

Estructura de selección doble if - else

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al finalizar la unidad, el alumno, mediante el uso de estructuras de selección, diseña algoritmos que involucren procesos selectivos.

TEMARIO

1. Estructura de selección doble if-else
2. Operadores lógicos y relacionales

ACTIVIDADES

Los alumnos desarrollan algoritmos que involucren estructuras de selección if-else.

1. Estructura de selección doble if - else

La *estructura de selección doble if...else (si...sino)* evalúa una condición lógica y en caso que resulte verdadera efectúa la acción **acciona**; de lo contrario, efectúa la acción **accionb**. Tanto **acciona** como **accionb** pueden ser **acciones simples** (una sola acción) o **acciones compuestas** (un bloque de acciones).

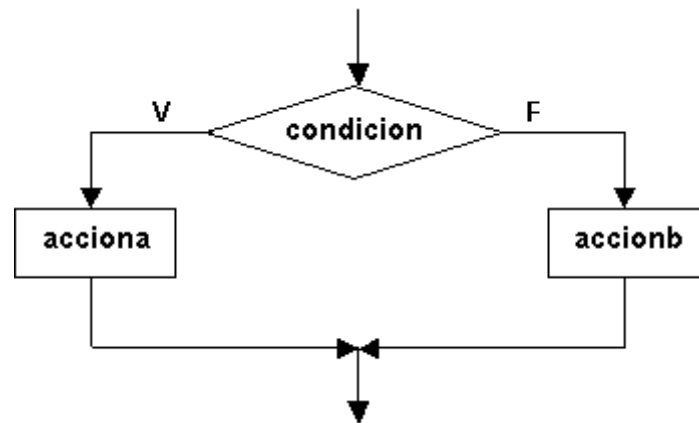


Figura 1 Diagrama de flujo de la estructura de selección if-else

En la tabla que sigue, se muestra el código y el pseudocódigo de la estructura de selección simple if. Note que, en el caso de bloques de acciones (más de una acción), estas deben estar encerradas entre llaves de bloque { }.

Código Java	Pseudocódigo
<pre> if(condicion) accionA; else accionB; </pre>	<pre> si(condicion) accionA sino accionB </pre>
<pre> if(condicion){ acciónA1; acciónA2; . . . accionAn; } else{ acciónB1; acciónB2; . . . accionBn; } </pre>	<pre> si(condicion){ acciónA1 acciónA2 . . . accionAn } sino{ acciónB1 acciónB2 . . . accionBn } </pre>

2. Operadores lógicos y relacionales

Son operadores que se utilizan para crear condiciones lógicas. Una condición lógica es una expresión lógica que puede ser verdadera (**true**) o falsa (**false**) y puede incluir operadores aritméticos.

2.1 Operadores relacionales

Los operadores relacionales se utilizan para escribir condiciones que describan la relación entre dos valores. El conjunto de operadores relacionales se muestran en la Tabla 1.

Tabla 1 Operadores relacionales

Operador	Significado
==	Igual a
!=	Diferente de
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que

2.2 Operadores lógicos

Son operadores que permiten relacionar varias expresiones lógicas. El conjunto de operadores lógicos se muestra en la Tabla 2.

Tabla 2 Operadores lógicos

Operador	Significado
	OR lógico (ó)
&&	AND lógico (y)
!	NOT lógico (no)

Las tablas de verdad de los operadores lógicos son las mismas de la lógica matemática, como se muestra en la Tabla 3.

Tabla 3 Tabla de verdad de los operadores lógicos

p	q	p && q	p q	!p
true	true	true	true	false
true	false	false	true	false
false	true	false	true	true
false	false	false	false	true

3. Problemas propuestos

Problema 1

Una tienda ha puesto en oferta la venta de un producto ofreciendo un porcentaje de descuento sobre el importe de la compra de acuerdo con la siguiente tabla:

Docenas adquiridas	Descuento
≥ 10	15%
< 10	11%

Adicionalmente, la tienda obsequia lapiceros de acuerdo con a la siguiente tabla:

Importe a pagar	Lapiceros
≥ 200	2 por cada docena
< 200	0

Dado el precio de la docena y la cantidad de docenas adquiridas, diseñe un programa que determine el importe de la compra, el importe del descuento, el importe a pagar y la cantidad de lapiceros de obsequio.

Algoritmo

Inicio

```
// Declaración de variables
entero cantidad, lapiceros
real impcom, impdes, imppag, precio

// Entrada de datos
Leer cantidad, precio

// Calcula el importe de la compra
impcom = cantidad*precio

// Calcula el importe del descuento
si(cantidad >= 10)
    impdes = 0.15*impcom
sino
    impdes = 0.11*impcom

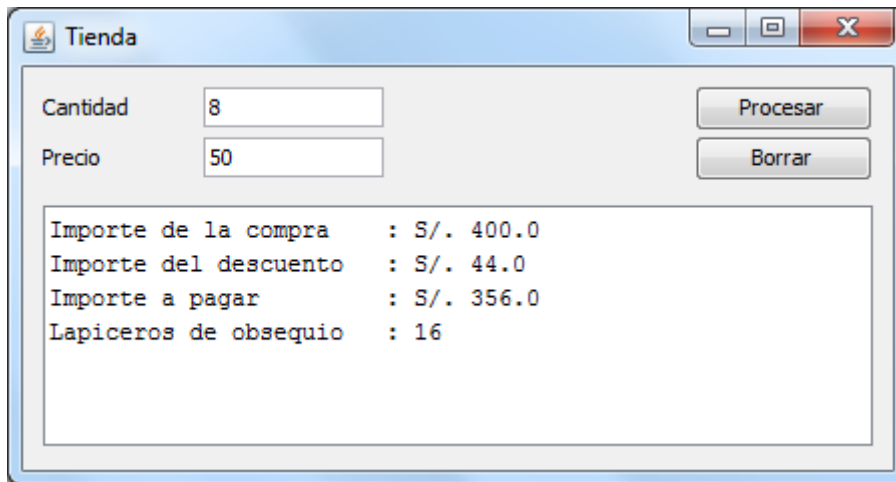
// Calcula el importe a pagar
imppag = impcom - impdes

// Calcula los lapiceros de obsequio
si(impag >= 200)
    lapiceros = 2*cantidad
sino
    lapiceros = 0

// Salida de resultados
Imprimir impcom, impdes, imppag, lapiceros
```

Fin

Programa



```
package cibertec;

import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

public class Tienda extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblCantidad;
    private JLabel lblPrecio;
    private JTextField txtCantidad;
    private JTextField txtPrecio;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;

    // Lanza la aplicación
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(
                "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        }
        catch (Throwable e) {
            e.printStackTrace();
        }
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Tienda frame = new Tienda();
                    frame.setVisible(true);
                }
            }
        });
    }
}
```

```

        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
});
}

// Crea la GUI
public Tienda() {
    setTitle("Tienda");
    setBounds(100, 100, 450, 239);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    lblCantidad = new JLabel("Cantidad");
    lblCantidad.setBounds(10, 13, 80, 14);
    getContentPane().add(lblCantidad);

    lblPrecio = new JLabel("Precio");
    lblPrecio.setBounds(10, 38, 80, 14);
    getContentPane().add(lblPrecio);

    txtCantidad = new JTextField();
    txtCantidad.setBounds(90, 10, 90, 20);
    getContentPane().add(txtCantidad);
    txtCantidad.setColumns(10);

    txtPrecio = new JTextField();
    txtPrecio.setBounds(90, 35, 90, 20);
    getContentPane().add(txtPrecio);
    txtPrecio.setColumns(10);

    btnProcesar = new JButton("Procesar");
    btnProcesar.addActionListener(this);
    btnProcesar.setBounds(335, 9, 89, 23);
    getContentPane().add(btnProcesar);

    btnBorrar = new JButton("Borrar");
    btnBorrar.addActionListener(this);
    btnBorrar.setBounds(335, 34, 89, 23);
    getContentPane().add(btnBorrar);

    scpScroll = new JScrollPane();
    scpScroll.setBounds(10, 69, 414, 120);
    getContentPane().add(scpScroll);

    txtS = new JTextArea();
    txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
    scpScroll.setViewportView(txtS);
}

// Direcciona eventos de tipo ActionEvent
public void actionPerformed(ActionEvent arg0) {
    if (arg0.getSource() == btnProcesar) {
        actionPerformedBtnProcesar(arg0);
    }
    if (arg0.getSource() == btnBorrar) {
        actionPerformedBtnBorrar(arg0);
    }
}

```

```

    }
}

// Procesa la pulsación del botón Procesar
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declaración de variables
    int cantidad, lapiceros;
    double impcom, impdes, imppag, precio;

    // Entrada de datos
    cantidad = Integer.parseInt(txtCantidad.getText());
    precio = Double.parseDouble(txtPrecio.getText());

    // Calcula el importe de la compra
    impcom = cantidad * precio;

    // Calcula el importe del descuento
    if (cantidad >= 10)
        impdes = 0.15 * impcom;
    else
        impdes = 0.11 * impcom;

    // Calcula el importe a pagar
    imppag = impcom - impdes;

    // Calcula los lapiceros de obsequio
    if (imppag >= 200)
        lapiceros = 2 * cantidad;
    else
        lapiceros = 0;

    // Salida de resultados
    txtS.setText("Importe de la compra      : S/. " + impcom + "\n");
    txtS.append("Importe del descuento    : S/. " + impdes + "\n");
    txtS.append("Importe a pagar          : S/. " + imppag + "\n");
    txtS.append("Lapiceros de obsequio     : " + lapiceros);
}

// Procesa la pulsación del botón Borrar
protected void actionPerformedBtnBorrar(ActionEvent arg0) {
    txtCantidad.setText("");
    txtPrecio.setText("");
    txtS.setText("");
    txtCantidad.requestFocus();
}
}

```

Problema 2

Una empresa de transportes cubre la ruta Lima-Huánuco en dos turnos: mañana y noche. Los precios de los pasajes se dan en la siguiente tabla:

Turno	Precio del pasaje
Mañana	S/. 37.5
Noche	S/. 45.0

Como oferta especial, la empresa aplica un porcentaje de descuento sobre el importe de la compra de acuerdo con la siguiente tabla:

Cantidad de pasajes	Descuento
≥ 15	8%
< 15	5%

Adicionalmente, la empresa obsequia caramelos de acuerdo con la siguiente tabla:

Importe a pagar	Caramelos
> 200	2 por cada boleto
≤ 200	0

Dado el turno y la cantidad de pasajes adquiridos por un cliente, diseñe un programa que determine el importe de la compra, el importe del descuento, el importe a pagar y la cantidad de caramelos de obsequio.

Algoritmo

Inicio

```
// Declaración de variables
entero turno, cantidad, caramelos
real impcom, impdes, imppag

// Entrada de datos
Leer turno, cantidad

// Calcula el importe compra
si (turno == 0)
    impcom = 37.5 * cantidad
sino
    impcom = 45.0 * cantidad

// Calcula el importe del descuento
si (cantidad >= 15)
    impdes = 0.08 * impcom
sino
    impdes = 0.05 * impcom

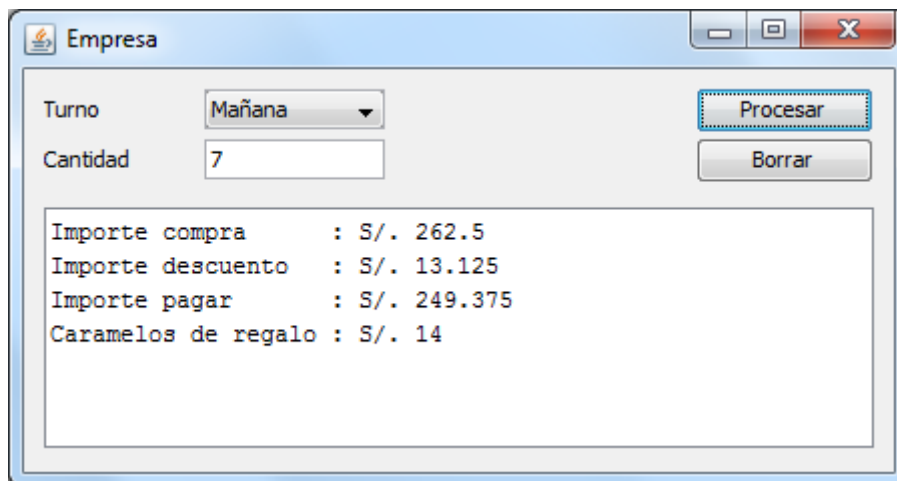
// Calcula el importe a pagar
imppag = impcom - impdes

// Calcula los caramelos de obsequio
si (imppag > 200)
    caramelos = 2 * cantidad
sino
    caramelos = 0

// Salida de resultados
Imprimir impdes, imppag, caramelos
```

Fin

Programa



```
package cibertec;

import java.awt.EventQueue;

public class Empresa extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblTurno;
    private JLabel lblCantidad;
    private JComboBox<String> cboTurno;
    private JTextField txtCantidad;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;

    // Lanza la aplicación
    public static void main(String[] args) {
        try {
            UIManager

.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        }
        catch (Throwable e) {
            e.printStackTrace();
        }
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Empresa frame = new Empresa();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    // Crea la GUI
```

```

public Empresa() {
    setTitle("Empresa");
    setBounds(100, 100, 450, 239);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    lblTurno = new JLabel("Turno");
    lblTurno.setBounds(10, 13, 80, 14);
    getContentPane().add(lblTurno);

    lblCantidad = new JLabel("Cantidad");
    lblCantidad.setBounds(10, 38, 80, 14);
    getContentPane().add(lblCantidad);

    cboTurno = new JComboBox<String>();
    cboTurno.setModel(new DefaultComboBoxModel<String>(new String[] {
        "Mañana", "Noche" }));
    cboTurno.setBounds(90, 10, 90, 20);
    getContentPane().add(cboTurno);

    txtCantidad = new JTextField();
    txtCantidad.setBounds(90, 35, 90, 20);
    getContentPane().add(txtCantidad);
    txtCantidad.setColumns(10);

    btnProcesar = new JButton("Procesar");
    btnProcesar.addActionListener(this);
    btnProcesar.setBounds(335, 9, 89, 23);
    getContentPane().add(btnProcesar);

    btnBorrar = new JButton("Borrar");
    btnBorrar.addActionListener(this);
    btnBorrar.setBounds(335, 34, 89, 23);
    getContentPane().add(btnBorrar);

    scpScroll = new JScrollPane();
    scpScroll.setBounds(10, 69, 414, 120);
    getContentPane().add(scpScroll);

    txtS = new JTextArea();
    txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
    scpScroll.setViewportView(txtS);
}

// Direcciona eventos de tipo ActionEvent
public void actionPerformed(ActionEvent arg0) {
    if (arg0.getSource() == btnProcesar) {
        actionPerformedBtnProcesar(arg0);
    }
    if (arg0.getSource() == btnBorrar) {
        actionPerformedBtnBorrar(arg0);
    }
}

// Procesa la pulsación del botón Procesar
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declaración de variables
    int turno, cantidad, caramelos;
    double impcom, impdes, imppag;

```

```

// Entrada de datos
turno = cboTurno.getSelectedIndex();
cantidad = Integer.parseInt(txtCantidad.getText());

// Calcula el importe de la compra
if (turno == 0)
    impcom = 37.5 * cantidad;
else
    impcom = 45.0 * cantidad;

// Calcula el importe del descuento
if (cantidad >= 15)
    impdes = 0.08 * impcom;
else
    impdes = 0.05 * impcom;

// Calcula el importe a pagar
imppag = impcom - impdes;

// Calcula los caramelos de obsequio
if (imppag > 200)
    caramelos = 2 * cantidad;
else
    caramelos = 0;

// Salida de resultados
txtS.setText("Importe compra      : S/. " + impcom + "\n");
txtS.append("Importe descuento  : S/. " + impdes + "\n");
txtS.append("Importe pagar       : S/. " + imppag + "\n");
txtS.append("Caramelos de regalo : S/. " + caramelos);
}

// Procesa la pulsación del botón Borrar
protected void actionPerformedBtnBorrar(ActionEvent arg0) {
    txtCantidad.setText("");
    txtS.setText("");
    txtCantidad.requestFocus();
}
}

```

Problema 3

El sueldo bruto de los empleados de una empresa se calcula sumando el sueldo básico más la bonificación por hijos.

El sueldo básico se calcula multiplicando las horas trabajadas por la tarifa horaria. La tarifa horaria depende de la categoría del empleado de acuerdo con la siguiente tabla:

Categoría	Tarifa horaria (S/.)
A	45.0
B	37.5

La bonificación por hijos se calcula de acuerdo con la siguiente tabla:

Número de hijos	Bonificación
Hasta 3	S/. 40.5 por cada hijo
Más de 3	S/. 35.0 por cada hijo

Por ley, todo empleado está sujeto a un porcentaje de descuento sobre el sueldo bruto de acuerdo con la siguiente tabla:

Sueldo bruto (S/.)	Descuento
≥ 3500	13.5%
< 3500	10.0%

Dadas la categoría y la cantidad de horas trabajadas de un empleado, diseñe un programa que determine el sueldo básico, el sueldo bruto, el descuento y el sueldo neto que le corresponden.

Algoritmo

Inicio

```
// Declaración de variables
entero hij, cat
real bonif, suelbas, suelbru, desc, suelnet, hor

// Entrada de datos
Leer hor, cat, hij

// Calcula el sueldo bruto
si (cat == 0 )
    suelbas = hor * 45.0
sino
    suelbas = hor * 37.5

// Calcula la bonificación por hijos
si (hij <= 3)
    bonif = 40.5 * hij
sino
    bonif = 35.0 * hij

// Calcula el sueldo bruto
suelbru = suelbas + bonif

// Calcula el descuento
si (suelbru >= 3500)
    desc = 0.135 * suelbru
sino
    desc = 0.10 * suelbru

// Calcula el sueldo neto
suelnet = suelbru - desc

// Salida de resultados
Imprimir suelbas, bonif, suelbru, desc, suelnet
```

Fin

Programa

```
package cibertec;

import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JComboBox;
import javax.swing.DefaultComboBoxModel;

public class Empleados extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblCategoría;
    private JLabel lblHoras;
    private JLabel lblHijos;
    private JComboBox<String> cboCategoría;
    private JTextField txtHoras;
    private JTextField txtHijos;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;

    // Lanza la aplicación
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(
                "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        }
        catch (Throwable e) {
            e.printStackTrace();
        }
    }
}
```

```

    }
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                Empleados frame = new Empleados();
                frame.setVisible(true);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

// Crea la GUI
public Empleados() {
    setTitle("Empleados");
    setBounds(100, 100, 450, 264);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    lblCategoria = new JLabel("Categoría");
    lblCategoria.setBounds(10, 13, 80, 14);
    getContentPane().add(lblCategoria);

    lblHoras = new JLabel("Horas");
    lblHoras.setBounds(10, 38, 80, 14);
    getContentPane().add(lblHoras);

    lblHijos = new JLabel("Hijos");
    lblHijos.setBounds(10, 63, 80, 14);
    getContentPane().add(lblHijos);

    cboCategoria = new JComboBox<String>();
    cboCategoria.setModel(new DefaultComboBoxModel<String>(new String[] {
        "A", "B" }));
    cboCategoria.setBounds(90, 10, 90, 20);
    getContentPane().add(cboCategoria);

    txtHoras = new JTextField();
    txtHoras.setBounds(90, 35, 90, 20);
    getContentPane().add(txtHoras);
    txtHoras.setColumns(10);

    txtHijos = new JTextField();
    txtHijos.setBounds(90, 60, 90, 20);
    getContentPane().add(txtHijos);
    txtHijos.setColumns(10);

    btnProcesar = new JButton("Procesar");
    btnProcesar.addActionListener(this);
    btnProcesar.setBounds(335, 9, 89, 23);
    getContentPane().add(btnProcesar);

    btnBorrar = new JButton("Borrar");
    btnBorrar.addActionListener(this);
    btnBorrar.setBounds(335, 34, 89, 23);
    getContentPane().add(btnBorrar);
}

```

```

        scpScroll = new JScrollPane();
        scpScroll.setBounds(10, 94, 414, 120);
        getContentPane().add(scpScroll);

        txtS = new JTextArea();
        txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
        scpScroll.setViewportView(txtS);
    }

    // Direcciona eventos de tipo ActionEvent
    public void actionPerformed(ActionEvent arg0) {
        if (arg0.getSource() == btnProcesar) {
            actionPerformedBtnProcesar(arg0);
        }
        if (arg0.getSource() == btnBorrar) {
            actionPerformedBtnBorrar(arg0);
        }
    }

    // Procesa la pulsación del botón Procesar
    protected void actionPerformedBtnProcesar(ActionEvent arg0) {
        // Declaración de variables
        int hij, cat;
        double bonif, suelbas, suelbru, desc, suelnet, hor;

        // Entrada de datos
        hor = Double.parseDouble(txtHoras.getText());
        cat = cboCategoria.getSelectedIndex();
        hij = Integer.parseInt(txtHijos.getText());

        // Calcula el sueldo básico
        if (cat == 0)
            suelbas = hor * 45.0;
        else
            suelbas = hor * 37.5;

        // Calcula la bonificación por hijos
        if (hij <= 3)
            bonif = 40.5 * hij;
        else
            bonif = 35.0 * hij;

        // Calcula el sueldo bruto
        suelbru = suelbas + bonif;

        // Calcula el descuento
        if (suelbru >= 3500)
            desc = 0.135 * suelbru;
        else
            desc = 0.10 * suelbru;

        // Calcula el sueldo neto
        suelnet = suelbru - desc;

        // Salida de resultados
        txtS.setText("Sueldo básico : S/. " + suelbas + "\n");
        txtS.append("Bonificación : S/. " + bonif + "\n");
        txtS.append("Sueldo bruto : S/. " + suelbru + "\n");
        txtS.append("Descuento : S/. " + desc + "\n");
    }

```

```

        txtS.append("Sueldo neto    : S/. " + suelnet);
    }

    // Procesa la pulsación del botón Borrar
    protected void actionPerformedBtnBorrar(ActionEvent arg0) {
        cboCategoria.setSelectedIndex(0);
        txtHoras.setText("");
        txtHijos.setText("");
        txtS.setText("");
        txtHoras.requestFocus();
    }
}

```

Problema 4

Los cálculos salariales de los vendedores de una empresa se calculan de la siguiente manera:

- Sueldo básico: S/.600
- Comisión: 7% del importe total vendido si es que el importe total vendido es mayor a S/.15000; en caso contrario, 5% del importe total vendido
- Bonificación: S/.25 por cada hijo si es que el número de hijos es menor a 5; en caso contrario, S/.22 por cada hijo
- Sueldo bruto: La suma del sueldo básico, más la comisión y más la bonificación
- Descuento: 15% del sueldo bruto si es que el sueldo bruto es mayor que S/.3500; en caso contrario, 11% del sueldo bruto
- Sueldo neto: La resta del sueldo bruto menos el descuento

Dado el importe total vendido y el número de hijos de un vendedor, diseñe un programa que determine el sueldo básico, la comisión, la bonificación, el sueldo bruto, el descuento y el sueldo neto.

Algoritmo

Inicio

```

// Declaración de variables
real impven, suelbas, comi, bonif, suelbru, desc, suelnet
entero hijos

// Entrada de datos
Leer impven, hijos

// Asigna el sueldo básico
suelbas = 600

// Calcula la comisión
si (impven > 15000)
    comi = 0.07 * impven
sino
    comi = 0.05 * impven

// Calcula la bonificación
si (hijos < 5)
    bonif= 25 * hijos
sino
    bonif= 22 * hijos

```



```

// Calcula el sueldo bruto
suelbru = suelbas + comi + bonif

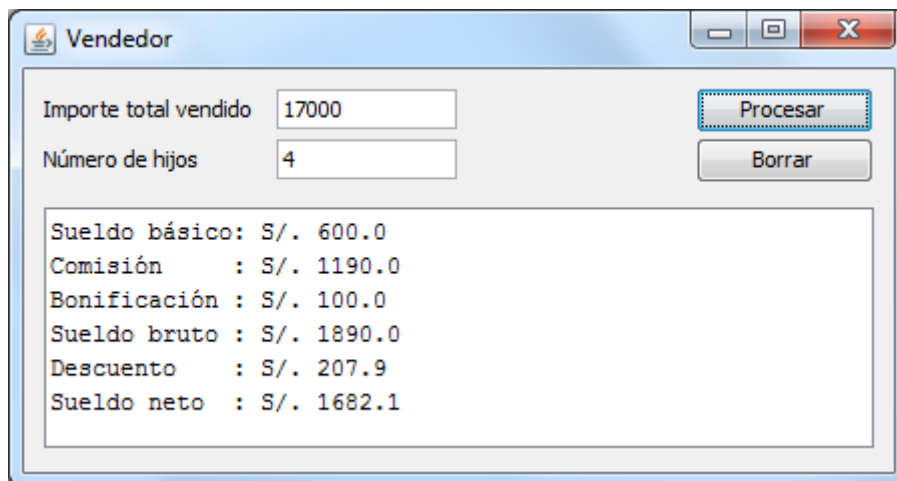
// Calcula el descuento
si (suelbru > 3500)
    desc = 0.15 * suelbru
sino
    desc = 0.11 * suelbru

// Calcula el sueldo neto
suelnet = suelbru - desc

// Salida de resultados
Imprimir suelbas, comi, bonif, suelbru, desc, suelnet
Fin

```

Programa



```

package cibertec;

import java.awt.EventQueue;

public class Vendedor extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblImporte;
    private JLabel lblHijos;
    private JTextField txtImporte;
    private JTextField txtHijos;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;

    // Lanza la aplicación
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(
                "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        }
        catch (Throwable e) {

```

```

        e.printStackTrace();
    }
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                Vendedor frame = new Vendedor();
                frame.setVisible(true);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

// Crea la GUI
public Vendedor() {
    setTitle("Vendedor");
    setBounds(100, 100, 450, 239);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    btnProcesar = new JButton("Procesar");
    btnProcesar.addActionListener(this);

    lblImporte = new JLabel("Importe total vendido");
    lblImporte.setBounds(10, 13, 115, 14);
    getContentPane().add(lblImporte);

    txtImporte = new JTextField();
    txtImporte.setBounds(126, 10, 90, 20);
    getContentPane().add(txtImporte);
    txtImporte.setColumns(10);
    btnProcesar.setBounds(335, 9, 89, 23);
    getContentPane().add(btnProcesar);

    btnBorrar = new JButton("Borrar");
    btnBorrar.addActionListener(this);

    lblHijos = new JLabel("Número de hijos");
    lblHijos.setBounds(10, 38, 115, 14);
    getContentPane().add(lblHijos);

    txtHijos = new JTextField();
    txtHijos.setBounds(126, 35, 90, 20);
    getContentPane().add(txtHijos);
    txtHijos.setColumns(10);
    btnBorrar.setBounds(335, 34, 89, 23);
    getContentPane().add(btnBorrar);

    scpScroll = new JScrollPane();
    scpScroll.setBounds(10, 69, 414, 120);
    getContentPane().add(scpScroll);

    txtS = new JTextArea();
    txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
    scpScroll.setViewportView(txtS);
}

```

```

// Direcciona eventos de tipo ActionEvent
public void actionPerformed(ActionEvent arg0) {
    if (arg0.getSource() == btnProcesar) {
        actionPerformedBtnProcesar(arg0);
    }
    if (arg0.getSource() == btnBorrar) {
        actionPerformedBtnBorrar(arg0);
    }
}

// Procesa la pulsación del botón Procesar
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declaración de variables
    double impven, suelbas, comi, bonif, suelbru, desc, suelnet;
    int hijos;

    // Entrada de datos
    impven = Double.parseDouble(txtImporte.getText());
    hijos = Integer.parseInt(txtHijos.getText());

    // Asignación del sueldo básico
    suelbas = 600;

    // Calcula la comisión
    if (impven > 15000)
        comi = 0.07 * impven;
    else
        comi = 0.05 * impven;

    // Calcula la bonificación
    if (hijos < 5)
        bonif = 25 * hijos;
    else
        bonif = 22 * hijos;

    // Calcula el sueldo bruto
    suelbru = suelbas + comi + bonif;

    // Calcula el descuento
    if (suelbru > 3500)
        desc = 0.15 * suelbru;
    else
        desc = 0.11 * suelbru;

    // Calculo el sueldo neto
    suelnet = suelbru - desc;

    // Salida de resultados
    txtS.setText("Sueldo básico: S/. " + suelbas + "\n");
    txtS.append("Comisión : S/. " + comi + "\n");
    txtS.append("Bonificación : S/. " + bonif + "\n");
    txtS.append("Sueldo bruto : S/. " + suelbru + "\n");
    txtS.append("Descuento : S/. " + desc + "\n");
    txtS.append("Sueldo neto : S/. " + suelnet);
}

// Procesa la pulsación del botón Borrar
protected void actionPerformedBtnBorrar(ActionEvent arg0) {
    txtImporte.setText("");
}

```

```

        txtHijos.setText("");
        txtS.setText("");
        txtImporte.requestFocus();
    }
}

```

Problema 5

Una institución benéfica recibe anualmente una donación y lo reparte entre un centro de salud, un comedor de niños y una parte lo invierte en la bolsa de acuerdo con lo siguiente:

- Si el monto de la donación es de \$10000 o más: 30% se destina al centro de salud, 50% al comedor de niños y el resto se invierte en la bolsa.
- Si el monto de la donación es menor de \$10000: 25% se destina al centro de salud, 60% al comedor de niños y el resto se invierte en la bolsa.

Dado el monto de la donación, diseñe un programa que determine el monto de dinero que recibirá cada rubro.

Algoritmo

Inicio

```

// Declaración de variables
real donacion, comedor, salud, bolsa

```

```

// Entrada de datos
Leer donacion

```

```

// Reparte la donación
si( donacion >= 10000 ){
    comedor = 0.30*donacion
    salud   = 0.50*donacion
    bolsa   = 0.20*donacion
}
sino{
    comedor = 0.25*donacion
    salud   = 0.60*donacion
    bolsa   = 0.15*donacion
}

```

```

// Salida de resultados
Imprimir comedor, salud, bolsa

```

Fin

Programa

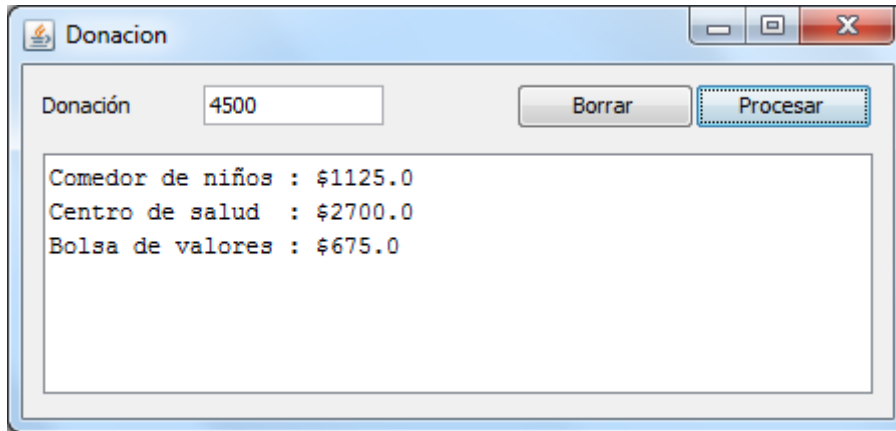
```

package cibertec;

import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;

```

```
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
```



```
public class Donacion extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblDonacion;
    private JTextField txtDonacion;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;

    // Lanza la aplicación
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(
                "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        }
        catch (Throwable e) {
            e.printStackTrace();
        }
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Donacion frame = new Donacion();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    // Crea la GUI
    public Donacion() {
        setTitle("Donacion");
        setBounds(100, 100, 450, 214);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().setLayout(null);
    }
}
```

```

lblDonacion = new JLabel("Donación");
lblDonacion.setBounds(10, 13, 80, 14);
getContentPane().add(lblDonacion);

txtDonacion = new JTextField();
txtDonacion.setBounds(90, 10, 90, 20);
getContentPane().add(txtDonacion);
txtDonacion.setColumns(10);

btnProcesar = new JButton("Procesar");
btnProcesar.addActionListener(this);
btnProcesar.setBounds(335, 9, 89, 23);
getContentPane().add(btnProcesar);

btnBorrar = new JButton("Borrar");
btnBorrar.addActionListener(this);
btnBorrar.setBounds(246, 9, 89, 23);
getContentPane().add(btnBorrar);

scpScroll = new JScrollPane();
scpScroll.setBounds(10, 44, 414, 120);
getContentPane().add(scpScroll);

txtS = new JTextArea();
txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
scpScroll.setViewportView(txtS);
}

// Direcciona eventos de tipo ActionEvent
public void actionPerformed(ActionEvent arg0) {
    if (arg0.getSource() == btnProcesar) {
        actionPerformedBtnProcesar(arg0);
    }
    if (arg0.getSource() == btnBorrar) {
        actionPerformedBtnBorrar(arg0);
    }
}

// Procesa la pulsación del botón Procesar
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declaración de variables
    double donacion, comedor, salud, bolsa;

    // Entrada de datos
    donacion = Double.parseDouble(txtDonacion.getText());

    // Reparte la donación
    if (donacion >= 10000) {
        comedor = 0.30 * donacion;
        salud = 0.50 * donacion;
        bolsa = 0.20 * donacion;
    }
    else {
        comedor = 0.25 * donacion;
        salud = 0.60 * donacion;
        bolsa = 0.15 * donacion;
    }

    // Salida de resultados

```

```

        txtS.setText("Comedor de niños : $" + comedor + "\n");
        txtS.append ("Centro de salud : $" + salud + "\n");
        txtS.append ("Bolsa de valores : $" + bolsa);
    }

    // Procesa la pulsación del botón Borrar
    protected void actionPerformedBtnBorrar(ActionEvent arg0) {
        txtDonacion.setText("");
        txtS.setText("");
        txtDonacion.requestFocus();
    }
}

```

Comentario

Note el uso de las llaves { } en el bloque **if** y en el bloque **else**, dado que cada bloque tiene más de una instrucción. En caso que el **if** o el **else** tuviera una sola instrucción, el uso de llaves es opcional.

Problema 6

En una oficina de empleos categorizan a los postulantes en función del sexo y de la edad de acuerdo con lo siguiente:

- Si la persona es de sexo femenino: categoría FA si tiene menos de 23 años y FB, en caso contrario.
- Si la persona es de sexo masculino: categoría MA si tiene menos de 25 años y MB, en caso contrario.

Dado el sexo y la edad de un postulante, diseñe un programa que determine su categoría.

Algoritmo

Inicio

```

// Declaración de variables
entero sexo, edad
cadena categoria

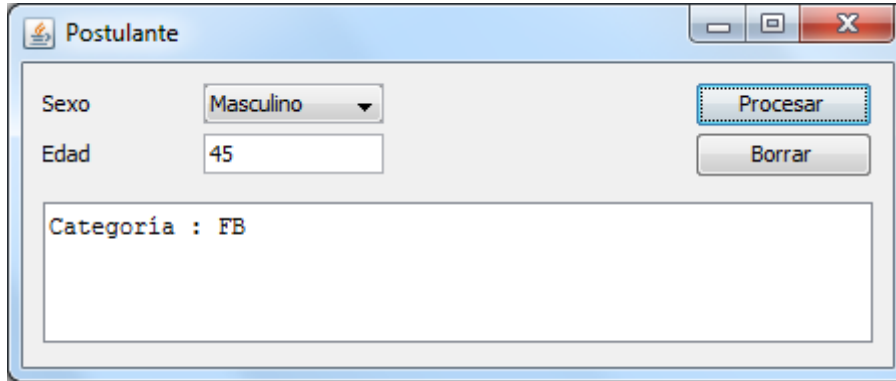
// Entrada de datos
Leer sexo, edad

// Determina la categoría
si( sexo == 0 ){
    si( edad < 23 )
        categoria = "FA"
    sino
        categoria = "FB"
}
sino{
    si( edad < 25 )
        categoria = "MA"
    sino
        categoria = "MB"
}

```

```
// Salida de resultados
Imprimir categoria
Fin
```

Programa



```
package cibertec;

import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JComboBox;
import javax.swing.DefaultComboBoxModel;

public class Postulante extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblSexo;
    private JLabel lblEdad;
    private JComboBox<String> cboSexo;
    private JTextField txtEdad;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;

    // Lanza la aplicación
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(
                "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        }
        catch (Throwable e) {
            e.printStackTrace();
        }
        EventQueue.invokeLater(new Runnable() {
```



```

        public void run() {
            try {
                Postulante frame = new Postulante();
                frame.setVisible(true);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

// Crea la GUI
public Postulante() {
    setTitle("Postulante");
    setBounds(100, 100, 450, 239);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    lblSexo = new JLabel("Sexo");
    lblSexo.setBounds(10, 13, 80, 14);
    getContentPane().add(lblSexo);

    lblEdad = new JLabel("Edad");
    lblEdad.setBounds(10, 38, 80, 14);
    getContentPane().add(lblEdad);

    cboSexo = new JComboBox<String>();
    cboSexo.setModel(new DefaultComboBoxModel<String>(new String[] {
        "Masculino", "Femenino" }));
    cboSexo.setBounds(90, 10, 90, 20);
    getContentPane().add(cboSexo);

    txtEdad = new JTextField();
    txtEdad.setBounds(90, 35, 90, 20);
    getContentPane().add(txtEdad);
    txtEdad.setColumns(10);

    btnProcesar = new JButton("Procesar");
    btnProcesar.addActionListener(this);
    btnProcesar.setBounds(335, 9, 89, 23);
    getContentPane().add(btnProcesar);

    btnBorrar = new JButton("Borrar");
    btnBorrar.addActionListener(this);
    btnBorrar.setBounds(335, 34, 89, 23);
    getContentPane().add(btnBorrar);

    scpScroll = new JScrollPane();
    scpScroll.setBounds(10, 69, 414, 120);
    getContentPane().add(scpScroll);

    txtS = new JTextArea();
    txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
    scpScroll.setViewportView(txtS);
}

// Direcciona eventos de tipo ActionEvent
public void actionPerformed(ActionEvent arg0) {

```

```

        if (arg0.getSource() == btnProcesar) {
            actionPerformedBtnProcesar(arg0);
        }
        if (arg0.getSource() == btnBorrar) {
            actionPerformedBtnBorrar(arg0);
        }
    }

    // Procesa la pulsación del botón Procesar
    protected void actionPerformedBtnProcesar(ActionEvent arg0) {
        // Declaración de variables
        int sexo, edad;
        String categoria;

        // Entrada de datos
        sexo = cboSexo.getSelectedIndex();
        edad = Integer.parseInt(txtEdad.getText());

        // Determina la categoría
        if (sexo == 0) {
            if (edad < 23)
                categoria = "FA";
            else
                categoria = "FB";
        }
        else {
            if (edad < 25)
                categoria = "MA";
            else
                categoria = "MB";
        }

        // Salida de resultados
        txtS.setText("Categoría : " + categoria);
    }

    // Procesa la pulsación del botón Borrar
    protected void actionPerformedBtnBorrar(ActionEvent arg0) {
        txtEdad.setText("");
        txtS.setText("");
        txtEdad.requestFocus();
    }
}

```

Problema 7

Una papelería ha puesto en oferta la venta de papel bond en paquetes de medio millar de acuerdo con los siguientes criterios:

- Para compras menores o iguales a 20 paquetes, se paga el precio normal.
- Para compras mayores de 20 paquetes, por los primeros 20 paquetes se paga el precio normal; pero, por los paquetes que exceden de 20, sólo se paga el 85% del precio normal.

Adicionalmente, para compras de más de 50 paquetes, el cliente recibe dos paquetes adicionales.

Dado el precio normal del paquete y la cantidad de paquetes adquiridos, diseñe un programa que determine el importe a pagar y la cantidad total de paquetes que recibirá el cliente

Algoritmo

Inicio

```
// Declaración de variables
real imppag, precio
entero cantidad, cantot

// Entrada de datos
Leer precio, cantidad

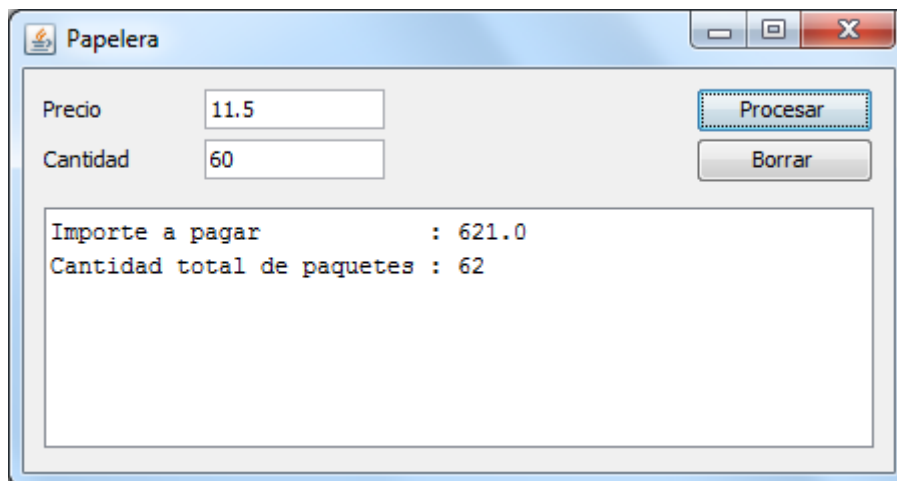
// Determina el importe a pagar
si(cantidad <= 20)
    imppag = precio*cantidad
sino
    imppag = precio*20 + 0.85*precio*(cantidad-20)

// Determina la cantidad total de paquetes
si(cantidad > 50)
    cantot = cantidad + 2
sino
    cantot = cantidad

// Salida de resultados
Imprimir imppag, cantot
```

Fin

Programa



```
package cibertec;

import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.UIManager;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
```

```
import java.awt.event.ActionEvent;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

public class Papelera extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblPrecio;
    private JLabel lblCantidad;
    private JTextField txtPrecio;
    private JTextField txtCantidad;
    private JButton btnProcesar;
    private JButton btnBorrar;
    private JScrollPane scpScroll;
    private JTextArea txtS;

    // Lanza la aplicación
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel
                ("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        }
        catch (Throwable e) {
            e.printStackTrace();
        }
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Papelera frame = new Papelera();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    // Crea la GUI
    public Papelera() {
        setTitle("Papelera");
        setBounds(100, 100, 450, 239);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().setLayout(null);

        lblPrecio = new JLabel("Precio");
        lblPrecio.setBounds(10, 13, 80, 14);
        getContentPane().add(lblPrecio);

        lblCantidad = new JLabel("Cantidad");
        lblCantidad.setBounds(10, 38, 80, 14);
        getContentPane().add(lblCantidad);

        txtPrecio = new JTextField();
        txtPrecio.setBounds(90, 10, 90, 20);
        getContentPane().add(txtPrecio);
        txtPrecio.setColumns(10);
```

```

txtCantidad = new JTextField();
txtCantidad.setBounds(90, 35, 90, 20);
getContentPane().add(txtCantidad);
txtCantidad.setColumns(10);

btnProcesar = new JButton("Procesar");
btnProcesar.addActionListener(this);
btnProcesar.setBounds(335, 9, 89, 23);
getContentPane().add(btnProcesar);

btnBorrar = new JButton("Borrar");
btnBorrar.addActionListener(this);
btnBorrar.setBounds(335, 34, 89, 23);
getContentPane().add(btnBorrar);

scpScroll = new JScrollPane();
scpScroll.setBounds(10, 69, 414, 120);
getContentPane().add(scpScroll);

txtS = new JTextArea();
txtS.setFont(new Font("Monospaced", Font.PLAIN, 12));
scpScroll.setViewportView(txtS);
}

// Direcciona eventos de tipo ActionEvent
public void actionPerformed(ActionEvent arg0) {
    if (arg0.getSource() == btnProcesar) {
        actionPerformedBtnProcesar(arg0);
    }
    if (arg0.getSource() == btnBorrar) {
        actionPerformedBtnBorrar(arg0);
    }
}

// Procesa la pulsación del botón Procesar
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declaración de variables
    double imppag, precio;
    int cantidad, cantot;

    // Entrada de datos
    precio = Double.parseDouble(txtPrecio.getText());
    cantidad = Integer.parseInt(txtCantidad.getText());

    // Determina el importe a pagar
    if (cantidad <= 20)
        imppag = precio*cantidad;
    else
        imppag = precio*20 + 0.85*precio*(cantidad-20);

    // Determina la cantidad total de paquetes
    if(cantidad > 50)
        cantot = cantidad + 2;
    else
        cantot = cantidad;

    // Salida de resultados
    txtS.setText("Importe a pagar          : " + imppag + "\n");
    txtS.append("Cantidad total de paquetes : " + cantot);
}

```

```
    }  
  
    // Procesa la pulsación del botón Borrar  
    protected void actionPerformedBtnBorrar(ActionEvent arg0) {  
        txtPrecio.setText("");  
        txtCantidad.setText("");  
        txtS.setText("");  
        txtPrecio.requestFocus();  
    }  
}
```