Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Organización de Lenguajes y Compiladores 2

Segundo Semestre 2009

Primer Proyecto de Laboratorio

Introducción

En la actualidad el internet ha tomado un gran auge en la vida de las personas, y son más personas las que desean aprender a construir sus propios sitios web aunque no sean personas dedicadas a la informática. Las personas que se encuentran más involucradas con los lenguajes de programación, saben que hay más allá de una página simple hecha con HTML puro. Por ello es importante que se sepa diferenciar entre las opciones que nos brinda HTML, PHP, JAVASCRIPT, entre otros; y poder obtener todas las utilidades de ellos para crear sitios web de mayor alcance y complejidad que permita abarcar un público mayor en la web.

El presente proyecto se encuentra orientado a realizar una aplicación que diseñe páginas web en un único lenguaje; para que luego sean traducidas a los lenguajes indicados para poder verse en los navegadores web. Todo este proceso de traducción se realizará en un servidor, con la idea de poder conectar varios proyectos en red y acceder a varios recursos de otras computadoras tales como bases de datos, servidores de correo, etc.

Objetivos

General:

- Combinar los conocimientos adquiridos en Compiladores 2 y en los otros cursos de sistemas, para crear un servidor web que contenga un compilador con sus fases completas para crear distintas páginas web.

Específicos:

- Crear una herramienta que permita el diseño de páginas web de una forma sencilla para el usuario.
- Diseñar y construir un compilador en línea que permita compilar archivos de entrada creados en java.
- Desarrollar la habilidad de conectar varias máquinas en red para que por medio del servidor creado puedan acceder a distintos recursos de otras máquinas distintas a la local.
- Desarrollar la habilidad del estudiante para elaborar proyectos largos, de una adecuada planificación para que aprendan la manera en la que tienen que trabajar.

Descripción General

El primer proyecto de laboratorio consiste en la elaboración de una herramienta que permita el diseño y creación de páginas web de una forma sencilla. Para ello, el proyecto se dividirá en dos partes: Aplicación de Escritorio y Servidor Web.

La aplicación de escritorio será la encargada del diseño de las páginas web, contendrá un drag & drop para que sea más fácil la creación de la página. También debe de tener una opción donde se visualice el código que se va creando y se pueda editar.

Al tener ya la página terminada, tendremos el código de la misma en Java, y se debe de traducir a HTML, CSS, PHP y JAVASCRIPT.

Para la traducción se implementará un servidor Web que tendrá instalada una aplicación, dicha aplicación será el compilador del proyecto.

Desde el servidor, se podrán subir los archivos de entrada, compilarlos y ejecutarlos si no hay ningún error.

Para generar todo esto se tienen varios requerimientos que se presentarán a continuación:

Características de la Aplicación

El proyecto consiste en dos partes, la primera parte es en relación a la aplicación. Como una pequeña introducción, la aplicación consistirá en un programa que sirva para el diseño de la página web, y para poder editar el código. El lenguaje que se manejará dentro de la aplicación será JAVA puro. La idea es que al abrir el código generado en la aplicación, desde un IDE como Netbeans el código pueda ejecutarse correctamente.

Interfaz Gráfica

La interfaz gráfica será parecida a la del IDE Netbeans, la idea es que los objetos puedan arrastrarse a la ventana de diseño, y al colocarse se llenen las propiedades del mismo. Por aparte se debe poder contar con una opción donde pueda apreciarse el código que se va generando, y donde se pueda modificar y agregarle más cosas.

Los componentes gráficos que deben de aparecer en la interfaz gráfica para poderse arrastrar son:

NOMBRE DE LOS COMPONENTES GRÁFICOS EN JAVA	PROPIEDADES
JLabel	ID Color de letra Tamaño de letra Texto Imagen Alineación Tipo (Título o Normal) Color de Fondo
JButton	ID Texto Alineación
JCheckBox	ID Texto Marcada o no Color
JRadioButton	ID Texto Marcada o no Color
JButtonGroup	ID Tipo (JcheckBox o JradioButton) Elementos que pertenecen a él
JComboBox	ID Contenido
JTextField	ID Texto Tamaño
JTextArea	ID Texto Tamaño
JPasswordField	ID Texto Tamaño
JTree	ID DefaultTreeModel

	Y cada uno de los DefaultMutableTreeNode indicando cual es el inicio.
JTable	ID Columnas Filas Texto Celda número:
JPanel	ID Ancho Alto Color de Fondo LineBorder Margen: EmptyBorder Borde y Margen combinados: CompoundBorder Bordes con esquinas redondeadas: roundedCorners
JFileChooser	ID Filtro de Archivo (extensiones, pueden ser más de 1)

Por aparte en la sección del código, el editor de texto debe tener un formato de colores de la siguiente manera:

- Errores: ROJO
- Palabras Reservadas: VERDE
- Comentarios: GRIS

Debe existir una ventana en donde se aprecien los errores que están en el código, dichos errores estarán identificados de la siguiente manera:

- Léxicos: AZUL
- Sintácticos: ANARANJADO
- Semánticos: CAFÉ

Deben poderse realizar las siguientes opciones por medio de la interfaz:

- Nuevo
- Abrir
- Guardar

- Guardar como
- Salir

Sección de Código de la Interfaz

Por aparte del código que se genere de cada uno de los objetos gráficos, existen otros códigos que se manejarán.

Los códigos son:

NOMBRE	CÓDIGO JAVA
JLabel	• Solo con texto: JLabel ID = new JLabel("texto"); • Solo con imagen: ImageIcon imagen = new ImageIcon("path imagen"); JLabel ID = new JLabel(imagen); • Texto e Imagen: ImageIcon imagen = new ImageIcon("path imagen"); JLabel ID = new JLabel("texto", imagen, alineación); • Para la alineación, si no se coloca se tomará como predeterminada LEFT. • Si en dado caso la alineación no se coloca en el constructor se agregará de la siguiente forma: JLabel ID = new JLabel("texto); ID.setHorizontalTextPosition(JLabel.CENTER); • Las opciones para la alineación son: - LEFT - CENTER - RIGTH Para una correcta funcionalidad, al enviarlas como parámetro deben de llevar la siguiente estructura: JLabel.OPCIÓN

	• Para colocarle color: ID.setForeground(Color.OPCIÓN); Las opciones para el color son las predeterminadas para java, así como se pueden crear colores de la forma en que java lo permite. • Para la opción de título o texto normal, el estudiante debe de elegir como manejarlo; tener en cuenta que pueden aparecer textos de 3 tamaños: 1, 2 y 3; por aparte del texto normal. • Para colocar el color de fondo: ID.setBackground(Color.OPCION);
JButton	JButton ID = new JButton("texto"; ID.setHorizontalTextPosition(AbstractButton.CENTER); • Las opciones de alineación son: - LEFT - CENTER - RIGTH Para una correcta funcionalidad al enviarlas como parámetro deben llevar la siguiente estructura: AbstractButton.OPCION
JCheckBox	JCheckBox ID = new JCheckBox("texto"); El color se maneja de la misma forma que el JLabel.
JRadioButton	JRadioButton ID = new JRadioButton("texto", Boolean); En el Boolean se colocará TRUE si se desea seleccionado, y FALSE si no. El color se maneja de la misma forma que el JLabel.
JButtonGroup	ButtonGroup ID = new ButtonGroup( ); ID.add(OPCION); La OPCION es el ID de algún JCheckBox o de un JRadioButton dependiendo de con cual de los dos estén trabajando.
JComboBox	JComboBox ID = new JComboBox( ) ; ID.addItem("texto de cada item");

JTextField

- Sin texto en el:
  JTextField ID = new JTextField(número);

- Con texto:

JTextField ID = new JTextField("texto", número);

JTextArea

- Sin texto en el:
  JTextArea ID = new JTextArea( );

- Con texto:

JTextArea ID = new JTextArea("texto");

JPasswordField

- JPasswordField ID = new JPasswordField( );

- JPasswordField ID = new JPasswordField("con contraseña");

JTree

Para el ejemplo se tomarán: abuelo, padre, hijo y tío:

➢ DefaultMutableTreeNode abuelo = new DefaultMutableTreeNode("abuelo");
DefaultTreeModel modelo = new DefaultTreeModel(abuelo);
JTree ID = new JTree(modelo);
DefaultMutableTreeNode padre = new DefaultMutableTreeNode("padre");
DefaultMutableTreeNode tio = new DefaultMutableTreeNode("tio");
DefaultMutableTreeNode hijo=new DefaultMutableTreeNode("hijo");
DefaultMutableTreeNode hija=new DefaultMutableTreeNode("hija");

JTable

➢ Se crea una tabla vacía y en la interfaz gráfica se le agregan los datos:

JTable ID = new JTable(filas,columnas);

➢ Para colocar el texto en cada celda se utilizará el código:

ID.editCellAt(fila, columna);

JPanel

➢ Para crear el panel:

JPanel ID = new JPanel( );
ID.setSize(ancho,alto);

	ID.add(objeto dentro del panel); ➢ Para el color de fondo: ID.setBackground(Color.OPCION); Las opciones de colores se manejan de la misma manera que en el JLabel. Más adelante se encuentra una descripción más amplia y específica acerca de los bordes en JPanel.

JFileChooser	`JFileChooser chooser = new JFileChooser();`
	`ExampleFileFilter filter = new`
	`ExampleFileFilter();`
	`filter.addExtension("jpg");`
	`filter.addExtension("gif");`
	`filter.setDescription("JPG & GIF Images");`
	`chooser.setFileFilter(filter);`
	`int returnVal =`
	`chooser.showOpenDialog(parent);`
	`if(returnVal ==`
	`JFileChooser.APPROVE_OPTION) {`
	`System.out.println("You chose to`
	`open this file: " +`
	
	`chooser.getSelectedFile().getName());`
	`}`

Se debe de tener en cuenta que debe de existir un JFrame para poder crear todo lo anterior.

Dichos elementos son los mínimos que deben de aparecer, el estudiante puede agregar más para que su aplicación sea más funcional.

Otros Códigos:

Por aparte de los códigos de los componentes presentados anteriormente, se manejarán los siguientes códigos de Java:

- Conexión a Base de Datos MySQL
  - Consultar Datos
  - Insertar Datos
  - Eliminar Datos
  - Modificar Datos
- Envío de mails
- Subir archivos

Ciclos:

Se podrán manejar los siguientes ciclos:

- For
- While
- Do-while

Condicionales:

Se contarán con las siguientes condiciones:

- If
- Else
- Else if
- Case

Operaciones:

- Suma
- Resta
- Multiplicación
- División
- Módulo
- Concatenación
- Substring
- IndexOf
- Convertir a Cadena
- Convertir a Entero

Declaración de Variables:

Se podrán declarar variables una por una, así como también en lista; también puede que no tengan valor inicial o que si lo tengan. Pueden existir los siguientes tipos:

- Cadena
- Entero
- Booleano

- Flotantes
- Vectores
- Matrices de 2 dimensiones

Los vectores y matrices pueden ser de objetos de clases declaradas en el mismo archivo.

Manejo de Eventos:

Para tener un buen manejo en JButton, JCheckBox, JRadioButton, etc; se han restringido los eventos que pueden utilizar para cada uno:

- JButton:
  - al hacer click
- JCheckBox
  - Al hacer click
- JRadioButton
  - Al hacer click
- JComboBox
  - Al hacer click
  - Al cambiar el ítem seleccionado

Métodos y Funciones

Se podrán manejar métodos y funciones de la manera en que java lo permite.

Librerías

Se debe de tomar en cuenta que para utilizar los elementos de las librerías de java, se debe importar las mismas. Al inicio de cada clase:

import librería

Paquetes

Se pueden tener distintos paquetes; y se debe de indicar a qué paquete pertenece cada clase; por ejemplo para las imágenes utilizadas, éstas pueden aparecer en otro paquete.

Clases

Habrán varios archivos para manejar las clases, una clase por archivo; podrán crearse objetos de otras clases que vayan a compilarse al mismo tiempo.

Para el efecto de los anteriores, el estudiante debe investigar el código que funcione para cada uno; si hay más de una opción se aplicarán todas las opciones.

Comentarios

Se deben poder reconocer los comentarios.

El estudiante puede agregar las opciones que considere necesarias para el buen funcionamiento de su aplicación gráfica, puede agregar más opciones de java si así lo requiere.

Servidor Web

Existirá un servidor web al que se le instalará una aplicación que permita subir archivos para poderlos trabajar desde el servidor.

Características del Servidor Web

Luego de haber generado el archivo JAVA en la aplicación; debe de existir un servidor web con las siguientes características:

Interfaz Gráfica:

El servidor web estará creado en JSP y debe de contar con el siguiente menú como mínimo:

- Inicio
- Subir Archivo
- Compilar Archivo
- Ver código de salida
- Ver salida real:
    - Este punto es donde ya debe verse la página funcionando tal y como estaba en la aplicación de escritorio.
    - Por tanto el compilador instalará las páginas en el Servidor Web para que puedan ser accedidas desde otra máquina en la red.
- Errores Léxicos
- Errores Sintácticos
- Errores Semánticos

Descripción:

El servidor web estará construido en JSP's y su función es la del compilador del proyecto.

Por ser el compilador, quiere decir que el archivo será compilado en el servidor, no puede ser compilado en ninguna otra aplicación.

Servidor de Correo

Para poder implementar la opción de enviar correo, deben de tener implementado un servidor de correo SMTP. Con la idea de que al enviar correos se puedan enviar a la máquina local utilizando la dirección LOCALHOST.

Conectarse en Red:

El día de la calificación, los auxiliares realizarán grupos al azar, con la idea de que se conecten en red y se puedan probar las siguientes funciones:

- Conexión a Base de Datos
- Envío de correos
- Subir archivos

Se calificará de forma individual y de forma grupal.

Para la implementación del servidor de correo, cada estudiante deberá de investigar y de probarlo en su máquina para estar seguros de que les funciona.

Deben de llevar los materiales necesarios para conectarse en red:

- Un router
- Cable de red (en caso de no tener wireless)

Traducción

Al tener el archivo de JAVA generado, se deben obtener los siguientes archivos:

Hoja de Estilo CSS

Las hojas de estilo son un mecanismo que describen cómo se mostrará un documento en pantalla, en este caso, las páginas web. El CSS a utilizar debe ser CSS3.

La hoja de estilo podrá decirnos los siguientes datos:

- Color de Texto
- Color de Títulos
- Tipo de Letra
- Color de Fondo
- Alineación del texto

Los códigos de donde se sacarán la información anterior provienen de:

- Color de Texto en el label.
- Tipo de letra en el label.
- Alineación de los objetos
- Color de fondo en el panel.

Cómo generar el CSS?

Los códigos que deben utilizar para generar el CSS son:

- Color:
  ID{ color: COLOR;}
  Lista ID{color:COLOR;}

- Fondo:
  ID{ background-color: COLOR;}

- Letra:
  ID{Font-family: Tipo de Letra;}

  Se debe tener en cuenta que para el tipo de letra, el alumno debe de guardar el tipo de letra en el lugar predeterminado, y colocar únicamente las letras disponibles; si esto no está realizado se verá penalizado en la calificación.

- Alineación:
  ID{text-align:OPCION}

Los colores pueden ser todos los de HTML, en base al color que se encuentra en el archivo JAVA.

El orden del archivo CSS no importa.

Las funciones deben de venir agrupadas según el ID, es decir si se tiene como ID "h1" no puede aparecer un h1 para el color de letra y uno para el color de fondo; las funciones deben aparecer agrupadas.

Código HTML

Es el lenguaje predominante para la construcción de páginas web; se escribe en base a etiquetas; para el caso del proyecto, los siguientes objetos serán traducidos a html:

NOMBRE EN JAVA	NOMBRE EN HTML
JLabel	Dependiendo de la opción que se ha marcado al crearlo, puede llevar etiqueta o no.
JButton	Botón `<INPUT type="button" value="texto a mostrar">`
JCheckBox	CheckBox <input type="checkbox" name="identificador" value="ejemplo"> texto a mostrar

JRadioButton	Radio Button <input type="radio" name="identificador" value="ejemplo" > texto a mostrar
JComboBox	Combo Box <select name="menu"> <option value="0" selected>Escoger un valor</option> <option value="1">one</option> <option value="2">two</option> <option value="3">three</option> </select>
JTextField y JTextArea	TextArea <TEXTAREA name="eltexto" rows="20" cols="80"> Primera línea del texto inicial. Segunda línea del texto inicial. </TEXTAREA> Se debe tener en cuenta que para el jtextfield el número de filas debe ser 1, y el de columnas del tamaño del texto.
JPasswordField	<input type="password" size="25">
JTable	Tabla <TABLE BORDER="1"> <TR> <TH>Cabecera 1</TH> <TH>Cabecera 2</TH> <TH>Cabecera3</TH> </TR> <TR> <TD>Dato 1</TD> <TD>Dato 2</TD> <TD>Dato 3</TD> </TR> <TR> <TD>Dato 4</TD> <TD>Dato 5</TD> <TD>Dato 6</TD> </TR> </TABLE>

JTree	El Jtree pasará a ser lista enlazada, teniendo en cuenta que la estructura debe de ser parecida a la del árbol original. - Uno - Uno - Dos - Tres - Dos - Tres
JPanel	El panel pasará a ser un DIV dentro de html: Tag HTML div. Los elementos y atributos desaprobados están usualmente marcados en rojo. Esto ayuda a identificarlos.
JFileChooser	<input type=file size=60 name="file1">

Explicación más detallada JPanel

Para generar un bloque DIV en HTML equivalente a un javax.swing.JPanel en Java, se deberá hacer uso de los siguientes métodos de la clase javax.swing.JPanel.

Dimensiones y posicionamiento del bloque

Método: javax.swing.JPanel.setBounds(x,y,anchura,altura);

EJEMPLO 1

JAVA

```
JPanel panel1 = new JPanel();
panel1.setBounds(100,150,500,300);
```

CSS

```
#panel1 {
        position: absolute;
        left: 100px;
        top: 150px;
        width: 500px;
        height: 300px;
}
```

HTML

```
<div id="panel1" name="panel1">
        <!—CONTENIDO DEL DIV -->
</div>
```

Color de fondo del bloque

Método: javax.swing.JPanel.setBackground(java.awt.Color(int,int,int));

EJEMPLO 2

JAVA

```
JPanel panel1 = new JPanel();
panel1.setBounds(100,150,500,300);

Color color = new Color(0,0,255); //azul
panel1.setBackground(color);
```

CSS

```
#panel1 {
        position: absolute;
        left: 100px;
        top: 150px;
```

```
        width: 500px;
        height: 300px;
        background-color: rgb(0,0,255);
}
```

HTML

```
<div id="panel1" name="panel1">
        <!—CONTENIDO DEL DIV -->
</div>
```

Color de fondo transparente para el bloque

Método: javax.swing.JPanel.setBackground(java.awt.Color(float,float,float,float));

EJEMPLO 3

JAVA

```
JPanel panel1 = new JPanel();
panel1.setBounds(100,150,500,300);

Color color = new Color(0,0,0,0); //transparente
panel1.setBackground(color);
```

CSS

```
 #panel1 {
        position: absolute;
        left: 100px;
        top: 150px;
        width: 500px;
        height: 300px;
        background-color: transparent;
}
```

HTML

```
<div id="panel1" name="panel1">
        <!—CONTENIDO DEL DIV -->
</div>
```

Bordes del bloque

Método: javax.swing.JPanel.setBorder(javax.swing.border.Border);

EJEMPLO 4

JAVA

```
JPanel panel1 = new JPanel();
panel1.setBounds(100,150,500,300);
```

```
//borde color amarillo con 3 pixeles de grosor
Color color = new Color(255,255,0);
Border border = BorderFactory.createLineBorder(color,3);
panel1.setBorder(border);
```

CSS

```
#panel1 {
        position: absolute;
        left: 100px;
        top: 150px;
        width: 500px;
        height: 300px;
        border-color: rgb(255,255,0);
        border-width: 3px;
        border-style: solid;
}
```

HTML

```
<div id="panel1" name="panel1">
        <!—CONTENIDO DEL DIV -->
</div>
```

Relleno del bloque

Método: javax.swing.JPanel.setBorder(javax.swing.border.Border);

EJEMPLO 5

JAVA

```
JPanel panel1 = new JPanel();
panel1.setBounds(100,150,500,300);

//relleno con 10px arriba, 20px izquierda, 30px abajo, 40px derecha
Border border = BorderFactory. createEmptyBorder (10, 20, 30, 40);
panel1.setBorder(border);
```

CSS

```
#panel1 {
        position: absolute;
        left: 100px;
        top: 150px;
        width: 500px;
        height: 300px;
        padding-top: 10px;
        padding-left: 20px;
        padding-bottom: 30px;
        padding-right: 40px;
}
```

HTML

```
<div id="panel1" name="panel1">
        <!—CONTENIDO DEL DIV -->
</div>
```

Bordes y relleno combinados en un bloque

Método: javax.swing.JPanel.setBorder(javax.swing.border.Border);

EJEMPLO 6

JAVA

```
JPanel panel1 = new JPanel();
panel1.setBounds(100,150,500,300);

//borde color amarillo con 3 pixeles de grosor
Color color = new Color(255,255,0);
Border border = BorderFactory.createLineBorder(color,3);

//relleno con 10px arriba, 20px izquierda, 30px abajo, 40px derecha
Border padding = BorderFactory. createEmptyBorder (10, 20, 30, 40);

//combinación de borde y relleno
        Border combinacion =
BorderFactory.createCompoundBorder(border,padding);

        panel1.setBorder(combinacion);
```

CSS

```
 #panel1 {
        position: absolute;
        left: 100px;
        top: 150px;
        width: 500px;
        height: 300px;
        border-color: rgb(255,255,0);
        border-width: 3px;
        border-style: solid;
        padding-top: 10px;
        padding-left: 20px;
        padding-bottom: 30px;
        padding-right: 40px;
}
```

HTML

```
<div id="panel1" name="panel1">
        <!—CONTENIDO DEL DIV -->
```

```
</div>
```

Bordes redondeados del bloque

Método: javax.swing.JPanel.setBorder(javax.swing.border.LineBorder);

EJEMPLO 7

JAVA

```
JPanel panel1 = new JPanel();
panel1.setBounds(100,150,500,300);

//borde color amarillo con 3 pixeles de grosor
Color color = new Color(255,255,0);
LineBorder border = new LineBorder(color,3,true); /*true indica que las esquinas son redondeadas*/
panel1.setBorder(border);
```

CSS

```
#panel1 {
        position: absolute;
        left: 100px;
        top: 150px;
        width: 500px;
        height: 300px;
        border-color: rgb(255,255,0);
        border-width: 3px;
        border-style: solid;
        -webkit-border-radius: 10px;
-moz-border-radius: 10px;
}
```

HTML

```
<div id="panel1" name="panel1">
        <!—CONTENIDO DEL DIV -->
</div>
```

Código en PHP

PHP es un lenguaje diseñado para la creación de páginas web dinámicas. En el proyecto, los elementos traducidos de Java a PHP serán:

Manejo de Bases de Datos MySQL:

- Conexión a Base de Datos:

  El código presentado es un ejemplo, a la hora de la traducción el código cambiará según el código Java.

```
<?php
function Conectarse()
{
   if
(!($link=mysql_connect("localhost","usuario","Password")))
   {
      echo "Error conectando a la base de datos.";
      exit();
   }
   if (!mysql_select_db("base_datos",$link))
   {
      echo "Error seleccionando la base de datos.";
      exit();
   }
   return $link;
}

$link=Conectarse();
echo "Conexión con la base de datos conseguida.<br>";

mysql_close($link); //cierra la conexion
?>
```

- Consultar Datos:

  La forma de presentar los datos dependerá de la forma en que éstos sean presentados en java.

```
 $result=mysql_query("select * from prueba",$link);
```

- Insertar Registros:

```
mysql_query("insert into prueba (Nombre,Apellidos) values
('$nombre','$apellidos')",$link);
```

- Borrar Registros:

```
mysql_query("delete from prueba where ID_Prueba = $id",$link);
```

- Modificar Registros:

  Para modificar un registro, se debe eliminar el registro anterior y guarder el Nuevo.

Envío de eMails:

Para el envío de mails, se debe investigar la manera de configurar PHP para poderlo conectar con el servidor SMTP que se indicó anteriormente.

Se debe utilizar PHPMailer; pueden encontrarlo en el siguiente link:

http://sourceforge.net/projects/phpmailer/files/

ya sea para Windows o Linux.

Luego deben de configurarlo (investigar cómo) para poder utilizarlo.

Finalmente pueden enviar mails utilizando las siguientes línea:

```
$ mail = new PHPMailer ();
$ mail-> De senderemail = $, $ mail-> FromName = $ sendername;
$ mail-> AddAddress ($ receiveremail, receivername dólares);
/ / Fill Nombre de Usuario y Contraseña para servidores que requieren
autenticación $ mail-> El nombre de usuario = $ smtp username, $ mail->
Password = $ smtp password; / / nombre del servidor SMTP $ mail-> Host = $
smtp_server, $ mail-> Mailer = "smtp";
$ mail-> Subject = $ mail_subject $ Mail-> Body = $ mail_body; if ($ mail->
Enviar ()) $ resultados = 'Mensaje de error "; más $ resultados =' Exito
mensaje ';
```

Subir Archivos:

Para subir archivos se necesitan dos códigos:

CODIGO DE SELECCIÓN:

```
<form action="upload.php" method="post" enctype="multipart/form-
data">
  <input name="archivo" type="file" size="35" />
  <input name="enviar" type="submit" value="Upload File" />
  <input name="action" type="hidden" value="upload" />
</form>
```

CÓDIGO DE SUBIDA:

```
$status = "";
if ($ POST["action"] == "upload") {
    // obtenemos los datos del archivo
    $tamano = $ FILES["archivo"]['size'];
    $tipo = $_FILES["archivo"]['type'];
    $archivo = $ FILES["archivo"]['name'];
    $prefijo = substr(md5(uniqid(rand())),0,6);

    if ($archivo != "") {
        // guardamos el archivo a la carpeta files
        $destino =  "files/".$prefijo."_".$archivo;
        if (copy($_FILES['archivo']['tmp_name'],$destino)) {
            $status = "Archivo subido: <b>".$archivo."</b>";
        } else {
            $status = "Error al subir el archivo";
        }
    } else {
        $status = "Error al subir archivo";
    }
}
```

Código en JavaScript:

Para el manejo de la lógica de la página, se utilizará javascript.

El código javascript es similar a código java, lo único es que se le deben agregar las opciones necesarias para que el código sea válido dentro de la página.

El código que debe de pasar de java a javascript es el siguiente:

- Ciclos
  - For
  - While
  - Do-while

- Condicionales
  - If
  - Else
  - Else If
  - Case

- Operaciones
  - Suma
  - Resta
  - Multiplicación
  - División
  - Concatenación
  - Substring
  - IndexOf
  - Convertir a cadena
  - Convertir a Entero

- Manejo de Variables
  - Cadena
  - Entero
  - Booleano
  - Flotantes
  - Vectores
  - Matrices de 2 dimensiones

Navegadores que deben soportar las páginas

- Internet Explorer 6, 7 y 8.
- Firefox 3x
- Google Chrome 2x
- Opera 9x

Lenguajes y Herramientas a Utilizar

La construcción del proyecto se realizará con los siguientes lenguajes y herramientas:

- **Aplicación de Inicio:**
  La aplicación inicial, que será la encargada de poder realizar la interfaz gráfica y darle los detalles al lenguaje de entrada JAVA, será construida en:

  Microsoft Visual Studio .NET 2008 o 2005, lenguaje C#

- De dicha aplicación se obtendrá el archivo de entrada JAVA, que será el que se subirá al servidor.

- **Servidor:**

  Como ya se mencionó antes, el servidor será implementado en JSPs, utilizando el lenguaje

  JAVA

  Deben de trabajar con las siguientes herramientas:

  Para la elaboración de los JSPs:

  - ✓ Oracle JDeveloper 11g

  Para el servidor J2EE

  - ✓ WebLogic 10x

  Servidor HTML y PHP

  - ✓ Apache HTTP Server 2.2

  Como el servidor es en otras palabras el compilador, deben de utilizar las siguientes herramientas para el análisis:

  - ✓ ANÁLISIS LÉXICO Y SINTÁCTICO:
    Gold Parser

- **Servidor de Correo SMTP:**

  Para el servidor de correo deben de utilizar

  QKSOFT SMTP Server

  Disponible en el siguiente link:
  http://www.qksoft.com/download.html

- **Cliente de Correos**:

  Microsoft Outlook Express

- **Envío de correos PHP:**

  Para el envío de correos deben de utilizar

  PHP MAILER

  Disponible en el siguiente link:
  http://sourceforge.net/projects/phpmailer/files/

Se les recuerda que los archivos JAVA generados, deben poderse ejecutar sin problemas desde cualquier IDE de java, o como archivos java puros.
Por tanto deben de tener instaladas las herramientas que permitan que los archivos funcionen correctamente.

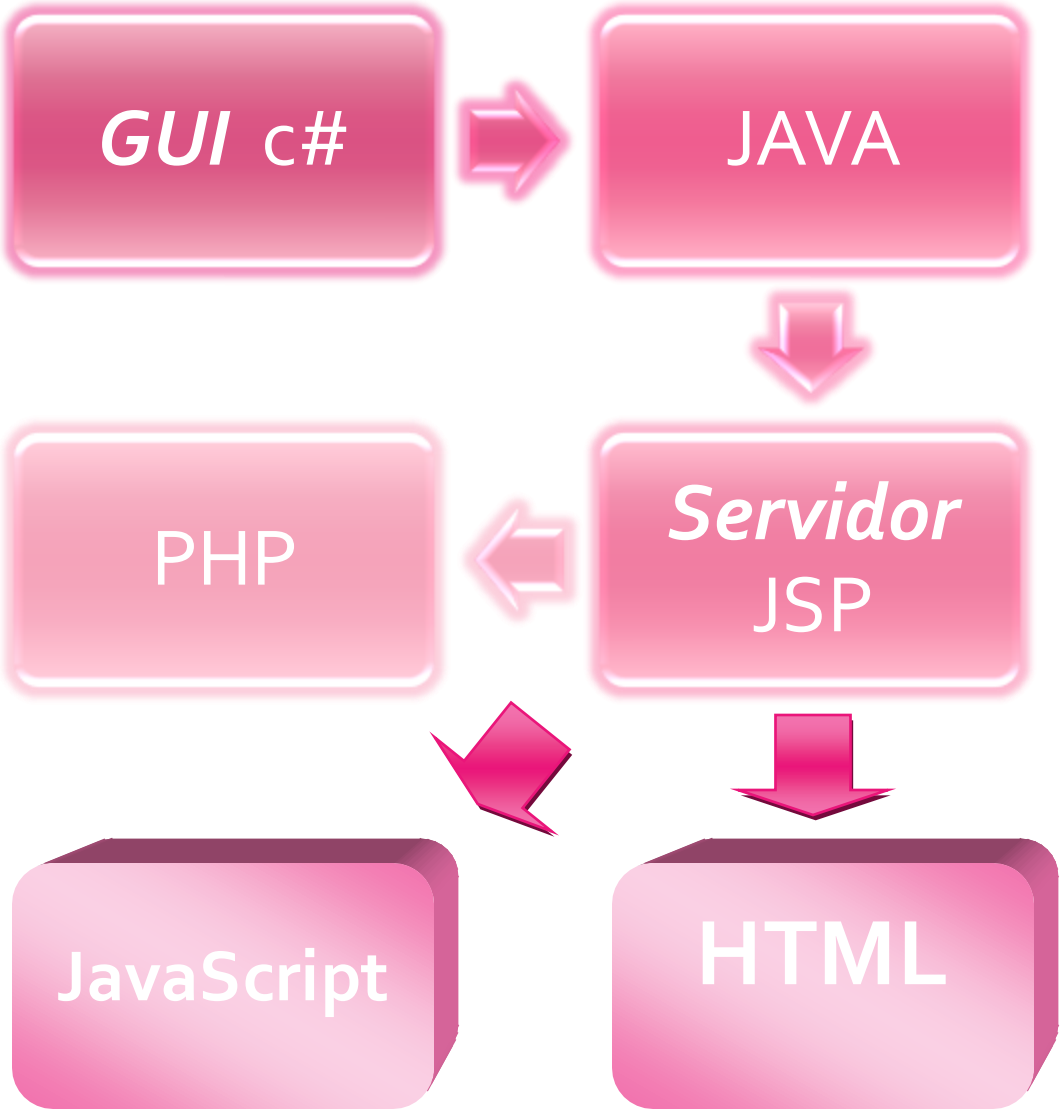
- **JDBC para la conexión con la base de datos MySQL desde Java**

  El JDBC para conectarse con bases de datos MySQL desde Java está disponible en:
  http://dev.mysql.com/downloads/connector/j/5.1.html

- **Herramientas para el envío de mails desde JAVA**

  - API Java Mail: http://java.sun.com/products/javamail/downloads/index.html
  - JAF Activation: http://java.sun.com/products/javabeans/glasgow/jaf.html

Diagrama general del proyecto



Datos Importantes

La calificación del proyecto es personal y deberá realizarse el día acordado para su respectiva entrega y calificación.

La aplicación de conocimientos correspondientes a cursos de semestres anteriores o el actual, es indispensable para realizar una parte o la totalidad de la práctica o los proyectos. Si existe alguna duda al respecto, se pueden dirigir a los auxiliares a quienes les pueden preguntar acerca de sus dudas.

Copias en cualquiera de las actividades teóricas o prácticas tendrán una nota de cero puntos y serán reportados al catedrático titular de su sección y a la Escuela de Ciencias y Sistemas para su respectiva sanción.

Para comunicarse con los auxiliares se estará utilizando la Universidad Virtual.

Todos los estudiantes (tanto los de la clase y laboratorio) deberán estar asignados en la Universidad Virtual en su respectiva sección del curso.

El laboratorio se aprueba con 61%.

Se deben de utilizar los lenguajes y herramientas indicados, de caso contrario se tendrá 0 de nota.

Copias de gramáticas y código, totales o parciales tienen 0 y serán reportados al catedrático de su sección y a la Escuela de Ciencias y Sistemas.

Código bajado de internet tiene 0.

Gramática bajada de internet tiene 0.

Fecha de entrega: 25 de septiembre 2009, a las 9 am en el lugar que indiquen los auxiliares.

Para tener derecho a entrega y calificación de proyecto, los alumnos deberán cumplir con la entrega de las fases del proyecto indicadas en este documento. Quien no las entregue no tiene derecho de calificación del proyecto.

Fases de entrega

El proyecto se ha dividido en pequeñas fases, para que vayan avanzando con el proyecto.

Dichas fases se entregarán en el formato que se les indicará por la UV, en PDF por correo a la siguiente dirección:

tareas.compi2@gmail.com

La entrega es antes de media noche de las fechas indicadas, correos recibidos después no serán válidos.

Las fases y fechas de entrega de cada una son:

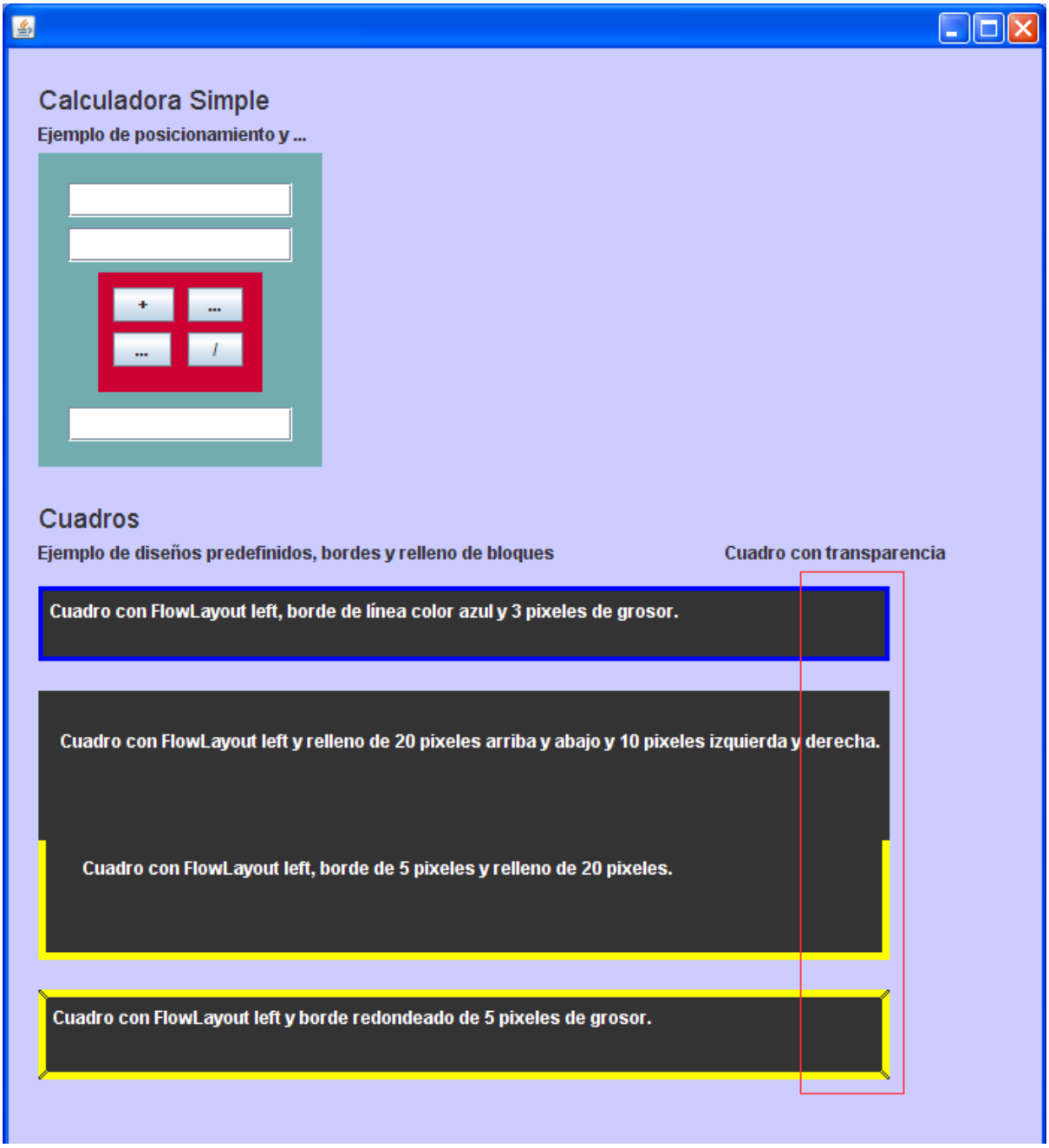
Fecha de Entrega	Descripción Fase
Miércoles 11 de agosto 2009	Gramática JAVA completa
Lunes 24 de agosto 2009	Interfaz gráfica construida IDE, gramática con acciones para levantar un archivo .java desde el IDE
Lunes 31 de agosto 2009	Creación completa de la parte web
Lunes 7 de septiembre 2009	Gramática con acciones para html y css completos.
Lunes 14 de septiembre 2009	Gramática con acciones para php y css completos
Lunes 21 de septiembre 2009	Gramática con acciones para javascript, configuraciones de red, correo y base de datos.
Viernes 25 de septiembre 2009	Entrega final proyecto

Ejemplos

A continuación se les presentarán 2 ejemplos detallados del archivo de entrada y las salidas del mismo.

Archivo de entrada 1

Interfaz:



Código Java:

```
package ejemplohtml;

public class CalculadoraFrame extends javax.swing.JFrame {
    private double valor1;
    private double valor2;

    public CalculadoraFrame() {
        initComponents();
        setSize(700,800);
    }

    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        jPanel3 = new javax.swing.JPanel();
        lienzo = new javax.swing.JPanel();
        jPanel1 = new javax.swing.JPanel();
        resultado = new javax.swing.JTextField();
        jPanel4 = new javax.swing.JPanel();
        suma = new javax.swing.JButton();
        resta = new javax.swing.JButton();
        multiplicacion = new javax.swing.JButton();
        division = new javax.swing.JButton();
        operando1 = new javax.swing.JTextField();
        operando2 = new javax.swing.JTextField();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jPanel9 = new javax.swing.JPanel();
        jPanel5 = new javax.swing.JPanel();
        jLabel5 = new javax.swing.JLabel();
        jPanel6 = new javax.swing.JPanel();
        jLabel6 = new javax.swing.JLabel();
        jPanel7 = new javax.swing.JPanel();
        jLabel7 = new javax.swing.JLabel();
        jPanel8 = new javax.swing.JPanel();
        jLabel8 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();

        javax.swing.GroupLayout jPanel3Layout = new
javax.swing.GroupLayout(jPanel3);
        jPanel3.setLayout(jPanel3Layout);
        jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
            .addGap(0, 100, Short.MAX_VALUE)
        );
        jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
            .addGap(0, 100, Short.MAX_VALUE)
        );
```

```
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setBounds(new java.awt.Rectangle(0, 0, 800, 600));

        lienzo.setBackground(new java.awt.Color(204, 204, 255));
        lienzo.setLayout(null);

        jPanel1.setBackground(new java.awt.Color(114, 172, 176));
        jPanel1.setLayout(null);

        resultado.setFont(new java.awt.Font("Arial", 0, 14));
        resultado.setHorizontalAlignment(javax.swing.JTextField.RIGHT);
        jPanel1.add(resultado);
        resultado.setBounds(20, 170, 150, 23);

        jPanel4.setBackground(new java.awt.Color(204, 0, 51));
        jPanel4.setLayout(null);

        suma.setText("+");
        suma.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                sumaActionPerformed(evt);
            }
        });
        jPanel4.add(suma);
        suma.setBounds(10, 10, 41, 23);

        resta.setText("-");
        resta.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                restaActionPerformed(evt);
            }
        });
        jPanel4.add(resta);
        resta.setBounds(60, 10, 37, 23);

        multiplicacion.setText("x");
        multiplicacion.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                multiplicacionActionPerformed(evt);
            }
        });
        jPanel4.add(multiplicacion);
        multiplicacion.setBounds(10, 40, 39, 23);

        division.setText("/");
        division.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                divisionActionPerformed(evt);
            }
        });
        jPanel4.add(division);
        division.setBounds(60, 40, 37, 23);

        jPanel1.add(jPanel4);
        jPanel4.setBounds(40, 80, 110, 80);

        operando1.setFont(new java.awt.Font("Arial", 0, 14));
        operando1.setHorizontalAlignment(javax.swing.JTextField.RIGHT);
        jPanel1.add(operando1);
```

```
        operando1.setBounds(20, 20, 150, 23);

        operando2.setFont(new java.awt.Font("Arial", 0, 14));
        operando2.setHorizontalAlignment(javax.swing.JTextField.RIGHT);
        jPanel1.add(operando2);
        operando2.setBounds(20, 50, 150, 23);

        lienzo.add(jPanel1);
        jPanel1.setBounds(20, 70, 190, 210);

        jLabel1.setFont(new java.awt.Font("Arial", 0, 18));
        jLabel1.setText("Cuadros");
        lienzo.add(jLabel1);
        jLabel1.setBounds(20, 300, 170, 30);

        jLabel2.setText("Ejemplo de posicionamiento y lógica");
        lienzo.add(jLabel2);
        jLabel2.setBounds(20, 50, 180, 14);

        jLabel3.setText("Ejemplo de diseños predefinidos, bordes y relleno
de bloques");
        lienzo.add(jLabel3);
        jLabel3.setBounds(20, 330, 570, 14);

        jLabel4.setFont(new java.awt.Font("Arial", 0, 18));
        jLabel4.setText("Calculadora Simple");
        lienzo.add(jLabel4);
        jLabel4.setBounds(20, 20, 170, 30);

        jPanel9.setBackground(new java.awt.Color(0,0,0,0));
        jPanel9.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(255, 51, 51)));

        javax.swing.GroupLayout jPanel9Layout = new
javax.swing.GroupLayout(jPanel9);
        jPanel9.setLayout(jPanel9Layout);
        jPanel9Layout.setHorizontalGroup(

jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
            .addGap(0, 68, Short.MAX_VALUE)
        );
        jPanel9Layout.setVerticalGroup(

jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
            .addGap(0, 348, Short.MAX_VALUE)
        );

        lienzo.add(jPanel9);
        jPanel9.setBounds(530, 350, 70, 350);

        jPanel5.setBackground(new java.awt.Color(51, 51, 51));
        jPanel5.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 255), 3));
        jPanel5.setLayout(new
java.awt.FlowLayout(java.awt.FlowLayout.LEFT));

        jLabel5.setForeground(new java.awt.Color(255, 255, 255));
        jLabel5.setText("Cuadro con FlowLayout left, borde de línea color
azul y 3 pixeles de grosor.");
```

```
        jPanel5.add(jLabel5);

        lienzo.add(jPanel5);
        jPanel5.setBounds(20, 360, 570, 50);

        jPanel6.setBackground(new java.awt.Color(51, 51, 51));
        jPanel6.setBorder(javax.swing.BorderFactory.createEmptyBorder(20,
10, 20, 10));
        jPanel6.setLayout(new
java.awt.FlowLayout(java.awt.FlowLayout.LEFT));

        jLabel6.setForeground(new java.awt.Color(255, 255, 255));
        jLabel6.setText("Cuadro con FlowLayout left y relleno de 20 pixeles
arriba y abajo y 10 pixeles izquierda y derecha.");
        jPanel6.add(jLabel6);

        lienzo.add(jPanel6);
        jPanel6.setBounds(20, 430, 570, 100);

        jPanel7.setBackground(new java.awt.Color(51, 51, 51));

jPanel7.setBorder(javax.swing.BorderFactory.createCompoundBorder(javax.swin
g.BorderFactory.createLineBorder(new java.awt.Color(255, 255, 0), 5),
javax.swing.BorderFactory.createEmptyBorder(20, 20, 20, 20)));
        jPanel7.setLayout(new
java.awt.FlowLayout(java.awt.FlowLayout.LEFT));

        jLabel7.setForeground(new java.awt.Color(255, 255, 255));
        jLabel7.setText("Cuadro con FlowLayout left, borde de 5 pixeles y
relleno de 20 pixeles.");
        jPanel7.add(jLabel7);

        lienzo.add(jPanel7);
        jPanel7.setBounds(20, 510, 570, 100);

        jPanel8.setBackground(new java.awt.Color(51, 51, 51));
        jPanel8.setBorder(new javax.swing.border.LineBorder(new
java.awt.Color(255, 255, 0), 5, true));
        jPanel8.setLayout(new
java.awt.FlowLayout(java.awt.FlowLayout.LEFT));

        jLabel8.setForeground(new java.awt.Color(255, 255, 255));
        jLabel8.setText("Cuadro con FlowLayout left y borde redondeado de 5
pixeles de grosor.");
        jPanel8.add(jLabel8);

        lienzo.add(jPanel8);
        jPanel8.setBounds(20, 630, 570, 60);

        jLabel9.setText("Cuadro con transparencia");
        lienzo.add(jLabel9);
        jLabel9.setBounds(480, 330, 150, 14);

        getContentPane().add(lienzo, java.awt.BorderLayout.CENTER);

        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void sumaActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_sumaActionPerformed
        valor1 = getOperando1();
```

```
        valor2 = getOperando2();
        double total = valor1 + valor2;
        setResultado(total);
    }//GEN-LAST:event_sumaActionPerformed

    private void restaActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_restaActionPerformed
        valor1 = getOperando1();
        valor2 = getOperando2();
        double diferencia = valor1 - valor2;
        setResultado(diferencia);
    }//GEN-LAST:event_restaActionPerformed

    private void multiplicacionActionPerformed(java.awt.event.ActionEvent
evt) {//GEN-FIRST:event_multiplicacionActionPerformed
        valor1 = getOperando1();
        valor2 = getOperando2();
        double producto = valor1 * valor2;
        setResultado(producto);
    }//GEN-LAST:event_multiplicacionActionPerformed

    private void divisionActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_divisionActionPerformed
        valor1 = getOperando1();
        valor2 = getOperando2();
        double division = valor1 / valor2;
        setResultado(division);
    }//GEN-LAST:event_divisionActionPerformed

    private double getOperando1() {
        return Double.parseDouble(operando1.getText());
    }

    private double getOperando2() {
        return Double.parseDouble(operando2.getText());
    }

    private void setResultado(double res) {
        resultado.setText(Double.toString(res));
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton division;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JPanel jPanel4;
    private javax.swing.JPanel jPanel5;
    private javax.swing.JPanel jPanel6;
    private javax.swing.JPanel jPanel7;
    private javax.swing.JPanel jPanel8;
    private javax.swing.JPanel jPanel9;
    private javax.swing.JPanel lienzo;
```

```
    private javax.swing.JButton multiplicacion;
    private javax.swing.JTextField operando1;
    private javax.swing.JTextField operando2;
    private javax.swing.JButton resta;
    private javax.swing.JTextField resultado;
    private javax.swing.JButton suma;
    // End of variables declaration//GEN-END:variables

}
```

Salidas:

HTML:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Documento sin tÃtulo</title>
<script type="text/javascript" src="javascript.js"></script>
<link href="css.css" rel="stylesheet" type="text/css" />
</head>

<body>
<div id="apDiv1">
  <div id="apDiv2"><label>Calculadora Simple</label></div>
  <div id="apDiv3"><label>Ejemplo de posicionamiento y
lÃ³gica</label></div>
  <div id="apDiv4">
    <div id="apDiv6">
      <input name="operando2" type="text" class="campoTexto" id="operando2"
/>
    </div>
    <div id="apDiv5">
      <input name="operando1" type="text" class="campoTexto" id="operando1"
/>
    </div>
    <div id="apDiv7">
      <div id="apDiv8">
        <input name="suma" type="submit" class="botonOperador" id="suma"
value="+" onclick="sumaActionPerformed()"/>
      </div>
      <div id="apDiv9">
        <input name="resta" type="submit" class="botonOperador" id="resta"
value="-" onclick="restaActionPerformed()" />
      </div>
      <div id="apDiv10">
      <input name="multiplicacion" type="submit" class="botonOperador"
id="multiplicacion" value="x" onclick="multiplicacionActionPerformed()"/>
      </div>
<div id="apDiv11">
  <input name="division" type="submit" class="botonOperador" id="division"
value="/" onclick="divisionActionPerformed()"/>
```

```
</div>
    </div>
    <div id="apDiv12">
      <form id="form1" name="form1" method="post" action="">
        <input name="resultado" type="text" class="campoTexto"
id="resultado" />
      </form>
    </div>
  </div>
  <div id="apDiv13"><label>Cuadros</label></div>
  <div id="apDiv14"><label>Ejemplo de diseÃ±os predefinidos, bordes y
relleno de bloques</label></div>
  <div id="apDiv15"><label>Cuadro con transparencia</label></div>
  <div id="apDiv16"><label>Cuadro con BorderLayout, borde de lÃnea color
azul y 3 pixeles de grosor.</label></div>
  <div id="apDiv17"><label>Cuadro con FlowLayout left y relleno de 20
pixeles arriba y abajo y 10 pixeles izquierda y derecha.</label></div>
  <div id="apDiv18"><label>Cuadro con FlowLayout left, borde de 5 pixeles y
relleno de 20 pixeles.</label></div>
  <div id="apDiv19"><label>Cuadro con FlowLayout left y borde redondeado de
5 pixeles de grosor.</label></div>
  <div id="apDiv20"></div>
</div>
</body>
</html>
```

JavaScript:

```
var valor1;
var valor2;

function sumaActionPerformed() {
      valor1 = getOperando1();
      valor2 = getOperando2();
      var total = valor1 + valor2;
      setResultado(total);
}

function restaActionPerformed() {
      valor1 = getOperando1();
      valor2 = getOperando2();
      var diferencia = valor1 - valor2;
      setResultado(diferencia);
}

function multiplicacionActionPerformed() {
      valor1 = getOperando1();
      valor2 = getOperando2();
      var producto = valor1 * valor2;
      setResultado(producto);
}

function divisionActionPerformed() {
      valor1 = getOperando1();
      valor2 = getOperando2();
      var division = valor1 / valor2;
      setResultado(division);
}
```

```
function getOperando1() {
	return parseFloat(document.getElementById('operando1').value);
}

function getOperando2() {
	return parseFloat(document.getElementById('operando2').value);
}

function setResultado(res) {
	document.getElementById('resultado').value = res;
}
```

CSS:

```
#apDiv1 {
	position:absolute;
	width:700px;
	height:800px;
	z-index:1;
	background-color: #CCF;
}
#apDiv2 {
	position:absolute;
	width:170px;
	height:30px;
	z-index:1;
	left: 20px;
	top: 20px;
}
#apDiv1 #apDiv2 label {
	font-family: Arial, Helvetica, sans-serif;
	font-size: 18px;
}
#apDiv3 {
	position:absolute;
	width:280px;
	height:20px;
	z-index:2;
	left: 20px;
	top: 50px;
}
#apDiv1 #apDiv3 label {
	font-family: Arial, Helvetica, sans-serif;
	font-size: 11px;
}
#apDiv4 {
	position:absolute;
	width:240px;
	height:270px;
	z-index:3;
```

```
        left: 20px;
        top: 90px;
        background-color: #72ACB0;
}
#apDiv5 {
        position:absolute;
        width:202px;
        height:20px;
        z-index:1;
        left: 20px;
        top: 20px;
}
#apDiv6 {
        position:absolute;
        width:200px;
        height:20px;
        z-index:2;
        left: 20px;
        top: 60px;
}
#apDiv7 {
        position:absolute;
        width:150px;
        height:90px;
        z-index:3;
        left: 40px;
        top: 110px;
        background-color: #C36;
}
#apDiv8 {
        position:absolute;
        width:60px;
        height:30px;
        z-index:1;
        left: 10px;
        top: 10px;
}
#apDiv9 {
        position:absolute;
        width:60px;
        height:30px;
        z-index:2;
        left: 80px;
        top: 10px;
}
#apDiv10 {
        position:absolute;
        width:60px;
        height:30px;
        z-index:3;
        left: 10px;
```

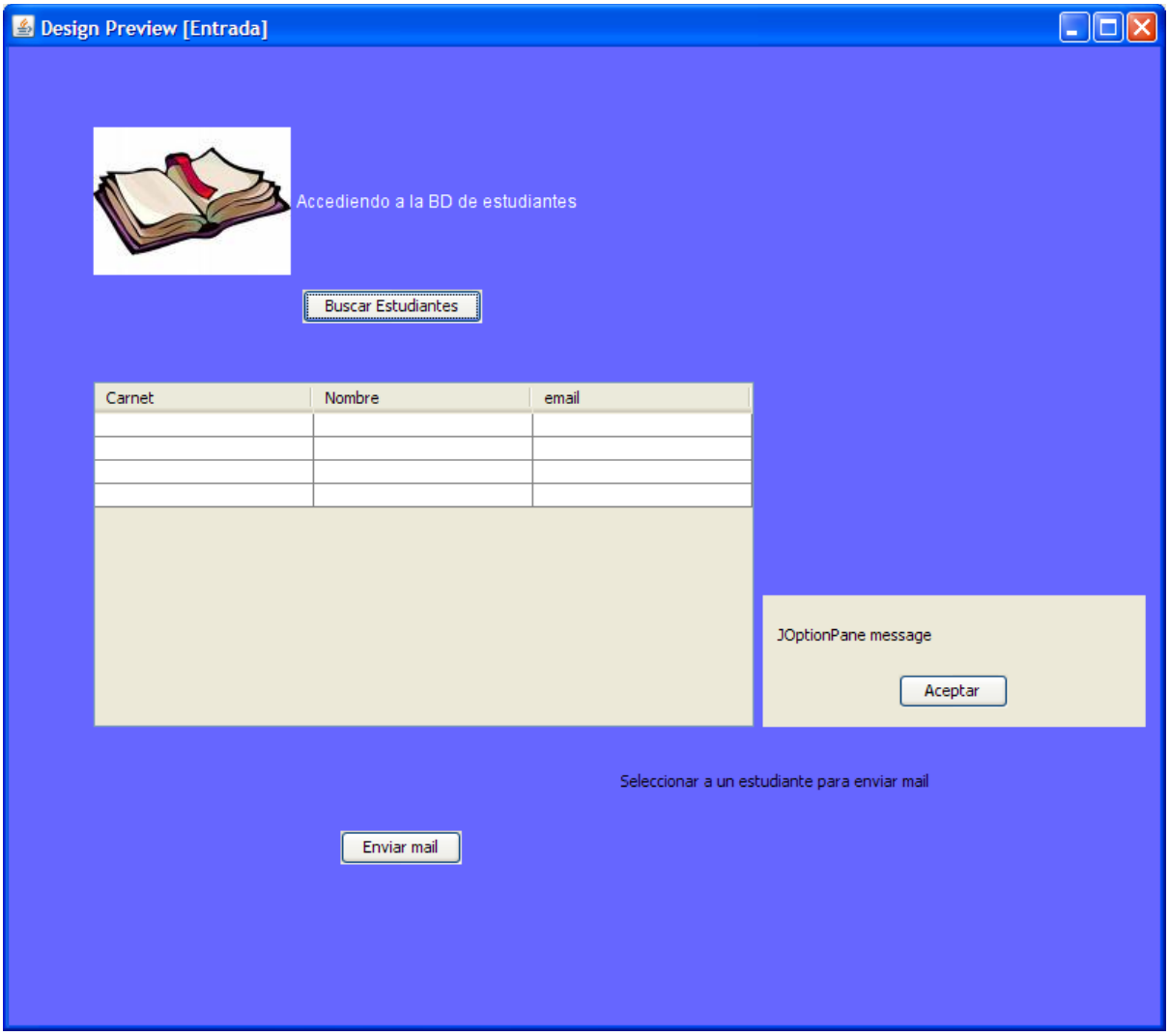
```
        top: 50px;
}
#apDiv11 {
        position:absolute;
        width:60px;
        height:30px;
        z-index:4;
        left: 80px;
        top: 50px;
}
#apDiv12 {
        position:absolute;
        width:200px;
        height:20px;
        z-index:4;
        left: 20px;
        top: 220px;
}
.campoTexto {
        height: auto;
        width: 100%;
        text-align: right;
}
.botonOperador{
        height: 100%;
        width: 100%;
}
#apDiv13 {
        position:absolute;
        width:160px;
        height:28px;
        z-index:4;
        left: 20px;
        top: 382px;
}
#apDiv14 {
        position:absolute;
        width:430px;
        height:20px;
        z-index:5;
        left: 20px;
        top: 410px;
}
#apDiv1 #apDiv13 label {
        font-family: Arial, Helvetica, sans-serif;
        font-size: 18px;
}
#apDiv15 {
        position:absolute;
        width:200px;
        height:20px;
```

```
        z-index:6;
        left: 480px;
        top: 410px;
}
#apDiv1 #apDiv15 label {
        font-family: Arial, Helvetica, sans-serif;
        font-size: 11px;
}
#apDiv1 #apDiv14 label {
        font-family: Arial, Helvetica, sans-serif;
        font-size: 11px;
}
#apDiv16 {
        position:absolute;
        width:570px;
        height:50px;
        z-index:7;
        left: 20px;
        top: 450px;
        background-color: #000;
        border-color: #00F;
        border-width: 3px;
        border-style: solid;
}
#apDiv1 #apDiv16 label {
        font-family: Arial, Helvetica, sans-serif;
        font-size: 11px;
        color: #FFF;
}
#apDiv17 {
        position:absolute;
        width:560px;
        height:20px;
        z-index:8;
        left: 20px;
        top: 520px;
        background-color: #000;
        padding-left: 10px;
        padding-right: 10px;
        padding-top: 20px;
        padding-bottom: 20px;
}
#apDiv18 {
        position:absolute;
        width:528px;
        height:12px;
        z-index:9;
        left: 22px;
        top: 590px;
        background-color: #000;
        border-color: #FF0;
```

```
        border-width: 5px;
        border-style: solid;
        padding: 20px;
}
#apDiv19 {
        position:absolute;
        width:580px;
        height:50px;
        z-index:10;
        left: 20px;
        top: 670px;
        background-color: #000;
        border-color: #FF0;
        border-width: 5px;
        border-style: solid;
        -webkit-border-radius: 10px;
  -moz-border-radius: 10px;
}
#apDiv20 {
        position:absolute;
        width:98px;
        height:308px;
        z-index:11;
        left: 520px;
        top: 440px;
        border-color: #F00;
        border-style: solid;
        border-width: 1px;
        background-color: transparent;
}
#apDiv1 #apDiv17 label {
        font-family: Arial, Helvetica, sans-serif;
        font-size: 11px;
        color: #FFF;
}
#apDiv1 #apDiv18 label {
        font-family: Arial, Helvetica, sans-serif;
        font-size: 11px;
        color: #FFF;
}
#apDiv1 #apDiv19 label {
        font-family: Arial, Helvetica, sans-serif;
        font-size: 11px;
        color: #FFF;
}
```

Ejemplo # 2

Vista Gráfica:



Entrada:

Hay 3 archivos JAVA de entrada.

Archivo 1:

```
package BaseDatos;

import javax.activation.*;
import javax.mail.*;
import javax.mail.internet.*;
import java.util.Properties;

public class Mail
{

  public Mail(String destino, String mensaje){
    SendAuthentication.Send(destino, mensaje);
  }
```

```
}

class SendAuthentication
{

public static void Send(String destino, String mensaje)
{

String host ="127.0.0.1";//Suponiendo que el servidor SMTPsea la propia máquina
String from ="Ivina Acuña";
Properties prop = new Properties();
prop.put("mail.smtp.host", host);
/*Esta línea es la que indica al API que debe autenticarse*/
prop.put("mail.smtp.auth", "true");
/*Añadir esta linea si queremos ver una salida detallada del programa*/
//prop.put("mail.debug", "true");

try{

SMTPAuthentication auth = new SMTPAuthentication();
Session session = Session.getInstance(prop, auth);
Message msg = getMessage(session, from, destino, mensaje);
System.out.println ("Enviando ...");


Transport.send(msg);
System.out.println ("Mensaje enviado!");

}

catch (Exception e)
{

ExceptionManager.ManageException(e);

}

}

private static MimeMessage getMessage(Session session, String from, String to, String mensaje)
{

try{

MimeMessage msg = new MimeMessage(session);
msg.setText(mensaje);
msg.addRecipient(Message.RecipientType.TO, new InternetAddress(to));
msg.setFrom(new InternetAddress(from,"JavaMail en accion"));
return msg;
```

```
}

catch (java.io.UnsupportedEncodingException ex)
{

ExceptionManager.ManageException(ex);
return null;

}

catch (MessagingException ex)
{

ExceptionManager.ManageException(ex);
return null;

}

}

}

class SMTPAuthentication extends javax.mail.Authenticator
{

public PasswordAuthentication getPasswordAuthentication()
{

String username = "ivi";

String password = "123";

return new PasswordAuthentication(username, password);

}

}

class ExceptionManager
{

public static void ManageException (Exception e)
{

System.out.println ("Se ha producido una exception");

System.out.println (e.getMessage());

e.printStackTrace(System.out);

}
```

```
}
```

Archivo 2:

```
package BaseDatos;
import java.sql.Connection;
import java.sql.Driver;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;


/**
 *
 * @author Ivi
 */
public class Conexion {
   String lista[] = new String[100];

   public static Connection getConnection() {
     Connection conn = null;
     Driver d = null;
     Class clase = null;

     try {
        //clase = Class.forName("org.postgresql.Driver");
        clase = Class.forName("com.mysql.jdbc.Driver");
        //conn = DriverManager.getConnection("jdbc:postgresql://127.0.0.1/open","saduser","sad");
        //conn =
DriverManager.getConnection("jdbc:oracle:thin:@5.6.111.94:1521:xe","db","password");
        conn = DriverManager.getConnection("jdbc:mysql://localhost/baseDatos","root", "123");
     } catch (Exception ex) {
        ex.printStackTrace();
     }

     return conn;
  }

public String[] getDatos() {
     Connection conn = getConnection();

     PreparedStatement ps = null;
     ResultSet rs = null;
              try {
        ps = conn.prepareStatement("SELECT * FROM estudiante ORDER BY camet");
        rs = ps.executeQuery();
        int x=0;
        while (rs.next()) {
           lista[0]=rs.getObject(0).toString();
```

```
            lista[x+1]=rs.getObject(1).toString();
            lista[x+2]=rs.getObject(3).toString();
            x = x +3;
          }

          rs.close();
          conn.close();
      } catch (SQLException ex) {
          ex.printStackTrace();
      }
      return lista;
}

}
```

Archivo 3:

```
package BaseDatos;

/**
 *
 * @author Ivi
 */
public class Entrada extends javax.swing.JFrame {
    Conexion con;
    /** Creates new form Entrada */
    public Entrada() {
        initComponents();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        tabla = new javax.swing.JTable();
        jButton2 = new javax.swing.JButton();
        jLabel2 = new javax.swing.JLabel();
        opcion = new javax.swing.JOptionPane();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jPanel1.setBackground(new java.awt.Color(102, 102, 255));
```

```
    jLabel1.setBackground(new java.awt.Color(102, 204, 255));
    jLabel1.setFont(new java.awt.Font("Arial", 0, 12));
    jLabel1.setForeground(new java.awt.Color(255, 255, 255));
    jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Imagenes/libro.jpg"))); //
NOI18N
    jLabel1.setText("Accediendo a la BD de estudiantes");

    jButton1.setText("Buscar Estudiantes");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
      }
    });

    tabla.setModel(new javax.swing.table.DefaultTableModel(
      new Object [][] {
        {null, null, null},
        {null, null, null},
        {null, null, null},
        {null, null, null}
      },
      new String [] {
        "Carnet", "Nombre", "email"
      }
    ) {
      boolean[] canEdit = new boolean [] {
        false, false, false
      };

      public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
      }
    });
    jScrollPane1.setViewportView(tabla);

    jButton2.setText("Enviar mail");
    jButton2.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
      }
    });

    jLabel2.setText("Seleccionar a un estudiante para enviar mail");

    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(
      jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
            .addGroup(jPanel1Layout.createSequentialGroup()
              .addGap(201, 201, 201)
              .addComponent(jButton1))
            .addGroup(jPanel1Layout.createSequentialGroup()
              .addGap(58, 58, 58)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                  .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                  .addComponent(opcion, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 342,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGroup(jPanel1Layout.createSequentialGroup()
              .addGap(227, 227, 227)
              .addComponent(jButton2)))
          .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
          .addContainerGap(419, Short.MAX_VALUE)
          .addComponent(jLabel2)
          .addGap(158, 158, 158))
    );
    jPanel1Layout.setVerticalGroup(
      jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(51, 51, 51)
        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 109,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jButton1)
        .addGap(40, 40, 40)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 236,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(opcion, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(30, 30, 30)
        .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 14,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(27, 27, 27)
        .addComponent(jButton2)
        .addContainerGap(110, Short.MAX_VALUE))
    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
```

```
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );

    pack();
  }// </editor-fold>

  private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    con.getConnection();
    String datos[] = con.getDatos();
    int x, y = datos.length/3, z,a=0;
    for(x=0;x<y;x++){
      for(z=0; z<3;z++){
        tabla.setValueAt(datos[a], x, z);
        a = a +1;
      }
    }
  }

  private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String destino, mensaje;
    Mail correo;
    destino = tabla.getValueAt(tabla.getSelectedRow(), tabla.getSelectedColumn()).toString();
    mensaje = "Comuniquese conmigo inmediatamente Atte.";
    correo = new Mail(destino, mensaje);
    opcion.showConfirmDialog(this, "Correo enviado exitosamente", "Aviso", 0);
  }

  /**
   * @param args the command line arguments
   */
  public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
      public void run() {
        new Entrada().setVisible(true);
      }
    });
  }

  // Variables declaration - do not modify
  private javax.swing.JButton jButton1;
  private javax.swing.JButton jButton2;
  private javax.swing.JLabel jLabel1;
```

```
    private javax.swing.JLabel jLabel2;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JOptionPane opcion;
    private javax.swing.JTable tabla;
    // End of variables declaration

}
```

Traducción:

Para la traducción se iniciará desde el tercer archivo presentado, y conforme se vayan creando los objetos de las otras clases se irá realizando la traducción de cada clase.