

**TECNOLOGICO NACIONAL DE MEXICO  
CAMPUS "OAXACA"**

**INSTITUTO TECNOLÓGICO DE OAXACA**

**INGENIERIA EN SISTEMAS COMPUTACIONALES.**

**NOMBRE DE LA MATERIA: TOPICOS AVANZADOS DE PROGRAMACION.**

**UNIDAD 1: "INTERFAZ GRAFICA DEL USUARIO".**

**ACTIVIDAD: EJERCICIOS DE LENGUAJE JAVA 8, 9 Y 10.**

**GRUPO: ISU**

**HORA: 09:00-10:00 AM**

**DOCENTE: M.C. LUIS ALBERTO ALONSO HERNÁNDEZ**

**ALUMNO: GONZALEZ PASCUAL MELVIN PAUL**

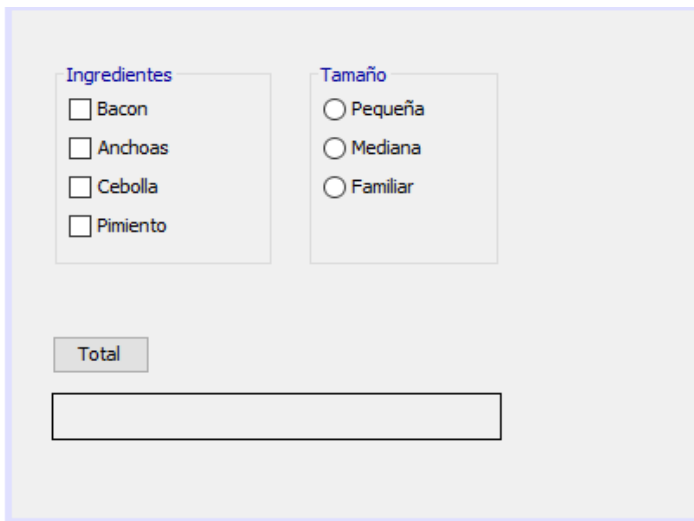
**FECHA DE ENTREGA: 14 DE NOVIEMBRE DEL 2020**

## Contenido

Ejemplo Lenguaje 8. ....	3
Ejercicio 1. ....	3
Ejemplo Lenguaje 9. ....	7
Ejercicio 1. ....	7
Ejemplo Lenguaje 10. ....	10
Ejercicio 1. ....	10
Ejercicio 2. ....	12

## Ejemplo Lenguaje 8.

### Ejercicio 1.

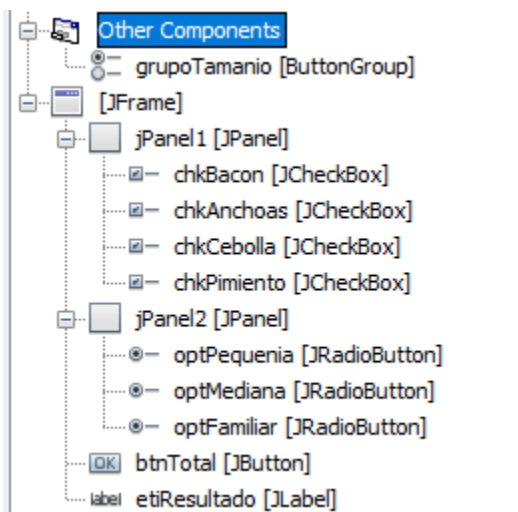


The screenshot shows a Java Swing window with a light gray background. It contains two panels: 'Ingredientes' and 'Tamaño'. The 'Ingredientes' panel has four checkboxes labeled 'Bacon', 'Anchoas', 'Cebolla', and 'Pimiento'. The 'Tamaño' panel has three radio buttons labeled 'Pequeña', 'Mediana', and 'Familiar'. Below these panels is a 'Total' button and a text field for the result.

Se implemento el siguiente diseño, en cual hace uso de check box y radio button, además de un botón salir y una etiqueta de texto para el cálculo.

Además, que se crearon en dos paneles extras para cada uno de los componentes.

Las variables se renombraron de la siguiente manera.



Estos fueron los componentes finales que se ocuparon para hacer los cálculos.

Se crearon los siguientes métodos:

El método para poner un logo.

```
//Metodo para el mensaje de bienvenida
public void asignar(){
    icono = new ImageIcon(getClass().getResource("imagenes/pz.jpg"));
    logopr = new ImageIcon(icono.getImage().getScaledInstance(100,100,Image.SCALE_DEFAULT));
}
```

Mientras que en el constructor agregaremos lo siguiente.

```

//Constructor
public Pizzeria() {
    asignar();
    //Creamos el mensaje de bienvenida
    JOptionPane.showMessageDialog(null, "Pizzeria Java\n Programadora: Melvin Paul Gonzalez Pascual", "Java Pizzeria", JOptionPane.PLAIN_MESSAGE, logope);
    initComponents();
    optFamiliar.setSelected(true); //Selección de tamaño familiar por defecto.
}

```

El método para calcular el costo de la pizza:

```

//Se hace los calculos del precio a pagar de la pizza.
public void calculos() {
    int tamaño = 0;
    double sabores = 0;
    if (optPequena.isSelected()) {
        tamaño = tamaño + 7;
    }
    if (optMediana.isSelected()) {
        tamaño = tamaño + 9;
    }
    if (optFamiliar.isSelected()) {
        tamaño = tamaño + 11;
    }
    if (chkBacon.isSelected()) {
        sabores = sabores + 1.50;
    }
    if (chkCebolla.isSelected()) {
        sabores = sabores + 1.00;
    }
    if (chkAnchoas.isSelected()) {
        sabores = sabores + 1.80;
    }
    if (chkPimientos.isSelected()) {
        sabores = sabores + 1.20;
    }
    //Se manda el texto a la etiqueta
    etiResultado.setText("Total de compra es:" + (tamaño + sabores) + "€");
}

```

Además del método para la Validación del teclado.

```

//Se valida la salida del programa
public void cerrar() {
    Object[] opciones = {"Aceptar", "Cancelar"};
    int eleccion = JOptionPane.showOptionDialog(rootPane, "¿Desea salir del programa?", "Mensaje de Confirmación",
        JOptionPane.YES_NO_OPTION,
        JOptionPane.QUESTION_MESSAGE, null, opciones, "Aceptar");
    if (eleccion == JOptionPane.YES_OPTION) {
        System.exit(0);
    } else {
    }
}

```

En los siguientes eventos:

```

private void btnTotalActionPerformed(java.awt.event.ActionEvent evt) {
    calculos();
}

//Evento para salir del Frame
private void formWindowClosing(java.awt.event.WindowEvent evt) {
    cerrar();
}

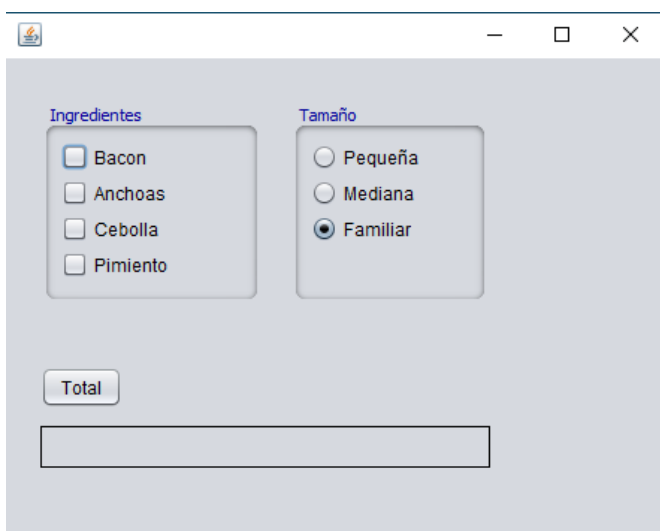
```

Al momento de ejecutar nuestro programa nos dice lo siguiente.

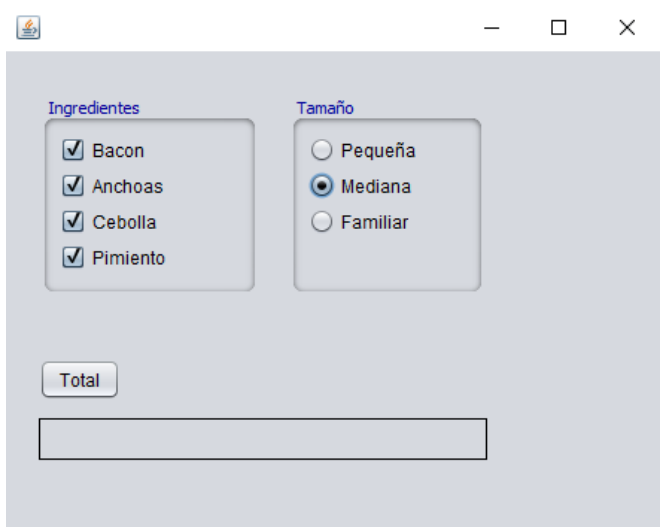


El cual es un mensaje de bienvenida antes de entrar al sistema por completo.

Al momento que se inicie, nos saldrá la opción familiar ya seleccionada, ya que se nos indica desde el programa.



Como se puede observar ya nos marca un tamaño establecido, además solo se puede seleccionar un tamaño, pero si podemos más de un ingrediente.



Por lo cual seleccionamos los siguiente y nos iremos al botón total, y nos dará la cantidad a pagar.

The screenshot shows a software window with two main sections: 'Ingredientes' and 'Tamaño'. Under 'Ingredientes', there are four checkboxes, all of which are checked: 'Bacon', 'Anchoas', 'Cebolla', and 'Pimiento'. Under 'Tamaño', there are three radio buttons: 'Pequeña', 'Mediana' (which is selected), and 'Familiar'. Below these sections is a 'Total' button and a text box displaying 'Total de compra es:14.5€'.

Nos calcula dependiendo del precio que se tenga para cada uno de los ingredientes y tamaño.

Al momento de querer salir del programa nos mandara un mensaje para confirmar la acción de la siguiente manera.

This screenshot shows the same software window as before, but with a 'Mensaje de Confirmacion' dialog box overlaid in the center. The dialog box has a question mark icon and the text '¿Desea salir del programa?'. It contains two buttons: 'Aceptar' and 'Cancelar'. The background window shows the 'Ingredientes' section with all items checked and the 'Tamaño' section with 'Pequeña' selected. The 'Total' button and the purchase total 'Total de compra es:14.5€' are also visible.

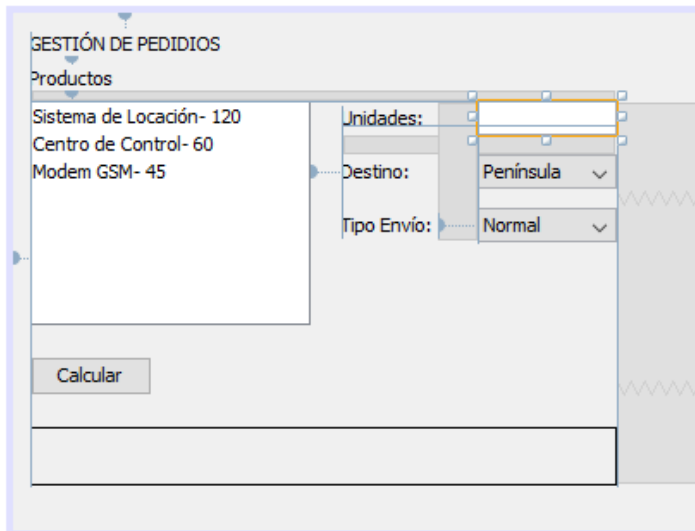
Al darle la opción salir, nos dejara abandonar por completo el programa.

Y de esa manera termina el programa.

## Ejemplo Lenguaje 9.

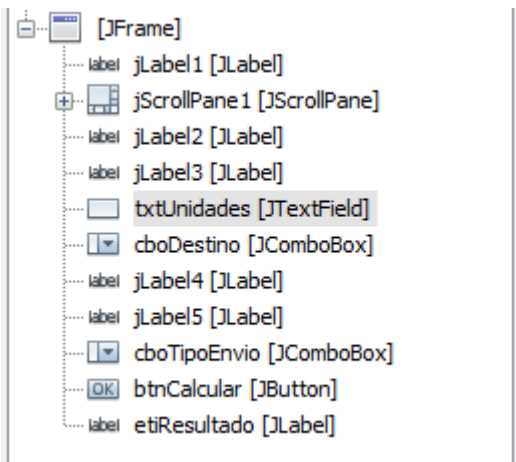
### Ejercicio 1.

En este ejercicio se nos indica hacer un programa en el cual se tenga una gestión de pedidos, el cual tendrá diferentes componentes y quedara de la siguiente manera.



Como se puede observar se usan varios componentes, como ScrollPane, ComboBox, cajas de textos, botón y etiqueta.

Los cuales se les asigno los siguientes nombres a sus variables.



Se crearon los siguientes métodos los cuales son los siguientes:

```
//Metodo que nos hace el calculo del envio
public void calculos() {
    int preciopro = 0;
    int total;
    int destino = 0;
    int envio = 0;
    int unidades = Integer.parseInt(txtUnidades.getText());
    //Se verifica que tenga un dato correspondiente, si no enviara un mensaje.
    if (lstProductos.getSelectedIndex() == -1) {
        JOptionPane.showMessageDialog(null, "No se ha seleccionado ningun producto");
    } else {
        //Hara todo el proceso del trabajo.
        if (lstProductos.getSelectedIndex() == 0) {
            preciopro = preciopro + 120;
        }
        if (lstProductos.getSelectedIndex() == 1) {
            preciopro = preciopro + 60;
        }
        if (lstProductos.getSelectedIndex() == 2) {
            preciopro = preciopro + 45;
        }
        if (cboDestino.getSelectedIndex() == 0) {
            destino = destino + 20;
        }
        if (cboDestino.getSelectedIndex() == 1) {
            destino = destino + 25;
        }
        if (cboDestino.getSelectedIndex() == 2) {
            destino = destino + 30;
        }
        if (cboTipoEnvio.getSelectedIndex() == 0) {
            envio = envio + 3;
        }
        if (cboTipoEnvio.getSelectedIndex() == 1) {
            envio = envio + 10;
        }
        //Se hace el calculo final
        total = (unidades * preciopro) + envio + destino;
        //Se envia el texto en la etiqueta correspondiente.
        etiResultado.setText("El total a pagar es: $" + total);
    }
}

//Metodo para la validacion de los datos.
public void verifNum(KeyEvent evt) {
    char validar = evt.getKeyChar();
    if (Character.isLetter(validar)) {
        getToolkit().beep();
        evt.consume();
        JOptionPane.showMessageDialog(rootPane, "Ingrese solo numeros");
    }
}
```

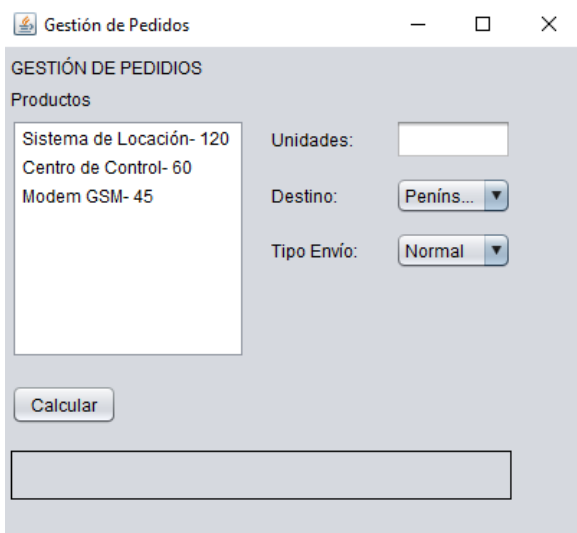
Y se utilizaron los siguientes eventos para ejecutar los métodos ya creados.

```
//Evento que nos hace el funcionamiento de los metodos.
private void btnCalcularActionPerformed(java.awt.event.ActionEvent evt) {
    calculos();
}

//Metodo para validar solo numeros.
private void txtUnidadesKeyTyped(java.awt.event.KeyEvent evt) {
    verifNum(evt);
}
```

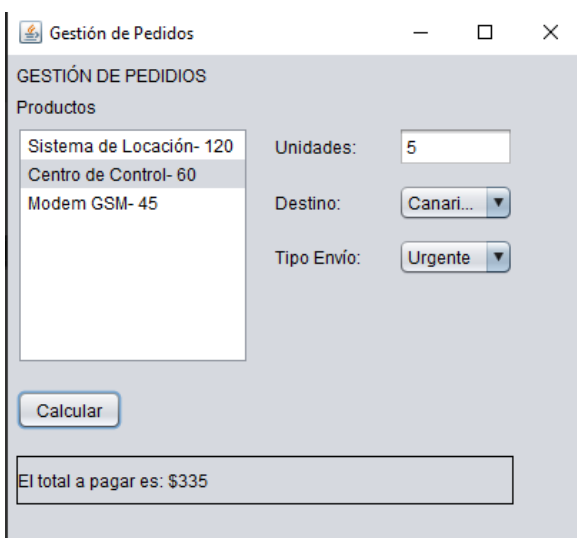


El funcionamiento del programa es el siguiente:



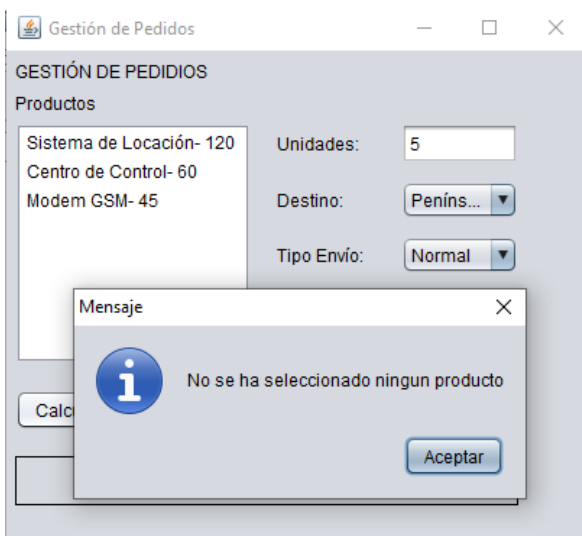
The screenshot shows a window titled "Gestión de Pedidos". Inside, there's a section "GESTIÓN DE PEDIDOS" with a "Productos" list containing "Sistema de Locación- 120", "Centro de Control- 60", and "Modem GSM- 45". To the right, there are input fields for "Unidades:" (empty), "Destino:" (dropdown menu showing "Peníns..."), and "Tipo Envío:" (dropdown menu showing "Normal"). A "Calcular" button is at the bottom left, and a large empty text box is at the bottom right.

Nos da los productos a seleccionar, las unidades y el tipo de envío, por lo cual haremos un pequeño calculo.



The screenshot shows the same window with "Unidades:" set to "5", "Destino:" set to "Canari...", and "Tipo Envío:" set to "Urgente". The "Calcular" button is now highlighted. The large text box at the bottom right now displays "El total a pagar es: \$335".

Al momento de realizar el cálculo nos dará la cantidad a pagar del costo del pedido.



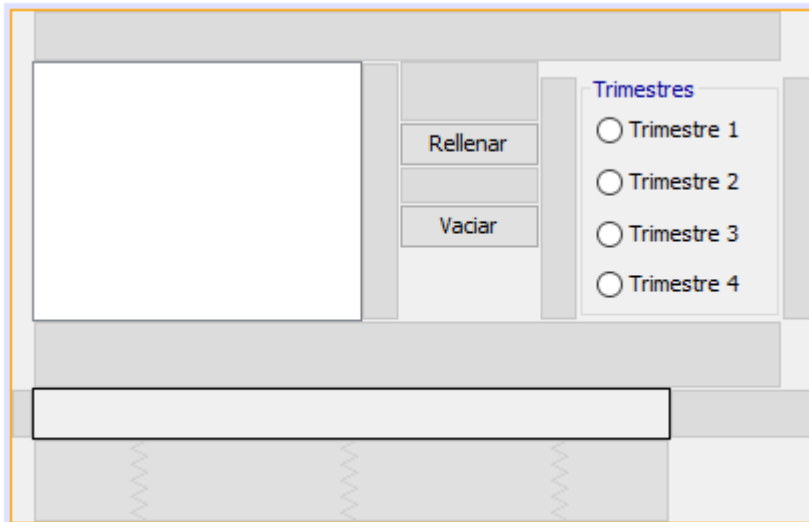
The screenshot shows the window with "Unidades:" set to "5", "Destino:" set to "Peníns...", and "Tipo Envío:" set to "Normal". An error message dialog box titled "Mensaje" is overlaid on the window. The message says "No se ha seleccionado ningun producto" (No product has been selected) and includes an "Aceptar" button.

Si no se selecciona una opción de los productos nos saldrá un mensaje en cual nos avisa.

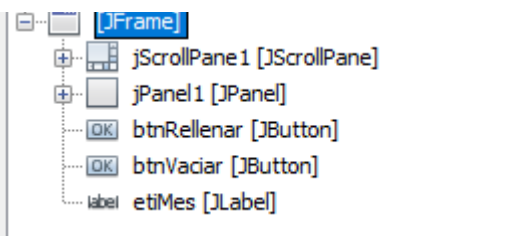
## Ejemplo Lenguaje 10.

### Ejercicio 1.

Para nuestro siguiente programa nos indica que cada radionbutton nos dará una selección de meses dependiendo el trimestre que selecciones, por lo cual se hizo el siguiente diseño en el cual s



Se les renombra a las siguientes variables.



Se crearon los siguientes métodos y eventos para el funcionamiento.

```
//Metodo para indicar los meses que se agregaran al ScrollPane
public void rellenar() {
    DefaultListModel modelo = new DefaultListModel();
    //Se selecciona esa opcion nos da los primeros meses
    if (optTri1.isSelected()) {
        modelo.addElement("Enero");
        modelo.addElement("Febrero");
        modelo.addElement("Marzo");
    }

    if (optTri2.isSelected()) {
        modelo.addElement("Abril");
        modelo.addElement("Mayo");
        modelo.addElement("Junio");
    }

    if (optTri3.isSelected()) {
        modelo.addElement("Julio");
        modelo.addElement("Agosto");
        modelo.addElement("Septiembre");
        //lstMeses.setModel(modelo);
    }
}
```

```

    if (optTri4.isSelected()) {
        modelo.addElement("Octubre");
        modelo.addElement("Noviembre");
        modelo.addElement("Diciembre");
        //lstMeses.setModel(modelo);
    }
    lstMeses.setModel(modelo);
}

```

Y los eventos para el funcionamiento.

```

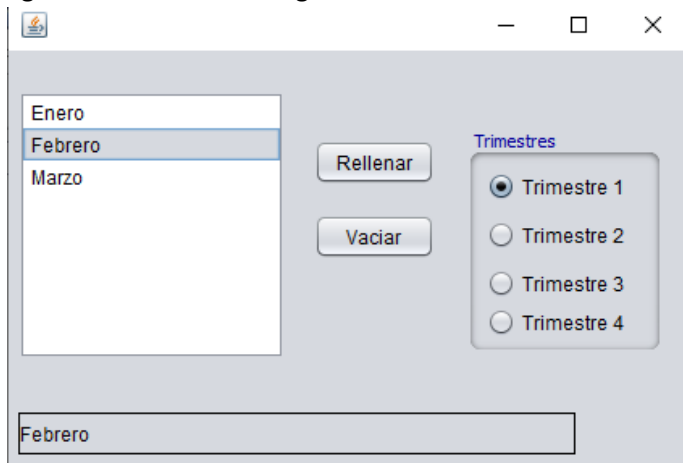
//Evento de acción para realizar la operación de relleno
private void btnRellenarActionPerformed(java.awt.event.ActionEvent evt) {
    rellenar();
}

//Nos proporciona el mes seleccionado y nos los envia a la etiqueta de texto
private void lstMesesMouseClicked(java.awt.event.MouseEvent evt) {
    etiMes.setText(lstMeses.getSelectedValue().toString());
}

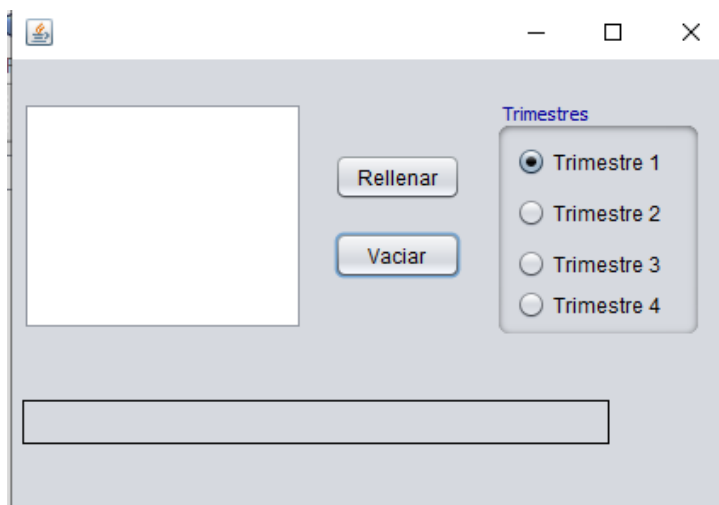
//Borra los elementos de la lista
private void btnVaciarActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultListModel modelo = new DefaultListModel();
    lstMeses.setModel(modelo);
}

```

El programa funciona de la siguiente manera.



Se selecciona un radio button y se marca el mes deseado y nos escribe en la etiqueta, además de la función del botón borrar que nos ayuda a limpiar.



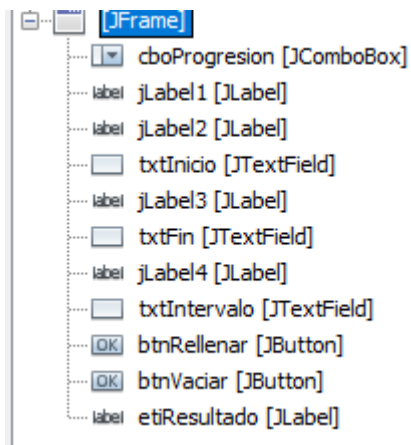
Nos borra los datos de la lista para poder seleccionar otra que deseemos.

## Ejercicio 2.

Se nos pide un programa el cual tenga un numero inicial y final, el cual le daremos un intervalo en el cual se ira iterando, por lo cual se hizo el siguiente diseño.

The image shows a Java Swing window titled "Progresión". Inside the window, there is a dropdown menu at the top. Below it, there are three labels: "Inicio:", "Fin:", and "Intervalo:", each followed by a text input field. Under these input fields, there are two buttons: "Rellenar" and "Vaciar". At the bottom of the window, there is a large rectangular area, likely a text area or a list, which is currently empty.

Se renombraron las siguientes variables.



Y se crearon los siguientes métodos para el funcionamiento.

```
//Metodo para obtener los numeros.
public void rellenar() {

    DefaultComboBoxModel modelo = new DefaultComboBoxModel();
    int i;
    int inicio = Integer.parseInt(txtInicio.getText());
    int fin = Integer.parseInt(txtFin.getText());
    int intervalo = Integer.parseInt(txtIntervalo.getText());
    if (fin > inicio) {
        for (inicio = inicio; inicio <= fin; inicio += intervalo) {
            modelo.addElement(inicio);
        }
        cboProgresion.setModel(modelo);
    } else {
        JOptionPane.showMessageDialog(null, "El tamaño de inicio debe ser menor al final");
    }
}
```

```
//Metodos para los eventos.
public void metodosteclado() {
    btnRellenar.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            btnRellenarActionPerformed(evt);
        }
    });
    btnVaciar.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            btnVaciarActionPerformed(evt);
        }
    });
    cboProgresion.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            cboProgresionActionPerformed(evt);
        }
    });
    txtFin.addKeyListener(new KeyAdapter() {
        public void keyTyped(KeyEvent evt) {
            txtFinKeyTyped(evt);
        }
    });
    txtInicio.addKeyListener(new KeyAdapter() {
        public void keyTyped(KeyEvent evt) {
            txtInicioKeyTyped(evt);
        }
    });
    txtIntervalo.addKeyListener(new KeyAdapter() {
        public void keyTyped(KeyEvent evt) {
            txtIntervaloKeyTyped(evt);
        }
    });
}
```

```
//Metodo de validacion de teclado
public void verif(KeyEvent evt){
    char validar = evt.getKeyChar();
    if(Character.isLetter(validar)){
        getToolkit().beep();
        evt.consume();
        JOptionPane.showMessageDialog(rootPane, "Ingrese solo numeros");
    }
}

//Metodo para borrar.
public void borrar() {
    DefaultComboBoxModel modelo = new DefaultComboBoxModel();
    cboProgresion.setModel(modelo);
}
}
```

Y los siguientes eventos.

```
//Eventos de acciones
public void btnRellenarActionPerformed(ActionEvent evt) {
    rellenar();
}

public void btnVaciarActionPerformed(ActionEvent evt) {
    borrar();
}

public void cboProgresionActionPerformed(ActionEvent evt) {
    etiResultado.setText(cboProgresion.getSelectedItem().toString());
}

//Eventos para validacion.
public void txtFinKeyTyped(KeyEvent evt){
    verif(evt);
}

public void txtInicioKeyTyped(KeyEvent evt){
    verif(evt);
}

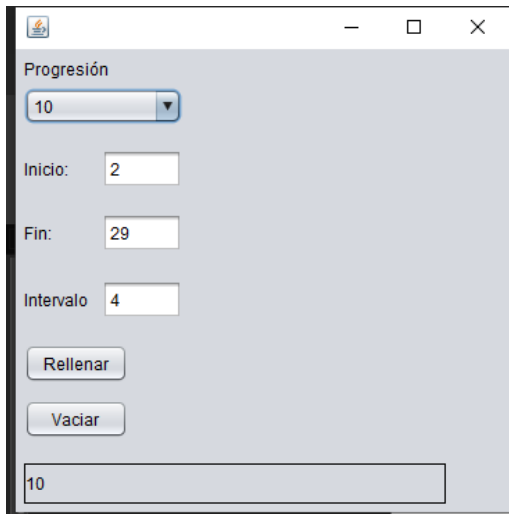
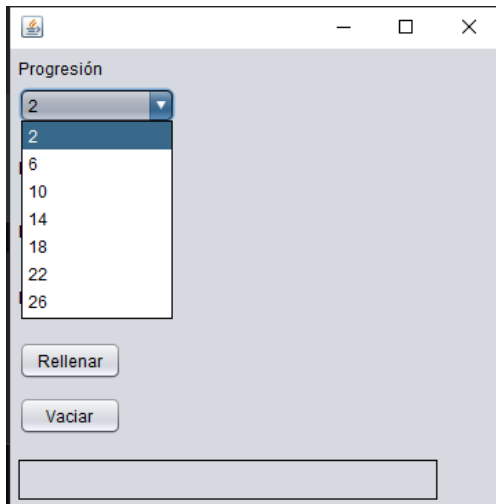
public void txtIntervaloKeyTyped(KeyEvent evt){
    verif(evt);
}
}
```

El funcionamiento del programa es el siguiente.

The screenshot shows a Java Swing window titled "Progresión". It has a light gray background. At the top, there is a dropdown menu with the value "2" selected. Below the dropdown are three text input fields. The first is labeled "Inicio:" and contains the number "2". The second is labeled "Fin:" and contains the number "29". The third is labeled "Intervalo" and contains the number "4". Below these input fields are two buttons: "Rellenar" and "Vaciar". At the bottom of the window is a large, empty rectangular text area.

Damos los parámetros a calcular.

Y verificamos la lista nueva, además que nos escribirá el dato seleccionado.



Con lo cual damos concluido este programa.