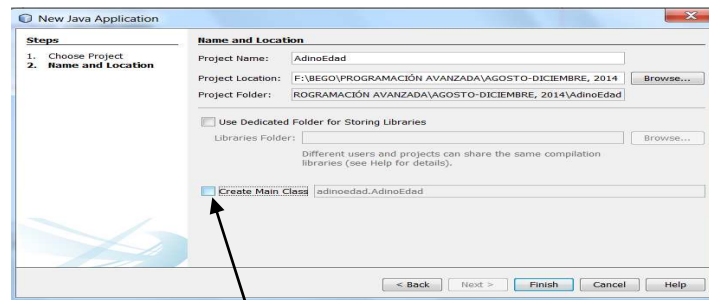


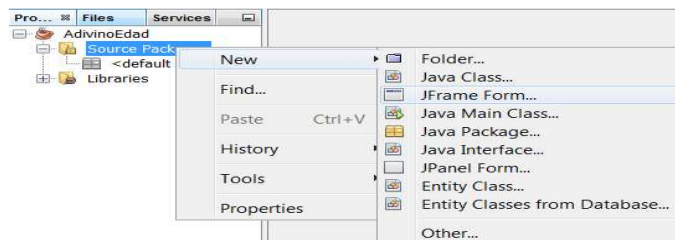
Interfaz gráfica - GUI

Para Empezar:

Se abre un nuevo proyecto normal pero se deselecciona "Create Main Class"



Se selecciona JFrame Form en lugar de Java Class.



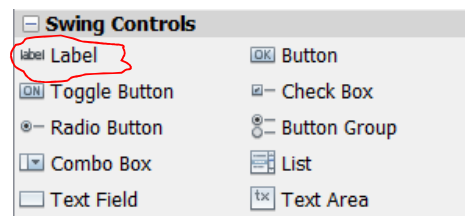
Terminación de la ejecución

Podemos utilizar el botón terminar y en su método escribimos: **System.exit(0);**

Label

Se utiliza para escribir título, rara vez se usa en la programación. Otro uso que se le puede dar es para mostrar resultados. El usuario no puede escribir en él con lo cual no puede modificar el contenido.

1. Seleccionamos Label y lo colocamos en la forma.
2. Botón derecho. Clic en Edit text y escribe el texto.

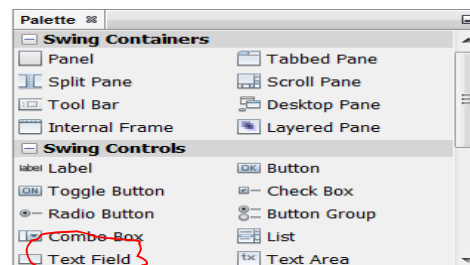


Si desea escribir un resultado en la etiqueta se utiliza el método **setText()**. Por ejemplo:
`lblResultado.setText("Resultado = " + r);`

Text Field

Es el cuadro usado para que el usuario escriba el dato que se le solicita.

1. Seleccionamos Text Field y lo colocamos en la forma.
2. Botón derecho. En Edit Text quitamos lo que muestra.
3. Seleccionamos Change Variable Name. Escribimos el nombre de la variable con prefijo txt. Por ejemplo, txtClave.
4. En el código obtenemos lo escrito en él usando el método **getText()**. Por ejemplo:
`clave = txtClave.getText();`



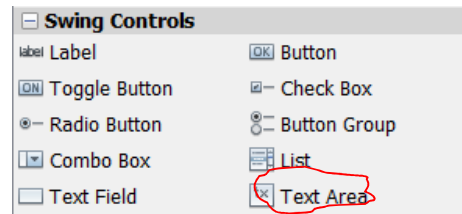
- Para establecer el foco (colocar el cursor) en el cuadro de texto usamos **requestFocus()**. Por ejemplo: `txtClave.requestFocus();`

Button

Es el botón que usa el usuario para que cuando hace clic en él se realiza alguna operación.

- Seleccionamos Button y lo colocamos en la forma.
- Botón derecho. Seleccionamos Change Variable Name. Escribimos el nombre de la variable con prefijo btn. Por ejemplo, `btnAceptar`.
- Botón derecho. En Edit Text y escribimos la acción que va a realizar. Por ejemplo, `Aceptar`.
- Para programar la acción que realizará hacemos doble clic en el botón y se muestra el encabezado del método:

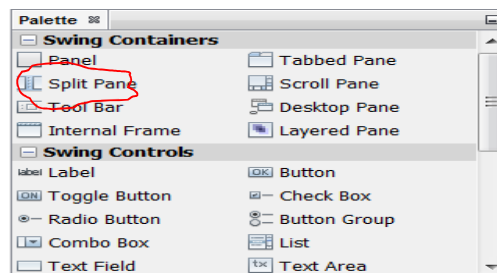
```
private void btnAceptarActionPerformed(java.awt.event.ActionEvent evt)
```



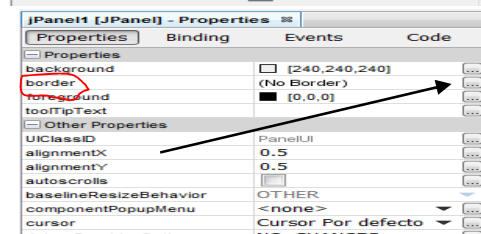
Panel

Se utiliza para agrupar controles

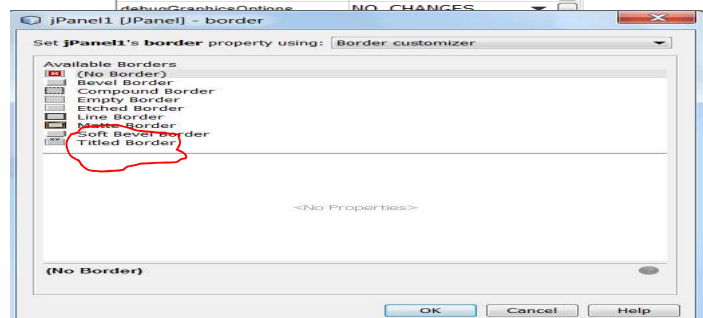
- Seleccionamos Panel y colócalo en la forma. Dale la dimensión que deseas.
- Para agregar un título utilizamos la ventana de propiedades.



- Hacemos clic en el botón con puntos de la propiedad "border". Se muestra la pantalla:



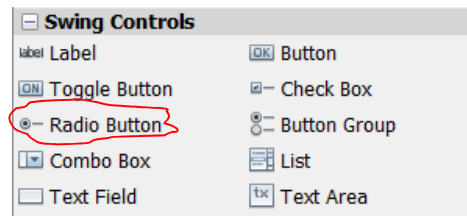
- Hacemos clic en "Titled Border" y en "Title" escribimos el título.



Radio Button

Son los botones que permiten seleccionar entre varias opciones una. También se les llama botones de opción.

1. Seleccionamos Radio Button y lo colocamos en la forma.
2. Botón derecho. Seleccionamos Change Variable Name. Escribimos el nombre de la variable con prefijo rbt. Por ejemplo, rbtFemenino.
3. Botón derecho. En Edit Text escribimos la opción a la que corresponde. Por ejemplo, Femenino.

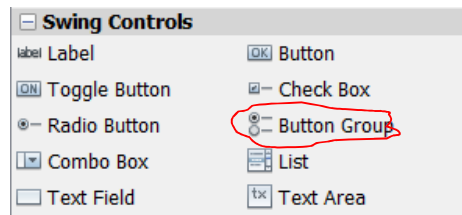


En la programación, para saber si fue seleccionado usamos el método booleano **isSelected()**: `rbtnMasculino.isSelected()`.

Si deseamos que un botón aparezca como seleccionado desde el principio, entonces en el constructor usamos el método **setSelected(true)**: `rbtnMasculino.setSelected(true)`.

Button Group

Para que sólo un botón pueda ser seleccionado a la vez se debe añadir un Button Group. Este objeto es invisible y no se ve en el formulario. Se puede ver en el Inspector, en Other Components.

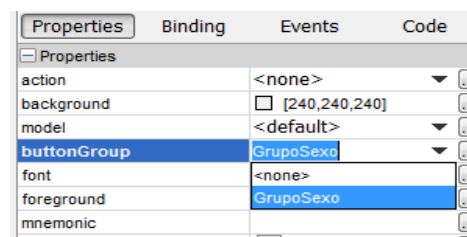


Le asignamos un nombre (GrupoSexo).



Ahora tenemos que hacer que todos los Radio Button pertenezcan a este grupo.

Seleccionamos cada uno de los Radio Button y cambiamos la propiedad `buttonGroup`, indicando que pertenece al grupo.

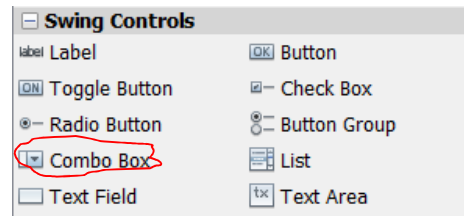


Para deseleccionar un Radio Button, en programación escribimos `grupoSexo.clearSelection()`.

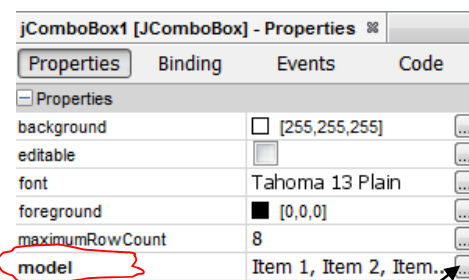
Combo Box

Se usa para mostrar una lista donde el usuario selecciona una opción de las que se muestran.

1. Seleccionamos Combo Box y lo colocamos en la forma.
2. Botón derecho. Seleccionamos Change Variable Name. Escribimos el nombre de la variable con prefijo cmb. Por ejemplo, cmbCarrera.



3. En **model** es donde se guardan los elementos que componen el ComboBox. De tal forma que, en la propiedad model se hace clic en los puntos de la derecha y se sustituyen Item 1, Item 2, etc. por los elementos que debe contener.

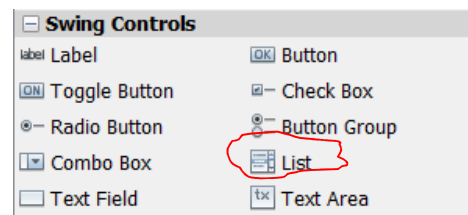


En programación para obtener el elemento seleccionado usamos el toString del método **getSelectedItem()**. Por ejemplo: cmbCarrera.getSelectedItem().toString();

List

Es similar al Combo Box aunque un uso muy común que se le da es el de desplegar en ella resultados de algunos cálculos o de información almacenada.

1. Seleccionamos List y lo colocamos en la forma.
2. Botón derecho. Seleccionamos Change Variable Name. Escribimos el nombre de la variable con prefijo lst. Por ejemplo, lstAlumnos



Podemos agregar elementos en la lista usando la propiedad model, igual que en el Combo Box.

En programación podemos sumarle datos a la lista, por ejemplo como resultado de la acción de hacer clic en un botón, de la siguiente forma:

```

javax.swing.DefaultListModel unaLista = new javax.swing.DefaultListModel();
String cad, texto;
int i;
cad = txtClave.getText() + " - " + txtNombre.getText(); //Lo que queremos agregar a la lista
for (i = 0; i < lstAlumnos.getModel().getSize(); i++)
{
    //Vaciamos todo lo que hay en el List (lstAlumnos) en el modelo unaLista
    texto= (String) lstAlumnos.getModel().getElementAt(i);
    unaLista.add(i, texto);
}
unaLista.add(i, cad);
lstAlumnos.setModel(unaLista); //Regresamos a la lista lo que tenía más lo nuevo (cad)

```

El proceso que se sigue es:

1. Bajamos todo lo que tenemos en la lista a la variable unaLista (for).
 La cantidad de elementos que hay en la lista la obtenemos con `getModel().getSize()`
 En la lista el primer elemento está en el lugar 0.
2. Agregamos a unaLista el nuevo elemento.
3. Usando el método `setModel`, guardamos unaLista en la lista.

Podemos usar `getSelectedIndex()` para conocer el índice del elemento seleccionado:
`System.out.println(lstAlumnos.getModel().getElementAt(lstAlumnos.getSelectedIndex()));`

Check Box

Normalmente se usan para que se puedan realizar varias selecciones a la vez.

Se manejan igual que los Radio Button.

MÉTODOS Y SUS ACCIONES¹

MÉTODO	OBJETO	ACCIÓN
exit(0)	System	Terminar la ejecución.
setText	Label	Mostrar el String en la forma.
getText	Text Field	Obtener el texto escrito en él.
requestFocus	Text Field	Colocar el foco cuadro de texto.
isSelected	Radio Button	Para saber si fue seleccionado.
setSelected(true)	Radio Button	Establecerlo seleccionado por default.
clearSeleccction()	Button Group	Deseleccionar un Radio Button.
getSelectedItem()	Combo Box List	Obtener el elemento seleccionado.
getModel().getSize()	List	Cantidad de elementos en la lista.
getModel().getElementAt(i)	List	Obtener el i-ésimo elemento de la lista (i comienza en 0).
add()	List	Agregar elementos al modelo.
setModel	List	Modificar la lista con el modelo que pasa como argumento.

¹ Se muestran sólo los métodos y se indica sólo los objetos en los que se utilizan en estas notas. Algunos métodos se pueden utilizar con varios objetos.