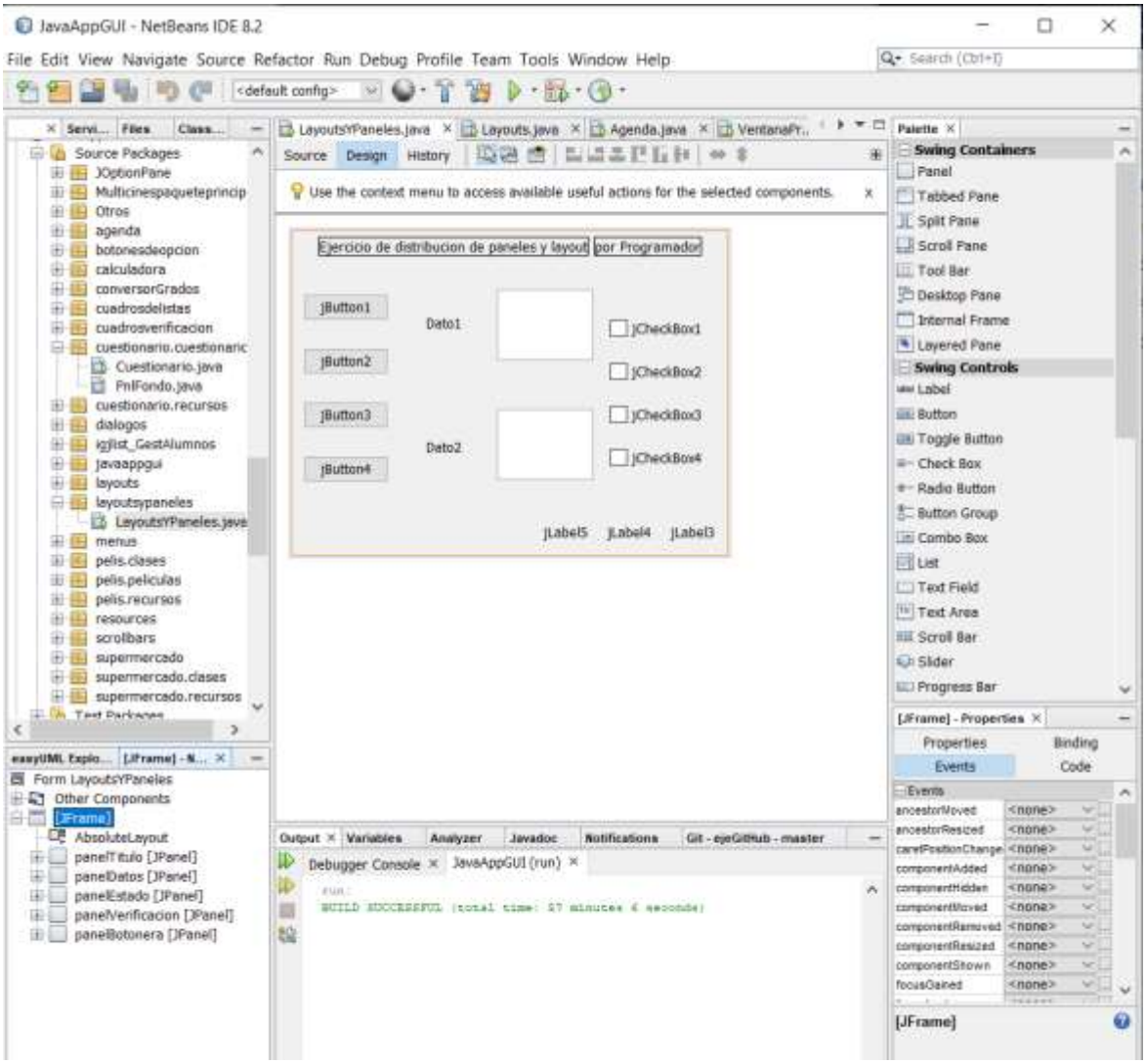
Resumen Programación 3ºEVA

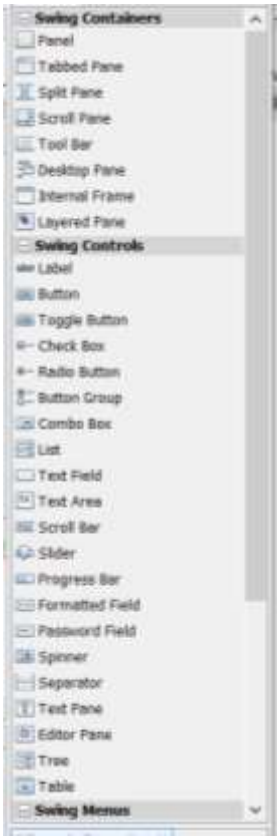
Tema: Interfaz Gráfica (GUI)

INTERFAZ GRÁFICA: *Conjunto de componentes gráficos que posibilitan la interacción entre el usuario y la aplicación*

***Entorno**:*



Resumen Programación 3ºEVA



CONTAINER: Objeto que hereda de la clase Container y se encarga de alojar componentes u otros containers.

*Ejs. (**JFrame**, **JPanel**).*

*COMPONENTES: Objetos (**JButton**, **JList**, **JLabel**...)*

*que heredan de la clase base **Button**, **List**,... (AWT)*

***FRAME**: Container principal que contendrá los componentes de la GUI*

***LAYOUT**: Componente que distribuye el espacio de un container especificando un criterio de distribución para los componentes.*

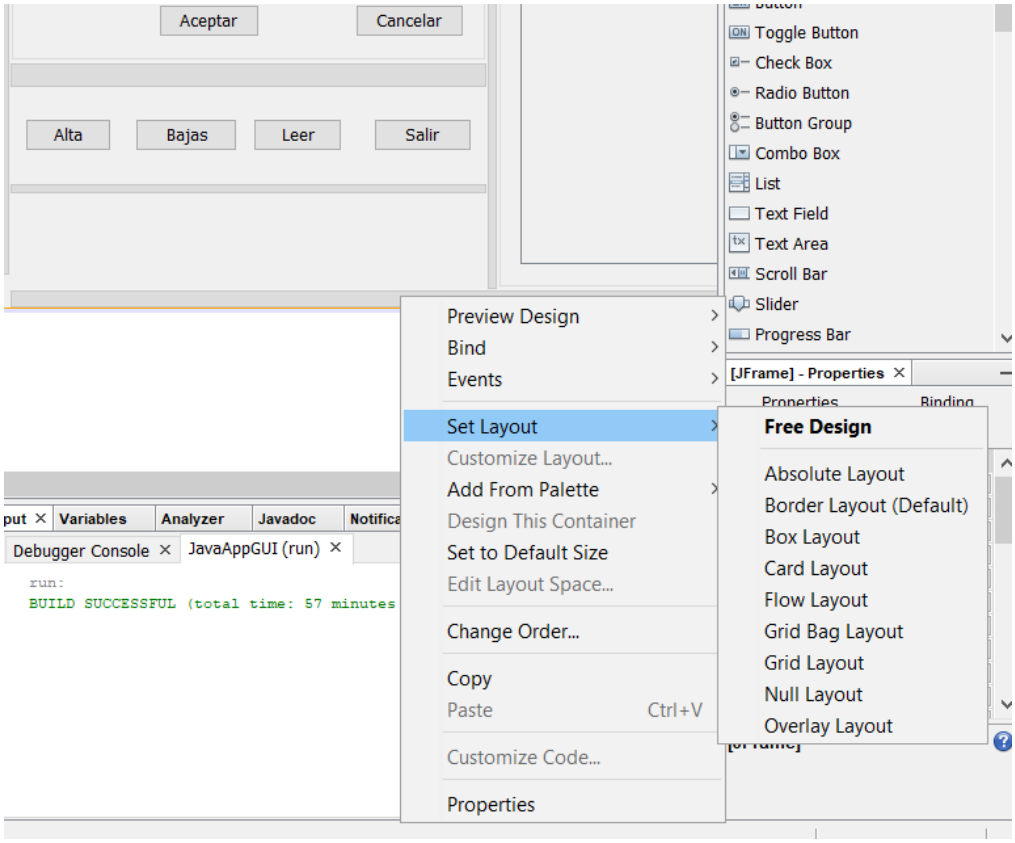
TIPOS:

A. *FREE DESIGN: Distribuye los componentes de forma que cada uno mantiene una situación relativa a otro y/o al propio container*
B. *ABSOLUTELAYOUT: Distribuye los componentes sin mantener una correlación entre ello y/o con el propio container, es decir sin ninguna restricción de posicionamiento*
C. *FLOWLAYOUT: Distribuye los componentes uno al lado del otro en la parte superior del container siguiendo una determinada alineación*
D. *BORDERLAYOUT: Divide el espacio del container en 5 regiones admitiendo un único componente por región*
   - *REGIONES:*

	NORTH	
WEST	*CENTER*	*EAST*
	SOUTH	

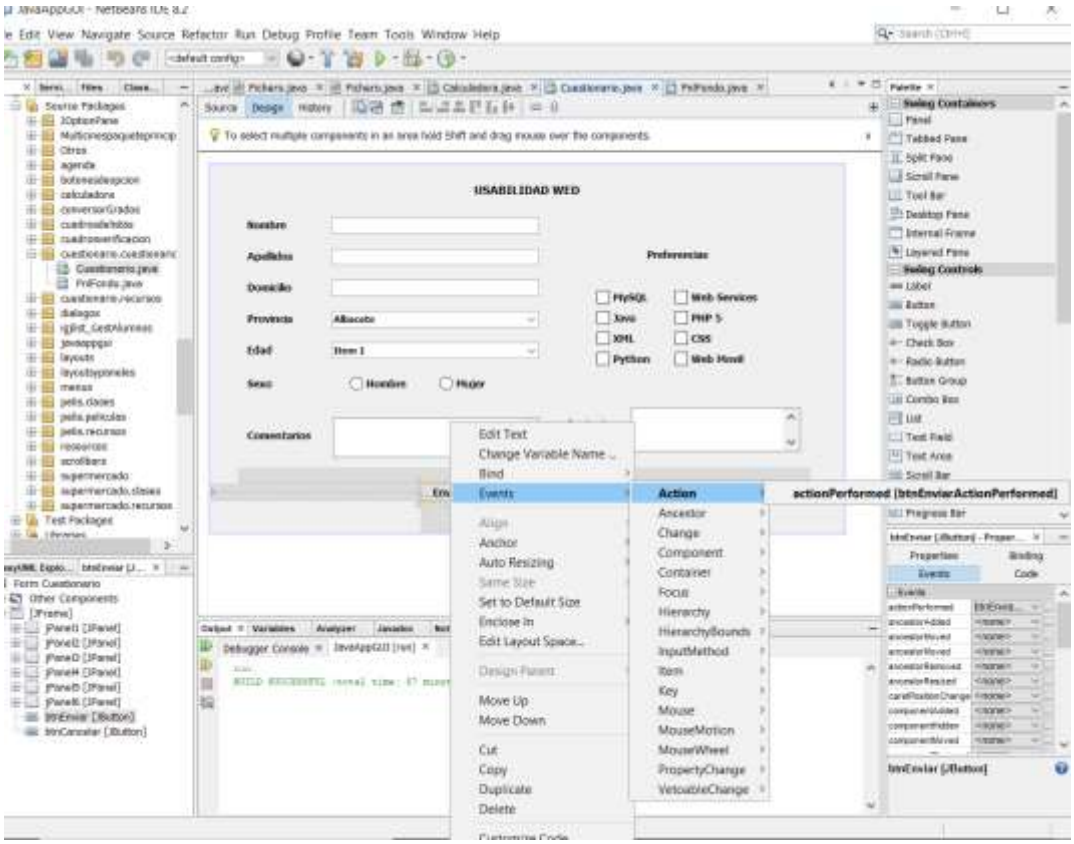
- *GridLayout: Distribuye el espacio en n filas y n columnas*
  //todas las celdas serán del mismo tamaño
- *GridBagLayout: Distribuye el espacio como el GridLayout, pero permitiendo a cada componente ocupar, más de una celda*

Botón Drcho. *Sobre el FRAME:*

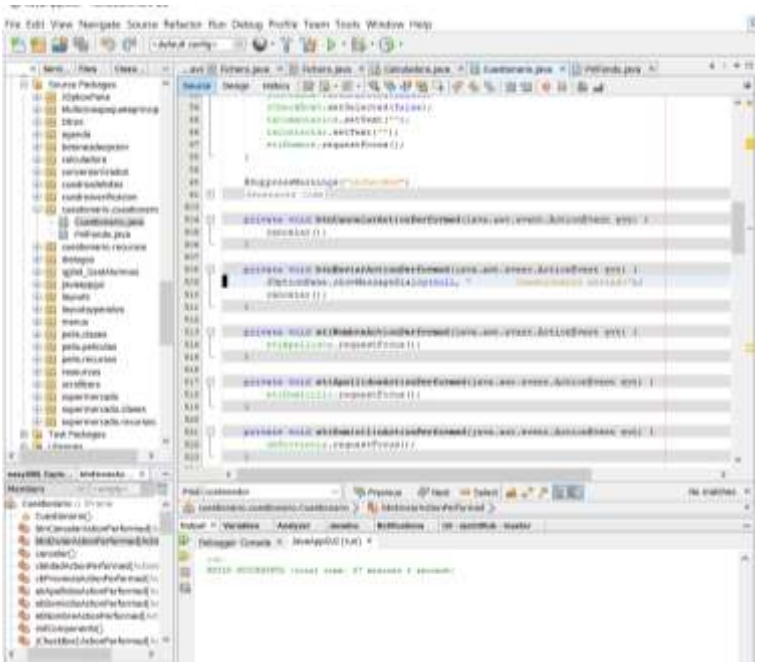


Resumen Programación 3ºEVA

EVENTOS:



Posteriormente se introducirá en el código la acción correspondiente a realizar.



Resumen Programación 3ºEVA

Creación de tablas (ej. Práctica - Supermercado):

Para trabajar con tablas debemos definir una variable de tipo DefaultTableModel y dos métodos para trabajar con la tabla, uno que diseñará el modelo de la misma y otro para insertar datos, además de uno que permitirá vaciar la misma.

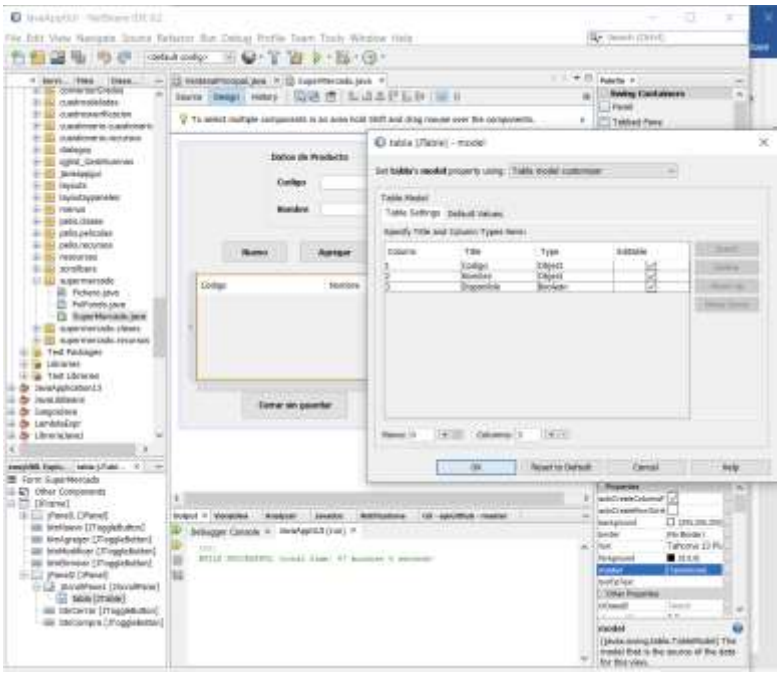
CREAR EL MODELO DE LA TABLA:

```
private DefaultTableModel crearModelo () {

    String [] columnas = new String [3];
    columnas [0] = "CIF";
    columnas [1] = "Razon Social";
    columnas [2] = "Direccion";

    DefaultTableModel modelo = new DefaultTableModel(null, columnas);

    return modelo;
}
```

//Modelo creado por el programador, podríamos usar Object[]

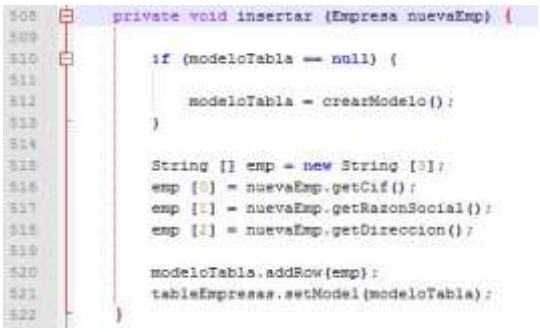
```
public Ej6Supermercado() {
    initComponents();
    fichero = new File ("productos.txt");
    textCodigo.setEnabled(false);
    textNombre.setEnabled(false);
    checkDisponible.setEnabled(false);
    buttonAgregar.setEnabled(false);
    buttonCancelar.setEnabled(false);
    modeloTabla = (DefaultTableModel)tableProductos.getModel();
}
```

//Modelo creado mediante GUI



Resumen Programación 3ºEVA

INSERTAR:

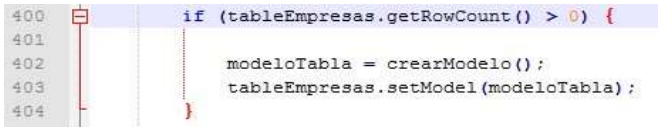


```
508    private void insertar (Empresa nuevaEmp) {
509
510        if (modeloTabla == null) {
511
512            modeloTabla = crearModelo();
513        }
514
515        String [] emp = new String [3];
516        emp [0] = nuevaEmp.getCif();
517        emp [1] = nuevaEmp.getRazonSocial();
518        emp [2] = nuevaEmp.getDireccion();
519
520        modeloTabla.addRow(emp);
521        tableEmpresas.setModel(modeloTabla);
522    }
```

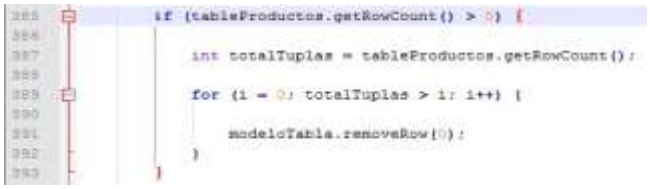
//Si el modelo fue creado mediante GUI no pondremos el if

VACIAR LA TABLA:

//Modelo creado por el programador



```
400        if (tableEmpresas.getRowCount() > 0) {
401
402            modeloTabla = crearModelo();
403            tableEmpresas.setModel(modeloTabla);
404        }
```



```
385        if (tableProductos.getRowCount() > 0) {
386
387            int totalTuplas = tableProductos.getRowCount();
388
389            for (i = 0; totalTuplas > i; i++) {
390
391                modeloTabla.removeRow(0);
392            }
393        }
```

//Modelo creado mediante GUI

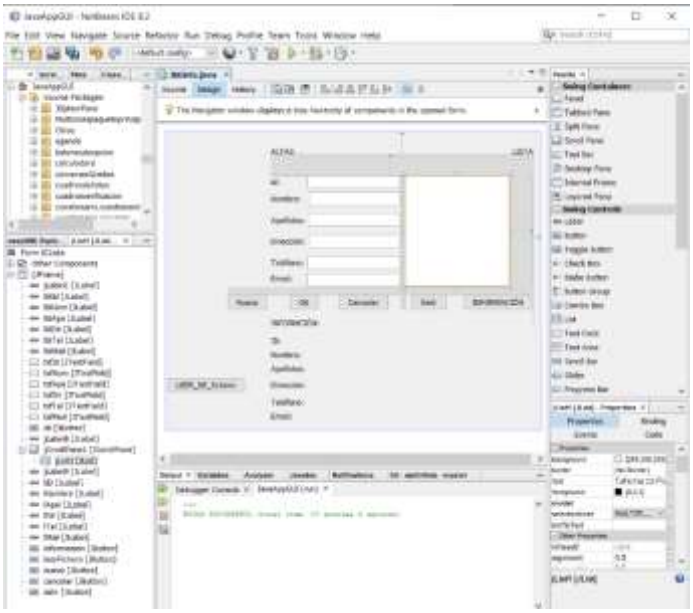
Creación de listas (Ej. Práctica - GestiónAlumnos):

Para trabajar con listas debemos definir una variable de tipo DefaultListModel y usaremos los siguientes 3 métodos:

CREAR MODELO:

En el constructor de la clase usaremos el generado mediante GUI pondremos la siguiente línea de código:

modeloLista = new DefaultListModel();



INSERTAR:

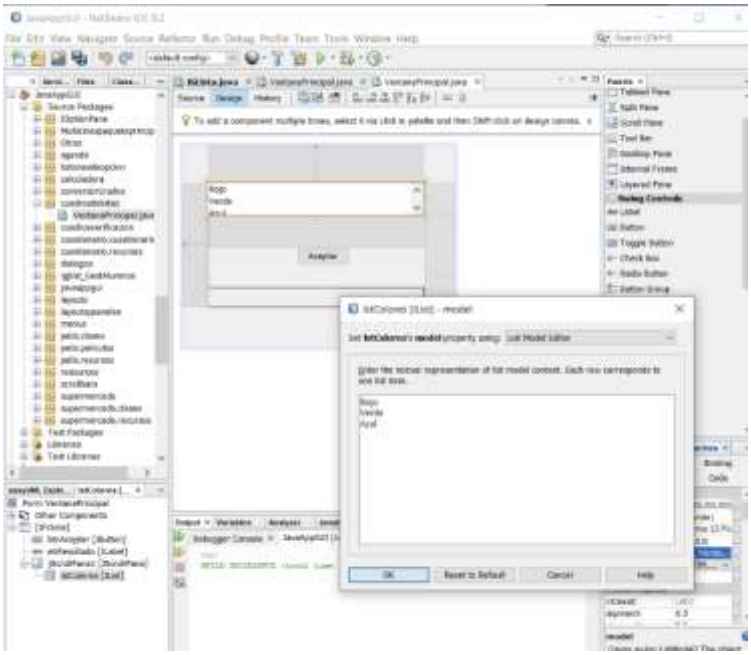
modeloLista.addElement(objeto);

jList.setModel(modeloLista);

VACIAR LA LISTA:

modeloLista.clear();

jList.setModel(modeloLista);



O en Entorno Gráfico otro tipo de lista(Ej. Cuadros de Listas)

Resumen Programación 3ºEVA

Creación de ComboBox (Ej: Práctica-Pelicula->modeloPelicula):

Para trabajar con ComboBox debemos definir una variable de tipo DefaultComboBoxModel y usaremos los siguientes métodos:

CREAR MODELO E INSERTAR:

```
private DefaultComboBoxModel crearModeloCmboxPersonas (File fichero) {

    DefaultComboBoxModel modelo = new DefaultComboBoxModel();
    ObjectInputStream ois = null;
    Persona pers;

    try {

        ois = new ObjectInputStream(new FileInputStream(fichero));

        modelo.addElement(" Selecciona una persona de la lista ");
        do {

            pers = (Persona) ois.readObject();
            modelo.addElement(pers.getNombre());
        } while (true);
    } catch (EOFException e) {
    } catch (IOException | ClassNotFoundException e) {
    } finally {

        if (ois != null) {

            try {
                ois.close();

            } catch (IOException e) {
            }
        }
    }

    return modelo;
}
```

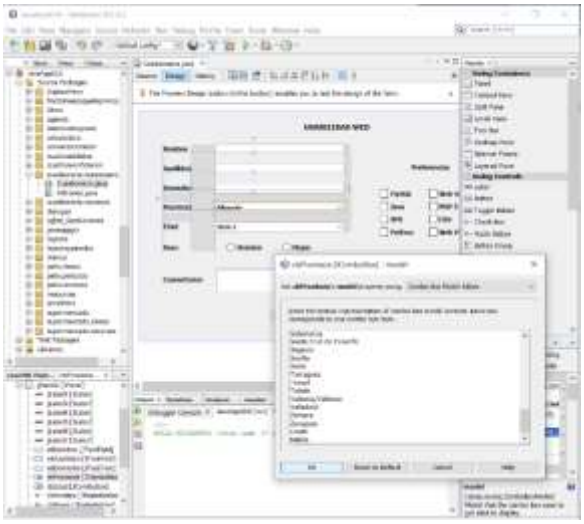
//Modelo creado por el programador con inserción de datos

VACIAR LA LISTA:

modeloComboBoxLista = crearModeloComboBoxLista(fichero); //Método anterior

comboBoxPeliculaMod.setModel(modeloComboBoxLista);

En entorno gráfico (Ej. Cuestionario)



Otros métodos de trabajo con GUI:

A. *setEnabled(boolean); //Activa/Desactiva el componente*
B. *getText(String); //Recupera el texto de un jLabel/jTextField*
C. *setText(String); //Cambia el texto de un jLabel/jTextField*
D. *isSelected(); //Devuelve un booleano para un jRadioButton/jCheckBox seleccionado o no*
E. *getSelectedItem(); //Devuelve un objeto*
F. *getSelectIndex(); //Devuelve el índice del ítem seleccionado*
G. *getRowCount(); //Devuelve el total de filas*
H. *removeRow(index); //Elimina la fila seleccionada*