1980**

- Protocolos de comunicación no eran objeto de interés de los desarrolladores
- Era suficiente realizar aplicaciones que se comunicaran entre sí en la misma máquina

**1990**

- Alcanzaron popularidad **COM** (Component Object Model) de **Microsoft** y **CORBA** (Common Object Request Broker Architecture) de **OMG** (Object Management Group)
- Modelos para escribir y encapsulan código binario
- No son fácilmente interoperables

**1999**

- Generalización de las redes locales
- **OMG** estable **IOP** (Internet Inter-ORB Protocol) para **CORBA**
- **Microsoft** creó **DCOM** (Distributed COM)
- **Sun Microsystems** lanzó **RMI** (Remote Method Interface)
- Se realizan llamadas bajo la forma **RPC** (Remote Procedure Call)

**Alrededor del año 2000**

- Surgen los **Servicios Web**
- Comunicación de aplicación a aplicación desde cualquier máquina sin importar el sistema operativo ni el lenguaje de programación
- Modelos de objetos distribuidos utilizando los estándares de Internet

Lámina 4
Roberto Gómez C.



## Elementos de un servicio web

---


- El Servicio.
- El proveedor de Servicios.
- El solicitante de Servicios.
- El registro de Servicios.



The diagram illustrates the components of a web service. At the center is 'XML Web Services'. To the top, 'Smart Clients' (represented by a laptop and a mobile phone) connect to the services. To the right, 'Developer Tools' (represented by a folder with a wrench) are used to create services. At the bottom, 'Servers' (represented by server racks) host the services.

Lámina 5

Roberto Gómez C.



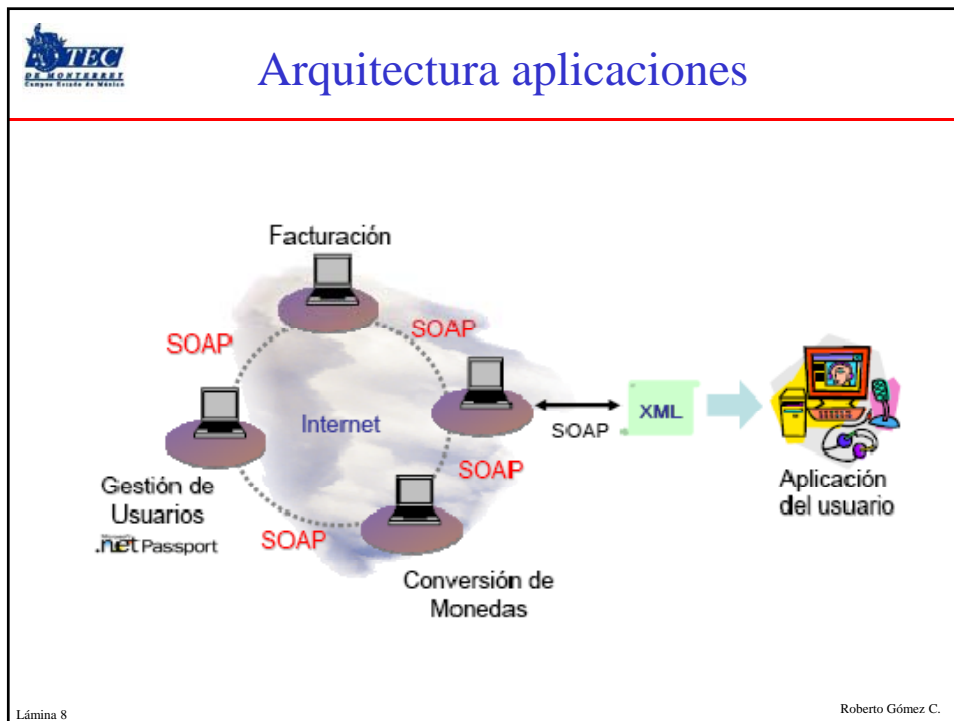
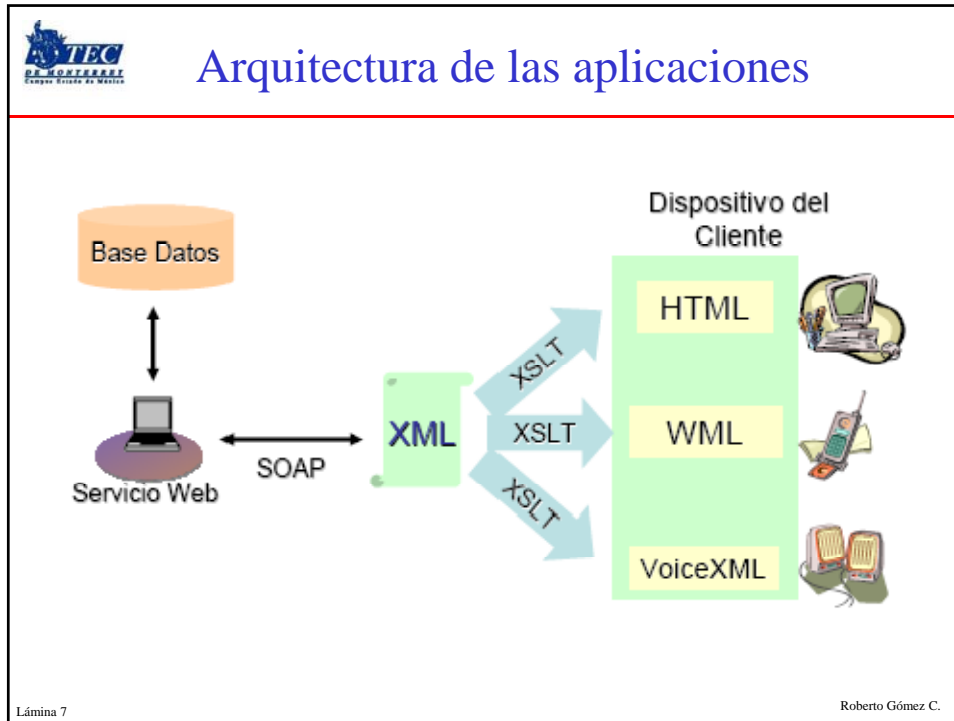
## El stack de tecnologías de Web Services


---

Stack de tecnología	Elecciones comunes
Descubrimiento	U D D I
Descripción	W S D L
Empaquetamiento	S O A P
Transporte	H T T P (Web)
Red	T C P / I P (Internet)

Lámina 6

Roberto Gómez C.





## Descubrimiento

- Para que un programa use un Servicio Web primero debe encontrarlo.
- Existe una red de servidores que proporcionan un directorio global de servicios Web: Universal Description Discovery and Integration (UDDI)
- Es el estándar de descubrimientos de servicios Web.

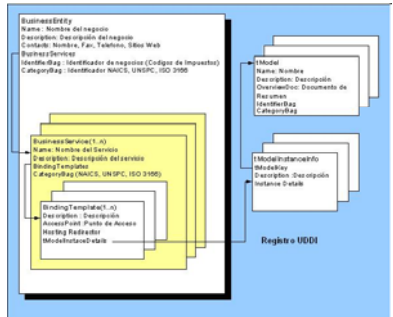
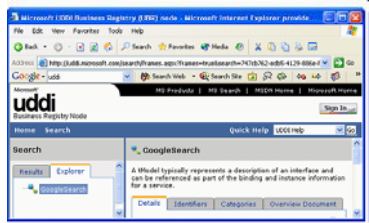




Lámina 9 Roberto Gómez C.



## Descripción

- Antes de crear un directorio de servicios web, es necesario poder describir un servicio.
- El Web Service Description Language (WSDL) es un lenguaje basado en XML que describe las funcionalidades que un servicio web ofrece.
- Una vez que un programa cuenta con la descripción WSDL de un servicio web, cuenta con todo lo necesario para usarlo.

WSDL

Data Types:  
<wsdl:types/>

Messages:  
<wsdl:message/>

Interfaces:  
<wsdl:portType/>

Services:  
<wsdl:binding/>  
<wsdl:service/>

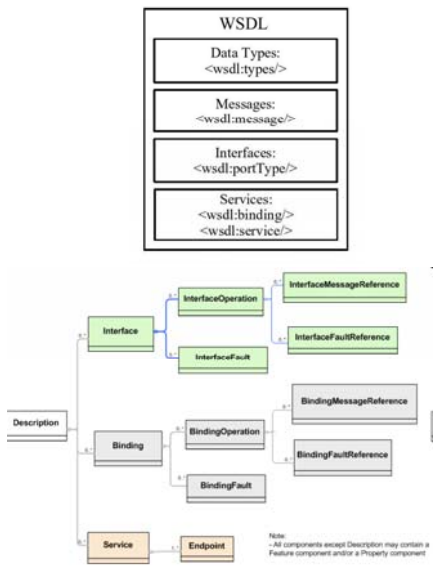


Lámina 10 Roberto Gómez C.




## Esquema WDSL

---

<pre style="background-color: #ffff00; padding: 5px;"> &lt;?xml version="1.0"&gt; &lt;definitions&gt;   &lt;types&gt;     ...   &lt;/types&gt;    &lt;message&gt;     ...   &lt;/message&gt;    &lt;portType&gt;     ...   &lt;/portType&gt;    &lt;binding&gt;     ...   &lt;/binding&gt; &lt;/definitions&gt;</pre>	<p>Comienzo del documento. Esta etiqueta agrupa a todas las demás.</p> <p style="color: red;">Se definen los tipos usados en el Web Service</p> <p style="color: green;">Se definen los métodos, parámetros para realizar la operación. Cada message es un método que implementa el envío o recepción de un mensaje y puede consistir en una o más partes (parámetros). NO se define el servicio a proporcionar.</p> <p style="color: purple;">Esta sección es la más importante, ya que se definen las operaciones (el SERVICIO) que pueden ser realizadas, y los mensajes que involucran (por ejemplo el mensaje de petición y el de respuesta).</p> <p style="color: brown;">Se definen el formato del mensaje y detalles del protocolo para cada portType.</p> <p style="color: blue;">Fin del documento WSDL</p>
---	---

Lámina 11
Roberto Gómez C.



## Ejemplo (definiciones)

---

```

<?xml version="1.0"?>
<definitions name="StockQuote"
  targetNamespace="http://example.com/stockquote.wsdl"
  xmlns:tns="http://example.com/stockquote.wsdl"
  xmlns:xsd1="http://example.com/stockquote.xsd"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
```

Lámina 12
Roberto Gómez C.



## Ejemplo (tipos)

```

<types>
  <schema targetNamespace="http://example.com/stockquote.xsd"
    xmlns="http://www.w3.org/2000/10/XMLSchema">
    <element name="TradePriceRequest">
      <complexType>
        <all>
          <element name="tickerSymbol" type="string"/>
        </all>
      </complexType>
    </element>
    <element name="TradePrice">
      <complexType>
        <all>
          <element name="price" type="float"/>
        </all>
      </complexType>
    </element>
  </schema>
</types>

```

Lámina 13

Roberto Gómez C.



## Ejemplo (mensajes y puertos)

```

<message name="GetLastTradePriceInput">
  <part name="body" element="xsd1:TradePriceRequest"/>
</message>


<message name="GetLastTradePriceOutput">
  <part name="body" element="xsd1:TradePrice"/>
</message>

<portType name="StockQuotePortType">
  <operation name="GetLastTradePrice">
    <input message="tns:GetLastTradePriceInput"/>
    <output message="tns:GetLastTradePriceOutput"/>
  </operation>
</portType>

```

Lámina 14

Roberto Gómez C.



## Ejemplo (binding)

---

```


<binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="GetLastTradePrice">
    <soap:operation soapAction="http://example.com/GetLastTradePrice"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>

<service name="StockQuoteService">
  <documentation>My first service</documentation>
  <port name="StockQuotePort" binding="tns:StockQuoteSoapBinding">
    <soap:address location="http://example.com/stockquote"/>
  </port>
</service>

</definitions>

```

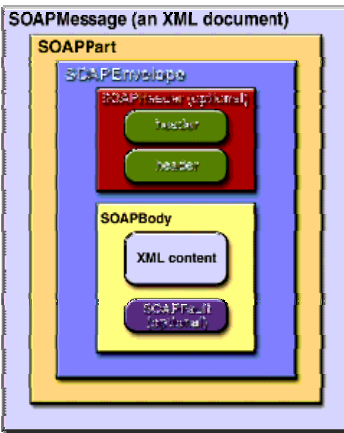
Lámina 15
Roberto Gómez C.



## Empaquetamiento

---


- Ya que los servicios web deben permitir acceso de cualquier programa escrito en cualquier lenguaje sobre cualquier sistema operativo, las peticiones y respuestas deben contar con un estándar independiente de la plataforma.
- El Simple Object Access Protocol (SOAP) es un lenguaje basado en XML que permite empaquetar llamadas de funciones y documentos como peticiones de servicios Web.



The diagram illustrates the structure of a SOAPMessage (an XML document). It is contained within a SOAPPart (yellow box). Inside the SOAPPart is a SOAPEnvelope (blue box). The SOAPEnvelope contains a SOAPHeader (red box) with 'header' and 'fault' elements, and a SOAPBody (yellow box) with 'XML content' and 'SOAPFault (optional)' elements.

Lámina 16
Roberto Gómez C.





## Estructura básica del protocolo

---

Comienza el envelope, (sobre) del mensaje

```

<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
  Soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  <soap:Header>
    ...
  </soap:Header>
  <soap:Body>
    ...
    ...
    ...
    <soap:Fault>
      ...
    </soap:Fault>
  </soap:Body>
</soap:Envelope>

```


Indica el comienzo del Header (encabezado). En esta sección se incluye información específica del mensaje, como puede ser la autenticación.

Aquí comienza el cuerpo del mensaje, en esta sección se incorpora toda la información necesaria para el nodo final. Por ejemplo, los parámetros para la ejecución, o la respuesta a una petición..

Cualquier tipo de fallo que se produzca será notificado en esta sección. La cual esta contenida dentro del cuerpo del mensaje.

Fin del mensaje SOAP

Lámina 17
Roberto Gómez C.



## Transporte

---

- Una vez que la petición ha sido empaquetada (por ejemplo como un mensaje SOAP) es necesario enviarla como un servicio web.
- Por su lado, el servicio web necesita una forma de enviar su respuesta.
- Servicios web es flexible con respecto a que se puede usar cualquier protocolo.
  - Nada nos impide utilizar SOAP sobre algún otro protocolo de Internet (SMTP, FTP, etc.).
  - Sin embargo HTTP es el más común, usado por ser un protocolo ampliamente difundido y que se encuentra menos restringido por firewalls

Lámina 18
Roberto Gómez C.



## Red

- Para enviar y recibir respuestas entre computadoras, se debe contar con una red de cierto tipo.
- En la mayor parte de los casos, Internet es dicha red.
- Esto abarca a algunas redes internas basadas en TCP/IP.

Lámina 19

Roberto Gómez C.



## Pasos servicios web

- El programa cliente construye la petición dentro de un mensaje SOAP.
- El mensaje SOAP es enviado al servicio Web como el cuerpo de una petición HTTP POST.
- El servicio Web desempaqueta la petición SOAP y la convierte en un comando que la aplicación puede entender.
- El servicio Web empaqueta la respuesta en otro mensaje SOAP, el cual se envía de regreso al cliente en respuesta a su petición HTTP.
- El cliente desempaqueta el mensaje SOAP para obtener el resultado de lo solicitado.

Lámina 20

Roberto Gómez C.




## ¿Quién ofrece Web Services?

---

- Google y Amazon han conducido experimentos con servicios Web.
- ¿Por qué?
  - Para permitir a sus socios de negocios desarrollar interfaces de usuarios y aplicaciones que usan datos y servicios de Google y Amazon.
- Se pueden bajar sus APIs de
  - <http://www.google.com/apis>
  - <http://www.amazon.com/webservices>

Lámina 21
Roberto Gómez C.



## Otras opciones

---

- Sistemas de información geográfica son candidatos perfectos para servicios web.
  - La Open Geospatial Consortium define varios estándares relevantes
    - Geographical Markup Language (GML) intercambia información.
    - Weab Feature Service trabaja con datos tipo GML
    - Web Map Service crea mapas (imágenes)
- Xmethods
  - <http://www.xmethods.com>
- Web Service List
  - <http://www.webservicelist.com/>
- Web ServiceX.Net
  - <http://www.webserviceX.net/WCF>

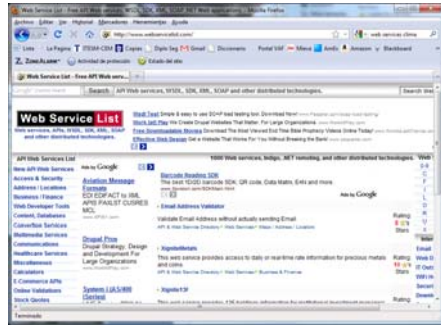


Lámina 22
Roberto Gómez C.



## Herramientas desarrollo Web Services

- Existen toolkits para diferentes lenguajes de programación.
  - C++
  - Python
  - Perl
  - PHP
  - Varios kits .NET
  - Apache AXIS

Lámina 23

Roberto Gómez C.

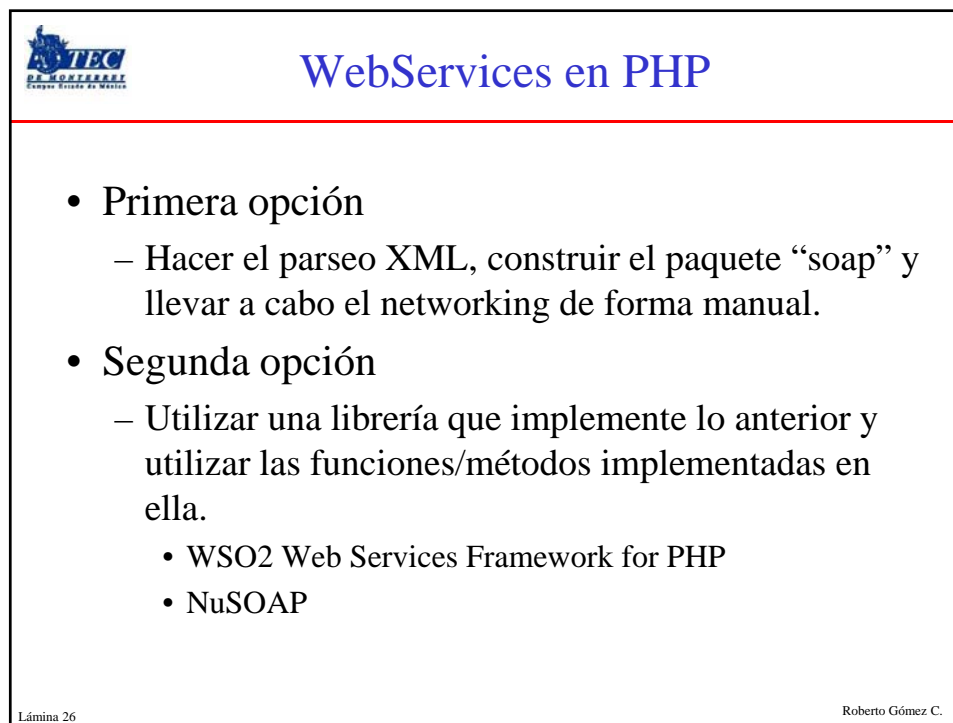
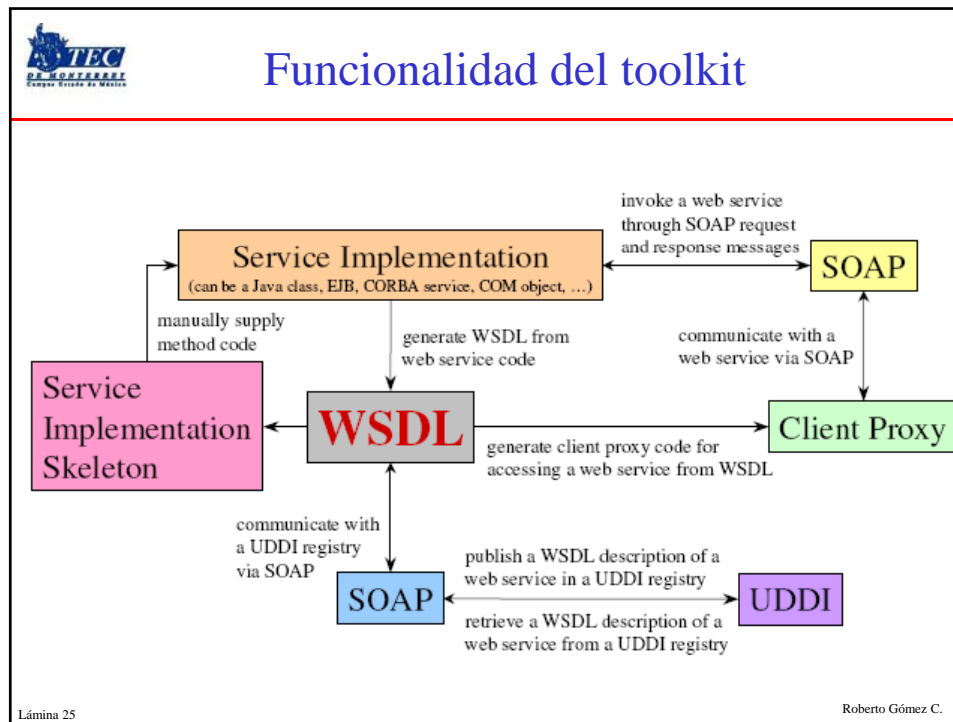


## Apache AXIS

- Es un toolkit para convertir aplicaciones Java en Servicios Web.
- Las herramientas ofrecidas permiten publicar un servicio en un servidor de aplicaciones en particular.
- Permiten convertir WSDL en stubs de cliente.
- Herramientas de tiempo real de Axis acepta peticiones SOAP y las redirige al servicio apropiado.

Lámina 24

Roberto Gómez C.





## NuSOAP

- Re-escritura de SOAPx.
- Desarrollado por NuSphere y Dietrich Ayala.
- Orientado a objetos
  - Conjunto de clases PHP, no requiere de extensiones.
- Permite a un desarrollador crear y consumir servicios web basados en SOAP.
- Su uso es simple y eficiente.
- <http://sourceforge.net/projects/nusoap/>

Lámina 27

Roberto Gómez C.



## Ejemplo uso web service (1/2)

```
<?php

ini_set('soap.wsdl_cache_enabled', '0');
require_once('lib/nusoap.php');

$cliente = new soapclient
('http://www.webservice.net/convertMetricWeight.asmx?wsdl','wsdl','','');


if ($sError = $cliente->getError()) {
    echo "No se pudo realizar la operaciÃ³n [" . $sError . "];"
}

$n=10;
$p1='centigram';
$p2='kilogram';

$params= array( 'MetricWeightValue' => '10',
                'fromMetricWeightUnit' => 'centigram',
                'toMetricWeightUnit' => 'kilogram');
```

Lámina 28

Roberto Gómez C.



## Ejemplo uso web service (2/2)


```

echo "Datos a enviar: 10 centigram, kilogram <br>";
$resp = $cliente->call("ChangeMetricWeightUnit",$params,"",false);

// Existe alguna falla en el servicio?
if ($cliente->fault)
{
    echo 'Error al llamar el metodo <br>'.$cliente->getError();
}
else
{
    echo "<br>Respuesta: <br>";
    print_r($resp);
}
?>

```

Lámina 29 Roberto Gómez C.



## Desarrollo Web Service en PHP

```

<?php
function area($x, $y)
{
    return $x * $y;
}
require_once('lib/nusoap.php');
$server = new nusoap_server();
$ns="http://localhost/WebSer/area";
$server->configurewsdl('ApplicationServices',$ns);
$server->wsdl->schematargetnamespace=$ns;
$server->register('area',
    array('largo' => 'xsd:int', 'ancho' => 'xsd:int'),
    array('return' => 'xsd:int'),
    $ns );

$HTTP_RAW_POST_DATA = isset($HTTP_RAW_POST_DATA) ?
    $HTTP_RAW_POST_DATA : "";
$server->service($HTTP_RAW_POST_DATA);
?>

```

Lámina 30 Roberto Gómez C.



## Llamando el Web Service en PHP

```
<?php
    echo "Ejemplo<br>";
    require_once('lib/nusoap.php');
    $cliente = new nusoap_client('http://localhost/WebSer/area.php');
    $res = $cliente->call('area', array('largo' => '3','ancho' => '4'));

    if ($cliente->fault)
    {
        echo "Error al llamar el metodo<br>". $res->getError();
    }
    else
    {
        echo "<br>El area es de: <br>";
        print_r($res);
    }
?>
```

Lámina 31

Roberto Gómez C.



## Web Services

Roberto Gómez Cárdenas  
rogomez@itesm.mx  
<http://homepage.cem.itesm.mx/rogomez>

Lámina 32

Roberto Gómez C.