

Desarrollo de un servicio web para el envío de correos utilizando C# y su correspondiente cliente consumidor utilizando Java

Versión 0.1

Jorge Iván Meza Martínez
<http://www.jorgeivanmeza.com/>
<jimezam@gmail.com>

2008

Introducción

Este documento describe como desarrollar un servicio web muy simple para el envío de correos electrónicos en el lenguaje de programación C# utilizando la herramientas de programación gratuita Visual Web Development Express en su versión 2008. De igual manera se describe como desarrollar un cliente que consuma este servicio web en el lenguaje de programación Java utilizando la herramienta de programación gratuita NetBeans en su versión 6,0,1.

El desarrollo de este software tiene fines académicos únicamente y no se espera que supla algún tipo de necesidad o que sea útil para ser utilizado en servidores en producción. No se toman ningún tipo de medidas de seguridad ni se contempla la utilización de servidores SMTP que requieran autenticación.

Desarrollo del servicio web utilizando C#.

Para el desarrollo del servicio web se utilizará **Visual Web Developer Express** en su versión 2008 el cual puede ser descargado gratuitamente de la siguiente dirección.

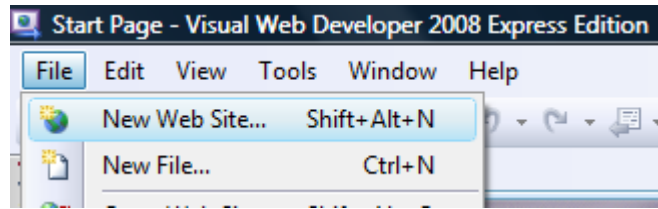
<http://www.microsoft.com/express/>

Se inicia el ambiente de desarrollo haciendo click sobre su correspondiente ícono en el escritorio, el cual se puede encontrar también a través del **Menú de Inicio > Todos los programas > Microsoft Visual Web Developer 2008 Express**.



Se inicia un nuevo proyecto seleccionando el menú **File > New Website** o de manera alternativa presionando las teclas **Mayúsculas+Alt+N**.

Desarrollo de un servicio web para el envío de correos utilizando C# y su correspondiente cliente consumidor utilizando Java,



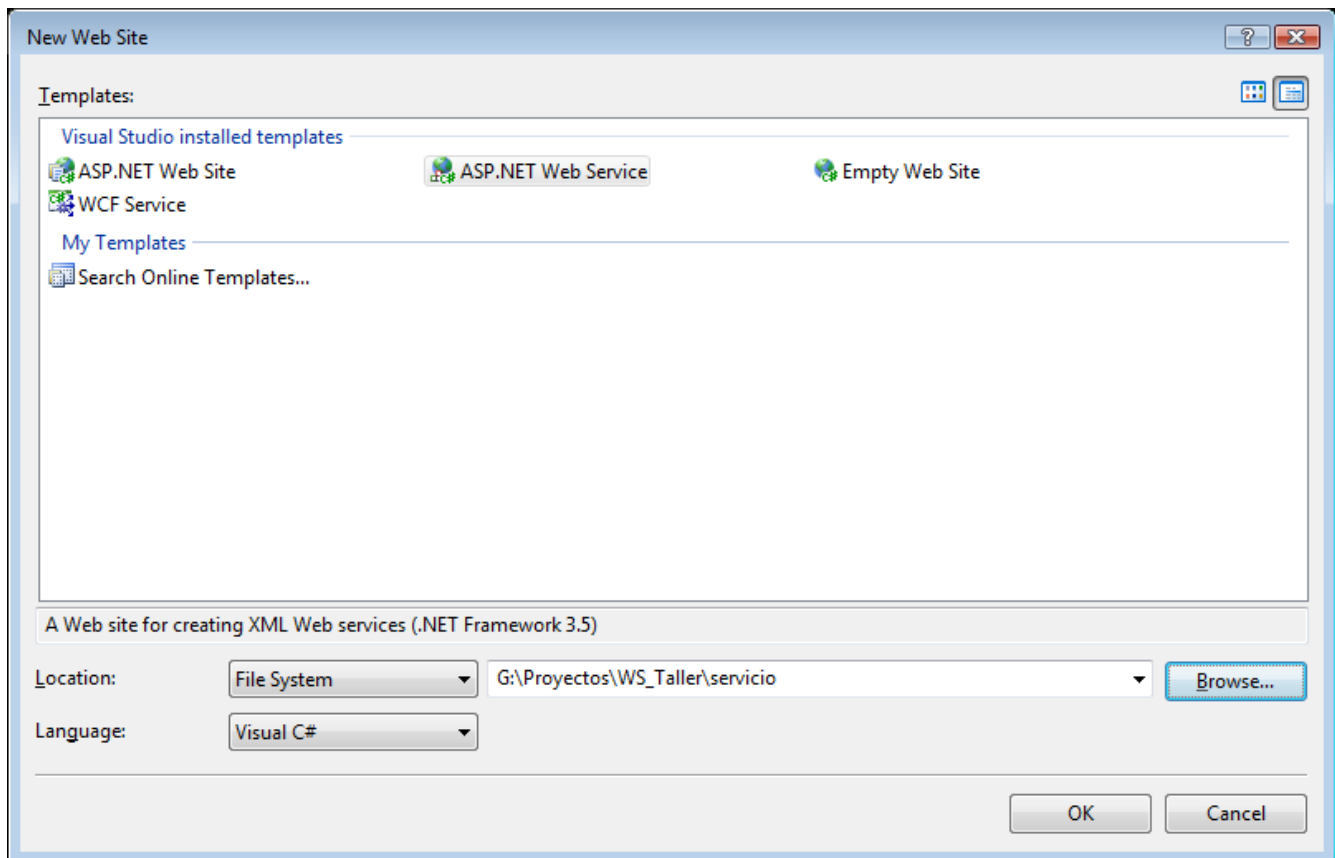
Se especifican las siguientes opciones en el diálogo de creación de sitios web.

Template: **ASP.NET Web Service**.

Location: **File System** = **G:\Proyectos\WS_Taller\servicio**.

Language: **Visual C#**.

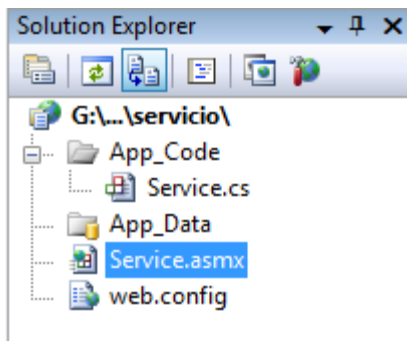
Presione el botón **OK** para aceptar los valores anteriores.



Téngase en cuenta que con los valores anteriores los archivos del proyecto se van a crear en la ruta especificada y utilizando el lenguaje de programación C#.NET. Personalice estas opciones según sus requerimientos.

Desarrollo de un servicio web para el envío de correos utilizando C# y su correspondiente cliente consumidor utilizando Java,

En el **Solution Explorer** abra el archivo **Service.asmx** haciendo doble click izquierdo sobre él o haciendo click derecho sobre él y eligiendo la opción **Open**.



Modifique el valor **Class** a **MailService** el cual es el servicio web que estamos implementando.

```
<%@ WebService Language="C#" CodeBehind="~/App_Code/Service.cs" Class="MailService" %>
```

Repita el proceso anterior abriendo ahora el archivo **Service.cs** que se encuentran bajo la rama **App_Code** del **Solution Explorer**. Realice las siguientes modificaciones al código.

Agregue una referencia al *namespace* **Net.Mail** al comienzo del código.

```
using System.Net.Mail;
```

Modifique el nombre de la clase y del constructor a **MailService**.

```
// ...  
public class MailService : System.Web.Services.WebService  
{  
    public MailService ()  
    {  
    }  
}  
// ...
```

Agregue la implementación del método **send** como *web service* para enviar el correo electrónico con el siguiente contenido.

```
[WebMethod]  
public string send(string fromUser, string toUser,  
                  string subject, string body,  
                  string smtpServer)  
{  
    // Indica el mensaje que se le va a retornar al usuario al final de la  
    // operación. "OK" significa que el envío del mensaje fue exitoso.  
  
    string response = "OK";
```

Desarrollo de un servicio web para el envío de correos utilizando C# y su correspondiente cliente consumidor utilizando Java,

```
// Se crea el objeto para el envío de correos electrónicos indicándole
// la dirección del servidor SMTP a utilizarse.

SmtpClient client = new SmtpClient(smtpServer);

// Se especifica la dirección del remitente del mensaje.

MailAddress from = new MailAddress(fromUser);

// Se especifica la dirección del receptor del mensaje.

MailAddress to = new MailAddress(toUser);

// Se especifica el contenido del mensaje.

MailMessage message = new MailMessage(from, to);
message.Body = body + Environment.NewLine + Environment.NewLine;
message.Body += "Mensaje enviado utilizando el servicio web MailService.";
message.BodyEncoding = System.Text.Encoding.UTF8;

// Se especifica el tema del mensaje.

message.Subject = subject;
message.SubjectEncoding = System.Text.Encoding.UTF8;

// Se realiza el envío el mensaje creado anteriormente.

try
{
    client.Send(message);
}
catch (SmtpException smtpException)
{
    // En caso de suceder problemas enviando el mensaje, la descripción del
    // problema sucedido es enviada al cliente que consumió el servicio.

    response = smtpException.Message;
}

// Se realiza la limpieza de los recursos.

message.Dispose();

// Se retorna el mensaje de respuesta: "OK" o el mensaje de error sucedido.

return response;
}
```

Para probar el servicio desde el **Visual Web Developer Express** utilice las opciones **Start Debugging** o **Start Without Debugging** bajo el menú **Debug** para iniciar el servicio con o sin información de depuración respectivamente. Opcionalmente es posible utilizar las combinaciones de teclas **F5** o **Ctrl+F5** para realizar las mismas acciones.

Es posible ejecutar el **ASP.NET Development Server** desde la línea de comando obviando la necesidad de tener abierto el **Visual Web Developer Express**, para esto ejecute los siguientes comandos en una ventana de **Símbolo del sistema**.

```
prompt> set DEVSRV=C:\Program Files\Common Files\microsoft
shared\DevServer\9.0\
```

Desarrollo de un servicio web para el envío de correos utilizando C# y su correspondiente cliente consumidor utilizando Java,

```
prompt> "%DEVSrv%"\webdev.webserver.exe /path:"G:\Proyectos\WS_Taller\servicio"  
/vpath:/MailService
```

La ubicación del programa `webdev.webserver.exe` puede variar según la versión del .NET Framework que se tenga instalada.

MailService

Las operaciones siguientes son compatibles. Para una definición formal, revise la [descripción de servicios](#).

- [send](#)

Este servicio Web utiliza <http://tempuri.org/> como espacio de nombres predeterminado.

Recomendación: cambiar el espacio de nombres predeterminado antes de hacer público el servicio Web XML.

Cada servicio Web XML necesita un espacio de nombres único para que las aplicaciones de cliente puedan distinguir este servicio los servicios Web XML publicados deberían utilizar un espacio de nombres más permanente.

Para consultar el documento WSDL¹ relacionado haga click en el enlace **descripción de servicios**, para probar el servicio web haga click sobre el servicio **send**.

MailService

Haga clic [aquí](#) para obtener una lista completa de operaciones.

send

Prueba

Haga clic en el botón 'Invocar', para probar la operación utilizando el protocolo HTTP POST.

| Parámetro | Valor |
|-------------|----------------------|
| fromUser: | <input type="text"/> |
| toUser: | <input type="text"/> |
| subject: | <input type="text"/> |
| body: | <input type="text"/> |
| smtpServer: | <input type="text"/> |

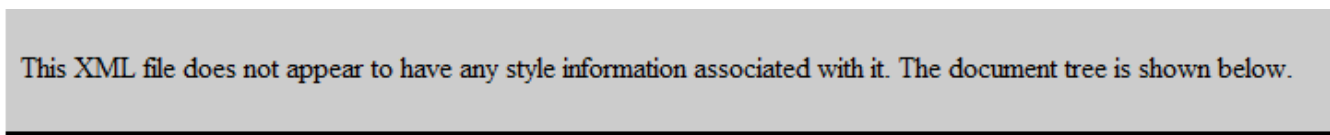
Especifique valores para los siguientes parámetros y presione el botón Invocar para probar el servicio web.

¹ **Web Services Description Language** - http://en.wikipedia.org/wiki/Web_Services_Description_Language

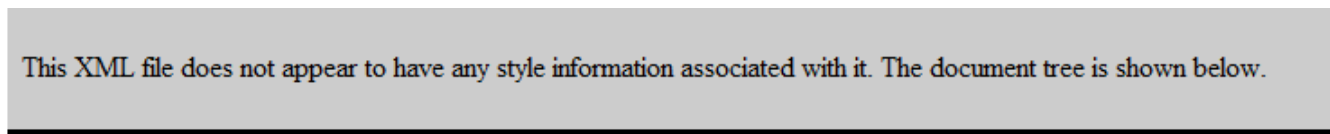
- fromUser** – dirección del usuario remitente.
- toUser** – dirección del usuario receptor.
- subject** – tema del mensaje.
- body** – cuerpo del mensaje.
- smtpServer** – dirección IP del servidor SMTP que se va a utilizar.

Nótese como estas variables coinciden con los parámetros de la función **send** implementada.

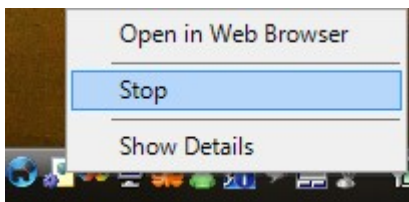
Si el envío del mensaje es exitoso deberá obtenerse la respuesta de éxito: OK, tal y como se muestra en la siguiente imagen.



En caso de suceder un error se deberá obtener como respuesta una descripción de este, de manera similar a la mostrada en la siguiente imagen.



Para detener el servicio web que se está probando haga click derecho sobre el icono del **ASP.NET Development Server** y elija la opción **Stop** tal y como se muestra a continuación.



Desarrollo del cliente del servicio web utilizando Java.

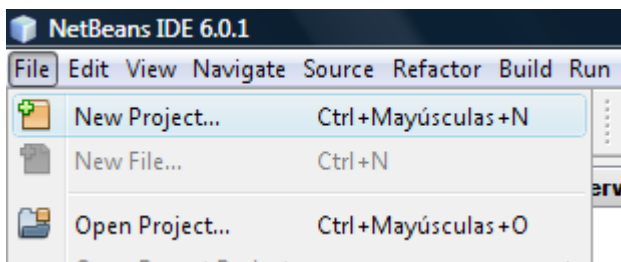
Para el desarrollo del cliente se utilizará **NetBeans** en su versión 6,0,1 el cual puede ser descargado gratuitamente de la siguiente dirección.

<http://www.netbeans.org/>

Se inicia el ambiente de desarrollo haciendo click sobre su correspondiente ícono en el escritorio, el cual se puede encontrar también a través del **Menú de Inicio > Todos los programas > NetBeans > NetBeans IDE 6,0,1**.

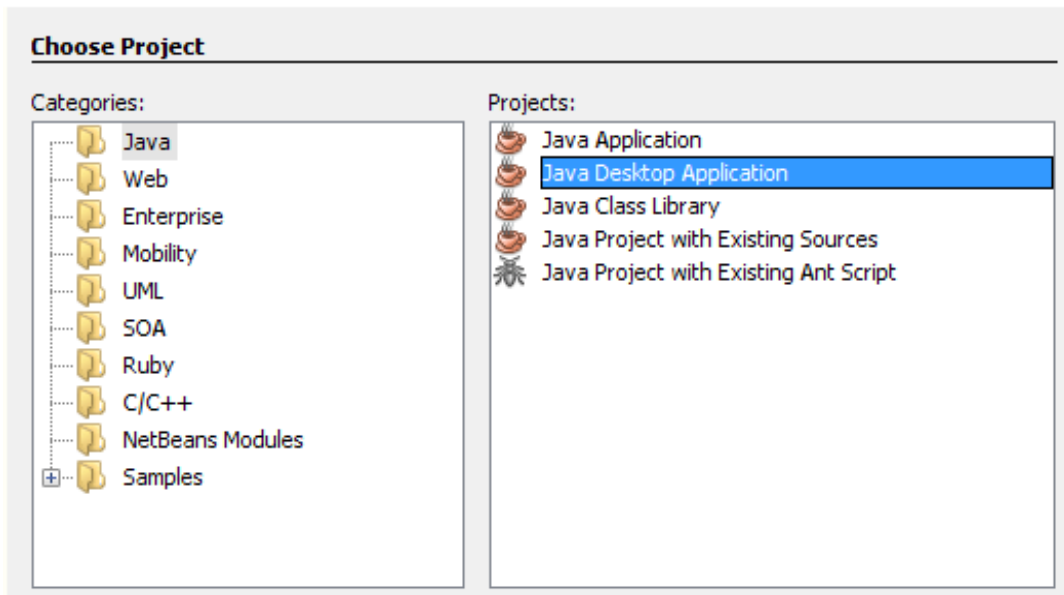


Cree un nuevo proyecto seleccionando el menú **File > New Project** o presionando la combinación de teclas **Mayúsculas+Ctrl+N**.



Seleccione una aplicación Java de escritorio: **Java > Java Desktop Application**. Presione el botón **Next** para continuar.

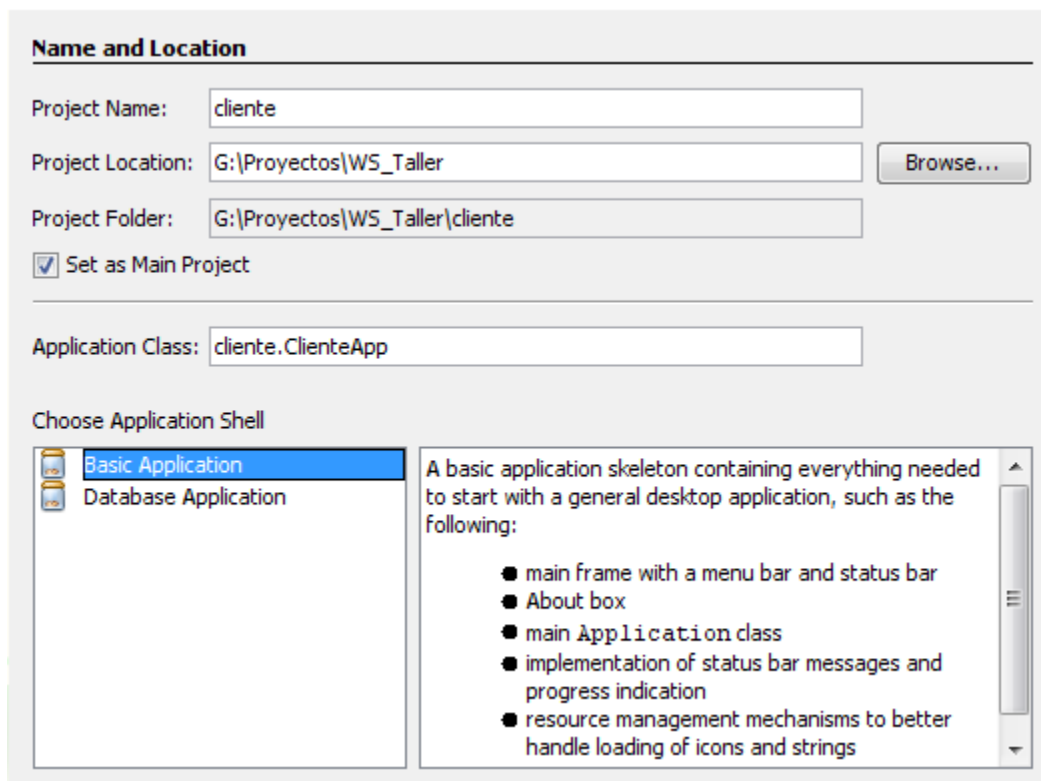
Desarrollo de un servicio web para el envío de correos utilizando C# y su correspondiente cliente consumidor utilizando Java,



Especifique las siguientes opciones del proyecto.

Project Name: **cliente**

Project Location: **G:\Proyectos\WS_Taller**



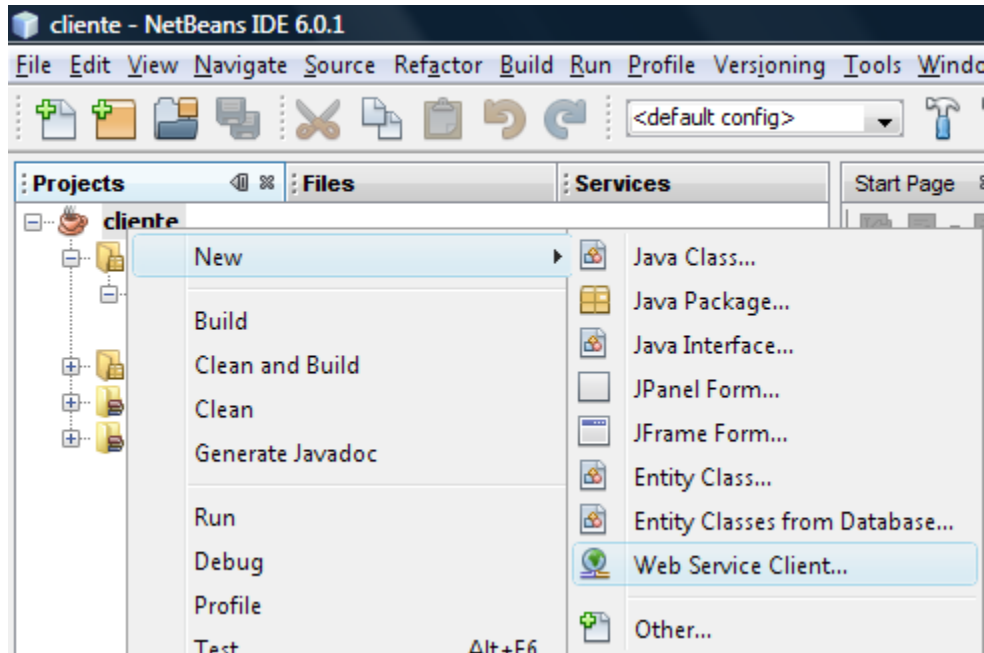
Tenga en cuenta que deberá personalizar la variable **Project Location** según sus propios

Desarrollo de un servicio web para el envío de correos utilizando C# y su correspondiente cliente consumidor utilizando Java,

requerimientos.

Presione el botón **Finish** para aceptar la creación del proyecto con la información anterior.

Agregue la referencia del servicio web al proyecto haciendo click derecho sobre el proyecto **cliente** y seleccionando los menús **New > Web Service Client ...**



Realice la especificación de los siguientes valores necesarios.

Seleccione la opción **WSDL URL**.

Especifique la ubicación del servicio web: <http://localhost/MailService/Service.asmx>.

Package: **cliente**

WSDL and Client Location

Specify the WSDL file of the Web Service.

Project:

Local File:

WSDL URL:

Specify a location for the client.

Project:

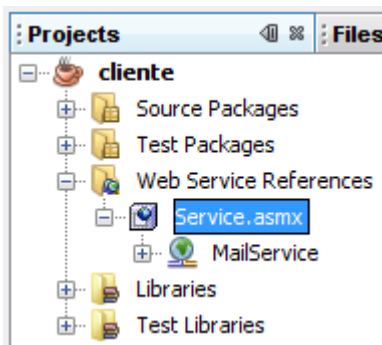
Package: ▼

Client Style: ▼

Tenga en cuenta que debe adaptar el valor del campo **WSDL URL** según su propia ubicación del servicio.

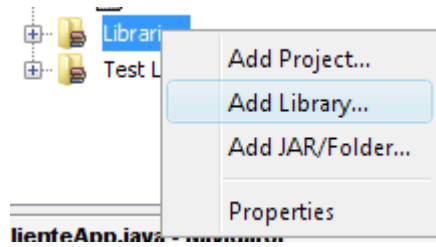
Presione el botón **Finish** para continuar.

Para confirmar que este paso se realizó exitosamente en el área del proyecto, bajo la rama **Web Service References** deberá aparecer el servicio recién referenciado: **Service.asmx**.

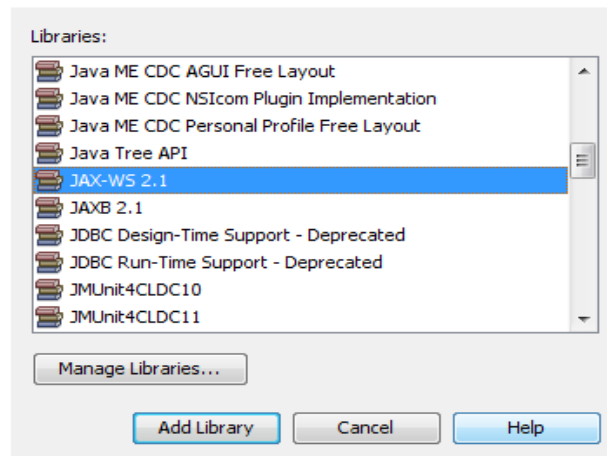


Agregue la librería **JAX-WS** al proyecto, para esto haga click derecho sobre la rama **Libraries** de la sección de **Projects** y elija la opción **Add Library ...**.

Desarrollo de un servicio web para el envío de correos utilizando C# y su correspondiente cliente consumidor utilizando Java,



Seleccione la librería **JAX-WS 2,1** y presione el botón **Add Library** para aceptar.



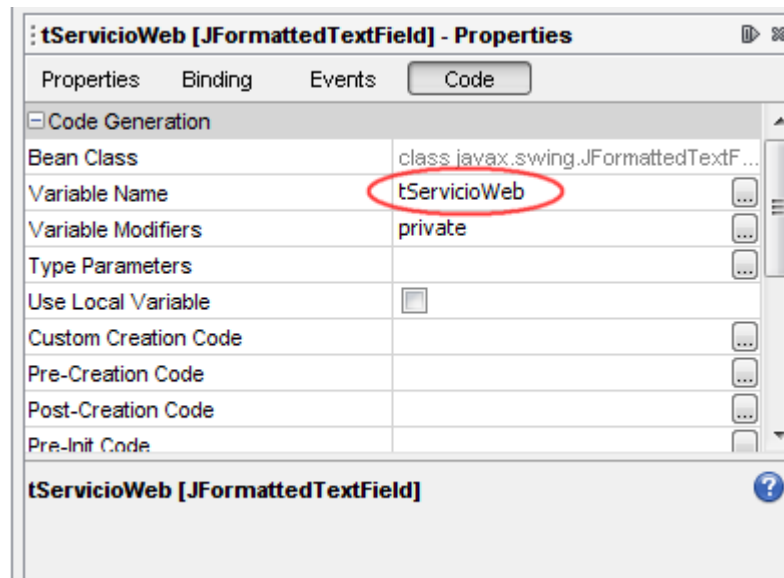
Edite el archivo **ClienteView.java** ubicado en la sección de **Projects** bajo el proyecto **cliente** en las ramas **Source Packages > cliente**. Utilice el modo **Design** (diseño) para editar gráficamente los componentes de la ventana.

Agregue los componentes necesarios para crear una interfaz de usuario como la mostrada a continuación.

Desarrollo de un servicio web para el envío de correos utilizando C# y su correspondiente cliente consumidor utilizando Java,

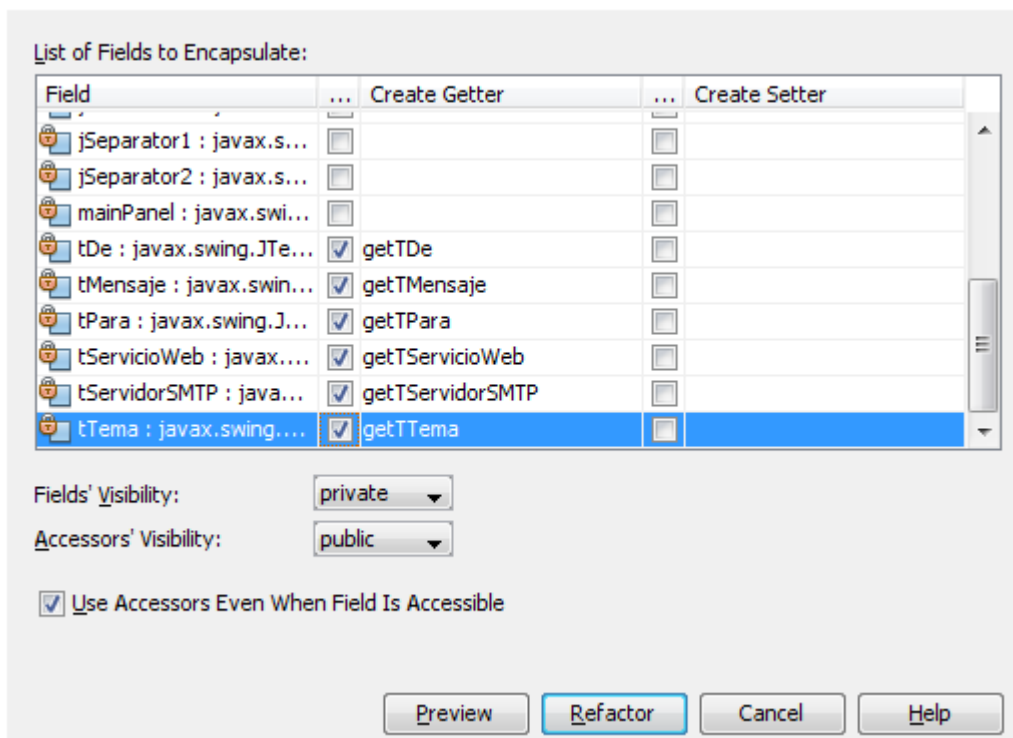


Los nombres especificados en la imagen son los nombres de los objetos **JtextField**. Para asignarlos seleccione el componente y en la sección de **Properties** modifique su valor en el campo **Variable Name**.



Cree los métodos **get** (obtener) para examinar los atributos de los JTextField recién creados: servicio web, servidor SMTP, de, para, tema y mensaje. Para esto abra el archivo **ClienteView.java** ubicado en el árbol de proyectos bajo la rama **Source Packages > cliente** y elija el modo **Source**.

En la vista del código elija el menú **Refactor > Encapsulate Fields ...** y seleccione la casilla de verificación de **Create Getter** (primera columna) para los atributos **tDe**, **tMensaje**, **tServicioWeb**, **tServidorSMTP** y **tTema** tal y como se muestra en la siguiente imagen. Presione el botón **Refactor** para continuar.



Modifique los métodos recién creados adaptándolos de la siguiente manera.

```
public String getTDe()
{
    return tDe.getText();
}

public String getTMensaje()
{
    return tMensaje.getText();
}

public String getTPara()
{
    return tPara.getText();
}
```

Desarrollo de un servicio web para el envío de correos utilizando C# y su correspondiente cliente consumidor utilizando Java,

```
public String getTServicioWeb()
{
    return tServicioWeb.getText();
}

public String getTServidorSMTP()
{
    return tServidorSMTP.getText();
}

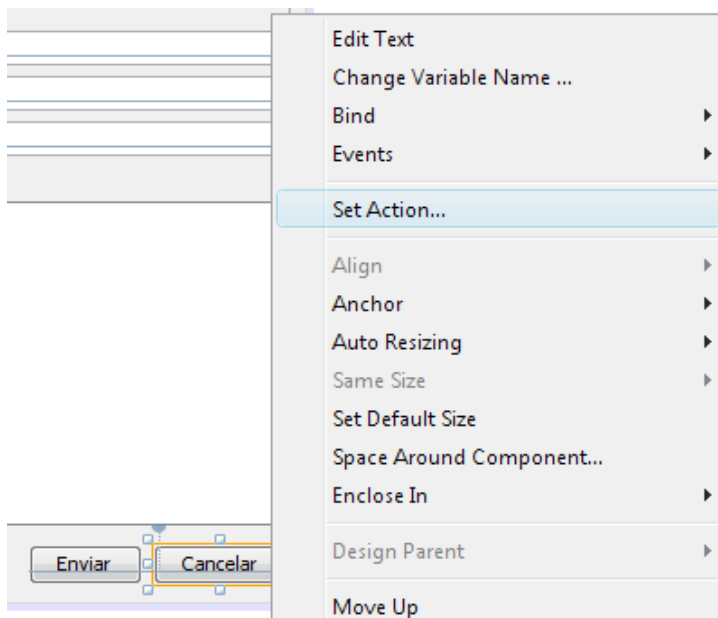
public String getTTema()
{
    return tTema.getText();
}
```

Cree el método **mostrarRespuesta** el cual estará encargado de mostrar el resultado, en éxito o fracaso del llamado al servicio web, el cual retorna una cadena de texto con el mensaje “OK” en éxito o un mensaje de error en fracaso.

```
public void mostrarRespuesta(boolean exito, String respuesta)
{
    String mensaje = ((exito) ? "Exito enviando el mensaje:" : "Fracaso enviando el mensaje:") +
        "\n\n" + respuesta;

    JOptionPane.showMessageDialog(this.getFrame(), mensaje);
}
```

Vuelva al modo **Design** del archivo **ClienteView.java** para asociar las acciones que se ejecutarán al presionarse los botones de **Enviar** y **Cancelar**, para ello de click derecho sobre el botón **Cancelar** y elija la opción **Set Action ...** del menú contextual.



Modifique los campos del diálogo emergente de la siguiente manera y presione el botón **Ok** para continuar.

Action: **Create New Action ...**

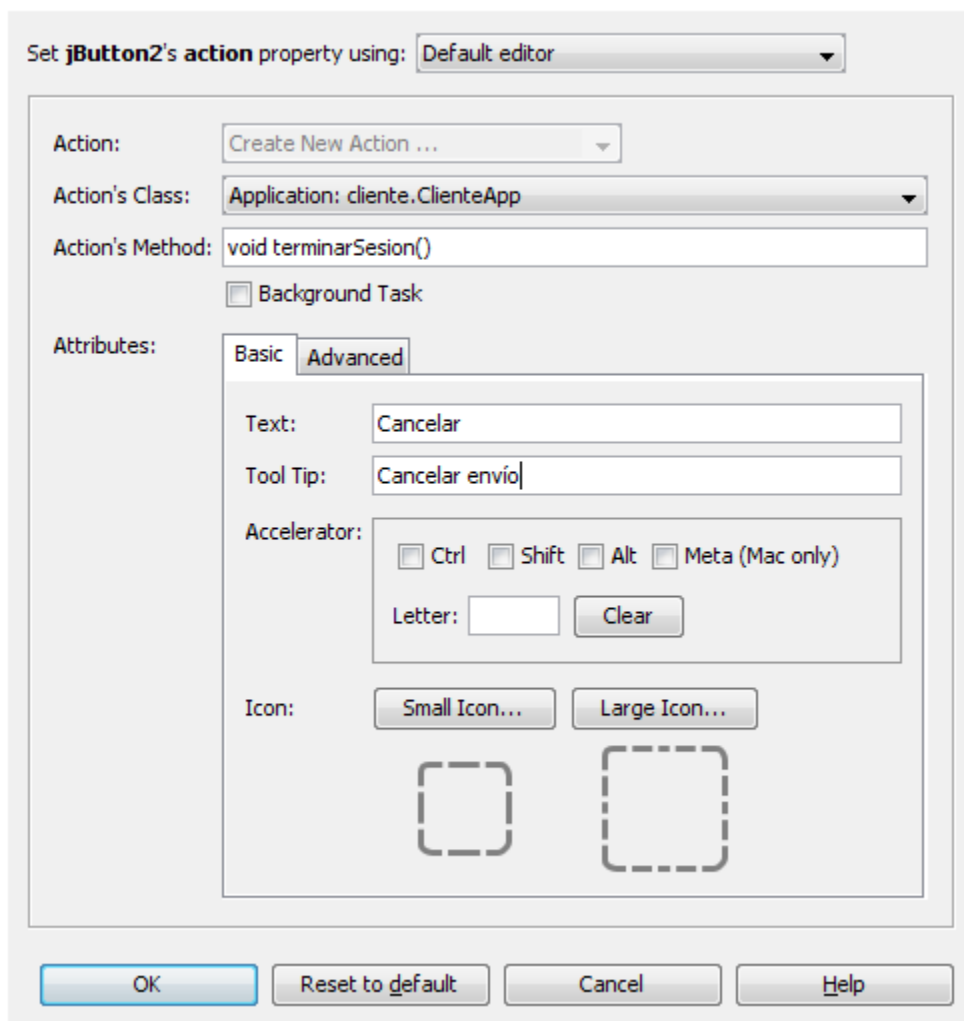
Action's Class: **Application.ClienteApp**

Action's Method: **terminarSesion**

Attributes:

Text: **Cancelar**

Tool Tip: **Cancelar envío**



El IDE abre automáticamente el archivo **ClienteApp.java** y agrega el método **terminarSesion**. Modifique su implementación tal y como se muestra a continuación.

Desarrollo de un servicio web para el envío de correos utilizando C# y su correspondiente cliente consumidor utilizando Java,

```
@Action
public void terminarSesion()
{
    this.exit();
}
```

Repita los mismos pasos ahora con el botón **Enviar** para modificar su comportamiento. De vuelta al modo **Design** del archivo ClienteView.java haga click derecho sobre el botón **Enviar** y seleccione la opción **Set Action ...** del menú contextual.

Modifique los campos del diálogo emergente de la siguiente manera y presione el botón **Ok** para continuar.

Action: **Create New Action ...**

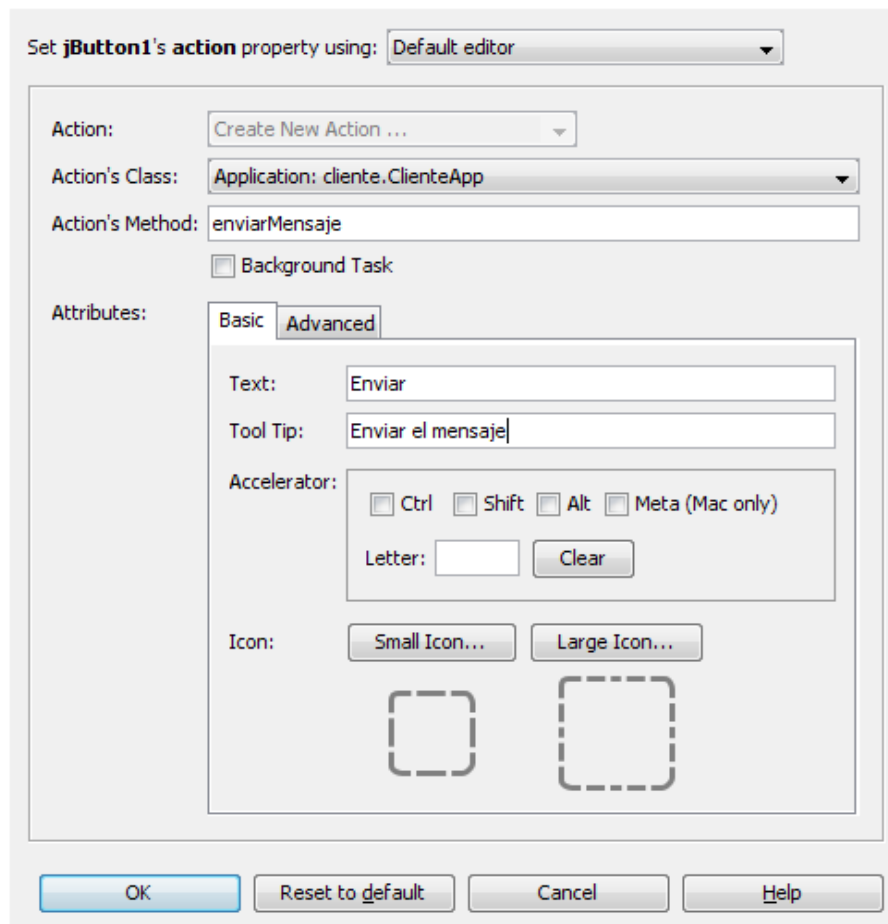
Action's Class: **Application.CienteApp**

Action's Method: **enviarMensaje**

Attributes:

Text: **Enviar**

Tool Tip: **Enviar el mensaje**



Desarrollo de un servicio web para el envío de correos utilizando C# y su correspondiente cliente consumidor utilizando Java,

El IDE abre automáticamente el archivo **ClienteApp.java** y agrega el método **enviarMensaje**. Modifique su implementación tal y como se muestra a continuación.

```
@Action
public void enviarMensaje()
{
    // Obtiene la información proporcionada por el usuario a través
    // del ClienteView.

    String wsService = ((ClienteView)this.getMainView()).getTServicioWeb();
    String smtpServer = ((ClienteView)this.getMainView()).getTServidorSMTP();
    String fromUser = ((ClienteView)this.getMainView()).getTDe();
    String toUser = ((ClienteView)this.getMainView()).getTPara();
    String subject = ((ClienteView)this.getMainView()).getTTema();
    String body = ((ClienteView)this.getMainView()).getTMensaje();

    // Determina si hubo o no éxito en el proceso del envío del mensaje

    boolean exito = false;

    // Almacena la respuesta que va a ser mostrada al cliente.

    String respuesta = "desconocido";

    try
    {
        // Crea la referencia al servicio web. El primer parámetro es el
        // URL donde se encuentra ubicado y el segundo es el nombre del
        // servicio que se va a invocar.

        cliente.MailService service = new cliente.MailService(new URL(wsService + "?WSDL"),
                                                                new QName("http://tempuri.org/", "MailService"));

        cliente.MailServiceSoap port = service.getMailServiceSoap();

        // Se realiza la solicitud de ejecución del método publicado en el
        // servicio web.

        respuesta = port.send(fromUser, toUser, subject, body, smtpServer);

        // Como todo salió bien así se indica en la variable de éxito.

        if (respuesta.equals("OK"))
            exito = true;
    }
    catch (Exception ex)
    {
        // Si sucede un problema, se almacena la cadena que lo describe
        // para mostrársela al usuario posteriormente.

        respuesta = ex.getLocalizedMessage();

        // Se modifica la variable de éxito para representar el fracaso.

        exito = false;
    }

    // Se muestra la respuesta obtenida durante el llamado al servicio
    // web al usuario final.

    ((ClienteView)this.getMainView()).mostrarRespuesta(exito, respuesta);
}
```

Desarrollo de un servicio web para el envío de correos utilizando C# y su correspondiente cliente consumidor utilizando Java,

Ejecutar el cliente y probar la aplicación.

Desde **NetBeans** presione el botón de **Run Main Project** (triángulo verde) o presione la tecla **F6**. Esto deberá compilar y ejecutar la aplicación si no se encontraron errores.

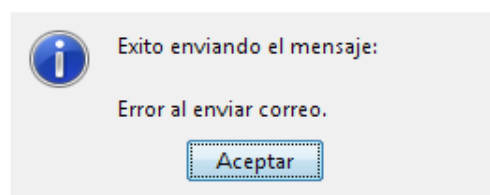
Especifique los campos de la aplicación con los correspondientes a su entorno y presione el botón **Enviar** para enviar un mensaje de prueba.



The image shows a dialog box titled "Servicio web" with the following fields and content:

- Servicio web:** `http://localhost:55587/servicio/Service.asmx`
- Servidor SMTP:** `localhost`
- De:** `jimezam@gmail.com`
- Para:** `destino@servidor.com`
- Tema:** `Mensaje de prueba`
- Mensaje:** `Este es un mensaje de prueba del servicio web.`
- Buttons: **Enviar** and **Cancelar**

Si el mensaje es enviado exitosamente se obtendrá el siguiente mensaje.



Desarrollo de un servicio web para el envío de correos utilizando C# y su correspondiente cliente consumidor utilizando Java,

En caso de producirse un problema se obtendrá un mensaje como el siguiente con la descripción del mismo.

