

Estructura de selección simple if

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 simple if

ACTIVIDADES

Los alumnos desarrollan algoritmos que involucran la estructuras de selección if.

1. Estructura de selección simple if

La *estructura de selección simple if* evalúa una *condición lógica* y, en caso resulte verdadera, efectúa la acción **acciona**. En caso que la condición resulte falsa, continúa con la siguiente instrucción del programa. La acción **acciona** puede ser una **acción simple** (una sola acción) o una **acción compuesta** (bloque de acciones).

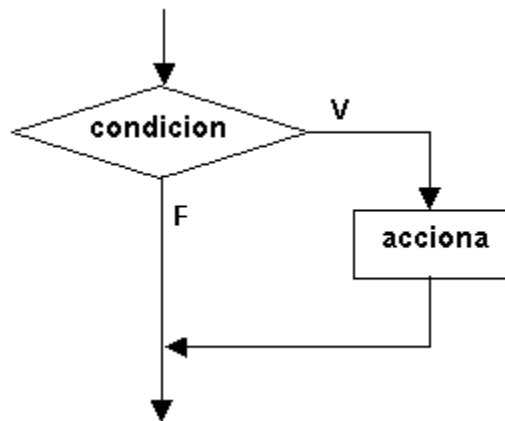


Figura 1 Diagrama de flujo de la estructura de selección simple if

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
<code>if(condicion) accionA;</code>	<code>si(condicion) accionA</code>
<code>if(condicion){ accionA1; accionA2; . . . accionAn; }</code>	<code>si(condicion){ accionA1 accionA2 . . . accionAn }</code>

2. Problemas propuestos

Problema 1

Una tienda vende un producto a precios unitarios que dependen de la cantidad de unidades adquiridas de acuerdo con la siguiente tabla:

Unidades adquiridas	Precio unitario
1 a 25	S/. 27.7
26 a 50	S/. 25.5
51 a 75	S/. 23.5
76 en adelante	S/. 21.5

Adicionalmente, si el cliente adquiere más de 50 unidades la tienda le descuenta el 15% del importe de la compra; en caso contrario, sólo le descuenta el 5% del importe compra.

Diseñe un programa que determine el importe de la compra, el importe del descuento y el importe a pagar por la compra de cierta cantidad de unidades del producto.

Algoritmo

Inicio

```
// Declaración de variables
entero unidades;
real impcom = 0, impdes, imppag

// Entrada de datos
Leer unidades

// Calcula el importe de la compra
si(unidades >= 1 && unidades <= 25)
    impcom = unidades*27.5
si(unidades >= 26 && unidades <= 50)
    impcom = unidades*25.5
si(unidades >= 51 && unidades <= 75)
    impcom = unidades*27.5
si(unidades >= 76)
    impcom = unidades*27.5

// Calcula el importe del descuento
si(unidades > 50)
    impdes = 0.15*impcom
sino
    impdes = 0.05*impcom

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

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

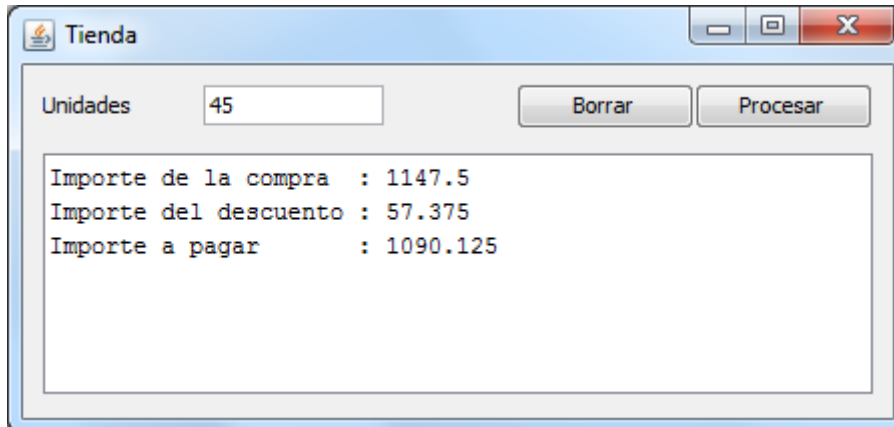
Fin

Programa

```
package cibertec;

import java.awt.EventQueue;
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 java.awt.Font;
```



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

    // Declaración de variables
    private JLabel lblUnidades;
    private JTextField txtUnidades;
    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, 214);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().setLayout(null);
    }
}
```

```

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

txtUnidades = new JTextField();
txtUnidades.setBounds(90, 10, 90, 20);
getContentPane().add(txtUnidades);
txtUnidades.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
    int unidades;
    double impcom = 0, impdes, imppag;

    // Entrada de datos
    unidades = Integer.parseInt(txtUnidades.getText());

    // Calcula el importe de la compra
    if (unidades >= 1 && unidades <= 25)
        impcom = unidades * 27.5;
    if (unidades >= 26 && unidades <= 50)
        impcom = unidades * 25.5;
    if (unidades >= 51 && unidades <= 75)
        impcom = unidades * 27.5;
    if (unidades >= 76)
        impcom = unidades * 27.5;

    // Calcula el importe del descuento
    if (unidades > 50)

```

```

        impdes = 0.15 * impcom;
    else
        impdes = 0.05 * impcom;

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

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

}

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

```

Problema 2

Una tienda ha decidido incentivar a sus clientes con un obsequio. Para ello, el cliente debe extraer un bolo de una urna que contiene 100 bolos numerados del 1 al 100. Luego, sobre la base del número del bolo se obtiene el obsequio de acuerdo con la siguiente tabla:

Número del bolo	Obsequio
1 a 20	Un lapicero
21 a 40	Un cuaderno de 100 hojas
41 a 60	Una caja de plumones
61 a 80	Un cuaderno espiral
81 a 99	Una agenda
100	Una mochila

Dado el número del bolo obtenido por un cliente, diseñe un programa que determine qué obsequio le corresponde. En caso que el número ingresado sea incorrecto, como obsequio mostrar "Ninguno".

Algoritmo

Inicio

```

// Declaración de variables
entero numero
cadena obsequio = "Ninguno"

// Entrada de datos
Leer numero

// Determina el obsequio
si(numero >= 1 && numero <= 20)
    obsequio = "Un lapicero"
si(numero >= 21 && numero <= 40)
    obsequio = "Un cuaderno de 100 hojas"

```

```

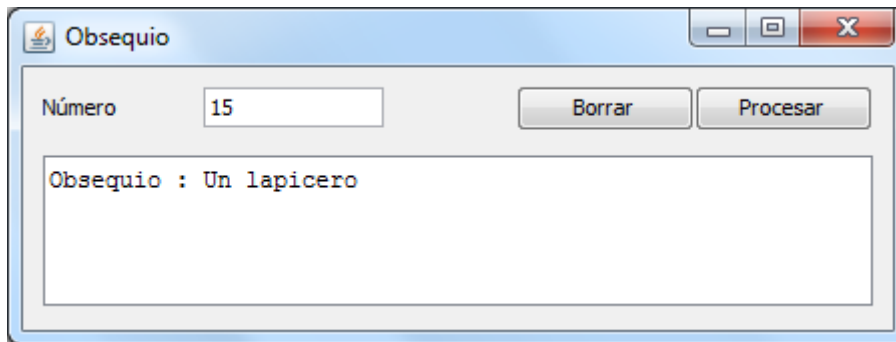
si(numero >= 41 && numero <= 60)
    obsequio = "Una caja de 12 plumones"
si(numero >= 61 && numero <= 80)
    obsequio = "Un cuaderno espiral"
si(numero >= 81 && numero <= 99)
    obsequio = "Una agenda"
si(numero == 100)
    obsequio = "Una mochila"

// Salida de resultados
Imprimir obsequio

```

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 Obsequio extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblNumero;
    private JTextField txtNumero;
    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 {
                    Obsequio frame = new Obsequio();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    // Crea la GUI
    public Obsequio() {
        setTitle("Obsequio");
        setBounds(100, 100, 450, 169);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().setLayout(null);

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

        txtNumero = new JTextField();
        txtNumero.setBounds(90, 10, 90, 20);
        getContentPane().add(txtNumero);
        txtNumero.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, 75);
        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 numero;
        String obsequio = "Ninguno";

        // Entrada de datos
        numero = Integer.parseInt(txtNumero.getText());

        // Determina el obsequio
        if (numero >= 1 && numero <= 20)
            obsequio = "Un lapicero";
        if (numero >= 21 && numero <= 40)
            obsequio = "Un cuaderno de 100 hojas";
        if (numero >= 41 && numero <= 60)
            obsequio = "Una caja de 12 plumones";
        if (numero >= 61 && numero <= 80)
            obsequio = "Un cuaderno espiral";
        if (numero >= 81 && numero <= 99)
            obsequio = "Una agenda";
        if (numero == 100)
            obsequio = "Una mochila";

        // Salida de resultados
        txtS.setText("Obsequio : " + obsequio);
    }

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

```

Problema 3

Una dulcería vende chocolates a los precios dados en la siguiente tabla:

Tipo de chocolate	Precio unitario
Primor	S/. 8.5
Dulzura	S/. 10.0
Tentación	S/. 7.0
Explosión	S/. 12.5

Como oferta, la tienda aplica un porcentaje de descuento sobre el importe de la compra, sobre la base de la cantidad de chocolates adquiridos, de acuerdo con la siguiente tabla:

Cantidad de chocolates	Descuento
< 5	4.0%
≥ 5 y < 10	6.5%
≥ 10 y < 15	9.0%
≥ 15	11.5%

Adicionalmente, si el importe a pagar es no menor de S/. 250, la tienda obsequia 3 caramelos por cada chocolate; en caso contrario, obsequia 2 caramelos por cada chocolates.

Dado el tipo de chocolate y la cantidad de unidades adquiridas, 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 tipo, cantidad, caramelos
real impcom = 0, impdes = 0, imppag

// Entrada de datos
Leer tipo, cantidad

// Calcula el importe de la compra
si(tipo == 0)
    impcom = 8.5*cantidad
si(tipo == 1)
    impcom = 10.0*cantidad
si(tipo == 2)
    impcom = 7.0*cantidad
si(tipo == 3)
    impcom = 12.5*cantidad

// Calcula el importe del descuento
si(cantidad < 5)
    impdes = 0.04*impcom
si(cantidad >= 5 && cantidad < 10)
    impdes = 0.065*impcom
si(cantidad >= 10 && cantidad < 15)
    impdes = 0.09*impcom
si(cantidad >= 15)
    impdes = 0.115*impcom

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

// Calcula la cantidad de caramelos de regalo
si(imppag < 250)
    caramelos = 2*cantidad
sino
    caramelos = 3*cantidad

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

Fin

Programa

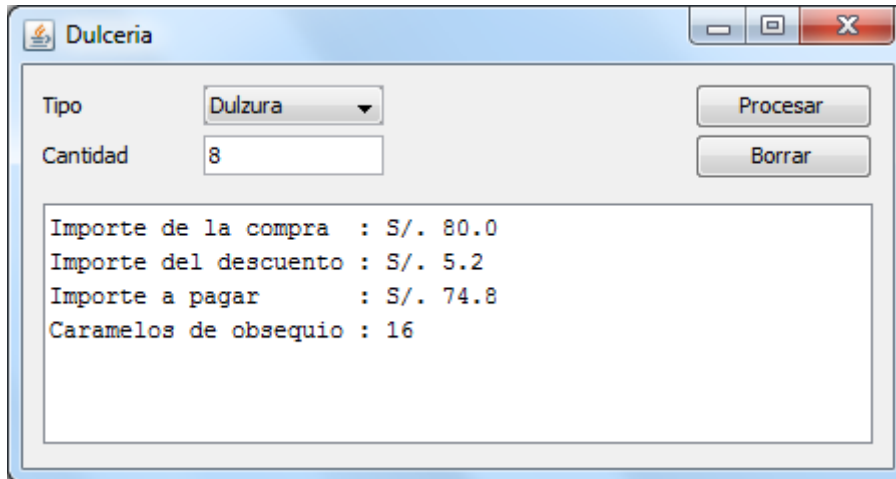
```
package cibertec;

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 Dulceria extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblTipo;
    private JLabel lblCantidad;
    private JComboBox<String> cboTipo;
    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 {
                    Dulceria frame = new Dulceria();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}

```

```

    }
    });
}

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

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

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

    cboTipo = new JComboBox<String>();
    cboTipo.setModel(new DefaultComboBoxModel<String>(new String[] {
        "Primor", "Dulzura", "Tentación", "Explosión" }));
    cboTipo.setBounds(90, 10, 90, 20);
    getContentPane().add(cboTipo);

    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 tipo, cantidad, caramelos;
    double impcom = 0, impdes = 0, imppag;

    // Entrada de datos
    tipo = cboTipo.getSelectedIndex();
    cantidad = Integer.parseInt(txtCantidad.getText());

    // Calcula el importe de la compra
    if (tipo == 0)
        impcom = 8.5 * cantidad;
    if (tipo == 1)
        impcom = 10.0 * cantidad;
    if (tipo == 2)
        impcom = 7.0 * cantidad;
    if (tipo == 3)
        impcom = 12.5 * cantidad;

    // Calcula el importe del descuento
    if (cantidad < 5)
        impdes = 0.04 * impcom;
    if (cantidad >= 5 && cantidad < 10)
        impdes = 0.065 * impcom;
    if (cantidad >= 10 && cantidad < 15)
        impdes = 0.09 * impcom;
    if (cantidad >= 15)
        impdes = 0.115 * impcom;

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

    // Calcula la cantidad de caramelos de regalo
    if (imppag < 250)
        caramelos = 2 * cantidad;
    else
        caramelos = 3 * cantidad;

    // 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("Caramelos de obsequio : " + caramelos);
}

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

```

Problema 4

Los cálculos salariales de los empleados de una empresa se efectúan de la siguiente manera:

- Sueldo bruto : horas trabajadas x tarifa horaria
- Descuento : 15% del sueldo bruto
- Sueldo neto : sueldo bruto – descuento

La tarifa horaria depende de la categoría del trabajador de acuerdo con la siguiente tabla:

Categoría	Tarifa horaria (S/.)
C1	45.0
C2	37.5
C3	35.0
C4	32.5

Dadas las horas trabajadas y la categoría de un empleado, diseñe un programa que determine la tarifa horaria, el sueldo bruto, el descuento y el sueldo neto del empleado.

Algoritmo**Inicio**

```
// Declaración de variables
entero categoria
real horas, suelbru, desc, suelnet, tarifa = 0

// Entrada de datos
Leer categoria, horas

// Determina la tarifa horaria
si(categoria == 0)
    tarifa = 45.0
si(categoria == 1)
    tarifa = 37.5
si(categoria == 2)
    tarifa = 35.0
si(categoria == 3)
    tarifa = 32.5

// Calcula el sueldo bruto
suelbru = horas*tarifa

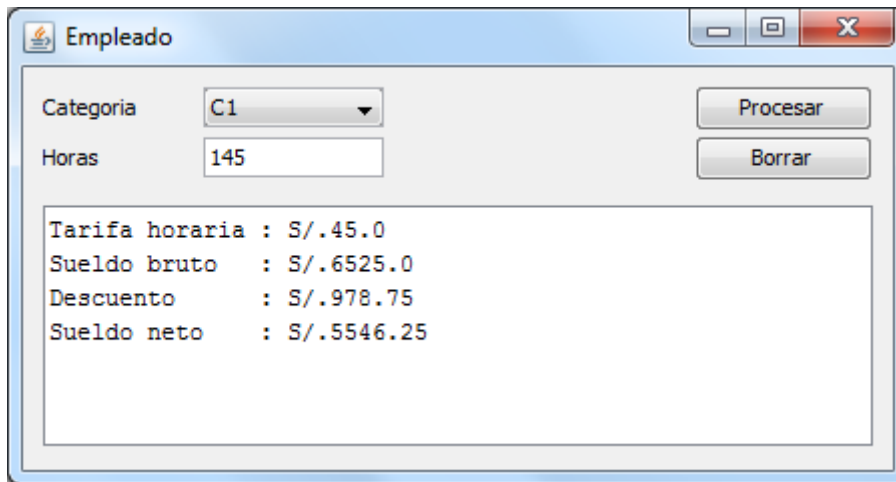
// Calcula el descuento
desc = 0.15*suelbru

// Calcula el sueldo neto
suelnet = suelbru - desc

// Salida de resultados
Imprimir tarifa, 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 Empleado extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

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

    // 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 {
```

```

        Empleado frame = new Empleado();
        frame.setVisible(true);
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}

});
}

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

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

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

    cboCategoría = new JComboBox<String>();
    cboCategoría.setModel(new DefaultComboBoxModel<String>(new String[] {
        "C1", "C2", "C3", "C4" }));
    cboCategoría.setBounds(90, 10, 90, 20);
    getContentPane().add(cboCategoría);

    txtHoras = new JTextField();
    txtHoras.setBounds(90, 35, 90, 20);
    getContentPane().add(txtHoras);
    txtHoras.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 categoria;
    double horas, suelbru, desc, suelnet, tarifa = 0;

    // Entrada de datos
    categoria = cboCategoria.getSelectedIndex();
    horas = Double.parseDouble(txtHoras.getText());

    // Determina la tarifa horaria
    if (categoria == 0)
        tarifa = 45.0;
    if (categoria == 1)
        tarifa = 37.5;
    if (categoria == 2)
        tarifa = 35.0;
    if (categoria == 3)
        tarifa = 32.5;

    // Calcula el sueldo bruto
    suelbru = horas * tarifa;

    // Calcula el descuento
    desc = 0.15 * suelbru;

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

    // Salida de resultados
    txtS.setText("Tarifa horaria : S/." + tarifa + "\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("");
    txtS.setText("");
    txtHoras.requestFocus();
}
}

```

Problema 5

Los ángulos se clasifican de la siguiente manera:

Magnitud	Clasificación
$\beta = 0^\circ$	Nulo
$0^\circ < \beta < 90^\circ$	Agudo
$\beta = 90^\circ$	Recto
$90^\circ < \beta < 180^\circ$	Obtuso
$\beta = 180^\circ$	Llano
$180^\circ < \beta < 360^\circ$	Cóncavo
$\beta = 360^\circ$	Completo

Diseñe un algoritmo que determine la clasificación de un ángulo dado en grados, minutos y segundos. Asuma que el ángulo está en el intervalo de 0° a 360° .

Algoritmo

Inicio

```
// Declaración de variables
entero grados, minutos, segundos
real beta
cadena tipo = ""

// Entrada de datos
Leer grados, minutos, segundos

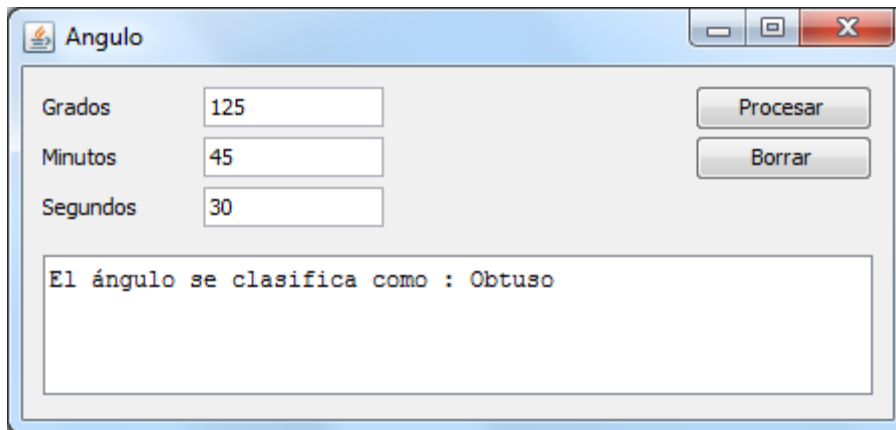
// Determina el ángulo en grados
beta = grados + minutos/60.0 + segundos/3600.0

// Determina el tipo de ángulo
si( beta == 0 )
    tipo = "Nulo"
si( beta > 0 && beta < 90 )
    tipo = "Agudo"
si( beta == 90 )
    tipo = "Recto"
si( beta > 90 && beta < 180 )
    tipo = "Obtuso"
si( beta == 180 )
    tipo = "Llano"
si( beta > 180 && beta < 360 )
    tipo = "Cóncavo"
si( beta == 360 )
    tipo = "Completo"

// Salida de resultados
Imprimir tipo
```

Fin

Programa



```
package cibertec;

import java.awt.EventQueue;
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 java.awt.Font;

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

    // Declaración de variables
    private JLabel lblGrados;
    private JLabel lblMinutos;
    private JLabel lblSegundos;
    private JTextField txtGrados;
    private JTextField txtMinutos;
    private JTextField txtSegundos;
    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 {
                    Angulo frame = new Angulo();

```

```

        frame.setVisible(true);
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}
});
}

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

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

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

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

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

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

    txtSegundos = new JTextField();
    txtSegundos.setBounds(90, 60, 90, 20);
    getContentPane().add(txtSegundos);
    txtSegundos.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, 70);
    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 grados, minutos, segundos;
        double beta;
        String tipo = "";

        // Entrada de datos
        grados = Integer.parseInt(txtGrados.getText());
        minutos = Integer.parseInt(txtMinutos.getText());
        segundos = Integer.parseInt(txtSegundos.getText());

        // Determina el ángulo en grados
        beta = grados + minutos / 60.0 + segundos / 3600.0;

        // Determina el tipo de ángulo
        if (beta == 0)
            tipo = "Nulo";
        if (beta > 0 && beta < 90)
            tipo = "Agudo";
        if (beta == 90)
            tipo = "Recto";
        if (beta > 90 && beta < 180)
            tipo = "Obtuso";
        if (beta == 180)
            tipo = "Llano";
        if (beta > 180 && beta < 360)
            tipo = "Cóncavo";
        if (beta == 360)
            tipo = "Completo";

        // Salida de resultados
        txtS.setText("El ángulo se clasifica como : " + tipo);
    }

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

```

Problema 6

El promedio final de un curso se obtiene sobre la base del promedio simple de tres prácticas calificadas. Para ayudar a los alumnos, el profesor del curso ha decidido incrementar en dos puntos las notas de las prácticas no menores que 10. Dadas las tres notas de práctica de un estudiante, diseñe un programa que determine el promedio final que le corresponde. Considere que la nota máxima es 20.

Algoritmo

Inicio

```
// Declaración de variables
real p1, p2, p3, promedio

// Entrada de datos
Leer p1, p2, p3

// Si amerita, añade 2 puntos a p1
si( p1 >= 10 ){
    p1 = p1 + 2
    si( p1 > 20 )
        p1 = 20
}

// Si amerita, añade 2 puntos a p2
si( p2 >= 10 ){
    p2 = p2 + 2
    si( p2 > 20 )
        p2 = 20
}

// Si amerita, añade 2 puntos a p3
si( p3 >= 10 ){
    p3 = p3 + 2
    si( p3 > 20 )
        p3 = 20
}

// Cálculo del promedio
promedio = (p1+p2+p3)/3

// Salida de resultados
Imprimir promedio
```

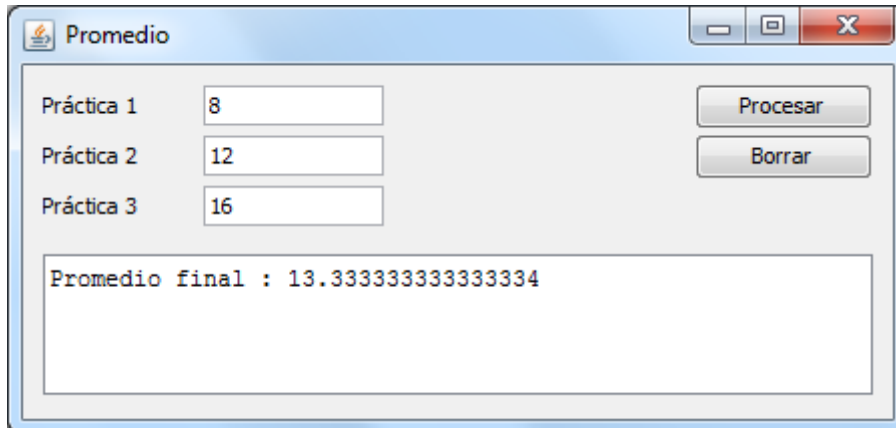
Fin

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 Promedio extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblPractica1;
    private JLabel lblPractica2;
    private JLabel lblPractica3;
    private JTextField txtPractica1;
    private JTextField txtPractica2;
    private JTextField txtPractica3;
    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 {
                    Promedio frame = new Promedio();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

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

```

getContentPane().setLayout(null);

lblPractica1 = new JLabel("Práctica 1");
lblPractica1.setBounds(10, 13, 80, 14);
getContentPane().add(lblPractica1);

lblPractica2 = new JLabel("Práctica 2");
lblPractica2.setBounds(10, 38, 80, 14);
getContentPane().add(lblPractica2);

lblPractica3 = new JLabel("Práctica 3");
lblPractica3.setBounds(10, 63, 80, 14);
getContentPane().add(lblPractica3);

txtPractica1 = new JTextField();
txtPractica1.setBounds(90, 10, 90, 20);
getContentPane().add(txtPractica1);
txtPractica1.setColumns(10);

txtPractica2 = new JTextField();
txtPractica2.setBounds(90, 35, 90, 20);
getContentPane().add(txtPractica2);
txtPractica2.setColumns(10);

txtPractica3 = new JTextField();
txtPractica3.setBounds(90, 60, 90, 20);
getContentPane().add(txtPractica3);
txtPractica3.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, 70);
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 promedio;
    int p1, p2, p3;

    // Entrada de datos
    p1 = Integer.parseInt(txtPractica1.getText());
    p2 = Integer.parseInt(txtPractica2.getText());
    p3 = Integer.parseInt(txtPractica3.getText());

    // Si amerita, añade 2 puntos a p1
    if (p1 >= 10) {
        p1 = p1 + 2;
        if (p1 > 20)
            p1 = 20;
    }

    // Si amerita, añade 2 puntos a p2
    if (p2 >= 10) {
        p2 = p2 + 2;
        if (p2 > 20)
            p2 = 20;
    }

    // Si amerita, añade 2 puntos a p3
    if (p3 >= 10) {
        p3 = p3 + 2;
        if (p3 > 20)
            p3 = 20;
    }

    // Calcula el promedio
    promedio = (p1 + p2 + p3) / 3.0;

    // Salida de resultados
    txtS.setText("Promedio final : " + promedio);
}

// Procesa la pulsación del botón Borrar
protected void actionPerformedBtnBorrar(ActionEvent arg0) {
    txtPractica1.setText("");
    txtPractica2.setText("");
    txtPractica3.setText("");
    txtS.setText("");
    txtPractica1.requestFocus();
}
}

```

Problema 7

Dadas las edades de tres personas, diseñe un programa que determine la edad mayor.

Algoritmo

Inicio

```

// Declaración de variables
entero edad1, edad2, edad3, edadMayor

```

```

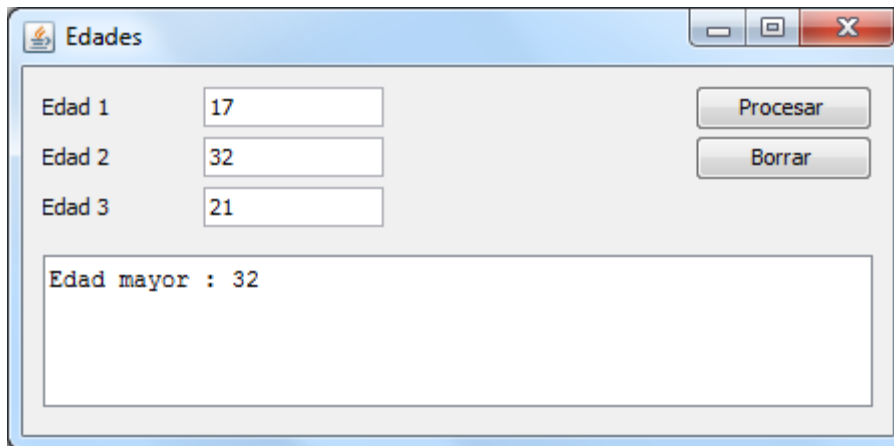
// Entrada de datos
Leer edad1, edad2, edad3

// Determina la edad mayor
edadMayor = edad1
si (edad2 > edadMayor)
    edadMayor = edad2
si (edad3 > edadMayor)
    edadMayor = edad3

// Salida de resultados
Imprimir edadMayor
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 Edades extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;

    // Declaración de variables
    private JLabel lblEdad1;
    private JLabel lblEdad2;
    private JLabel lblEdad3;
    private JTextField txtEdad1;
    private JTextField txtEdad2;
    private JTextField txtEdad3;
    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 {
                Edades frame = new Edades();
                frame.setVisible(true);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

// Crea la GUI
public Edades() {
    setTitle("Edades");
    setBounds(100, 100, 450, 222);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    lblEdad1 = new JLabel("Edad 1");
    lblEdad1.setBounds(10, 13, 80, 14);
    getContentPane().add(lblEdad1);

    lblEdad2 = new JLabel("Edad 2");
    lblEdad2.setBounds(10, 38, 80, 14);
    getContentPane().add(lblEdad2);

    lblEdad3 = new JLabel("Edad 3");
    lblEdad3.setBounds(10, 63, 80, 14);
    getContentPane().add(lblEdad3);

    txtEdad1 = new JTextField();
    txtEdad1.setBounds(90, 10, 90, 20);
    getContentPane().add(txtEdad1);
    txtEdad1.setColumns(10);

    txtEdad2 = new JTextField();
    txtEdad2.setBounds(90, 35, 90, 20);
    getContentPane().add(txtEdad2);
    txtEdad2.setColumns(10);

    txtEdad3 = new JTextField();
    txtEdad3.setBounds(90, 60, 90, 20);
    getContentPane().add(txtEdad3);
    txtEdad3.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, 76);
        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 edad1, edad2, edad3, edadMayor;

        // Entrada de datos
        edad1 = Integer.parseInt(txtEdad1.getText());
        edad2 = Integer.parseInt(txtEdad2.getText());
        edad3 = Integer.parseInt(txtEdad3.getText());

        // Determina la edad mayor
        edadMayor = edad1;
        if (edad2 > edadMayor)
            edadMayor = edad2;
        if (edad3 > edadMayor)
            edadMayor = edad3;

        // Salida de resultados
        txtS.setText("Edad mayor : " + edadMayor);
    }

    // Procesa la pulsación del botón Borrar
    protected void actionPerformedBtnBorrar(ActionEvent arg0) {
        txtEdad1.setText("");
        txtEdad2.setText("");
        txtEdad3.setText("");
        txtS.setText("");
        txtEdad1.requestFocus();
    }
}

```