



UNIVERSIDAD DE SONSONATE

USO

SCIENTIAE ET BONIS ARTIBUS

INGENIERÍA EN SISTEMAS COMPUTACIONALES
LABORATORIO DE DESARROLLO DE
SOFTWARE I

Guía 1

Swing



INTRODUCCIÓN

Interfaz gráfica.

Una interfaz gráfica de usuario (GUI) presenta un mecanismo amigable al usuario para interactuar con un programa; ya que a este último le proporciona una "apariencia visual" única. Al proporcionar distintos programas en los que los componentes de la interfaz de usuario sean consistentes e intuitivos, los usuarios pueden familiarizarse con un programa incluso antes de utilizarlo.

Las GUI se crean a partir de componentes de la GUI también conocidos como controles o widgets (accesorios de ventana). Un componente de la GUI es un objeto con el cual interactúa el usuario mediante el ratón, el teclado u otra forma de entrada, como el reconocimiento de voz.

Swing.

Swing es una biblioteca gráfica para Java. Incluye widgets para interfaz gráfica de usuario tales como cajas de texto, botones, desplegables y tablas. Swing es un conjunto de paquetes contruidos en la parte más alta de AWT (Abstract Windows Toolkit) que proporciona un gran número de clases preconstruidas. Todas estas clases son algunos de los componentes de la GUI de Swing del paquete javax.swing. La mayoría de los componentes de Swing, como se les llama comúnmente, están escritos, se manipulan y muestran completamente en Java (los denominados componentes puros de Java).

Los componentes originales de la GUI del paquete java.awt del AWT (Abstract Windows Toolkit) están enlazados directamente con las herramientas de interfaz gráfica de usuario de la plataforma local. Cuando se ejecuta un programa de Java con una GUI de AWT en distintas plataformas Java, los componentes de la GUI del programa se muestran de manera distinta en cada plataforma.

Características:

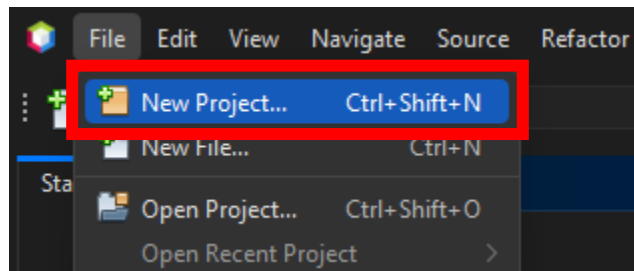
- ✓ Implementada sin usar código nativo (100% Java), con lo cual la GUI se verá de la misma forma en distintas plataformas.

- ✓ Basada en la arquitectura MVC.
- ✓ Genera interfaces gráficas, independientemente de la plataforma en que se ejecute la aplicación.
- ✓ Los contenedores contienen componentes, que son los controles básicos.
- ✓ No se usan posiciones fijas de los componentes, sino que están situados a través de una disposición controlada.
- ✓ El común denominador de más bajo nivel se acerca al teclado, ratón y manejo de eventos.
- ✓ Alto nivel de abstracción respecto al entorno de ventanas en que se ejecute la aplicación.
- ✓ La arquitectura de la aplicación es dependiente del entorno de ventanas, en vez de tener un tamaño fijo.

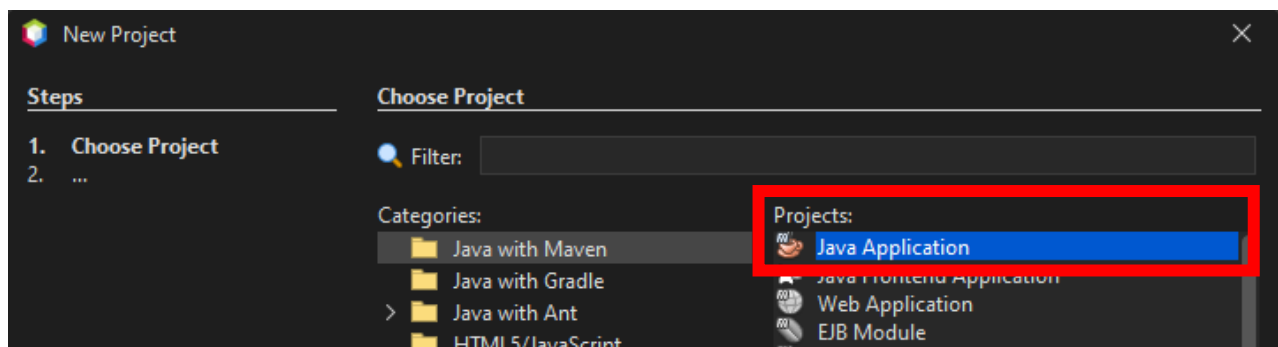
DESARROLLO

Creación de proyecto de Swing en NetBeans.

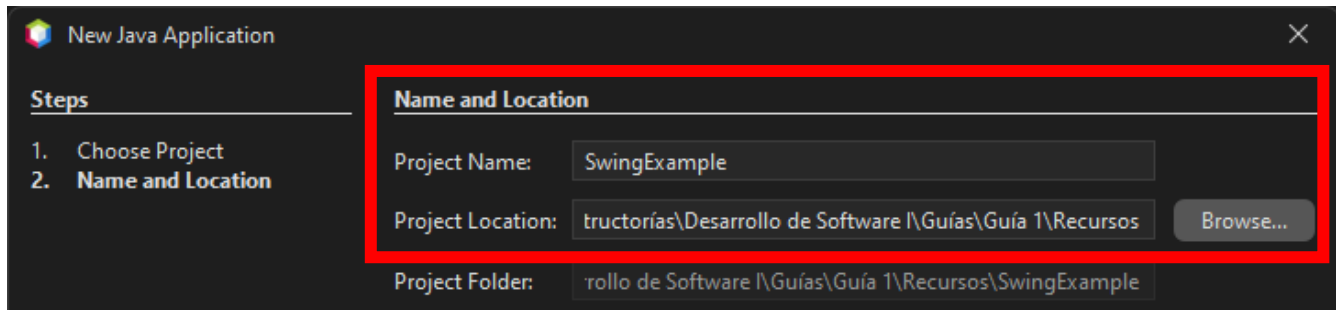
- En la barra de opciones superior, dar clic sobre **File**, luego en **New Project**.



- Se selecciona **Java Application** y luego en **Next**.

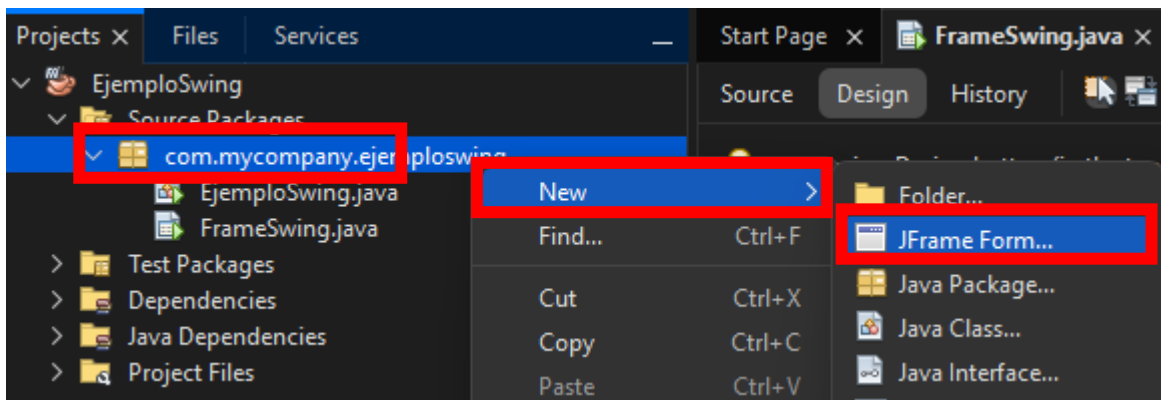


- Colocamos el nombre del proyecto, seleccionamos la ruta donde deseemos guardarlo y clic en **Finish**.

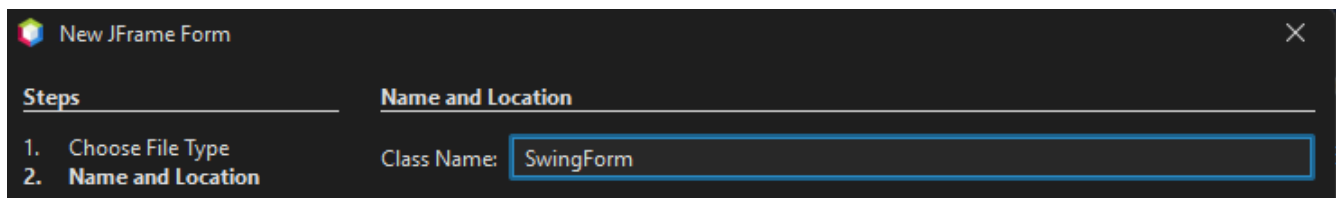


Trabajar con Swing.

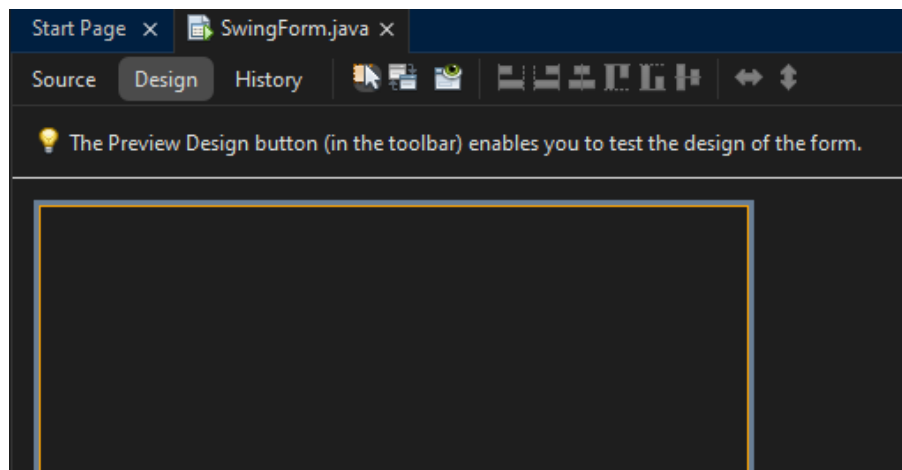
- Crear un formulario JFrame, dando clic derecho en el paquete por defecto del proyecto, clic en **New**, y seleccionamos **JFrame Form**. (Si JFrame Form no se encuentra en la lista, dar clic en **Other**, buscar la categoría **Swing GUI Forms** y seleccionar **JFrame Form**).



- Colocar el nombre y clic en **Finish**.

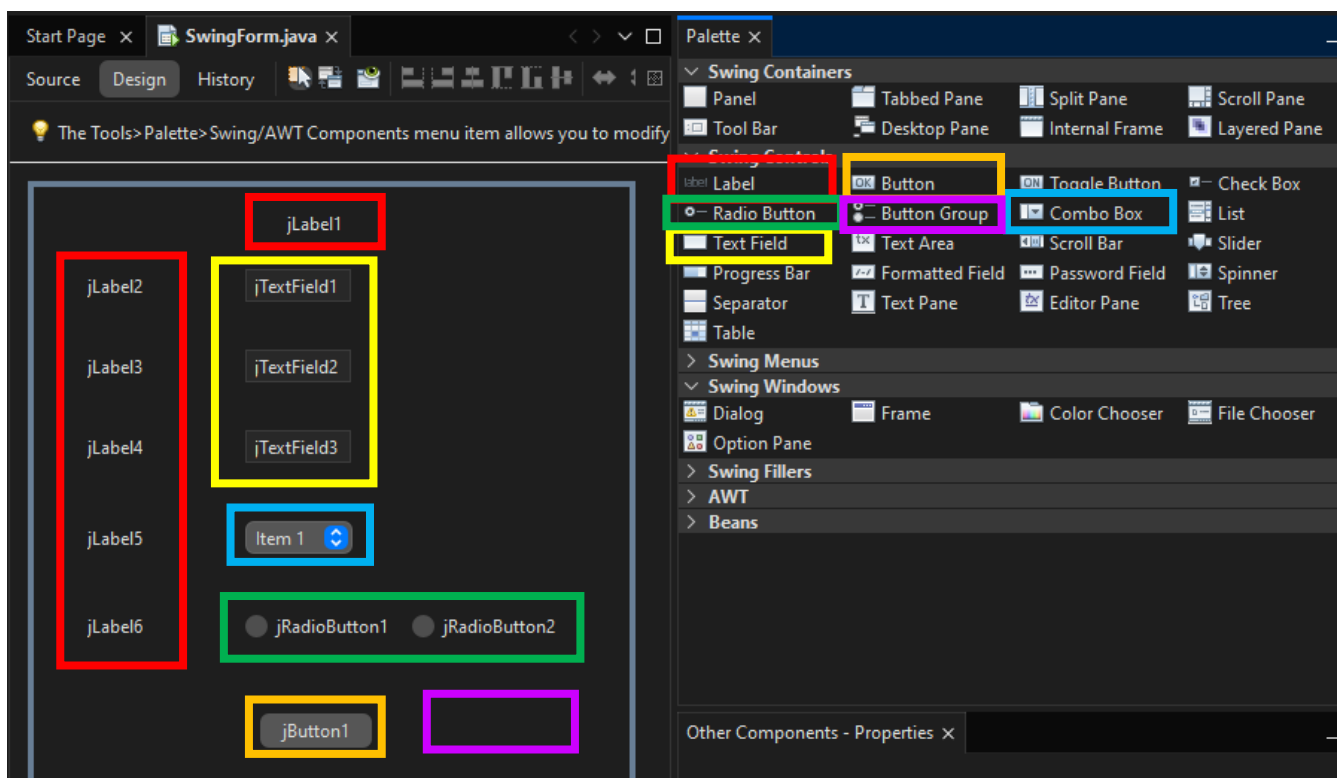


- Finalmente ya podremos trabajar en el lienzo del Frame.



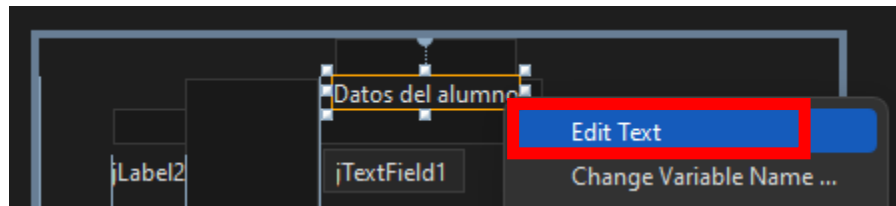
Agregar componentes y acciones.

- Para agregar componentes como botones, labels u otros, simplemente lo hacemos arrastrando y soltando desde la paleta de componentes. Agregaremos los siguientes componentes a nuestra aplicación:

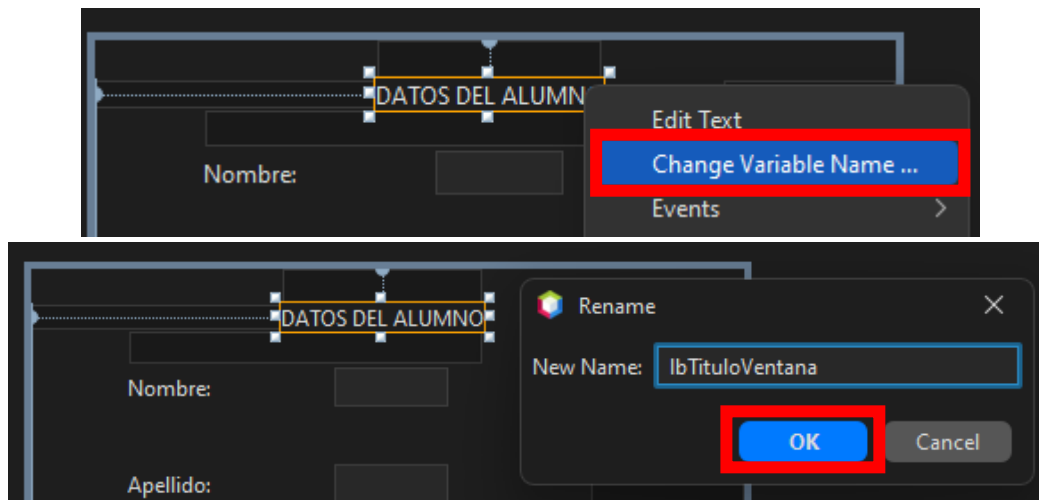


* El ButtonGroup solo se arrastrará directo al panel.

- En la propiedad text de cada elemento se cambia el texto que muestra cada etiqueta, clic derecho sobre el label y clic izquierdo en **Edit Text**.



- Lo siguiente es cambiar el nombre de la variable de cada elemento, esto es muy importante para saber cuales son las variables que se van a operar. Clic derecho sobre el elemento y luego en **Change Variable Name**, luego colocar el nombre y clic en **OK**.

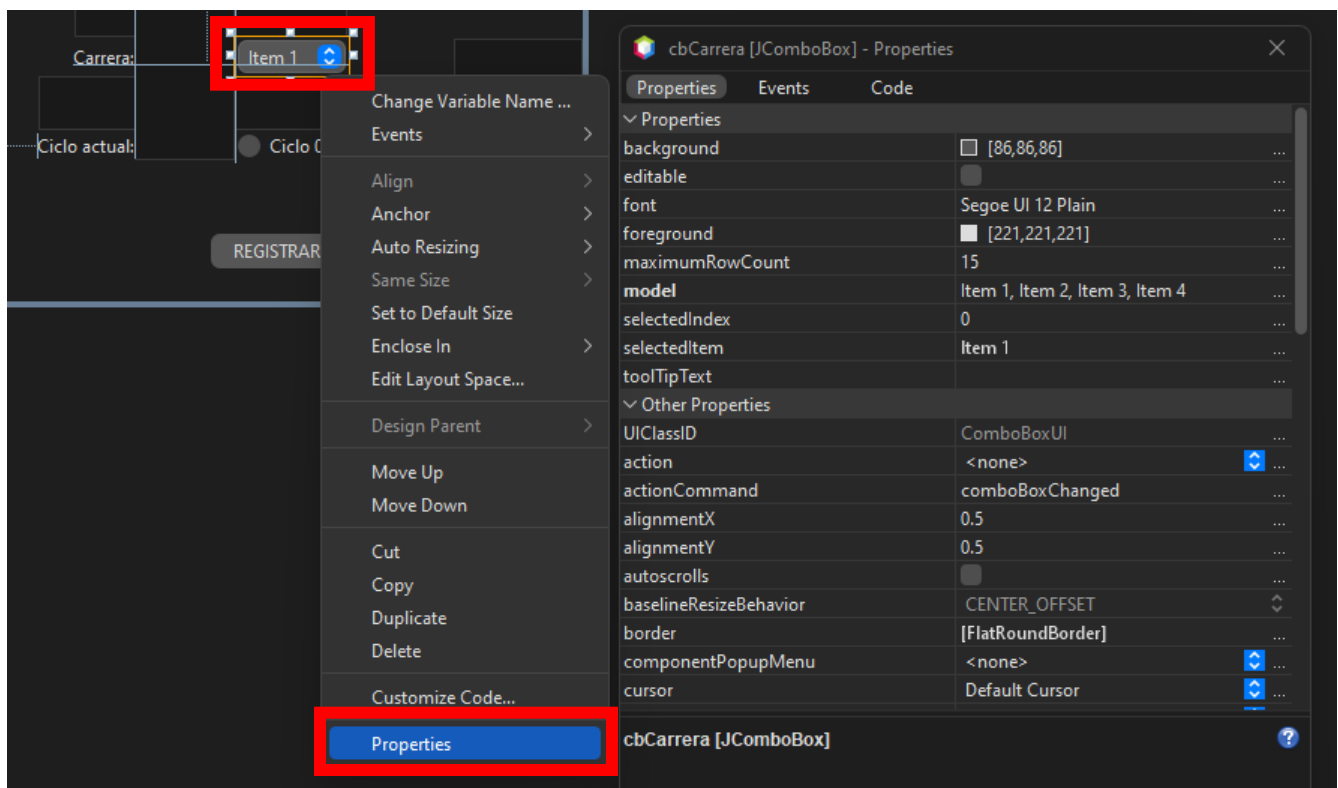


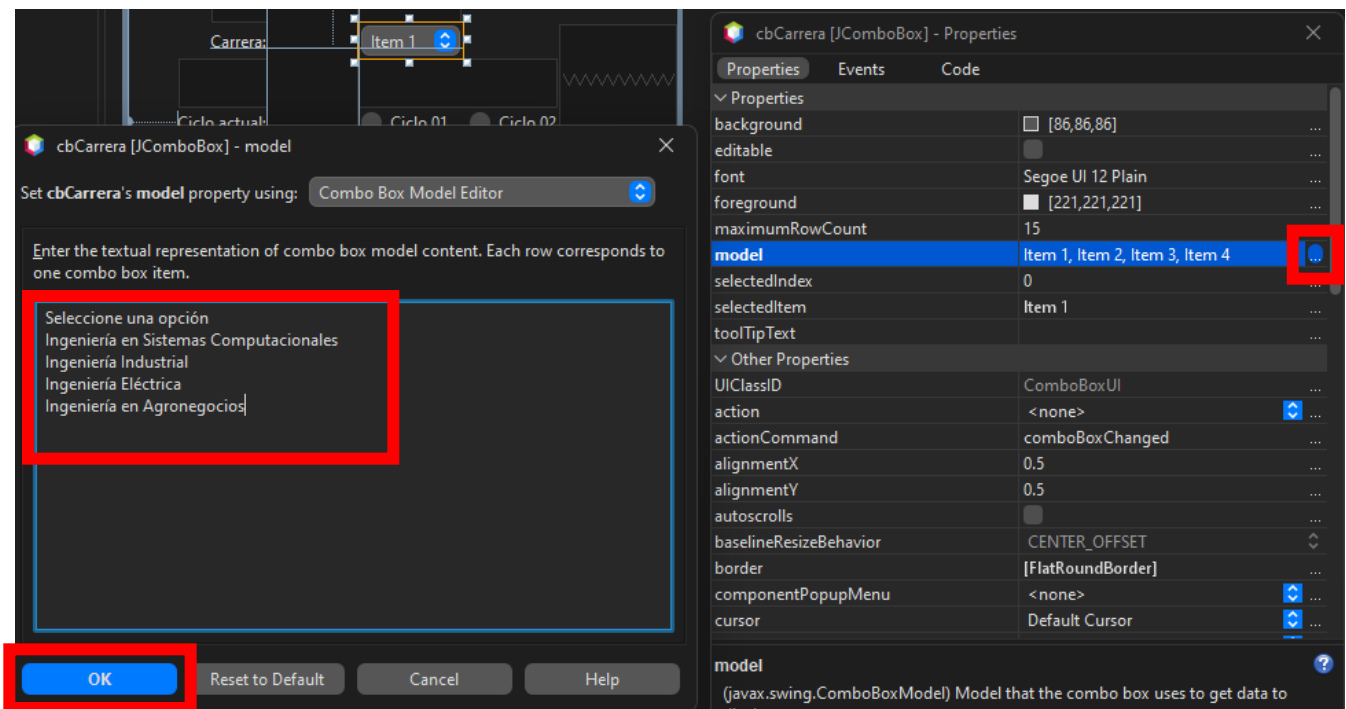
- Para los nombres de las variables usaremos una nomenclatura que sea ordenada y podamos recordar, por ejemplo utilizando forma CamelCase y escribiendo una abreviatura de dos letras del tipo de elemento.

Etiqueta	Label	lb	lbTituloVentana
Caja de texto	TextField	tf	tfNombre
Selector de opciones	ComboBox	cb	cbListaOpciones
Área de texto	TextArea	ta	taComentarios
Selector de opciones múltiples	CheckBox	ck	ckOpcionesMult
Selector de opciones tipo radio	RadioButton	rb	rbOpcionesRad
Botón	Button	bt	btEnviar

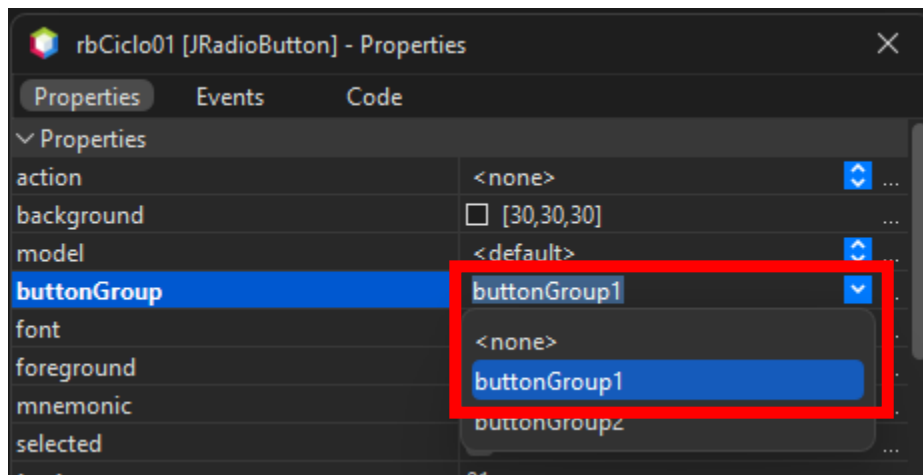
- Para añadir las opciones al ComboBox, damos clic derecho sobre él, luego en **Properties** y en el apartado de **model**, clic en los tres puntitos (...), posteriormente se abrirá un cuadro de dialogo donde escribiremos las opciones que el usuario podrá seleccionar, en este caso serán las carreras de Ingeniería. No olvidar agregar la opción por defecto.

- Seleccione una opción
- Ingeniería en Sistemas Computacionales
- Ingeniería Industrial
- Ingeniería Eléctrica
- Ingeniería en Agronegocios



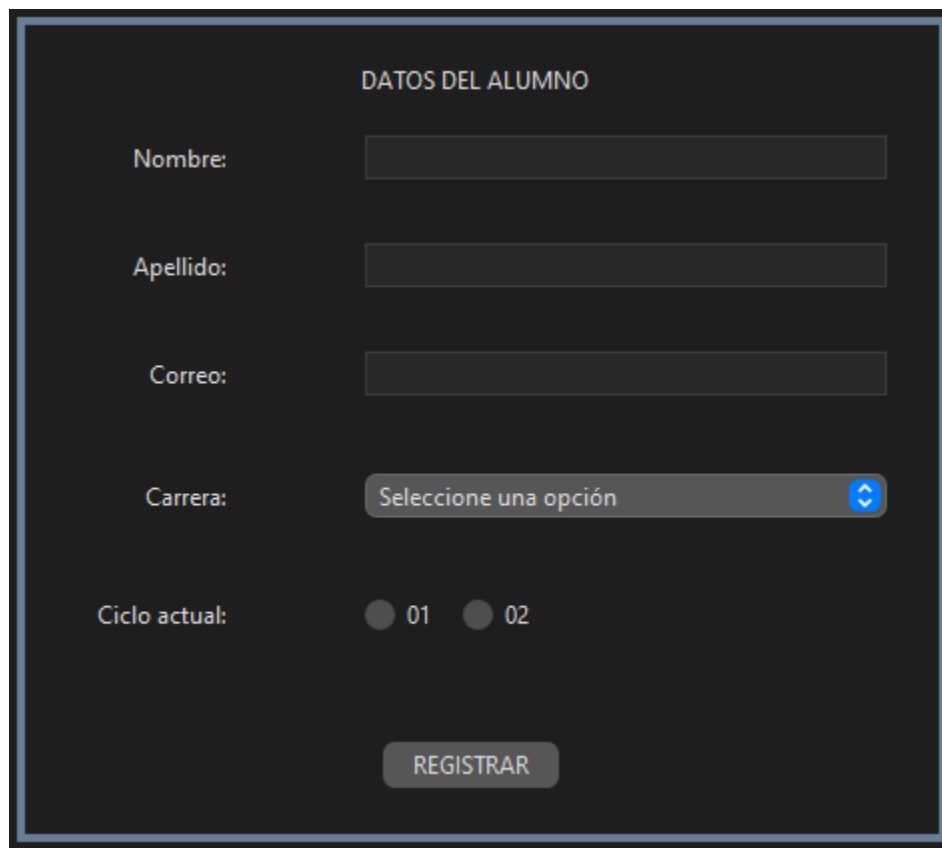


- Para enlazar los RadioButtons, debemos seleccionar en las propiedades, un grupo en común al que pertenecerán (en este caso) los dos RadioButtons. Hacer eso con ambos radios, seleccionar `buttonGroup1` en los dos, esto asegurará que solo se pueda seleccionar un RadioButton del grupo a la vez.



- Luego agregar un ActionListener a uno de los RadioButtons, seleccionando uno y dando clic en la pestaña **Eventos** de las propiedades, luego buscar el evento **ActionPerformed**

Al finalizar, el frame debe quedarnos similar a esto:



Programación.

El siguiente paso es programar la función que se va a desencadenar cuando se ejecute el evento de presionar el botón de **REGISTRAR**, para ello damos **doble click sobre el botón** y buscamos el método **ActionPerformed** y escribiremos lo siguiente:

```
String cicloActual = "";
String carrera = "";

JOptionPane.showMessageDialog(this, "Registro exitoso de "
    + this.tfNombre.getText() + " "
    + this.tfApellido.getText()
    + " estudiante de la carrera de "
    + this.cbCarrera.getSelectedItem()
    + " con el correo "
    + this.tfCorreo.getText()
    + " en el ciclo "
    + this.rbCiclo01.getText()
);
```

EJERCICIOS

Ejercicio 1: Realizar validaciones en el caso que el selector de opciones, los botones y las cajas de texto queden vacías.

Ejercicio 2: Haciendo uso de los elementos de GUI crear el formulario siguiente.

Nombre	Carrera	Ciclo	CUM
David Rajo			

Materia	UV	Nota	Estado
			Estado
			Estado
			Estado
			Estado
			Estado

Calcular

Al hacer clic en el botón calcular deberá mostrar en los estados dependiendo de la nota si está **aprobada** o **reprobada** la materia y un mensaje en la parte de abajo que muestre el nombre, el número de materias aprobadas y reprobadas, así como el CUM de esas materias. Trabajar únicamente con las materias de este ciclo de la carrera de Ing. en Sistemas Computacionales.

David Rajo su estado actual es de # materias aprobadas y # reprobadas y su Cum actual es de #.##

Este mensaje será un label, no una ventana, usar:
`miLabel.setText("Nuevo texto");`

¡Importante!

El cálculo del CUM es:

$$\frac{((\text{NotaMateria1} \times \text{UVMateria1}) + (\text{NotaMateria2} \times \text{UVMateria2}) + (\text{NotaMateria3} \times \text{UVMateria3}) + (\text{NotaMateria4} \times \text{UVMateria4}) + (\text{Materia5} \times \text{UVMateria5}))}{(\text{UVMateria1} + \text{UVMateria2} + \text{UVMateria3} + \text{UVMateria4} + \text{UVMateria5})}$$