

# DESARROLLO DE APLICACIONES MOVILES I

## ANDROID – SEMANA 12



**Curso:** Desarrollo de aplicaciones móviles I  
Equipo de Profesores del Curso



# Logro y Temas de la Unidad III

## Logro de la Unidad de Aprendizaje

Al término de la unidad, el alumno elabora aplicaciones que se conectan a la web utilizando http y consumen servicios web.

## Temas

### Networking

- ✓ 3.1.1 Uso del protocolo HTTP para consumir web services.
- ✓ 3.1.2 Obtención de data de archivos XML.
- ✓ 3.1.3 Descarga de data binaria (imágenes).

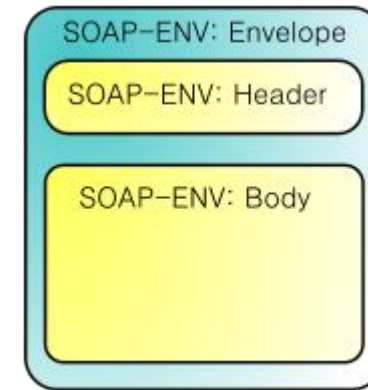


# Consumiendo Web services



- ✓ SOAP (Simple Object Access Protocol)

## Estructura de mensaje SOAP



- ✓ SOAP fue creado por Microsoft, IBM y otros. Está actualmente bajo el auspicio de la W3C. Es uno de los protocolos utilizados en los servicios Web.
- ✓ Archivos XML (Extensible Markup Language)

```
<?xml version="1.0" encoding=" UTF-8 " standalone=" yes "?>  
<ficha>  
  <nombre> Gabriel </nombre>  
  <apellido> Molina </apellido>  
  <direccion> Alfredo Vargas #36 </direccion>  
</ficha>
```

# Estructura Mensaje SOAP

- ✓ Un mensaje SOAP es un documento XML ordinario con una estructura definida en la especificación del protocolo.

- ✓ Envelope (obligatoria)
- ✓ Header
- ✓ Body (obligatoria)
- ✓ Fault

## ValidarUsuario

### Test

The test form is only available for requests from the local machine.

### SOAP 1.1

The following is a sample SOAP 1.1 request and response. The [placeholders](#) shown

```
POST /planificadorvirtual/LoginUsuario.aspx HTTP/1.1
Host: planificador.nisabusiness.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/ValidarUsuario"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ValidarUsuario xmlns="http://tempuri.org/">
      <pLoginUsuario>string</pLoginUsuario>
      <pContrasenia>string</pContrasenia>
    </ValidarUsuario>
  </soap:Body>
</soap:Envelope>
```

# Definición del Mensaje SOAP (WSDL)

- ✓ WSDL son las siglas de Web Services Description Language, un formato XML que se utiliza para describir servicios Web .
- ✓ WSDL se usa a menudo en combinación con SOAP y XML Schema.

## Estructura del WSDL

### Tipos de Datos

- ✓ `<types>`: Esta sección define los tipos de datos usados en los mensajes.

### Mensajes

- ✓ `<message>`: Aquí definimos los elementos de mensaje.

### Tipos de Puerto

- ✓ `<portType>`: Con este apartado definimos las operaciones permitidas y los mensajes intercambiados en el Servicio.

### Bindings

- ✓ `<binding>`: Especificamos los protocolos de comunicación usados.

### Servicios

- ✓ `<service>`: Conjunto de puertos y dirección de los mismos.



# Ejemplo (WSDL)

## Estructura del WSDL

### Tipos de Datos (<types>)

### Mensajes (<message>)

### Tipos de Puerto (<portType>)

### Bindings

### Servicios

```
<wsdl:definitions xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://tempuri.org/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://tempuri.org/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
      <s:element name="ValidarUsuario">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="pLoginUsuario" type="s:string"/>
            <s:element minOccurs="0" maxOccurs="1" name="pContraseña" type="s:string"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="ValidarUsuarioResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="ValidarUsuarioResult" type="s:string"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="ValidarUsuarioSoapIn">
    <wsdl:part name="parameters" element="tns:ValidarUsuario"/>
  </wsdl:message>
  <wsdl:message name="ValidarUsuarioSoapOut">
    <wsdl:part name="parameters" element="tns:ValidarUsuarioResponse"/>
  </wsdl:message>
  <wsdl:portType name="LoginUsuarioSoap">
    <wsdl:operation name="ValidarUsuario">
      <wsdl:input message="tns:ValidarUsuarioSoapIn"/>
      <wsdl:output message="tns:ValidarUsuarioSoapOut"/>
    </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>
```



# Ejemplo (WSDL) continuación...

Estructura del WSDL

**Bindings**

**Servicios**

```
▼ <wsdl:binding name="LoginUsuarioSoap" type="tns:LoginUsuarioSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
  ▼ <wsdl:operation name="ValidarUsuario">
    <soap:operation soapAction="http://tempuri.org/ValidarUsuario" style="document"/>
    ▼ <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    ▼ <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
▼ <wsdl:binding name="LoginUsuarioSoap12" type="tns:LoginUsuarioSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http"/>
  ▼ <wsdl:operation name="ValidarUsuario">
    <soap12:operation soapAction="http://tempuri.org/ValidarUsuario" style="document"/>
    ▼ <wsdl:input>
      <soap12:body use="literal"/>
    </wsdl:input>
    ▼ <wsdl:output>
      <soap12:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
▼ <wsdl:service name="LoginUsuario">
  ▼ <wsdl:port name="LoginUsuarioSoap" binding="tns:LoginUsuarioSoap">
    <soap:address location="http://planificador.nisabusiness.com/planificadorvirtual/LoginUsuario">
    </wsdl:port>
  ▼ <wsdl:port name="LoginUsuarioSoap12" binding="tns:LoginUsuarioSoap12">
    <soap12:address location="http://planificador.nisabusiness.com/planificadorvirtual/LoginUsuar">
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

# CASO PRÁCTICO

## CONSUMIENDO WEB SERVICE

Usar la librería ksoap2

<https://code.google.com/p/ksoap2-android/>

Name ▴



ksoap2-android-assembly-3.3.0-jar-with-dependencies.jar



CIBERTEC



# Crear el web service

Utilizar Visual studio para crear el web service

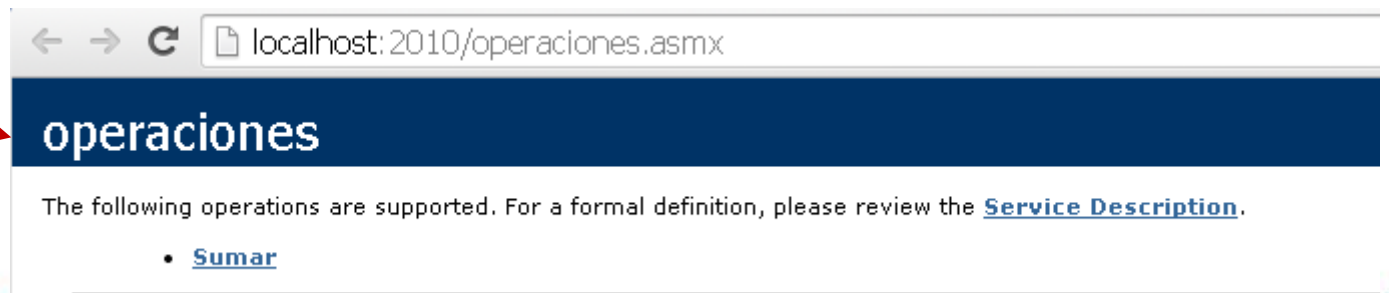


Metodo sumar 2 parámetros

```
operaciones.asmx.cs X LoginUsuario.asmx.cs
AppIntegradorASPNET.operaciones
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;

namespace AppIntegradorASPNET
{
    /// <summary>
    /// Summary description for operaciones
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // To allow this Web Service to be called from script, using ASP.NET
    // [System.Web.Script.Services.ScriptService]
    public class operaciones : System.Web.Services.WebService
    {
        [WebMethod]
        public String Sumar(Int32 pNumero1, Int32 pNumero2)
        {
            return (pNumero1 + pNumero2).ToString();
        }
    }
}
```

Ejecutando WS



# Clases de acceso a web service

## SoapObject

Permite realizar crear la conexión al servicio basado en el namespace y el nombre de método

```
SoapObject oRequest=new SoapObject("http://tempuri.org/", "Sumar");
```

## addProperty

Propiedad de SoapObject para agregar parámetros

```
oRequest.addProperty("pNumero1",1);  
oRequest.addProperty("pNumero2",2);
```

## SoapSerializationEnvelope

Permite inicializar el enlace de datos con el web service, serializar los datos para iniciar comunicación.

```
SoapSerializationEnvelope oSerializador=new SoapSerializationEnvelope(SoapEnvelope.VER11);  
oSerializador.setOutputSoapObject(oRequest);  
oSerializador.dotNet=true;
```

## HttpTransportSE

Permite realizar la llamada al servicio web utilizando la url y el objeto de serialización

```
HttpTransportSE oEjecutarWebService=new HttpTransportSE("http://planificador.nisabusiness.com/operaciones.asmx");  
oEjecutarWebService.call("http://tempuri.org/Sumar", oSerializador);  
SoapObject Resultado=(SoapObject)oSerializador.bodyIn;  
StrResultado=Resultado.getProperty(0).toString(); //Permite recuperar el dato.
```



# CONFIGURAR LA LIBRERÍA

CONFIG GRADLE



# Pasos

## Paso 1 Descargar la libreria

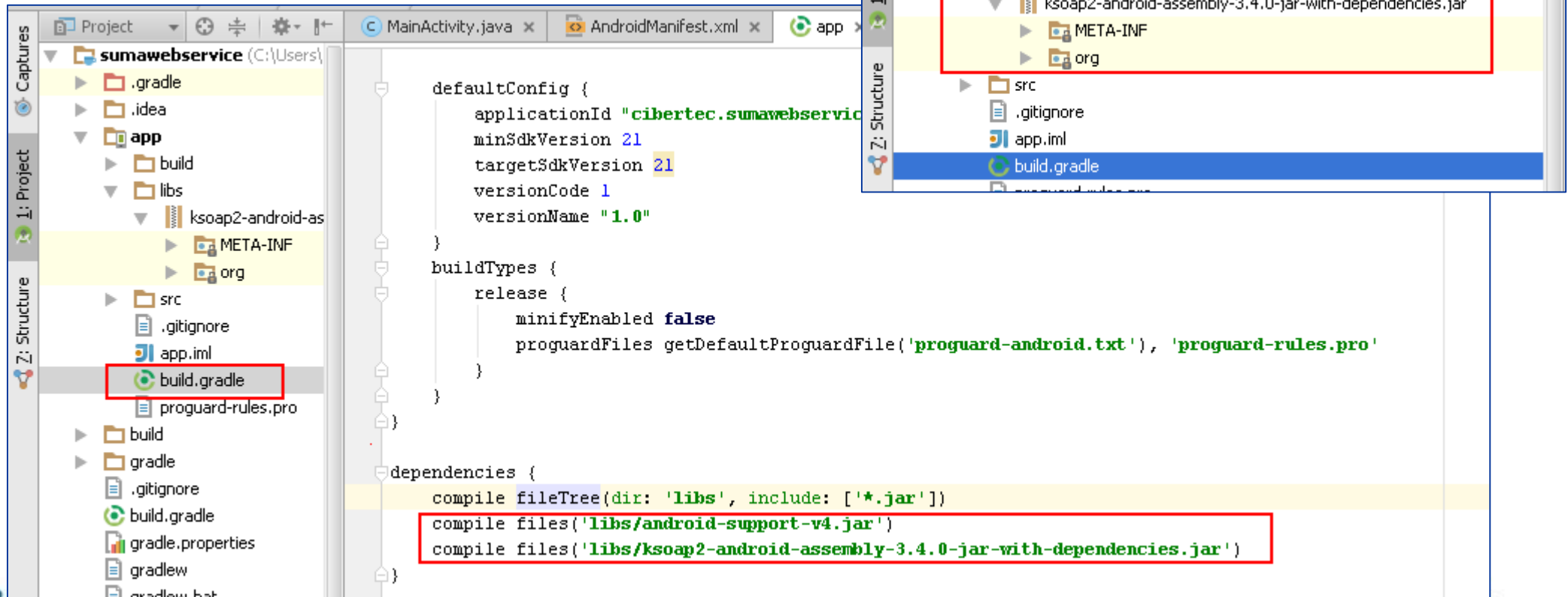
En google buscar “ksoap for android studio” y escoger el que muestra la ruta: [code.google.com/p/ksoap2-android/](http://code.google.com/p/ksoap2-android/)

## Paso 2 pegar la librería en la carpeta libs

En la vista Project ir a la carpeta app/libs

## Paso 3 Configurar la dependencia

En el archivo de configuración build.gradle

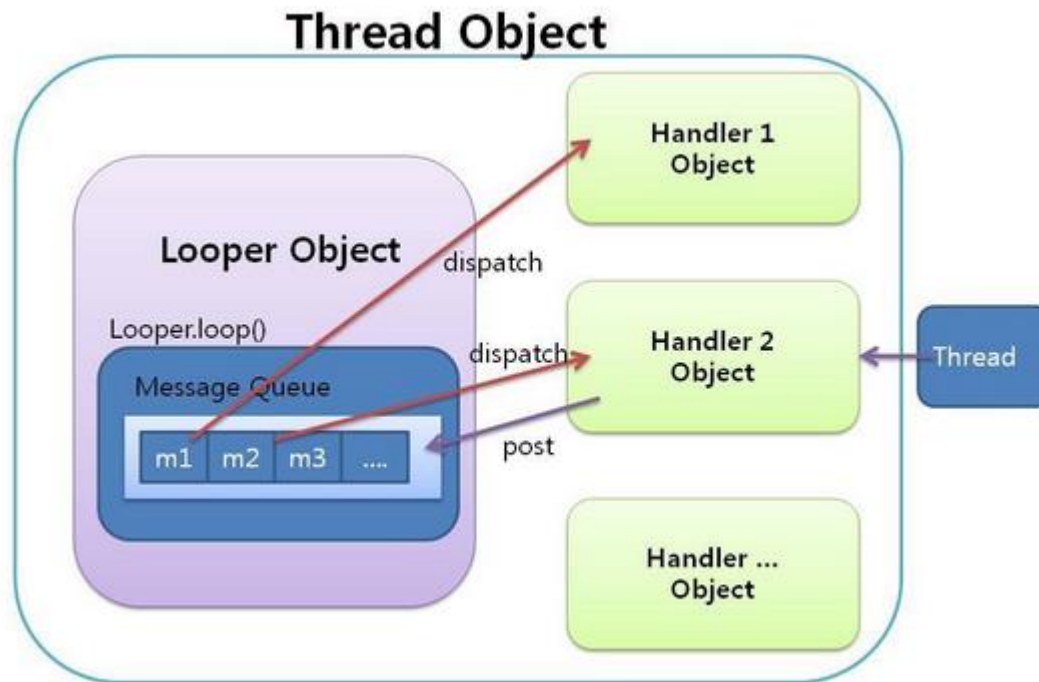


compile files('libs/android-support-v4.jar')

compile files('libs/ksoap2-android-assembly-3.4.0-jar-with-dependencies.jar')

# Thread Manejando Hilos

**Thread (Hilo).**- Clase que permite ejecutar algún proceso en segundo plano.



**Importante!**

El hilo no puede modificar ni insertar datos en el hilo principal (aplicación) esto causaría error

# Thread Ejemplo

Variable oHilo

Evento de ejecución

Metodo main de ejecución del hilo

Acceso a interfaz grafica actual

Ejecutar el Hilo

Thread oHilo=new Thread(  
new Runnable()  
{  
public void run()  
{  
//Metodo de ejecución  
//Acceso a interfaz grafica actual  
runOnUiThread(new Runnable()  
{  
public void run()  
{  
}  
});  
}  
});  
oHilo.start(); //Ejecutar el hilo

# Thread Sintaxis

Variable oHilo

Evento de ejecución

Metodo main de ejecución del hilo

Acceso a interfaz grafica actual

Ejecutar el Hilo

```
Thread oHilo=new Thread(  
    new Runnable()  
{  
    public void run()  
    {  
        //Metodo de ejecución  
        //Acceso a interfaz grafica actual  
        runOnUiThread(new Runnable()  
        {  
            public void run()  
            {}  
        });  
    }  
});  
oHilo.start(); //Ejecutar el hilo
```



# Thread código de ejemplo

```
obtnSumar=(Button)findViewById(R.id.btncalcular);  
obtnSumar.setOnClickListener(new View.OnClickListener() {
```

```
@Override
```

```
public void onClick(View v) {
```

```
// TODO Auto-generated method stub
```

```
Thread oHilo=new Thread()
```

```
{
```

```
String StrResultado="";
```

```
public void run()
```

```
{
```

```
oedtnumero1=(EditText)findViewById(R.id.edtnumero1);
```

```
oedtnumero2=(EditText)findViewById(R.id.edtnumero2);
```

```
SoapObject oRequest=new SoapObject(NAMESPACE, NOMBRE_METODO);
```

```
oRequest.addProperty("pNumero1",Integer.valueOf(oedtnumero1.getText().toString()));
```

```
oRequest.addProperty("pNumero2",Integer.valueOf(oedtnumero2.getText().toString()));
```

```
SoapSerializationEnvelope oSerializador=new SoapSerializationEnvelope(SoapEnvelope.VER11);
```

```
oSerializador.setOutputSoapObject(oRequest);
```

```
oSerializador.dotNet=true;
```

```
HttpTransportSE oEjecutarWebService=new HttpTransportSE(URL);
```

```
try
```

```
{
```

```
oEjecutarWebService.call(SOAP_ACTION, oSerializador);
```

```
SoapObject Resultado=(SoapObject)oSerializador.bodyIn;
```

```
StrResultado=Resultado.getProperty(0).toString();
```

```
}
```

```
catch(Exception ex)
```

```
{
```

```
StrResultado=ex.toString();
```

```
}
```

```
runOnUiThread(new Runnable()
```

```
{
```

```
public void run()
```

```
{
```

```
oTvResultado=(TextView)findViewById(R.id.tvresultado);
```

```
oTvResultado.setText(StrResultado);
```

```
Toast.makeText(getApplicationContext(), StrResultado, Toast.LENGTH_LONG).show();
```

```
}
```

```
});
```

```
}
```

```
};  
oHilo.start();
```

```
});  
}
```

```
EditText oedtnumero1=null;
```

```
EditText oedtnumero2=null;
```

```
TextView oTvResultado=null;
```

```
Button obtnSumar=null;
```

```
Button obtnCancelar=null;
```

```
private static String NAMESPACE = "http://tempuri.org/";
```

```
private static String URL = "http://planificador.nisabusiness.com/operaciones.asmx";
```

```
private static String NOMBRE_METODO = "Sumar";
```

```
private static String SOAP_ACTION = "http://tempuri.org/Sumar";
```

2

3

1



# GRACIAS



**SEDE MIRAFLORES**

Calle Diez Canseco Cdra. 2  
con Pasaje Tello  
T: 419 - 2900

**SEDE NORTE**

Av. Carlos Izaguirre 233  
Independencia  
T: 614 - 1717

**SEDE SAN MIGUEL**

Av. Federico Gallese 847  
San Miguel  
T: 632 - 4900

[www.cibertec.com.pe](http://www.cibertec.com.pe)