

Guía práctica N12:

Objetivo:

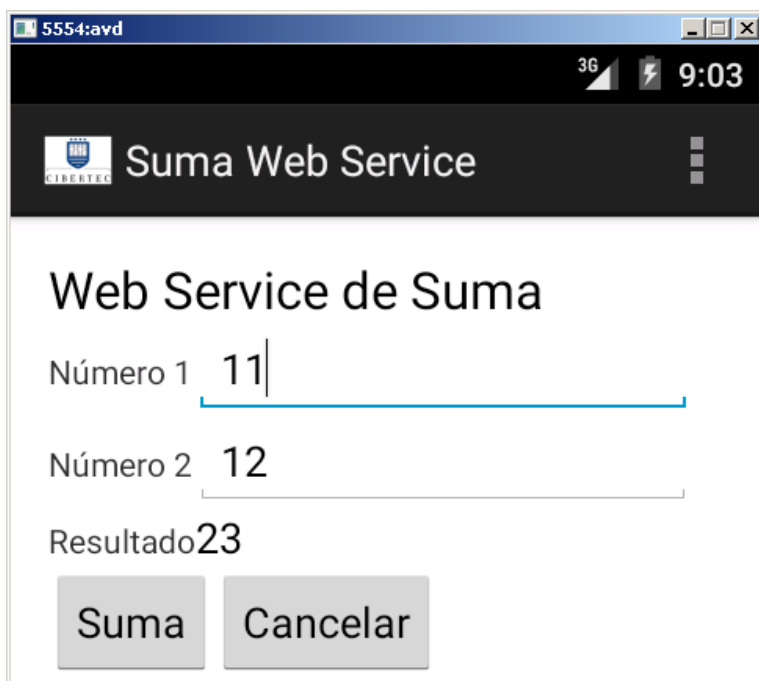
Al finalizar la práctica el participante conoce y aplica sentencias de networking haciendo uso de clases de comunicación SOAP:

- Crear web services haciendo uso visual studio 2010
- Descargar y configurar la librería externa de `ksoap2` para acceder a web services.
- Hacer uso de Thread para ejecutar llamadas asíncronas.
-

Importante: Se creará dos aplicaciones: el servicio web que será desarrollado usando visual studio .net con lenguaje C# y publicarlo en el servidor. Posterior a ello se debe crear la aplicación en android.

Indicaciones

Se desea desarrollar una aplicación que permita realizar operaciones de cálculo en línea de operaciones de suma, para ello se debe ingresar dos números que deben ser sumados.



Datos del proyecto:

Nombre de aplicación: Suma Web Service

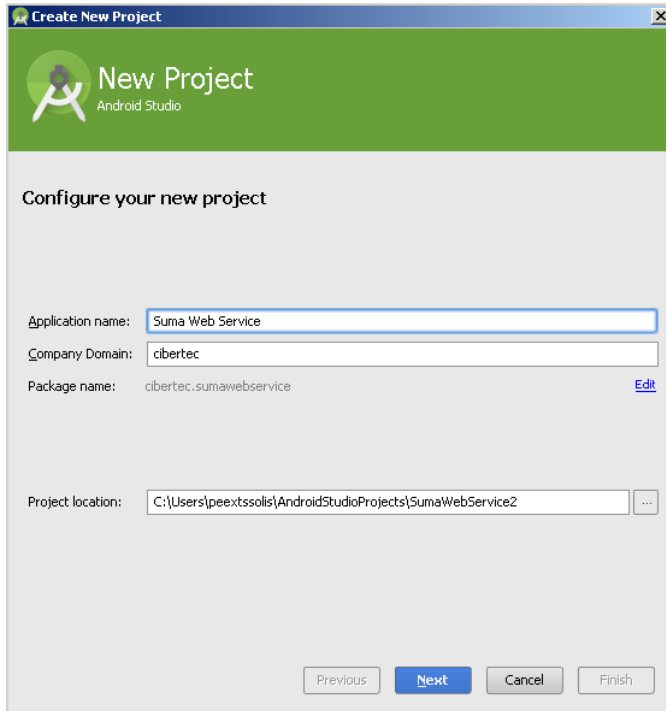
Nombre de proyecto: SumaWebService

Package Name: cibertec.apps.sumawebservice

Target SDK: API 21: Android 5.0 (Lollipop)

Compile with: API 21: Android 5.0 (Lollipop)

Theme: None



Create New Project

New Project
Android Studio

Configure your new project

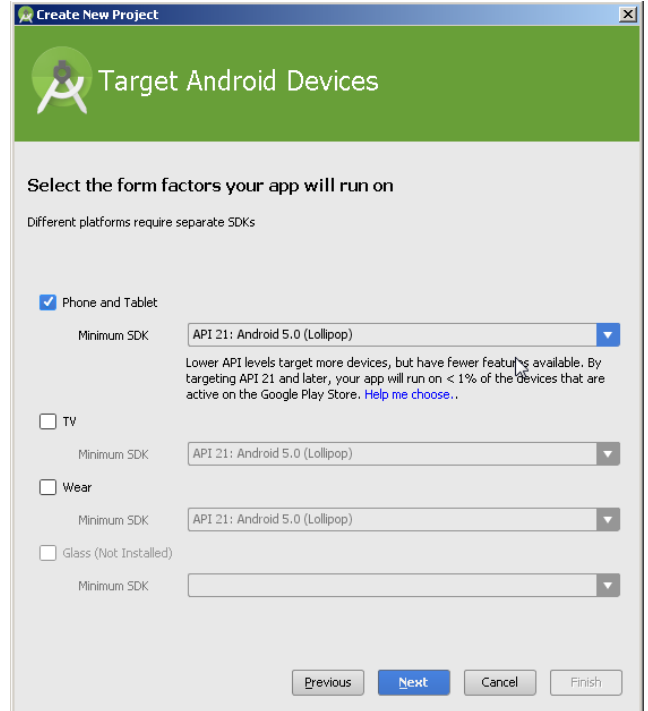
Application name: Suma Web Service

Company Domain: cibertec

Package name: cibertec.sumawebservice [Edit](#)

Project location: C:\Users\peextssolis\AndroidStudioProjects\SumaWebService2 ...

Previous Next Cancel Finish



Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms require separate SDKs

☒ Phone and Tablet

Minimum SDK: API 21: Android 5.0 (Lollipop)

Lower API levels target more devices, but have fewer features available. By targeting API 21 and later, your app will run on < 1% of the devices that are active on the Google Play Store. [Help me choose..](#)

☐ TV

Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Glass (Not Installed)

Minimum SDK:

Previous Next Cancel Finish

Permisos de la aplicación

No olvidarse de otorgar permisos:

`android.permission.WRITE_EXTERNAL_STORAGE`

 Uses Permission

Attributes for Uses Permission

The

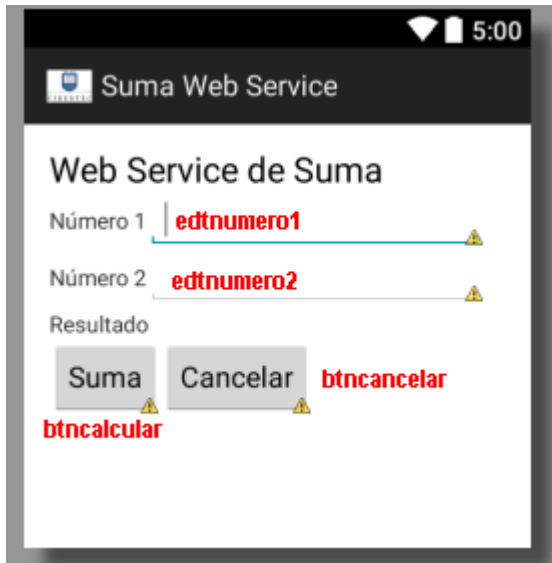
tag requests a {@link #AndroidManifestPermission <permission>} that the containing package must be granted in order for it to operate correctly.

Name

`android.permission.INTERNET`

Max SDK version

Diseño de pantalla



No olvidarse de almacenar los títulos en el archivo string.xml

Código fuente

Variables a nivel de activity

```
public class MainActivity extends Activity {  
  
    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";  
}
```

Código fuente de método onCreate

```
obtnAceptar=(Button)findViewById(R.id.btn calcular);
obtnAceptar.setOnClickListener(new View.OnClickListener() {

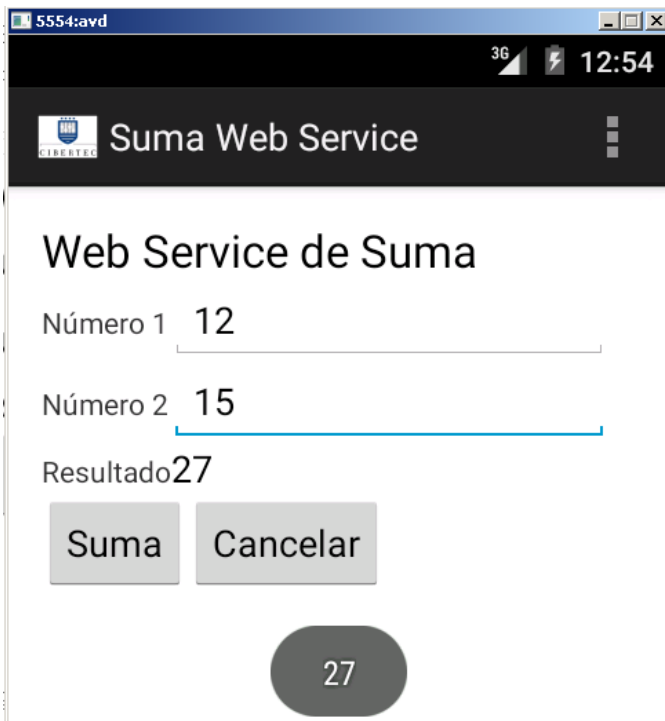
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        Thread oHilo=new Thread()
        {
            String StrResultado="";
            public void run()
            {
                oedtnumero1=(EditText)findViewById(R.id.edtnumero1);
                oedtnumero2=(EditText)findViewById(R.id.edtnumero2);
                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.VERSION1);
                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();
    }
});
```

Pruebas de funcionalidad

La primera vez que se carga la aplicación, se debe ingresar datos enteros como se muestra en el ejemplo la aplicación requiere 2 valores y posterior a ello dar click en el botón suma.



Al dar click en suma se muestra el resultado como se muestra:



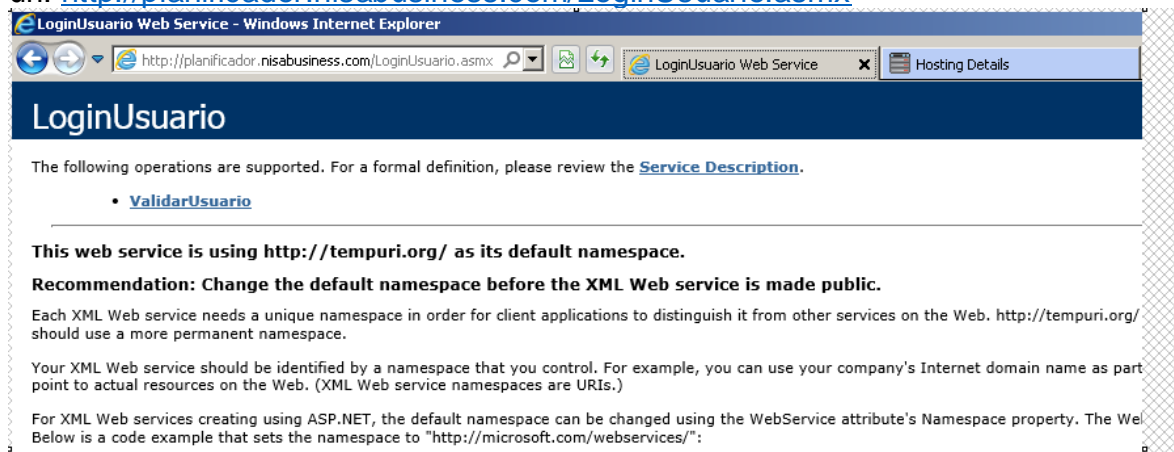
Práctica N12.1

Desarrollar una aplicación con base de datos que permita validar el usuario del sistema. Para ello se tiene los siguientes datos:

| Tabla: Usuario_Android | | <div> <div>dbo.Usuario_Android</div> <div> <div>Columns</div> <div> CodigoUsuario (PK, int, not null) Nombres (varchar(50), not null) LoginUsuario (varchar(50), not null) Contrasenia (varchar(150), not null) Correo (varchar(150), not null) </div> </div> </div> | | | | | | | | | | | | | | | | | | | | | |
|------------------------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|-------------|---------------------------|--|---------------|---------|--------------|-------------|--------|---|---|------------------------|--------|-----|---------------------------|---|---|------------------|-------|-----|--------------------------|
| Registro de usuarios | | <div> <div>Results</div> <div>Messages</div> <table> <tr> <th></th><th>CodigoUsuario</th><th>Nombres</th><th>LoginUsuario</th><th>Contrasenia</th><th>Correo</th></tr> <tr> <td>1</td><td>1</td><td>Juan Jose Arana Flores</td><td>jarana</td><td>123</td><td>jarana@nisaitbusiness.com</td></tr> <tr> <td>2</td><td>2</td><td>Maria Elena Jara</td><td>mjara</td><td>123</td><td>mjara@nisaitbusiness.com</td></tr> </table> </div> | | | | | CodigoUsuario | Nombres | LoginUsuario | Contrasenia | Correo | 1 | 1 | Juan Jose Arana Flores | jarana | 123 | jarana@nisaitbusiness.com | 2 | 2 | Maria Elena Jara | mjara | 123 | mjara@nisaitbusiness.com |
| | CodigoUsuario | Nombres | LoginUsuario | Contrasenia | Correo | | | | | | | | | | | | | | | | | | |
| 1 | 1 | Juan Jose Arana Flores | jarana | 123 | jarana@nisaitbusiness.com | | | | | | | | | | | | | | | | | | |
| 2 | 2 | Maria Elena Jara | mjara | 123 | mjara@nisaitbusiness.com | | | | | | | | | | | | | | | | | | |

Web service existente

url: <http://planificador.nisabusiness.com/LoginUsuario.asmx>



The following operations are supported. For a formal definition, please review the [Service Description](#).

- ValidarUsuario

This web service is using <http://tempuri.org/> as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. <http://tempuri.org/> should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part point to actual resources on the Web. (XML Web service namespaces are URIs.)

For XML Web services creating using ASP.NET, the default namespace can be changed using the WebService attribute's Namespace property. The Web Below is a code example that sets the namespace to "http://microsoft.com/webservices/":

Parametros Web service

```
POST /planificadorvirtual/LoginUsuario.asmx 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:xsd=
  <soap:Body>
    <ValidarUsuario xmlns="http://tempuri.org/">
      <pLoginUsuario>string</pLoginUsuario>
      <pContrasenia>string</pContrasenia>
    </ValidarUsuario>
  </soap:Body>
</soap:Envelope>
```

Datos del proyecto:

Nombre de aplicación: WS Login Usuario

Nombre de proyecto: WSLoginUsuario

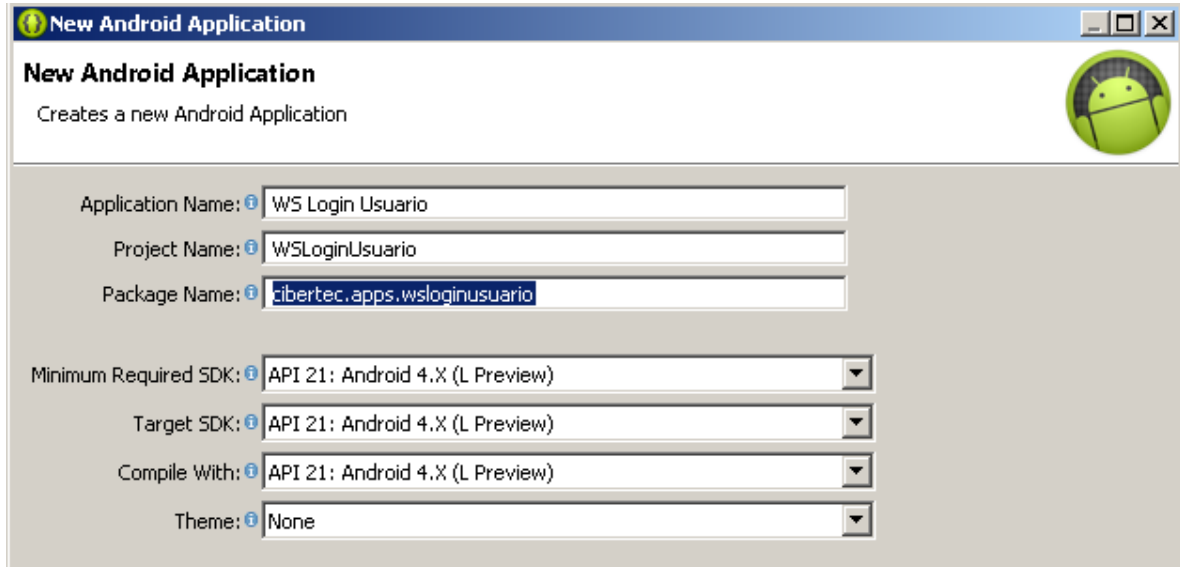
Package Name: cibertec.apps.wsloginusuario

Requerimiento mínimo de SDK: API 21: Android 4.X (L Preview)

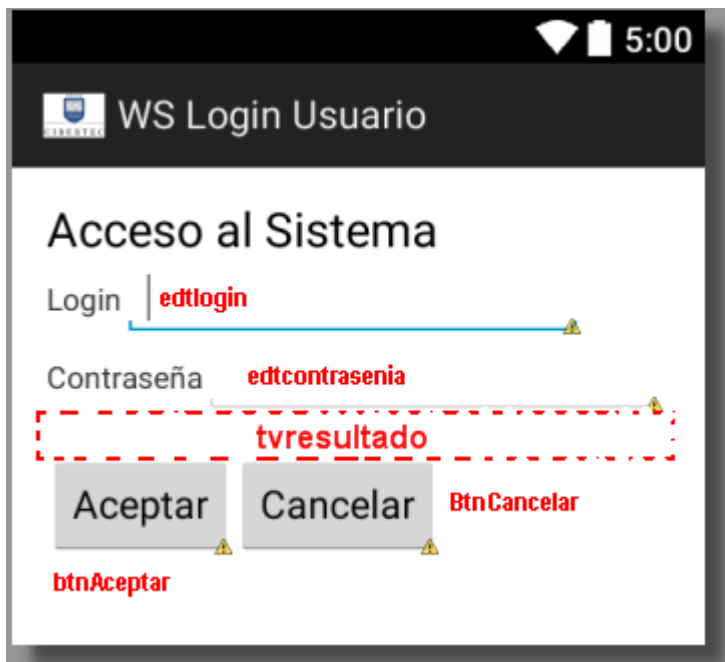
Target SDK: API 21: Android 4.x (L Preview)

Compile with: API 21: Android 4.x (L Preview)

Theme: None



Diseño de pantalla



Prueba de funcionalidad



5554:avd

3G 2:01

 WS Login Usuario

Acceso al Sistema

Login jarana

Contraseña 123

1|Juan Jose Arana Flores|jarana|123|
jarana@nisaitbusiness.com

Aceptar Cancelar