



Centro de Investigación en Computación del IPN

Java – Diplomado

Modulo 2

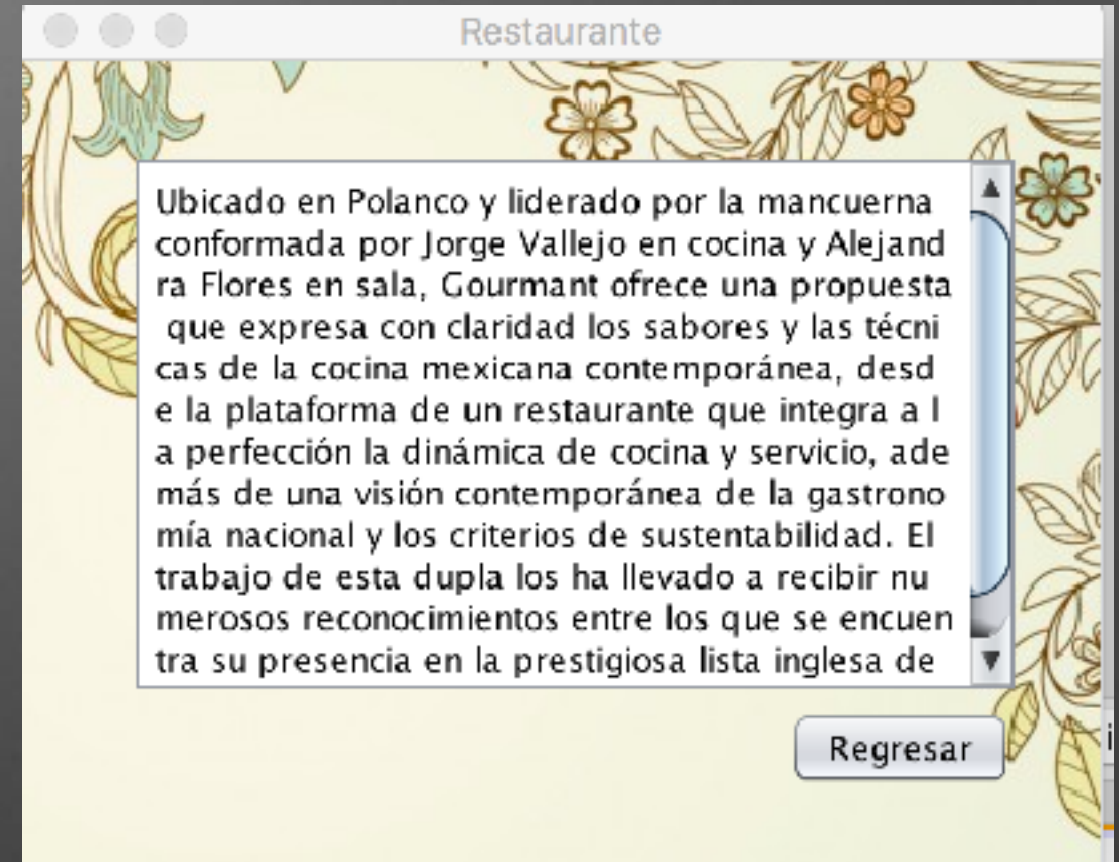
Dra. Erandi Castillo Montiel

Restaurante



Acerca de

- JFrame (fondo, icono)
- TextArea
- Lectura de un archivo de text



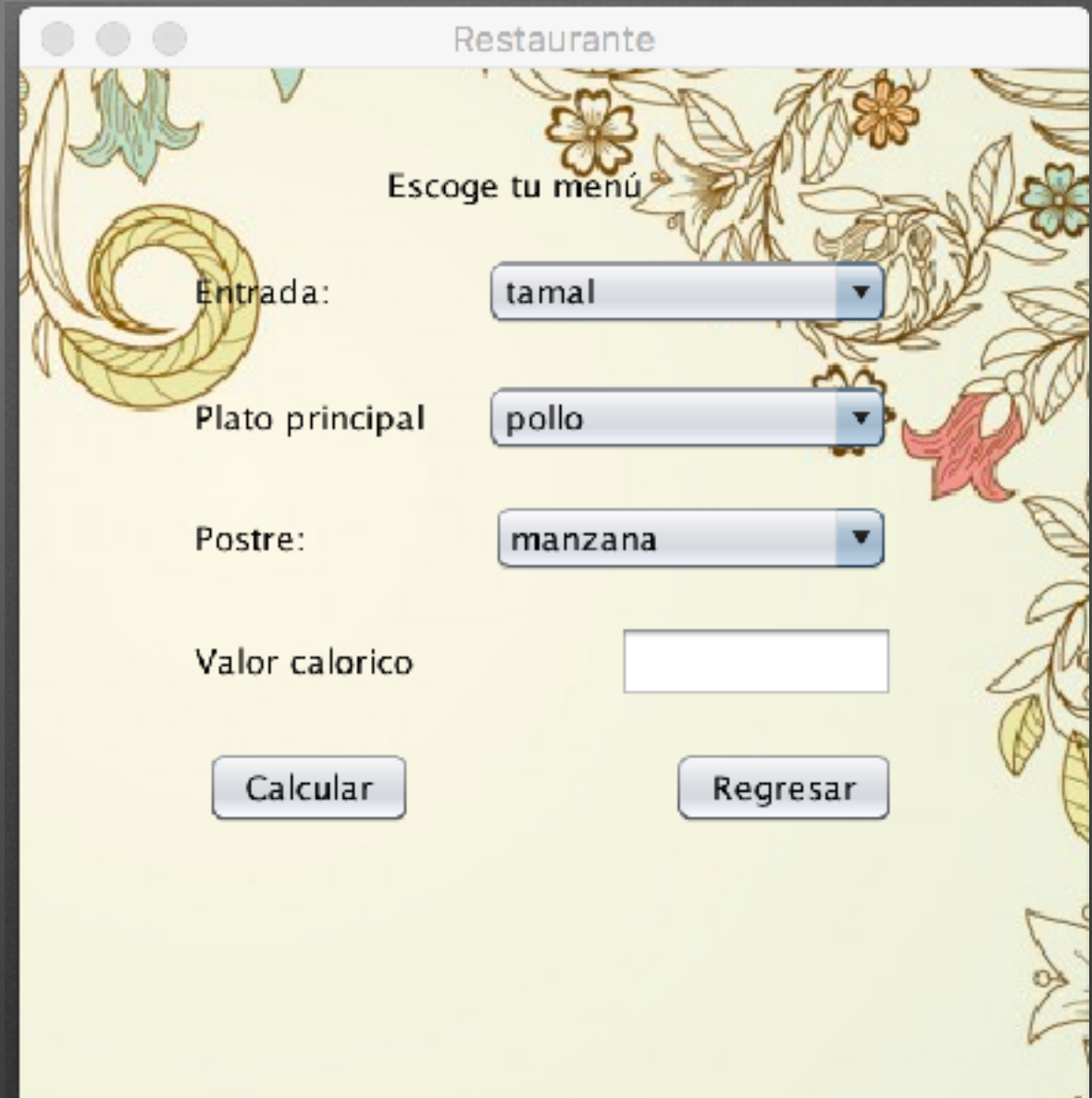
Nuestros expertos

- JForm (fondo, icono)
- 2 RadioButton
- TextArea
- Mensaje de seleccionar opción
- Lectura de un archivo de texto determinada por la elección del usuario



Valor calórico

- JForm (fondo y icono)
- ComboBox, con carga automática
- Clase llamada menú que calcula las calorías



Restaurante

Escoge tu menú

Entrada:

Plato principal

Postre:

Valor calórico

Carga ComboBox

En una clase llamada Carga - metodo cargaEntrada

```
public DefaultComboBoxModel cargaEntrada (){  
    DefaultComboBoxModel model = new DefaultComboBoxModel();  
    // edades[i-min]=String.valueOf(i);  
    model.addElement("tamal");  
    model.addElement("godita");  
    model.addElement("tlatoyo");  
  
    return model;  
}
```

Dentro del constructor del JForm OpcionesUI

```
public OpcionesUI() {  
    initComponents();  
    setLocationRelativeTo(null);  
    setResizable(false);  
    setTitle("Restaurante");  
    this.setIconImage(new ImageIcon(getClass().getResource("/imagenes/icono.png")).getImage());  
    ((JPanel) getContentPane()).setOpaque(false); //opcionesRestaurante2  
    ImageIcon uno = new ImageIcon(this.getClass().getResource("/imagenes/opcionesRestaurante2.jpg"));  
    JLabel fondo = new JLabel();  
    fondo.setIcon(uno);  
    getLayeredPane().add(fondo, JLayeredPane.FRAME_CONTENT_LAYER);  
    fondo.setBounds(0,0,uno.getIconWidth(),uno.getIconHeight());  
    Carga menus = new Carga();  
    entradaCB.setModel(menus.cargaEntrada());  
    principalCB.setModel(menus.cargaPrincipal());  
    postreCB.setModel(menus.cargaPostre());  
}
```

Lectura Archivos

```
public String leerAcercaDe() {  
  
    File archivo = null;  
    FileReader fr = null;  
    BufferedReader br = null;  
    String result = "";  
    try {  
        // Apertura del fichero y creacion de BufferedReader para poder  
        // hacer una lectura comoda (disponer del metodo readLine()).  
        archivo = new File("src/informacionRestaurante/acercaDe.txt");  
        fr = new FileReader(archivo);  
        br = new BufferedReader(fr);  
  
        // Lectura del fichero  
        String linea = "";  
        while ((linea = br.readLine()) != null) {  
            // System.out.println("datos: " + linea);  
            result = linea;  
        }  
        // return linea;  
    } catch (Exception e) {  
        e.printStackTrace();  
    } finally {  
        // En el finally cerramos el fichero, para asegurarnos  
        // que se cierra tanto si todo va bien como si salta  
        // una excepcion.  
  
        try {  
            if (null != fr) {  
                fr.close();  
            }  
        } catch (Exception e2) {  
            e2.printStackTrace();  
        }  
    }  
    return result;  
}
```

Clase Menu - calcular total calorías

```
private void calculaBtnActionPerformed(java.awt.event.ActionEvent evt) {  
    double resultado;  
    Menu client = new Menu(entradaCB.getSelectedIndex(), principalCB.getSelectedIndex(),  
                           postreCB.getSelectedIndex());  
    resultado=client.calculaCalorias();  
  
    valorText.setText(String.valueOf(resultado));  
}
```


Alumno - JList

The screenshot shows a Java Swing window with a light blue background. At the top, there are three standard macOS window control buttons (red, yellow, green). The window contains the following elements:

- Form Fields:**
 - Nombre:** A text field containing the text "luis".
 - Apellidos:** A text field containing the text "castillo montiel".
 - Edad:** A spinner field with the value "22" and a downward arrow on the right.
- Buttons:**
 - Agregar:** A button with a light blue gradient and rounded corners.
 - Eliminar:** A button with a light blue gradient and rounded corners.
- Datos:** A label positioned above a summary of the current data.
 - Nombre:** luis
 - Apellidos:** castillo montiel
 - Edad:** 22
- JList:** A vertical list box on the right side of the window containing three items: "erandi", "luis", and "luis". The second "luis" item is currently selected, highlighted with a darker background.

- Clase Alumno

Nombre(String), Apellido (String), Edad (Integer)

- Carga automatica del ComboBox edad
- Crear un ModelList personalizado

Clase Carga- método cargaEdad

```
public DefaultComboBoxModel cargaEdad (int min,int max){  
    DefaultComboBoxModel model = new DefaultComboBoxModel();  
    String edades[]=null;  
    for (int i = min; i <max; i++) {  
        //      edades[i-min]=String.valueOf(i);  
        model.addElement(String.valueOf(i));  
    }  
    return model;  
}
```

ModelList modificado

```
public class ModelListModificado extends AbstractListModel {  
  
    private ArrayList<Alumno> lista = new ArrayList<>();  
  
    @Override  
    public int getSize() {  
        return lista.size();  
    }  
  
    @Override  
    public Object getElementAt(int index) {  
        Alumno p = lista.get(index);  
        return p.getNombre();  
    }  
  
    public void addAlumno(Alumno p) {  
        lista.add(p);  
        //notifica al Jlis que un elemento ha sido modificado  
        this.fireIntervalAdded(this, getSize(), getSize() + 1);  
    }  
  
    public void eliminarAlumno(int index0) {  
        lista.remove(index0);  
        this.fireIntervalRemoved(index0, getSize(), getSize() + 1);  
    }  
  
    public Alumno getAlumno(int index) {  
        return lista.get(index);  
    }  
}
```


Constructor de AlumnoUI

```
public AlumnoUI() {  
    initComponents();  
    Carga datos= new Carga();  
    cmbEdad.setModel(datos.cargaEdad(18,99));  
    // listModel= new DefaultListModel();  
    listModel = new ModelListModificado();  
    listAlumno.setModel(listModel); //Jlist  
}
```

Botones

```
private void btnAgregarActionPerformed(java.awt.event.ActionEvent evt) {  
    Alumno nuevo = new Alumno();  
    nuevo.setNombre(textNombre.getText());  
    nuevo.setApellido(textApellidos.getText());  
    nuevo.setEdad(Integer.parseInt(cmbEdad.getSelectedItem().toString()));  
    listModel.addAlumno(nuevo);  
    // listModel.addElement(nuevo);  
}  
  
private void listAlumnoValueChanged(javax.swing.event.ListSelectionEvent evt) {  
    int index= listAlumno.getSelectedIndex();  
  
    if(index!=-1){  
        Alumno selectAlumno=(Alumno)listModel.getAlumno(index);  
  
        textNombre1.setText(selectAlumno.getNombre());  
        textApellidos1.setText(selectAlumno.getApellido());  
        textEdad1.setText(String.valueOf(selectAlumno.getEdad()));  
    }  
}  
  
private void btnEliminarActionPerformed(java.awt.event.ActionEvent evt) {  
    int index= listAlumno.getSelectedIndex();  
    if (index!=-1) {  
        listModel.eliminarAlumno(index);  
    }  
}
```

Opciones de ventanas Dialogo



Opciones

```
private void btnOp1ActionPerformed(java.awt.event.ActionEvent evt) {  
    JOptionPane.showMessageDialog(null, "Operación realizada correctamente");  
  
    //1. componente padre  
    //2. mensaje  
}
```

```
private void btnOp2ActionPerformed(java.awt.event.ActionEvent evt) {  
    JOptionPane.showMessageDialog(null, "Mensaje dentro de la ventana",  
        "Mensaje en la barra de titulo", JOptionPane.WARNING_MESSAGE);  
    //1. componente padre  
    //2. mensaje  
    //3. titulo  
    //4. icono  
}
```

```
private void btnOp3ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // Image image = new ImageIcon(getClass().getResource("/imagenes/icono.png")).getImage();  
    // this.setIconImage(image);  
    MyIcon icon = new MyIcon();  
    JOptionPane.showMessageDialog(null, "Mensaje dentro de la ventana",  
        "Titulo", JOptionPane.DEFAULT_OPTION, icon);  
  
}
```

```
private void btnOp4ActionPerformed(java.awt.event.ActionEvent evt) {  
String respuesta = JOptionPane.showInputDialog("Escribe tu nombre"); // texto interno  
//String respuesta = JOptionPane.showInputDialog("¿Cual es su nombre?", "atxy2k"); //con texto inicial  
//String respuesta = JOptionPane.showInputDialog(null, "Escribe tu nombre"); //indica el componente padre  
  
    System.out.println("respuesta"+ respuesta);  
}
```



```
private void btnOp5ActionPerformed(java.awt.event.ActionEvent evt) {  
    String respuesta = JOptionPane.showInputDialog(null,  
        "Escriba nuevamente su nombre", "Error!", JOptionPane.ERROR_MESSAGE);  
    System.out.println("respuesta" + respuesta);  
}
```

```
private void btnOp6ActionPerformed(java.awt.event.ActionEvent evt) {  
    String[] carreras = {  
        "Ingeniería en sistemas computacionales",  
        "Ingeniería industrial",  
        "Ingeniería en mecatrónica",  
        "Ingeniería en informática",  
        "Ingeniería petroquímica"  
    };  
    MyIcon icon = new MyIcon();  
    String resp = (String) JOptionPane.showInputDialog(null, "Seleccione una carrera a cursar",  
        "Carrera", JOptionPane.DEFAULT_OPTION, icon, carreras, carreras[0]);  
    System.out.println("respuesta"+ resp);  
}
```

```
private void btnOp7ActionPerformed(java.awt.event.ActionEvent evt) {  
    //int resp = JOptionPane.showConfirmDialog(null, "¿Está seguro?");  
    //cuarto parametro es el tipo de respuesta que puede dar  
    int resp = JOptionPane.showConfirmDialog(null, "¿Esta seguro?",  
        "Alerta!", JOptionPane.YES_NO_OPTION);  
  
    System.out.println("respuesta "+ resp);  
}
```

```
private void btnOp9ActionPerformed(java.awt.event.ActionEvent evt) {  
    MyIcon icon = new MyIcon();  
    String[] options = {"Opcion A", "Opcion B", "Opcion C", "Opcion D"};  
    int seleccion = JOptionPane.showOptionDialog(null, "Es necesario que seleccione una opcion",  
        "Titulo", JOptionPane.DEFAULT_OPTION, JOptionPane.QUESTION_MESSAGE, icon, options, options[0]);  
    System.out.println("respuesta "+ seleccion);  
}
```

```
private void btnOp8ActionPerformed(java.awt.event.ActionEvent evt) {  
    //se agrega icono  
    int resp = JOptionPane.showConfirmDialog(null, "¿Esta seguro?", "Alerta!",  
        JOptionPane.YES_NO_OPTION, JOptionPane.ERROR_MESSAGE);  
    System.out.println("respuesta "+ resp);  
}
```

```
import java.awt.Graphics;
import java.awt.Image;
import javax.swing.Icon;
import javax.swing.ImageIcon;

public class MyIcon implements Icon{

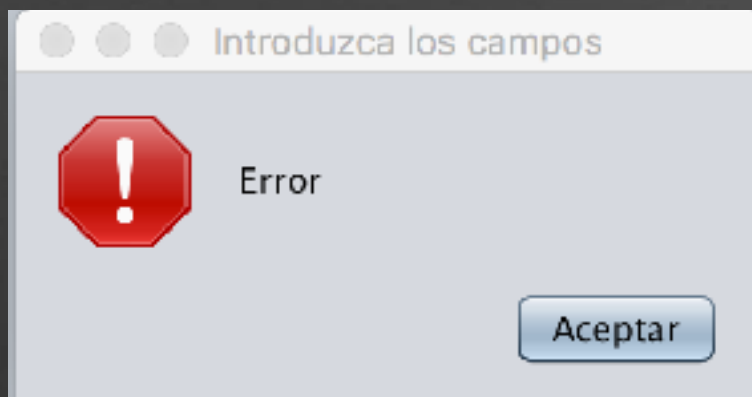
    @Override
    public int getIconWidth() {
        return 5;
    }

    @Override
    public int getIconHeight() {
        return 5;
    }

    @Override
    public void paintIcon(Component c, Graphics g, int x, int y) {
        Image image = new ImageIcon(getClass().getResource("/imagenes/icono.png")).getImage();
        g.drawImage(image, x, y, c);
    }

}
```

Dialogo personalizado



Crear un JDialog y modificar su constructor

```
public class AgregarAlumno extends javax.swing.JDialog {

    public final static int ACCEPT_OPTION = 1;
    public final static int CANCEL_OPTION = 2;
    private int SELECT_OPTION = CANCEL_OPTION;
    private Alumno alum = null;

    /**
     * Creates new form AgregarAlumno
     */
    public AgregarAlumno(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        initComponents();
        Carga edad = new Carga();
        cmbEdad.setModel(edad.cargaEdad(18, 45));
    }
}
```


Funcionalidad de los botones

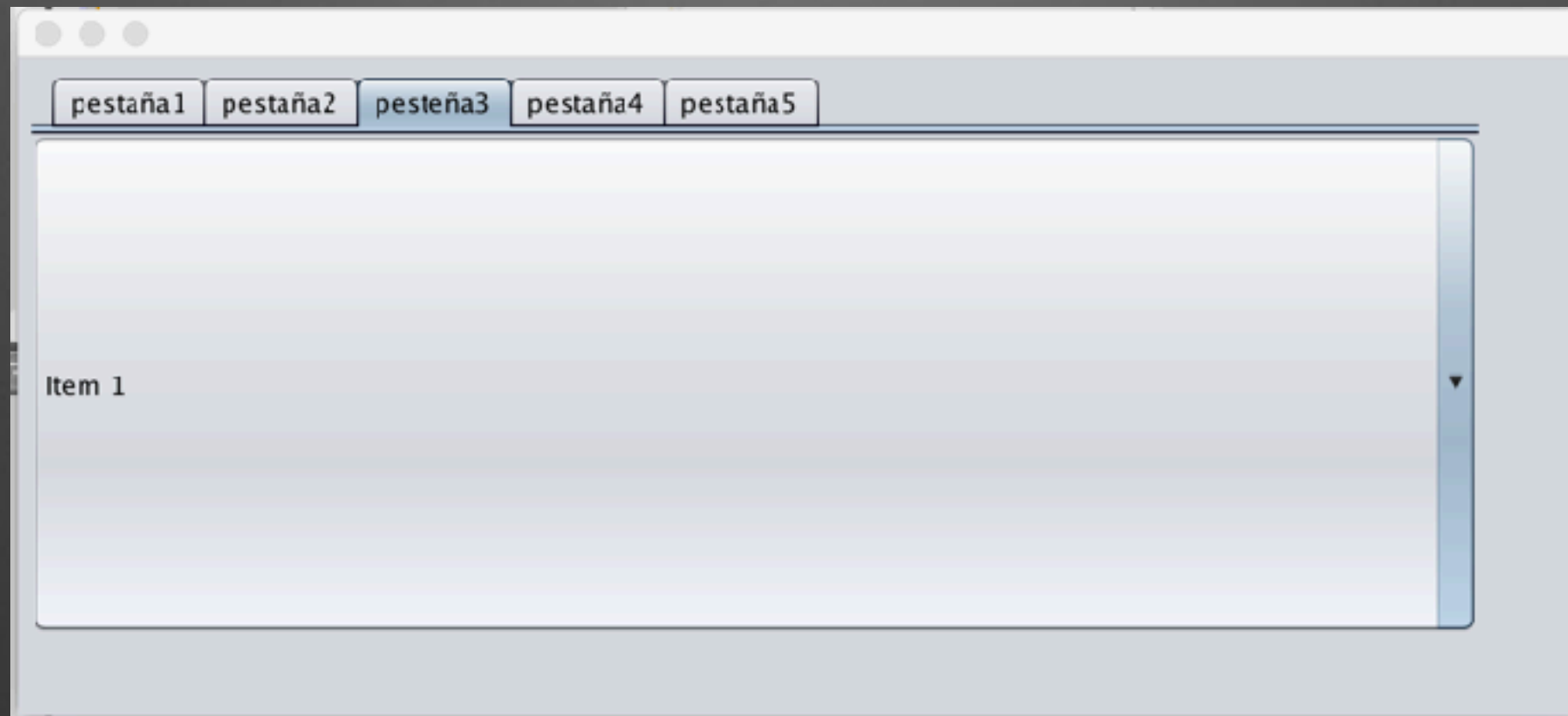
```
private void btnAceptarActionPerformed(java.awt.event.ActionEvent evt) {  
    String nombre = textNombre.getText();  
    String apellido = textApellido.getText();  
    String edad = cmbEdad.getSelectedItem().toString();  
    if (!nombre.isEmpty() && !apellido.isEmpty()  
        && !edad.isEmpty()) {  
        alum = new Alumno(nombre, apellido, Integer.parseInt(edad));  
        SELECT_OPTION = ACCEPT_OPTION;  
        this.setVisible(false);  
    } else {  
        JOptionPane.showMessageDialog(null, "Error", "Introduzca los campos", JOptionPane.ERROR_MESSAGE);  
    }  
}  
  
private void btnCancelarActionPerformed(java.awt.event.ActionEvent evt) {  
    SELECT_OPTION = CANCEL_OPTION;  
    this.setVisible(false);  
    System.out.println("Se cancelo la operación");  
}
```

En la clase padre FormAgregarAlumno

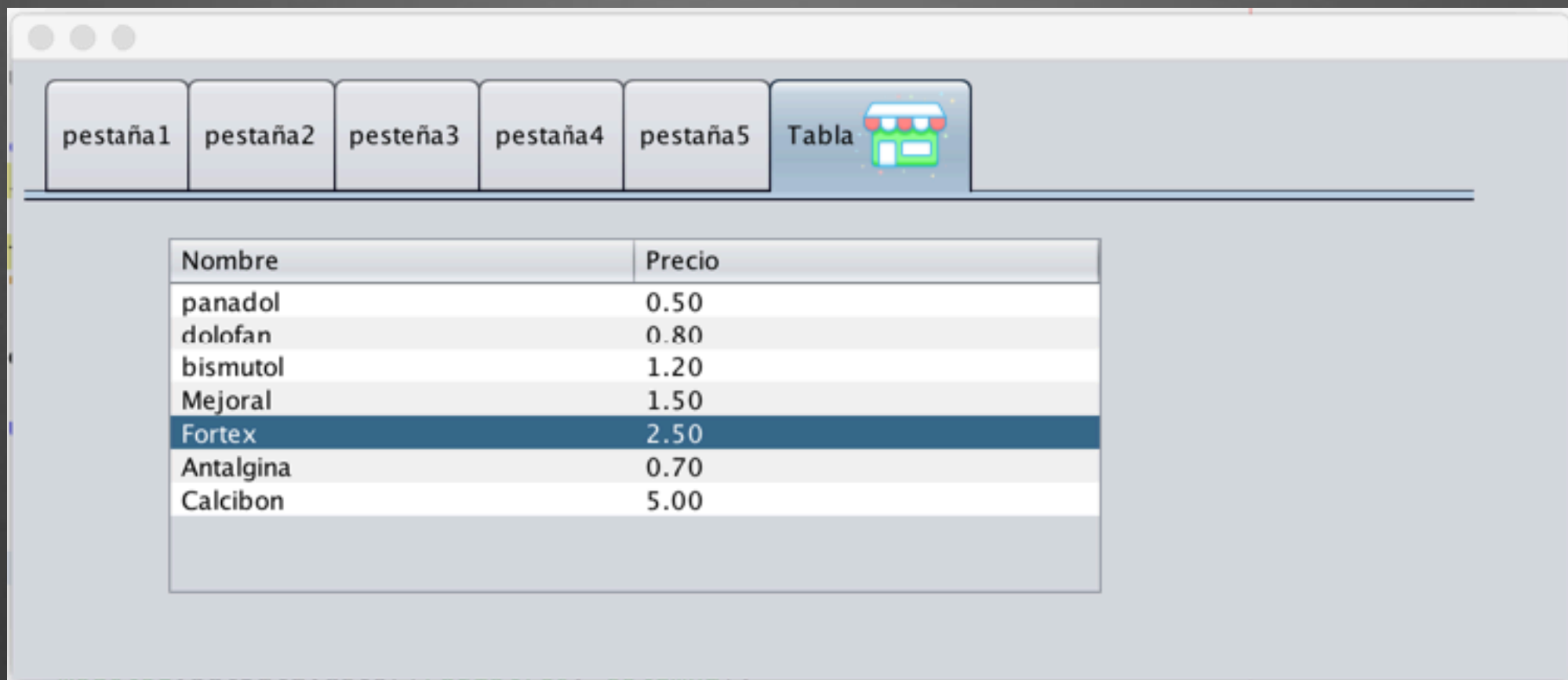
```
private void btnAgregarActionPerformed(java.awt.event.ActionEvent evt) {  
    AgregarAlumno dialogo = new AgregarAlumno(null,true);  
    //1. Componente padre  
    //2. queda suspendida la ventana anterior  
    dialogo.setVisible(true);  
    if(dialogo.getSelectedOption()==AgregarAlumno.ACCEPT_OPTION){  
        Alumno registrado = dialogo.getAlum();  
        JOptionPane.showMessageDialog(rootPane,"se acaba de registrar a"+ registrado.getNombre());  
    }else{  
        ///no hace nada  
        JOptionPane.showMessageDialog(rootPane,"no se registro");  
    }  
}
```

Implementar la misma metodología pero para el menú del restaurante

Tabbed Pane



Table



Nombre	Precio
panadol	0.50
dolofan	0.80
bismutol	1.20
Mejoral	1.50
Fortex	2.50
Antalgina	0.70
Calcibon	5.00

```
public class TabbedPanel extends javax.swing.JFrame {  
    String[] columna = {"Nombre", "Precio"};  
  
    String[][] registros = {{ "panadol", "0.50"}, {"dolofan", "0.80"}, {"bismutol", "1.20"},  
        {"Mejoral", "1.50"}, {"Fortex", "2.50"}, {"Antalgina", "0.70"}, {"Calcibon", "5.00"} };  
  
    DefaultTableModel modelo1 = new DefaultTableModel();  
  
    public TabbedPanel() {  
        initComponents();  
  
        ImageIcon imagen = new ImageIcon(getClass().getResource("/imagenes/farmacy2.png"));  
        jTabbedPane1.setIconAt(5, imagen);  
        jTable1.setModel(modelo1);  
        //asigna losvalores a la tabla  
        modelo1.setDataVector(registros, columna);  
    }  
}
```

