

## Integración de Servicios Web con Apache AXIS2 y .NET

En este ejercicio se creará un cliente con la plataforma VS.NET que invocará a un servicio Web realizado sobre Apache AXIS 2.

Para que un cliente pueda consumir un Servicio Web desarrollado en una plataforma distinta se debe tener en cuenta utilizar estilos de documentos compatibles. En los recursos aportados para el tema se pueden encontrar varios documentos referentes a este aspecto.

Existe un grupo responsable de la interoperatividad entre los Servicios Web y que se puede localizar en la URL <http://www.ws-i.org/>.

El plugin WTP posee una herramienta que avisa en caso de que el Servicio Web generado no sea compatible con el WS-I. Por defecto al usar AXIS2 el método es compatible, pero con AXIS puedes utilizar métodos no compatibles.

### Pasos a seguir

#### 1. Obtención del documento WSDL del Servicio Web a consumir

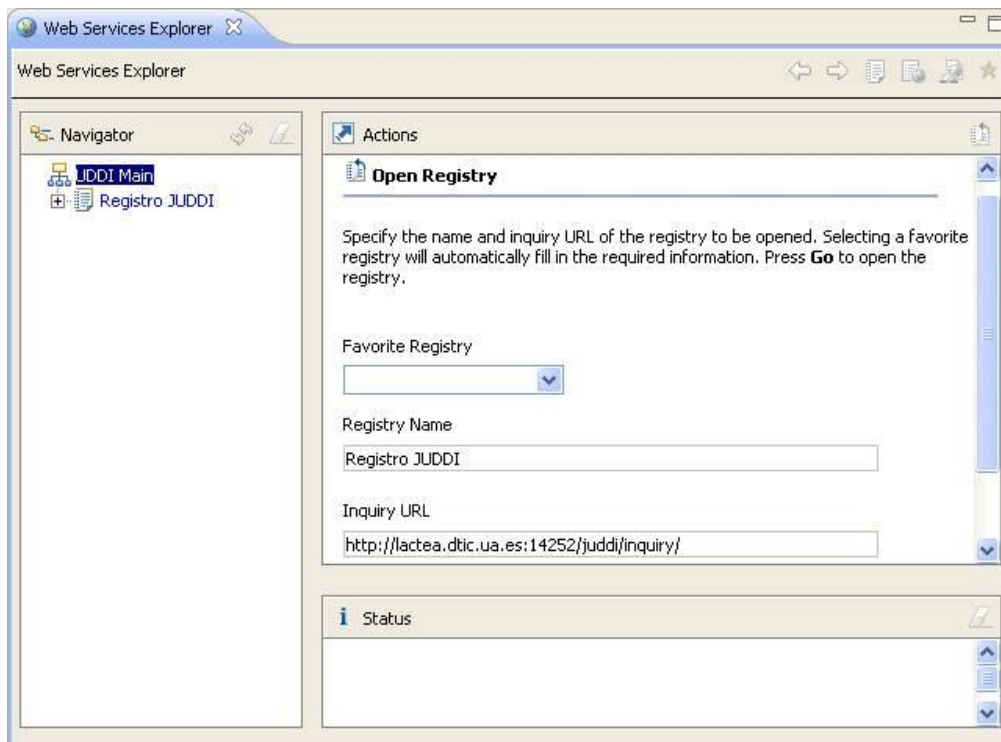
Existen varias formas de obtener el documento WSDL que describe el Servicio Web y que será necesario para generar la clase Proxy del cliente.


##### 1.1. Acceder a la URL del servicio utilizando el navegador y copiar de forma manual el documento WSDL. Para ello debe de estar activo el servidor.

```
<?xml version='1.0' encoding='UTF-8'>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:ns="http://master"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:ns1="http://org.apache.axis2/xsd"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" targetNamespace="http://master">
  <wsdl:documentation>Please Type your service description here</wsdl:documentation>
  <wsdl:types>...</wsdl:types>
  <wsdl:message name="celsiusToFahrenheitRequest">...</wsdl:message>
  <wsdl:message name="celsiusToFahrenheitResponse">...</wsdl:message>
  <wsdl:message name="fahrenheitToCelsiusRequest">...</wsdl:message>
  <wsdl:message name="fahrenheitToCelsiusResponse">...</wsdl:message>
  <wsdl:portType name="ConverterPortType">...</wsdl:portType>
  <wsdl:binding name="ConverterSoap11Binding" type="ns:ConverterPortType">...</wsdl:binding>
  <wsdl:binding name="ConverterSoap12Binding" type="ns:ConverterPortType">...</wsdl:binding>
  <wsdl:binding name="ConverterHttpBinding" type="ns:ConverterPortType">...</wsdl:binding>
  <wsdl:service name="Converter">...</wsdl:service>
</wsdl:definitions>
```

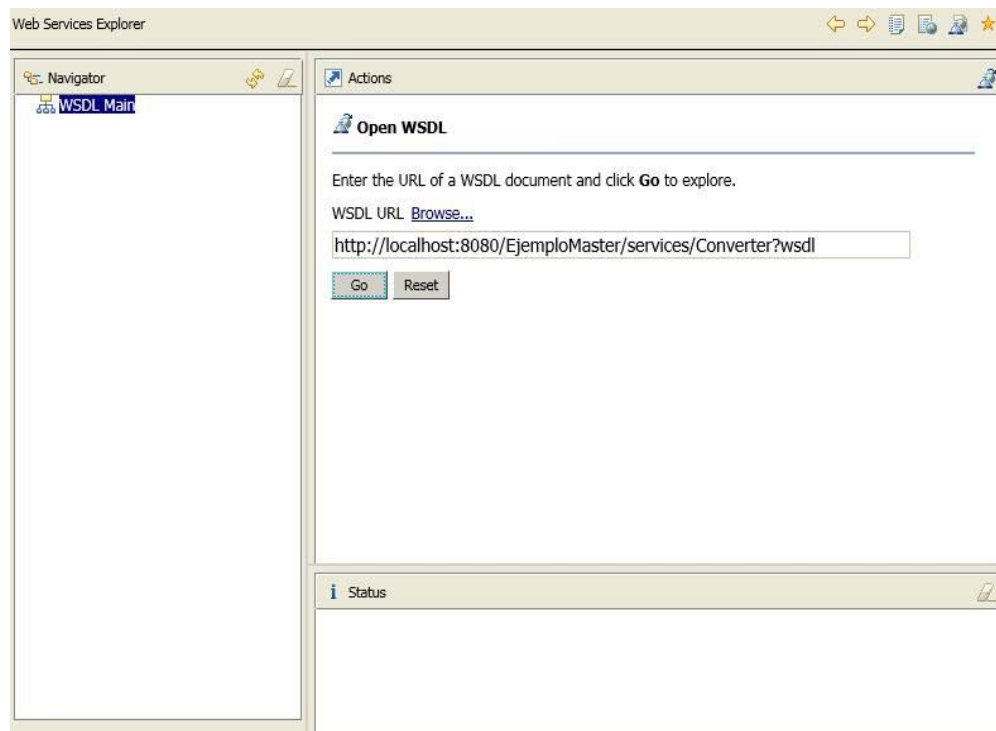
##### 1.2. Otra forma es utilizando una de las herramientas que provee la plataforma Eclipse con el plugin WTP (*Web Service Explorer*).

###### 1.2.1. Abrimos el explorador de Servicios Web *Run* → *Launch the Web Service Explorer*. Lo primero que se mostrará en el explorador es el árbol de servidores UDDI disponibles. En esta opción podemos crearnos acceso a nuestros propios registro UDDI.

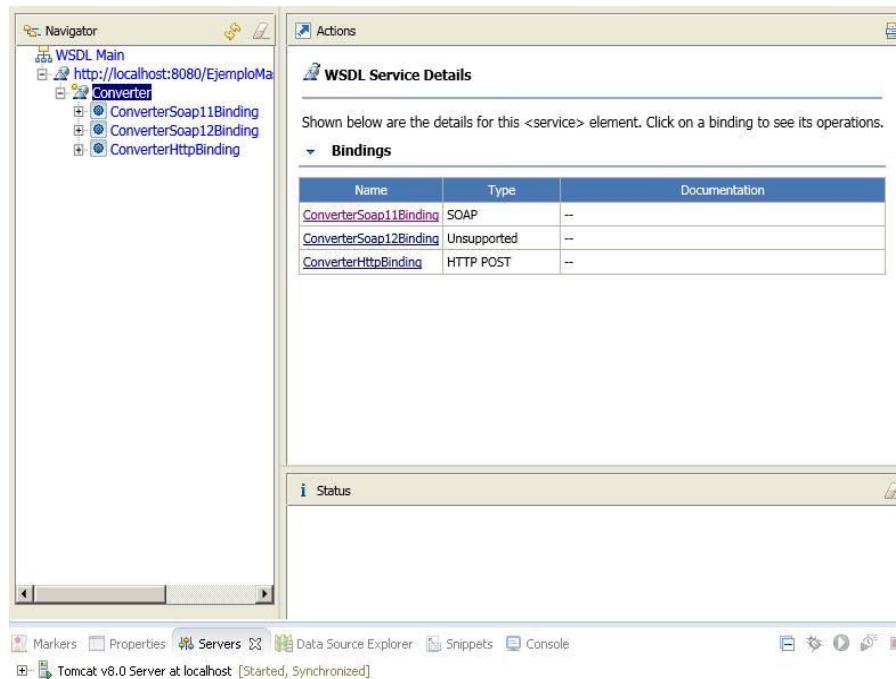


1.2.2. En el explorador, en los iconos de la esquina superior se puede cambiar la vista al explorador de documentos WSDL . Pulsando en el icono obtendremos dicha vista.

1.2.3. Introduciendo la WSDL y pulsando en *Go* se obtendrán los datos de documento WSDL.



- 1.2.4. Seleccionado el elemento de la URL localizada aparecen varias opciones para obtener el documento WSDL.



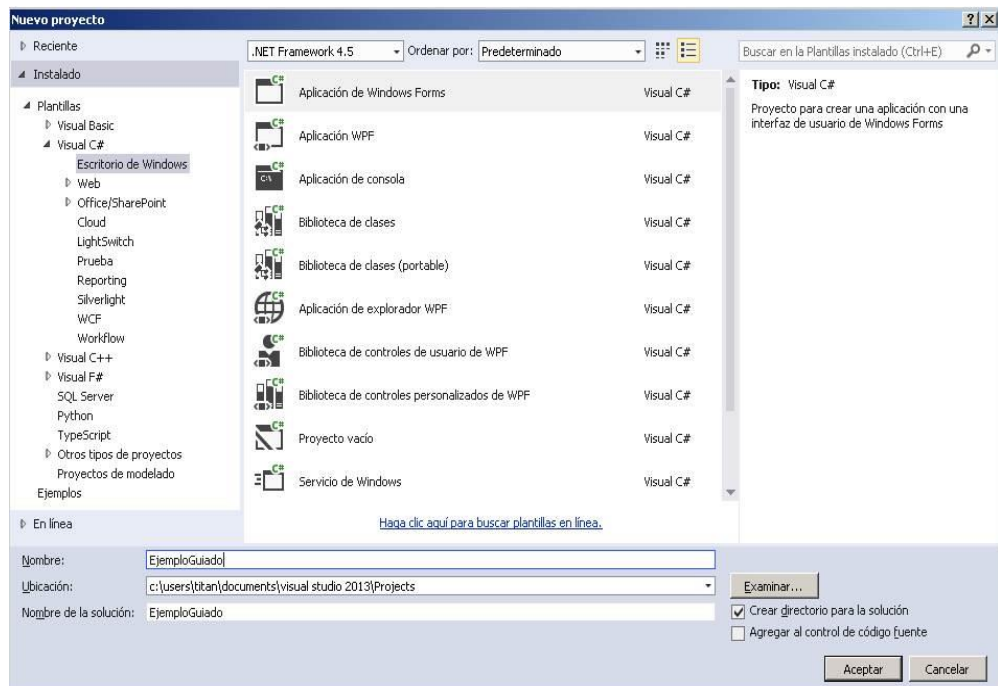
- 1.2.5. Importamos el documento WSDL con la opción *Import WSDL To File System* a la localización que nos interese.

## 2. Generar cliente desde .NET

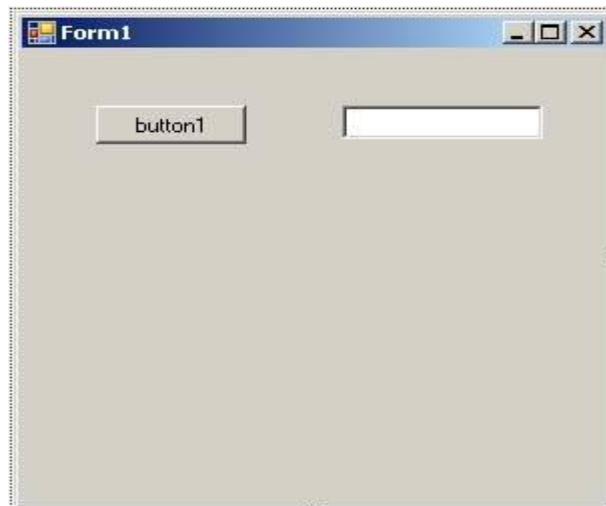
### 2.1. Generación del cliente desde el entorno VS.NET

En este caso, se puede utilizar directamente la URL que apunta al documento WSDL del servicio. Los pasos a seguir son los siguientes.

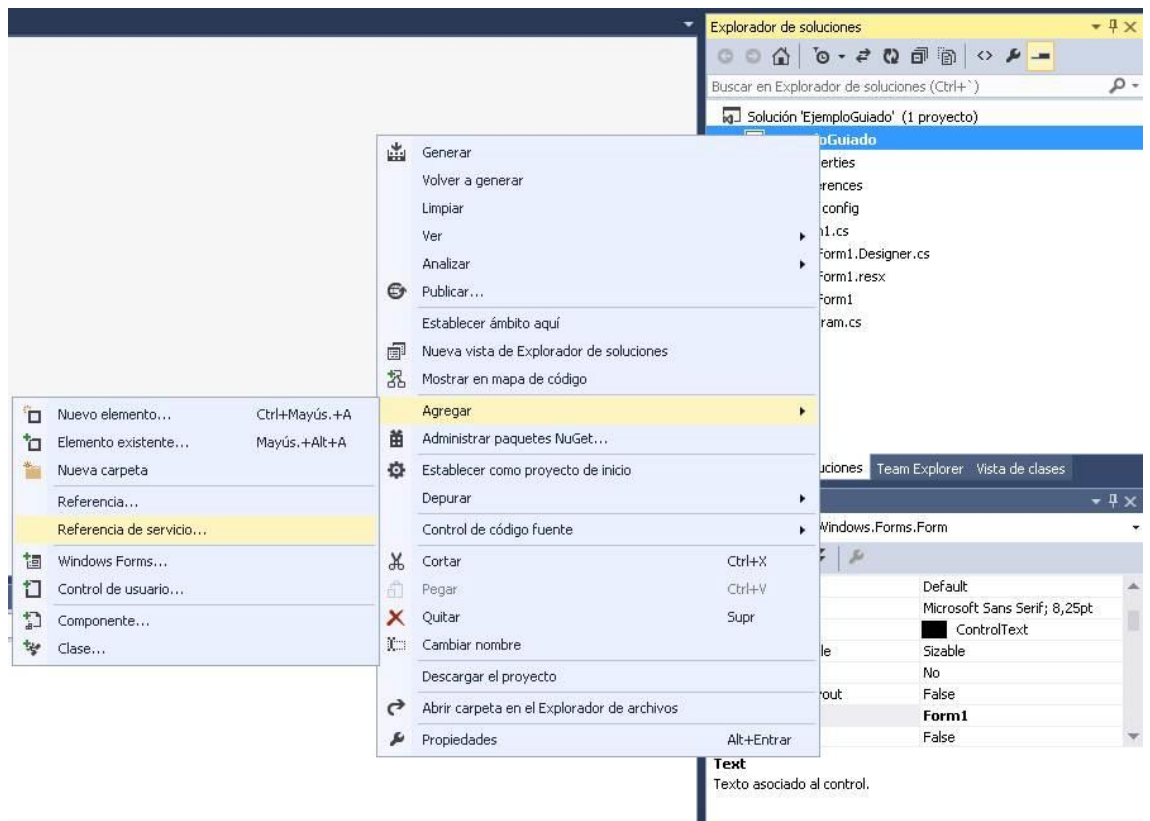
- 2.1.1. Abrimos el entorno VS.NET y generamos un proyecto de escritorio en C#.



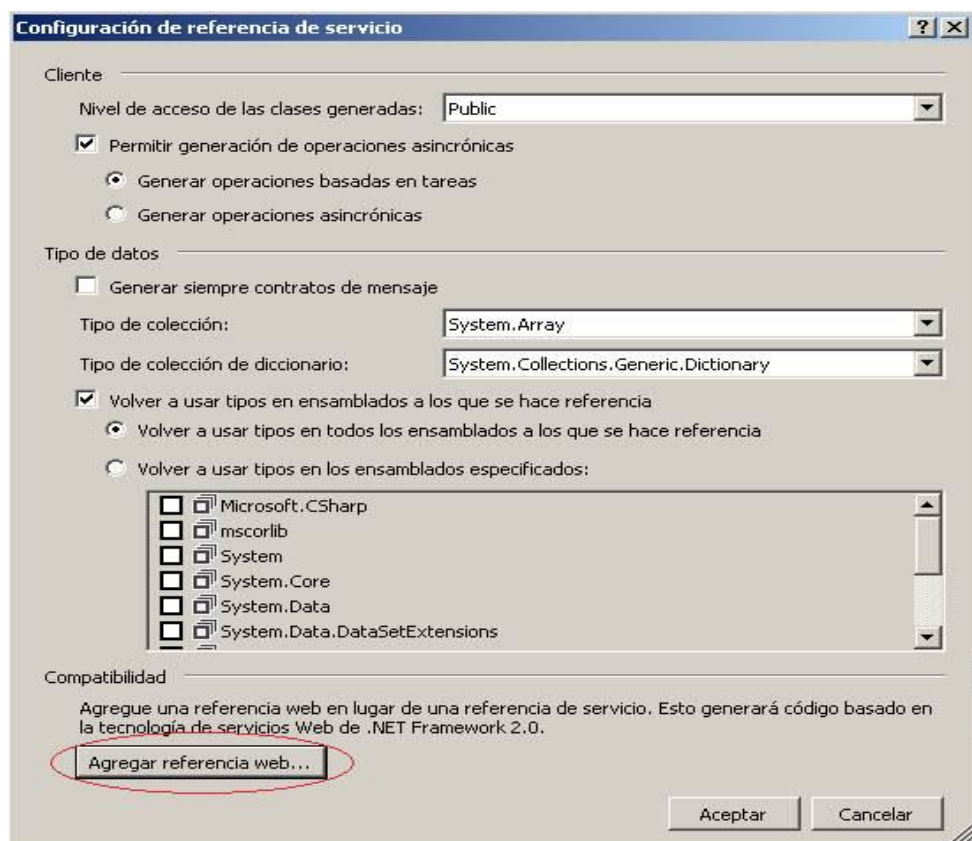
2.1.2. Creamos una aplicación simple de formulario con un control *button* y un *TextBox*.



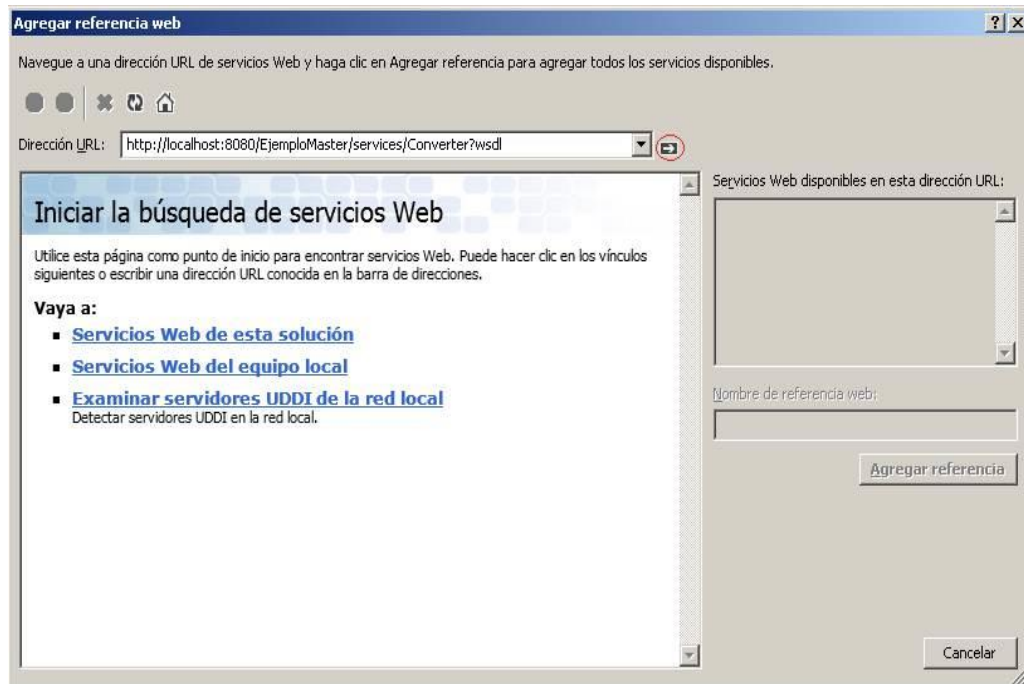
2.1.3. Dentro de la aplicación, en el explorador de aplicaciones seleccionamos el proyecto y pulsamos el botón derecho. A continuación seleccionamos la opción *Agregar Referencia de Servicio...*



2.1.4. En la ventana que aparece pulsar el botón *Avanzadas* situado en la parte inferior. Esta acción mostrará otra ventana sobre la que debemos pulsar en el botón *Agregar Referencia Web...*



2.1.5. Para agregar la referencia Web introducimos la URL del documento WSDL del servicio, pulsamos la flecha y pulsamos en *Agregar Referencia*.



Si ha funcionado deberías de obtener una pantalla similar a la siguiente.



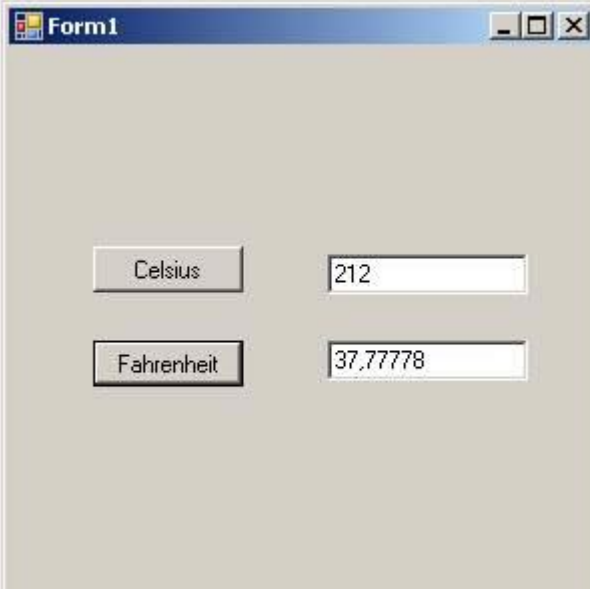
Una vez hecho esto, se debe introducir el Nombre de referencia Web, en este caso vamos a usar Converter para que el código de que tenemos a continuación funciona correctamente y apretamos Agregar referencia.

- 2.1.6. Dentro de la aplicación se habrá creado la referencia con el nombre que le hemos dado y que podemos usar para acceder al servicio como si se tratase de una clase local. Llamaremos al constructor para crear una instancia e invocando a sus métodos, aunque de una forma un poco diferente a como se hacía desde Java. El valor que debería devolver la llamada se define como un argumento por referencia del método. A continuación se muestra un ejemplo del código que se ejecutaría al pulsar el botón del formulario.

```
private void button1_Click(object sender, EventArgs e)
{
    Converter.Converter cn = new Converter.Converter();
    float result;
    bool resultB;

    cn.celsiusToFahrenheit(100, true, out result, out resultB);
    textBox1.Text = result.ToString();
}
```

Ejemplo de métodos funcionando.



- 2.1.7. Como ejercicio, invocar el resto de métodos del servicio más los que se hayan añadido para los grados centígrados.