



UNIVERSIDAD DE  
COSTA RICA

# Consumir Servicios Web SOAP

CI-2454 Servicios Web  
Prof. Braulio José Solano Rojas  
ECCI, UCR

# Tener un buen IDE es ideal

- La forma más productiva de consumir un Servicio Web SOAP es por medio de un Entorno Integrado de Desarrollo (IDE) que nos genere clases *proxy* o código fuente.
- Esto hace el trabajo más sencillo al programador. Evita el manejo manual del flujo XML y permite abstraer el servicio como una biblioteca o clase local de forma transparente.



.NET

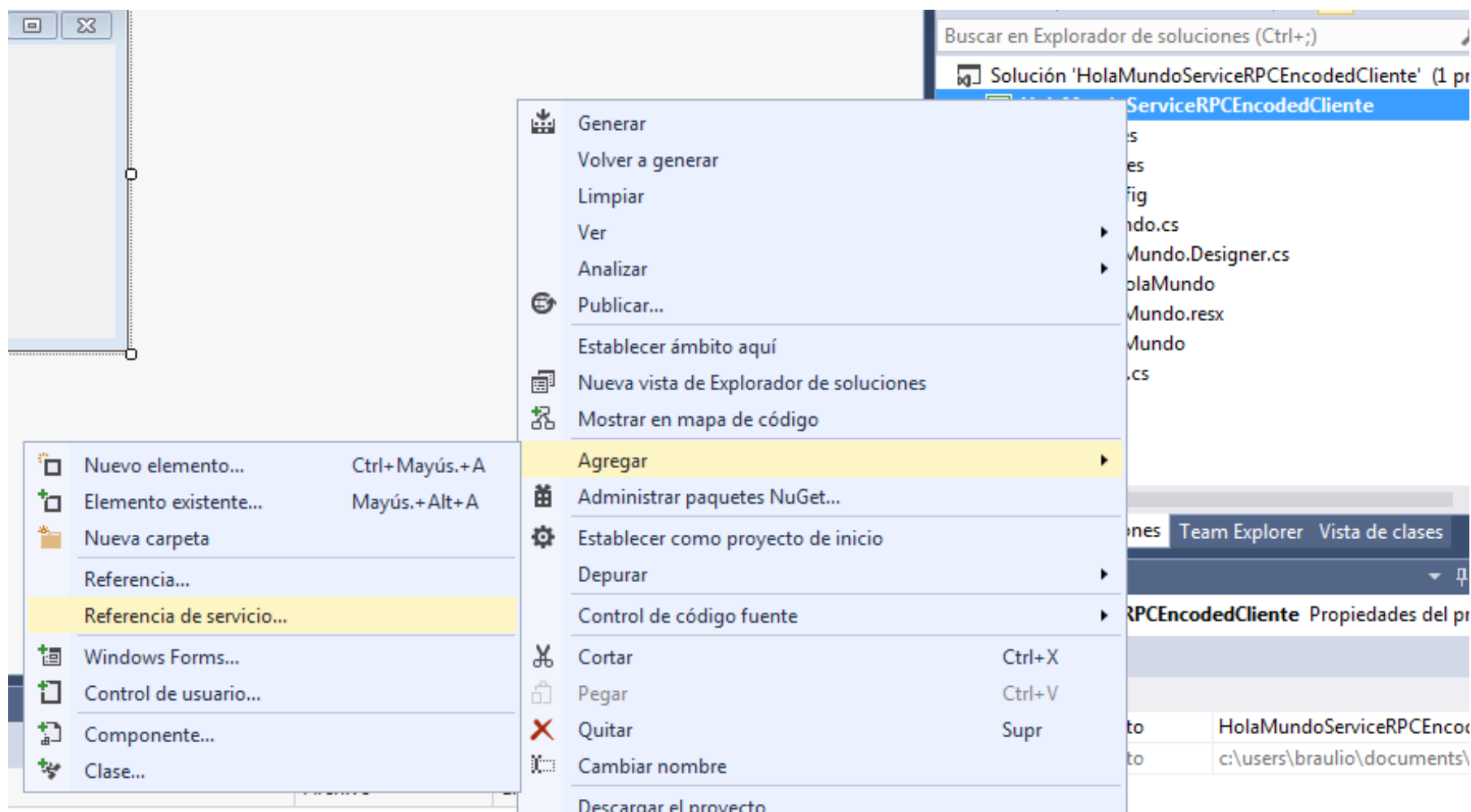


# Consumición en .NET

- En Microsoft Visual Studio, primero se debe crear una solución o proyecto de la naturaleza que se requiera. Es decir, puede ser una aplicación de ventanas, una aplicación web u otra.
- Luego en el momento que se requiera agregamos al solución o proyecto una referencia de servicio, tal como se muestra en la filmina siguiente. En versiones anteriores de .NET y antes de WCF (con ASMX) no se agrega una referencia a un servicio si no más bien una referencia web.



# Consumición en .NET

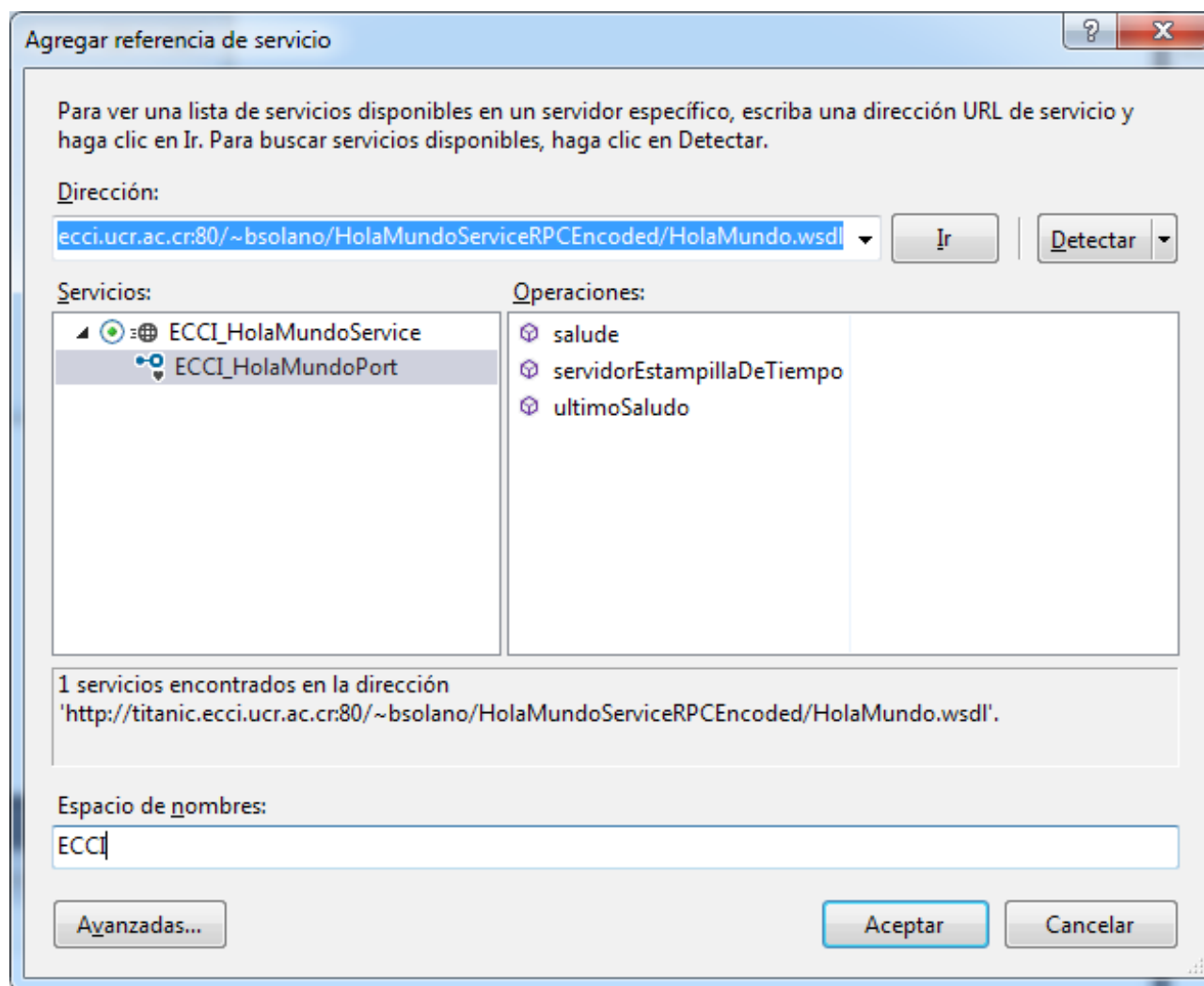


# Consumición en .NET

- Luego simplemente agregamos en el diálogo que se nos muestra la referencia web al WSDL.
- La referencia nos debería mostrar los métodos disponibles en el servicio.
- Además, debemos escoger un espacio de nombres adecuado para la clase con los métodos que va a generar el IDE.



# Consumición en .NET



# Consumición en .NET

- Una vez agregado el servicio Visual Studio genera código proxy que permite utilizar los métodos en el servicio de forma transparente.
- El ejemplo de código en la filmina siguiente muestra la utilización del servicio como un objeto local. Se abstrae lo remoto.





# Consumición en .NET

```
private void btnSaludar_Click(object sender, EventArgs e)
{
    ECCI.ECCI_HolaMundoPortClient holaMundo = new ECCI.ECCI_HolaMundoPortClient();
    MessageBox.Show(holaMundo.salude(txtNombre.Text));
}
}
```

string ECCI\_HolaMundoPortClient.salude(string nombre)



# Consumición con sesión en .NET

- Si es necesario sostener estado en el servidor existen varias posibilidades. Una de ellas es *cookies*. Siempre que el servidor utilice HTTP el cliente puede utilizar *cookies* para mantener dicha sesión. En ASMX (la forma de consumir un WSDL y su servicio asociado antes de WCF) dicho manejo se hacía a través de un objeto `CookieContainer`. En WCF dicho manejo se hace a través de configuración. Se modifica el archivo `App.config` de la solución o proyecto.



# Consumición con sesión en .NET

- Modificación en App.config:
  - Se agrega allowCookies="True" en el *binding*.

```
<system.serviceModel>
  <bindings>
    <basicHttpBinding>
      <binding name="ECCI_HolaMundoBinding" allowCookies="True" />
    </basicHttpBinding>
  </bindings>
</client>
```



# Consumición con Mono en Linux

- Mono es una alternativa de software libre que implementa el estándar de industria de .NET.
- MonoDevelop es un IDE para trabajar con dicha plataforma. Es posible consumir un Servicio Web con ese IDE de forma sencilla. Basta con “Agregar referencia web” a la solución o proyecto. Claramente utiliza la forma primera (ASMX) que utilizaba Visual Studio.



# Java

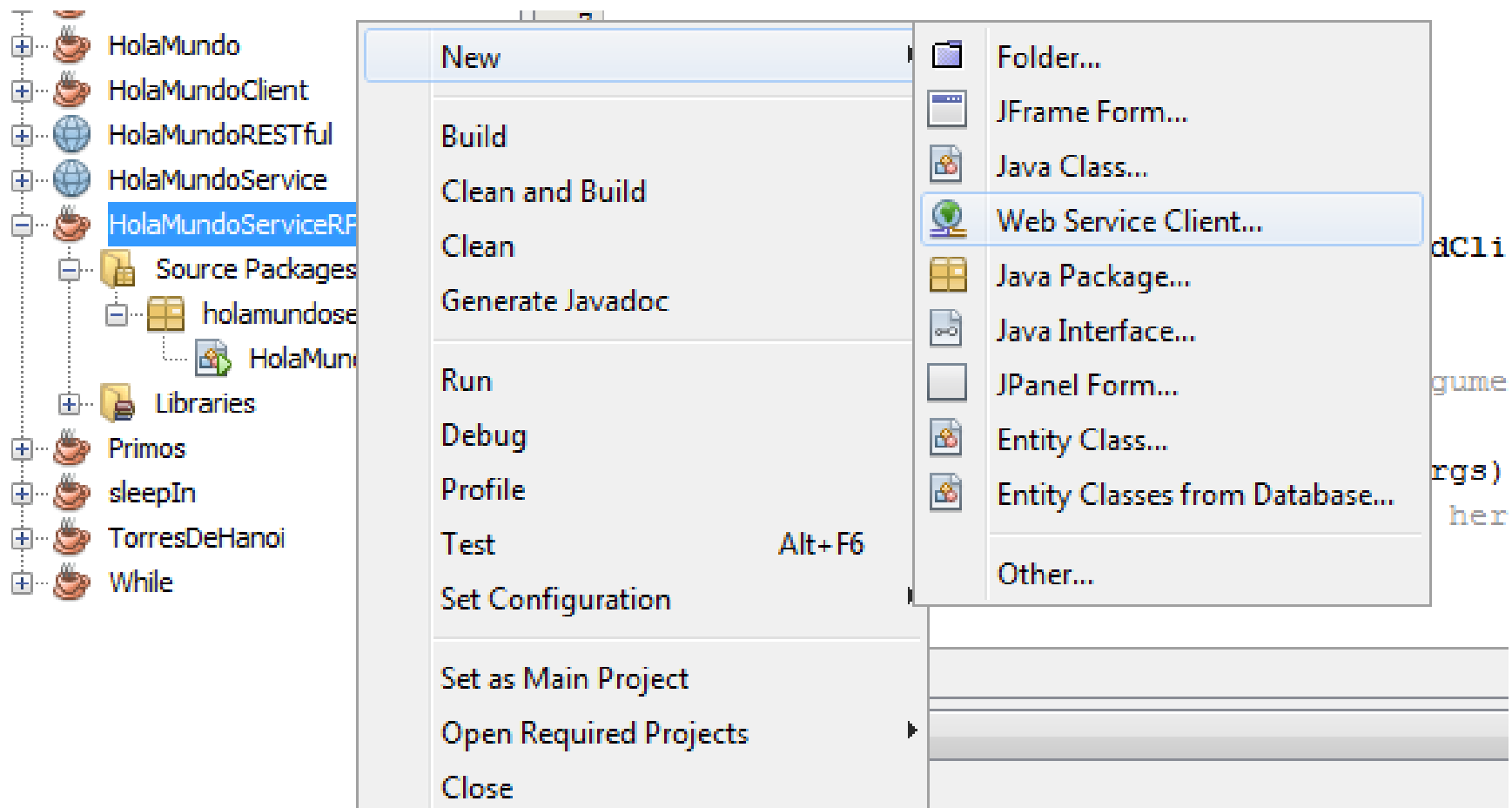


# Consumición en Java con Netbeans

- En primer lugar es necesario en Netbeans crear un proyecto Java que puede ser tan simple como una *Java Application*, más elaborado como u otro que se requiera.
- Luego a dicha aplicación cliente le agregamos un nuevo cliente de Servicio Web haciendo clic derecho en el proyecto, tal como se muestra en la filmina siguiente.



# Consumición en Java con Netbeans



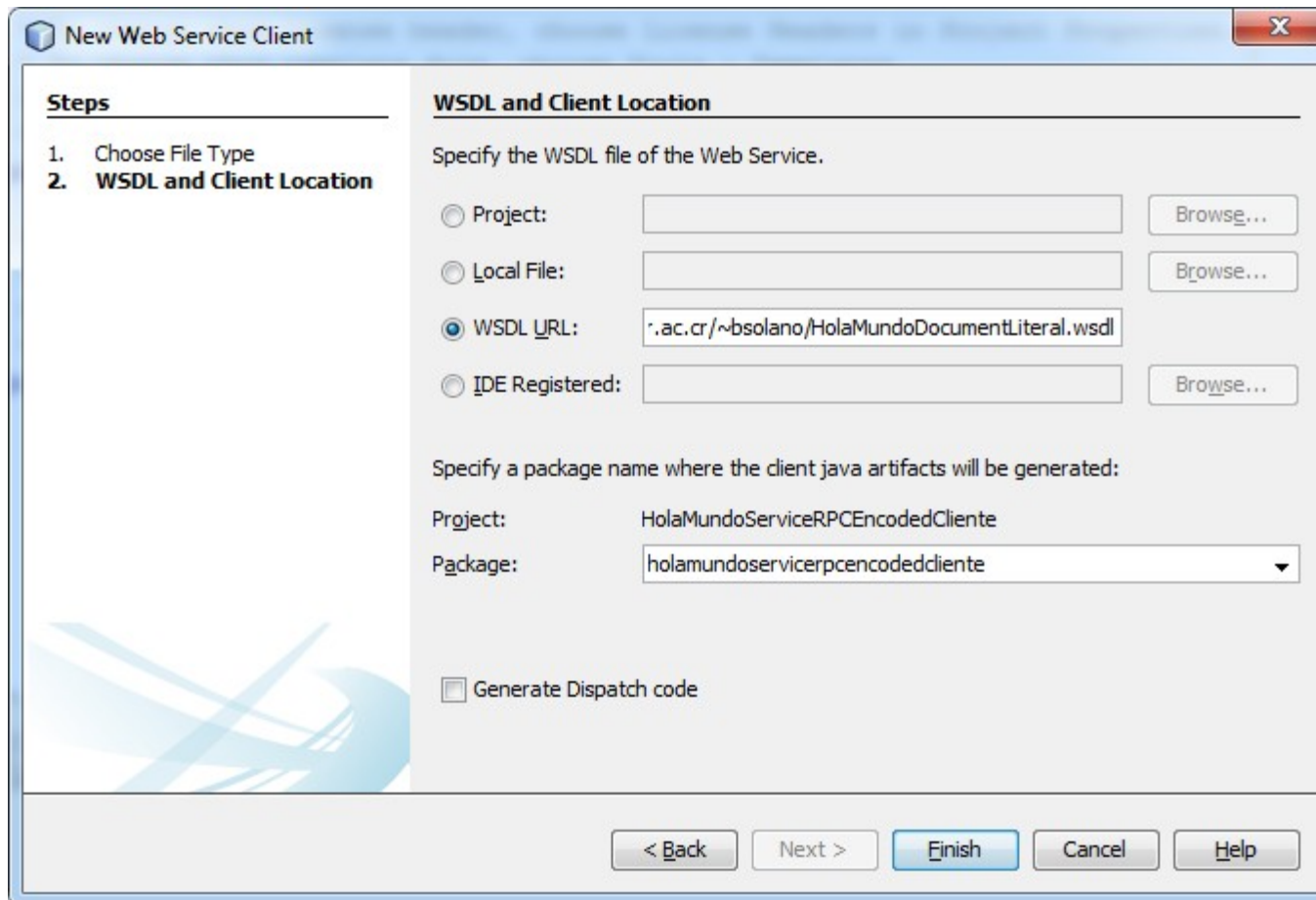
# Consumición en Java con Netbeans

- Netbeans muestra un diálogo para la creación de código cliente del servicio.
- Debemos indicar un WSDL y el nombre del paquete para el código a generar.





# Consumición en Java con Netbeans



The screenshot shows the 'New Web Service Client' dialog box in NetBeans. The 'Steps' pane on the left indicates the current step is '2. WSDL and Client Location'. The main area is titled 'WSDL and Client Location' and contains the following fields and options:

- Specify the WSDL file of the Web Service:**
  - ☐ Project: [Empty text box] [Browse...]
  - ☐ Local File: [Empty text box] [Browse...]
  - ☒ WSDL URL: [http://localhost:8080/HolaMundoDocumentLiteral.wsdl] [Browse...]
  - ☐ IDE Registered: [Empty text box] [Browse...]
- Specify a package name where the client java artifacts will be generated:**
  - Project: HolaMundoServiceRPCEncodedCliente
  - Package: holamundoservicrpcencodedcliente
- ☐ Generate Dispatch code

At the bottom, there are navigation buttons: '< Back', 'Next >', 'Finish' (highlighted in blue), 'Cancel', and 'Help'.



# Consumición en Java con Netbeans

- Para invocar al servicio podemos utilizar el código generado de forma transparente como si se tratase de una clase local.
- Podemos crear una variable privada en cualquiera de nuestras clases que sea una instancia del servicio.
- Luego en los métodos de la instancia podemos hacer uso del servicio.



# Consumición en Java con Netbeans

```
19 private ecci_holamundo.ECCIHolaMundo holaMundo;  
20 /**  
21  * Creates new form HolaMundoClientGUI  
22  */  
23 public HolaMundoClientGUI() {  
24     initComponents();  
25     ecci_holamundo.ECCIHolaMundo_Service service = new ecci_holamundo.ECCIHolaMundo_Service();  
26     holaMundo = service.getECCIHolaMundoPort();  
  
94 private void btnSaludarActionPerformed(java.awt.event.ActionEvent evt) {  
95     JOptionPane.showMessageDialog(this, holaMundo.salude(txtNombre.getText()));  
96 }
```



# Consumición en Java con sesión

- Al igual que con .NET, en Java, si es necesario sostener estado en el servidor existen varias posibilidades. Una de ellas es *cookies*. Siempre que el servidor utilice HTTP el cliente puede utilizar *cookies* para mantener dicha sesión.
- En Java cambiamos la configuración del *binding* por medio de código tal como se muestra en la filmina siguiente.



# Consumición en Java con sesión

```
27 | | ((BindingProvider) holaMundo).getRequestContext().put(BindingProvider.SESSION_MAINTAIN_PROPERTY, true);
```



# PHP



# Consumición en PHP

- PHP posee desde la versión cinco clases para la utilización de Servicios Web. Estas clases son limitadas por lo que algunos programadores las extienden por herencia o bien utilizan marcos de desarrollo que las extienden para ofrecer mayor funcionalidad.
- La clase para consumir Servicios Web en PHP es **SoapClient**. La siguiente filmina muestra un ejemplo de consumición con PHP.



# Consumición en PHP

```
39 $client = new SoapClient('http://titanic.ecci.ucr.ac.cr/~bsolano/HolaMundoServiceRPCEncoded/HolaMundo.wsdl',array(
40     "trace"      => 1,
41     "exceptions" => 0));
42
43 $client->salude("braulio");
44 print "<pre>\n";
45 print "Request :\n".htmlspecialchars($client->__getLastRequestHeaders()).htmlspecialchars($client->__getLastRequest())."\n";
46 print "Response :\n".htmlspecialchars($client->__getLastResponseHeaders()).htmlspecialchars($client->__getLastResponse())."\n";
47 print "</pre>";
48 $client->ultimoSaludo();
49 print "<pre>\n";
50 print "Request :\n".htmlspecialchars($client->__getLastRequestHeaders()).htmlspecialchars($client->__getLastRequest())."\n";
51 print "Response :\n".htmlspecialchars($client->__getLastResponseHeaders()).htmlspecialchars($client->__getLastResponse())."\n";
52 print "</pre>";
```





# ¡Gracias por su atención!

## ¿Preguntas?

