

**UNIDAD DE  
APRENDIZAJE****3****SEMANA****6**

# **Estructura de selección doble encadenada if – else – 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**

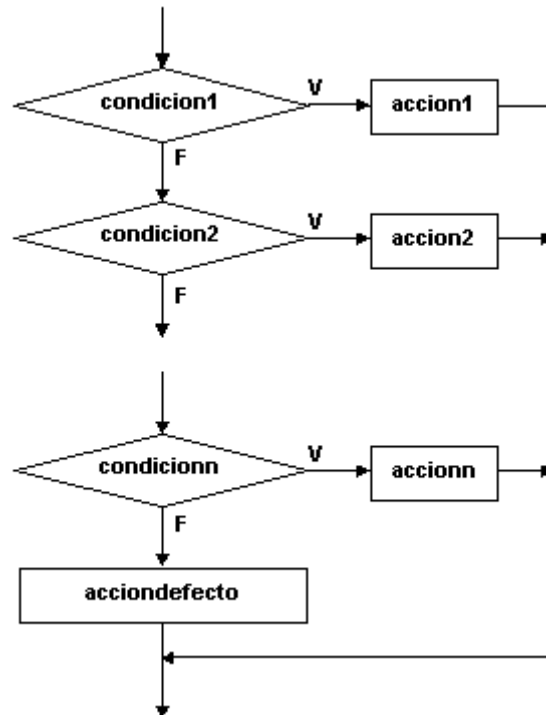
1. Estructura de selección doble encadenada if-else-if

## **ACTIVIDADES**

Los alumnos desarrollan algoritmos que involucren la estructura if-else-if.

## 1. Estructura de selección doble encadenada if-else-if

La *estructura de selección doble encadenada if...else...if* evalúa un conjunto de condiciones en orden descendente, pasando de una condición a otra siempre que la condición anterior sea falsa, y en el momento que encuentre una condición verdadera, efectúa la acción correspondiente a dicha condición y abandona el resto de la estructura. La estructura tiene una acción por defecto que se efectúa en el caso que todas las condiciones sean falsas.



**Figura 1** Estructura de selección doble en cascada

En las tablas que siguen, se muestran el código Java y el pseudocódigo correspondientes a los dos formatos de escritura de la estructura if-else-if.

### Estructura de Selección if – else –if (Formato 1)

Código Java	Pseudocódigo
<pre> if( condicion1)     accion1; else     if( condicion2 )         accion2;     else         if( condicion3 )             accion3;             .             .             .         else             if( condicionn )                 accionn;             else                 acciondefecto; </pre>	<pre> si( condicion1)     accion1 sino     si( condicion2 )         accion2     sino         si( condicion3 )             accion3             .             .             .         sino             si( condicionn )                 accionn             sino                 acciondefecto </pre>

--	--

### Estructura de Selección if – else –if (Formato 2)

Código Java	Pseudocódigo
<pre> if( condicion1)     accion1; else if( condicion2 )     accion2; else if( condicion3 )     accion3; . . . else if( condicionn )     accionn; else     acciondefecto; </pre>	<pre> si( condicion1)     accion1 sino si( condicion2 )     accion2 sino si( condicion3 )     accion3 . . . sino si( condicionn )     accionn sino     acciondefecto </pre>

En el caso de acciones compuestas, estas deben estar encerradas entre llaves de bloque {}.

## 2. Problemas resueltos

### Problema 1

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

Dada la medida de un ángulo en grados, minutos y segundos, diseñe un algoritmo que determine la clasificación del ángulo. Asuma que el ángulo está en el intervalo de  $0^\circ$  a  $360^\circ$ .

### 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

```

```

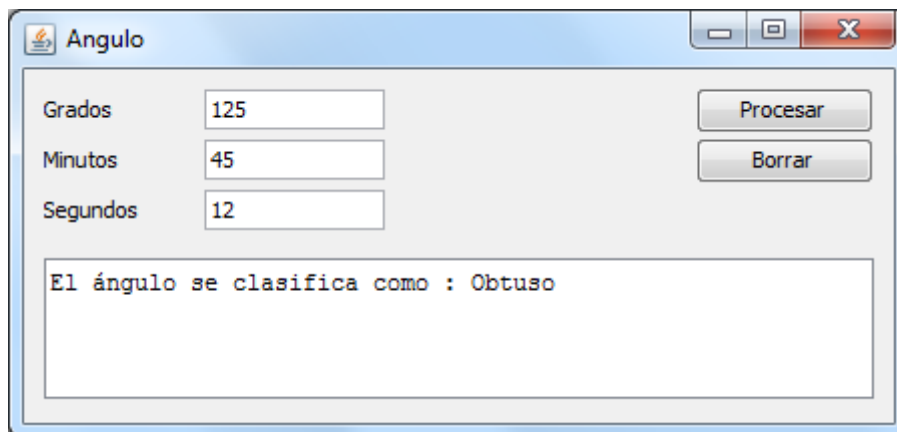
    beta = grados + minutos/60.0 + segundos/3600.0

    // Determina el tipo de ángulo
    si( beta == 0 )
        tipo = "Nulo"
    sino si(beta < 90 )
        tipo = "Agudo"
    sino si( beta == 90 )
        tipo = "Recto"
    sino si( beta < 180 )
        tipo = "Obtuso"
    sino si( beta == 180 )
        tipo = "Llano"
    sino si( beta < 360 )
        tipo = "Cónico"
    sino
        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";
    else if (beta < 90)
        tipo = "Agudo";
    else if (beta == 90)
        tipo = "Recto";
    else if (beta < 180)
        tipo = "Obtuso";
}

```

```

        else if (beta == 180)
            tipo = "Llano";
        else if (beta < 360)
            tipo = "Cóncavo";
        else
            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 2

Resuelva el problema 1 verificando que los datos ingresados sean correctos. Para el efecto, se requiere que la cantidad de grados sea de 0 a 360 y que la cantidad de minutos y de segundos sean de 0 a 59.

### Solución 1: Validación usando la estructura if – else -- if

```

// 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());

    // Verifica los grados
    if (grados < 0 || grados > 360) {
        JOptionPane.showMessageDialog(this, "Grados debe ser de 0 a 360");
        txtGrados.requestFocus();
        txtGrados.selectAll();
    }
    // Verifica los minutos
    else if (minutos < 0 || minutos > 59) {
        JOptionPane.showMessageDialog(this, "Minutos debe ser de 0 a 59");
        txtMinutos.requestFocus();
        txtMinutos.selectAll();
    }
    // Verifica los segundos
    else if (segundos < 0 || segundos > 59) {
        JOptionPane.showMessageDialog(this, "Segundos debe ser de 0 a 59");
        txtSegundos.requestFocus();
        txtSegundos.selectAll();
    }
}

```

```

    }
    // Continúa con datos válidos
    else {
        // Determina el ángulo en grados
        beta = grados + minutos / 60.0 + segundos / 3600.0;

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

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

```

## Solución 2: Validación usando la estructura if

```

// 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());

    // Verifica los grados
    if (grados < 0 || grados > 360) {
        JOptionPane.showMessageDialog(this, "Grados debe ser de 0 a 360");
        txtGrados.requestFocus();
        txtGrados.selectAll();
        return; // Salir si grados es incorrecto
    }

    // Verifica los minutos
    if (minutos < 0 || minutos > 59) {
        JOptionPane.showMessageDialog(this, "Minutos debe ser de 0 a 59");
        txtMinutos.requestFocus();
        txtMinutos.selectAll();
        return; // Salir si minutos es incorrecto
    }

    // Verifica los segundos

```



```

if (segundos < 0 || segundos > 59) {
    JOptionPane.showMessageDialog(this, "Segundos debe ser de 0 a 59");
    txtSegundos.requestFocus();
    txtSegundos.selectAll();
    return; // Salir si segundos es incorrecto
}

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

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

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

```

### Problema 3

En una universidad, los alumnos están clasificados en cuatro categorías. A cada categoría le corresponde una pensión mensual distinta dada en la siguiente tabla:

Categoría	Pensión
A	S/. 550
B	S/. 500
C	S/. 460
D	S/. 400

Semestralmente, la universidad efectúa rebajas en las pensiones de sus estudiantes a partir del segundo ciclo basándose en el promedio ponderado del ciclo anterior en porcentajes dados en la tabla siguiente:

Promedio	Descuento
0 a 13.99	No hay descuento
14.00 a 15.99	10 %
16.00 a 17.99	12 %
18.00 a 20.00	15 %

Dado el promedio ponderado y la categoría de un estudiante, diseñe un programa que determine cuánto de rebaja recibirá sobre su pensión actual y a cuánto asciende su nueva pensión.

## Algoritmo

### Inicio

```
// Declaración de variables
entero categoria
real actualpen, nuevapen, descuento, promedio

// Entrada de datos
Leer categoria, promedio

// Calcula la pensión actual
si( categoria == 0 )
    actualpen = 550
sino si ( categoria == 1 )
    actualpen = 500
sino si ( categoria == 2 )
    actualpen = 460
sino
    actualpen = 400

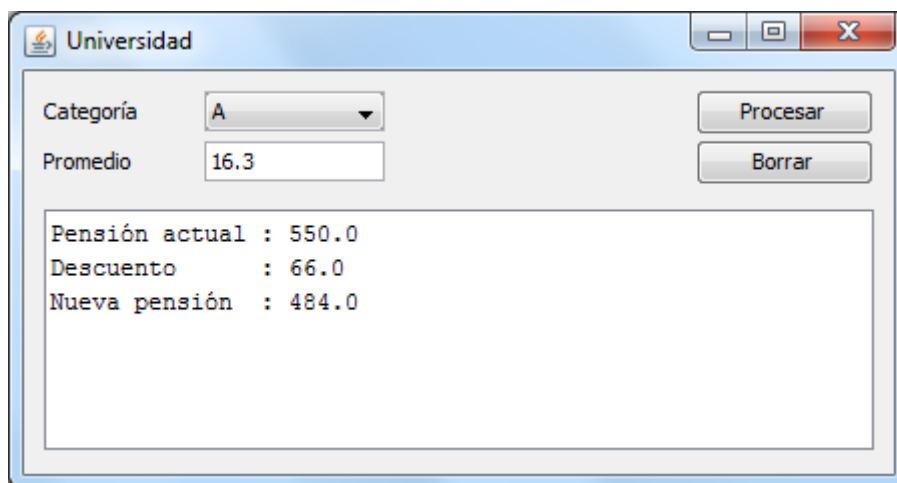
// Calcula el descuento
si( promedio <= 13.99 )
    descuento = 0
sino si( promedio <= 15.99 )
    descuento = 0.10*actualpen
sino si( promedio <= 17.99 )
    descuento = 0.12*actualpen
sino
    descuento = 0.15*actualpen

// Calcula la nueva pensión
nuevapen = actualpen - descuento

// Salida de resultados
Imprimir actualpen, nuevapen
```

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

    // Declaración de variables
    private JLabel lblCategoria;
    private JLabel lblPromedio;
    private JComboBox<String> cboCategoria;
    private JTextField txtPromedio;
    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 {
                    Universidad frame = new Universidad();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

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

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

```

```

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

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

txtPromedio = new JTextField();
txtPromedio.setBounds(90, 35, 90, 20);
getContentPane().add(txtPromedio);
txtPromedio.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 actualpen, nuevapen, descuento, promedio;

    // Entrada de datos
    categoria = cboCategoria.getSelectedIndex();
    promedio = Double.parseDouble(txtPromedio.getText());

    // Calcula la pensión actual
    if (categoria == 0)
        actualpen = 550;
    else if (categoria == 1)
        actualpen = 500;
}

```

```

else if (categoria == 2)
    actualpen = 460;
else
    actualpen = 400;

// Calcula el descuento
if (promedio <= 13.99)
    descuento = 0;
else if (promedio <= 15.99)
    descuento = 0.10 * actualpen;
else if (promedio <= 17.99)
    descuento = 0.12 * actualpen;
else
    descuento = 0.15 * actualpen;

// Calcula la nueva pensión
nuevapen = actualpen - descuento;

// Salida de resultados
txtS.setText("Pensión actual : " + actualpen + "\n");
txtS.append("Descuento      : " + descuento + "\n");
txtS.append("Nueva pensión  : " + nuevapen);
}

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

```

#### Problema 4

Una empresa de préstamos tiene el siguiente esquema de cobros:

Monto del préstamo (S/.)	Número de cuotas
Hasta 5000	2
Más de 5000 hasta 10000	4
Más de 10000 hasta 15000	6
Más de 15000	10

Si el monto del préstamo es mayor de S/. 10000, la empresa cobra 3% de interés mensual; en caso contrario, cobra 5% de interés mensual.

Dado el monto del préstamo de un cliente, diseñe un programa que determine el número de cuotas, el monto de la cuota mensual y el monto del interés total entre todas las cuotas.

#### Algoritmo

##### Inicio

```

// Declaración de variables
real montoprestamo, montointeres, tasainteres, montocuota
entero cuotas

```

```

// Entrada de datos
Leer montoprestamo

// Obtiene el número de cuotas
si( montoprestamo <= 5000 )
    cuotas = 2
sino si( montoprestamo <= 10000 )
    cuotas = 4
sino si( montoprestamo <= 15000 )
    cuotas = 6
sino
    cuotas = 10

// Obtiene la tasa de interés
si( montoprestamo > 10000 )
    tasainterres = 0.03
sino
    tasainterres = 0.05

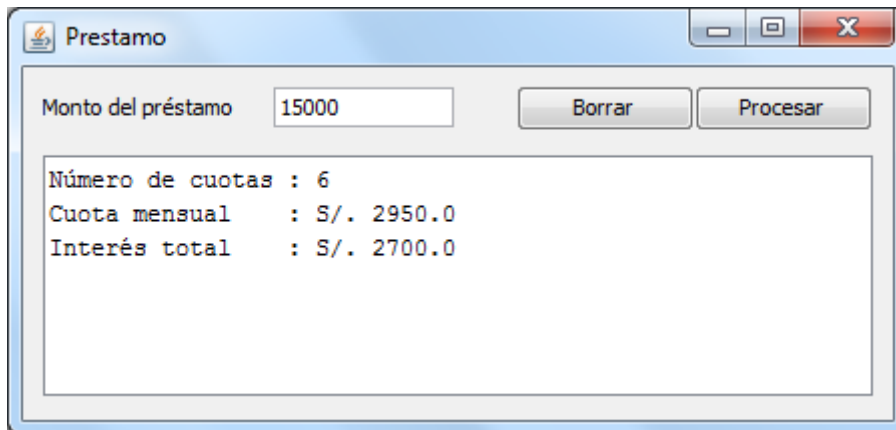
// Calcula el monto del interés total
montointeres = tasainterres*montoprestamo*cuotas

// Calcula el monto de la cuota
montocuota = (montoprestamo + montointeres)/cuotas

// Salida de resultados
Imprimir cuotas, montocuota, montointeres
Inicio

```

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

    // Declaración de variables
    private JLabel lblMontoPrestamo;
    private JTextField txtMontoPrestamo;
    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 {
                    Prestamo frame = new Prestamo();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

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

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

        lblMontoPrestamo = new JLabel("Monto del préstamo");
        lblMontoPrestamo.setBounds(10, 13, 115, 14);
        getContentPane().add(lblMontoPrestamo);

        txtMontoPrestamo = new JTextField();
        txtMontoPrestamo.setBounds(125, 10, 90, 20);
        getContentPane().add(txtMontoPrestamo);
        txtMontoPrestamo.setColumns(10);

        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 montoprestamo, montointeres, tasainteres, montocuota;
        int cuotas;

        // Entrada de datos
        montoprestamo = Double.parseDouble(txtMontoPrestamo.getText());

        // Obtiene el número de cuotas
        if (montoprestamo <= 5000)
            cuotas = 2;
        else if (montoprestamo <= 10000)
            cuotas = 4;
        else if (montoprestamo <= 15000)
            cuotas = 6;
        else
            cuotas = 10;

        // Obtiene la tasa de interés
        if (montoprestamo > 10000)
            tasainteres = 0.03;
        else
            tasainteres = 0.05;

        // Calcula el monto del interés total
        montointeres = tasainteres * montoprestamo * cuotas;

        // Calcula el monto de la cuota
        montocuota = (montoprestamo + montointeres) / cuotas;

        // Salida de resultados
        txtS.setText("Número de cuotas : " + cuotas + "\n");
        txtS.append("Cuota mensual : S/. " + montocuota + "\n");
        txtS.append("Interés total : S/. " + montointeres);
    }

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

```



```

        txtS.setText("");
        txtMontoPrestamo.requestFocus();
    }
}

```

### Problema 5

Una empresa evalúa a sus empleados bajo dos criterios: puntualidad y rendimiento. En cada caso, el empleado recibe un puntaje que va de 1 a 10, de acuerdo con los siguientes criterios:

**Puntaje por puntualidad:-** está en función de los minutos de tardanza de acuerdo con la siguiente tabla:

Minutos de tardanza	Puntaje
0	10
1 a 2	8
3 a 5	6
6 a 9	4
Más de 9	0

**Puntaje por rendimiento:-** está en función de la cantidad de observaciones efectuadas al empleado por no cumplir sus obligaciones de acuerdo con la siguiente tabla:

Observaciones efectuadas	Puntaje
0	10
1	8
2	5
3	1
Más de 3	0

El puntaje total del empleado es la suma del puntaje por puntualidad más el puntaje por rendimiento. Basándose en el puntaje total, el empleado recibe una bonificación anual de acuerdo con la siguiente tabla:

Puntaje total	Bonificación
Menos de 11	S/. 2.5 por punto
11 a 13	S/. 5.0 por punto
14 a 16	S/. 7.5 por punto
17 a 19	S/. 10.0 por punto
20	S/. 12.5 por punto

Dados los minutos de tardanza y el número de observaciones de un empleado, diseñe un programa que determine el puntaje por puntualidad, el puntaje por rendimiento, el puntaje total y la bonificación que le corresponden.

### Algoritmo

#### Inicio

```
// Declaración de variables
```

```
entero minutosTar, numeroObs, puntajePun, puntajeRen, puntajeTot
```

```

    real bonificacion

    // Entrada de datos
    Leer minutosTar, numeroObs

    // Determina el puntaje por puntualidad
    si(minutosTar == 0)
        puntajePun = 10
    sino si(minutosTar <= 2)
        puntajePun = 8
    sino si(minutosTar <= 5)
        puntajePun = 6
    sino si(minutosTar <= 9)
        puntajePun = 4
    sino
        puntajePun = 0

    // Determina el puntaje por rendimiento
    si(numeroObs == 0)
        puntajeRen = 10
    sino si(numeroObs == 1)
        puntajeRen = 8
    sino si(numeroObs == 2)
        puntajeRen = 5
    sino si(numeroObs == 3)
        puntajeRen = 1
    sino
        puntajeRen = 0

    // Determina el puntaje total
    puntajeTot = puntajePun + puntajeRen

    // Determina la bonificación
    si(puntajeTot < 11)
        bonificacion = 2.5*puntajeTot
    sino si(puntajeTot <= 13)
        bonificacion = 5.0*puntajeTot
    sino si(puntajeTot <= 16)
        bonificacion = 7.5*puntajeTot
    sino si(puntajeTot <= 19)
        bonificacion = 10.0*puntajeTot
    sino
        bonificacion = 12.5*puntajeTot

    // Salida de resultados
    Imprimir puntajePun, puntajeRen, puntajeTot, bonificacion
Fin

```

## Programa

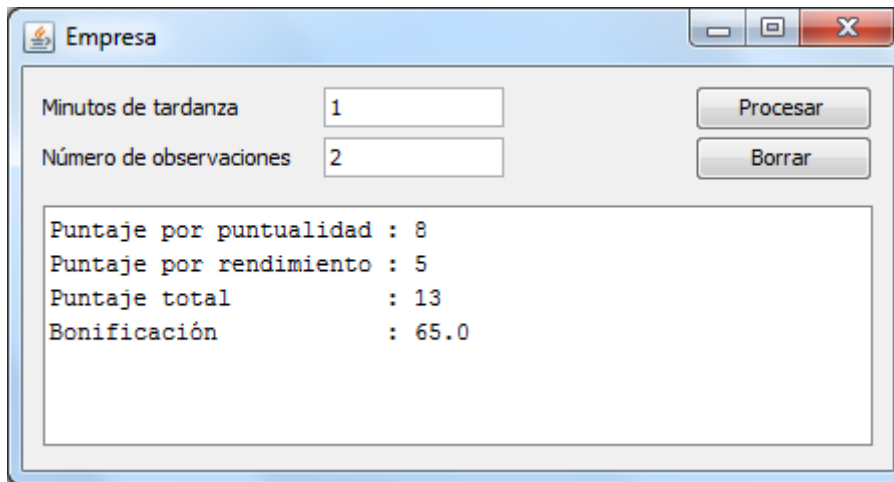
```

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

    // Declaración de variables
    private JLabel lblMinutosTardanza;
    private JLabel lblNumeroObservaciones;
    private JTextField txtMinutosTardanza;
    private JTextField txtNumeroObservaciones;
    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);

lblNumeroObservaciones = new JLabel("Número de observaciones");
lblNumeroObservaciones.setBounds(10, 38, 140, 14);
getContentPane().add(lblNumeroObservaciones);

lblMinutosTardanza = new JLabel("Minutos de tardanza");
lblMinutosTardanza.setBounds(10, 13, 140, 14);
getContentPane().add(lblMinutosTardanza);

txtMinutosTardanza = new JTextField();
txtMinutosTardanza.setBounds(150, 10, 90, 20);
getContentPane().add(txtMinutosTardanza);
txtMinutosTardanza.setColumns(10);

txtNumeroObservaciones = new JTextField();
txtNumeroObservaciones.setBounds(150, 35, 90, 20);
getContentPane().add(txtNumeroObservaciones);
txtNumeroObservaciones.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 minutosTar, numeroObs, puntajePun, puntajeRen, puntajeTot;
    double bonificacion;

    // Entrada de datos
    minutosTar = Integer.parseInt(txtMinutosTardanza.getText());

```

```
numeroObs = Integer.parseInt(txtNumeroObservaciones.getText());

// Determina el puntaje por puntualidad
if (minutosTar == 0)
    puntajePun = 10;
else if (minutosTar <= 2)
    puntajePun = 8;
else if (minutosTar <= 5)
    puntajePun = 6;
else if (minutosTar <= 9)
    puntajePun = 4;
else
    puntajePun = 0;

// Determina el puntaje por rendimiento
if (numeroObs == 0)
    puntajeRen = 10;
else if (numeroObs == 1)
    puntajeRen = 8;
else if (numeroObs == 2)
    puntajeRen = 5;
else if (numeroObs == 3)
    puntajeRen = 1;
else
    puntajeRen = 0;

// Determina el puntaje total
puntajeTot = puntajePun + puntajeRen;

// Determina la bonificación
if (puntajeTot < 11)
    bonificacion = 2.5 * puntajeTot;
else if (puntajeTot <= 13)
    bonificacion = 5.0 * puntajeTot;
else if (puntajeTot <= 16)
    bonificacion = 7.5 * puntajeTot;
else if (puntajeTot <= 19)
    bonificacion = 10.0 * puntajeTot;
else
    bonificacion = 12.5 * puntajeTot;

// Salida de resultados
txtS.setText("Puntaje por puntualidad : " + puntajePun + "\n");
txtS.append("Puntaje por rendimiento : " + puntajeRen + "\n");
txtS.append("Puntaje total : " + puntajeTot + "\n");
txtS.append("Bonificación : " + bonificacion);
}

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

## Problema 6

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, basándose en 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 chocolate.

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, impdes, imppag

// Entrada de datos
Leer tipo, cantidad

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

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

```

    sino
        impdes = 0.115*impcom

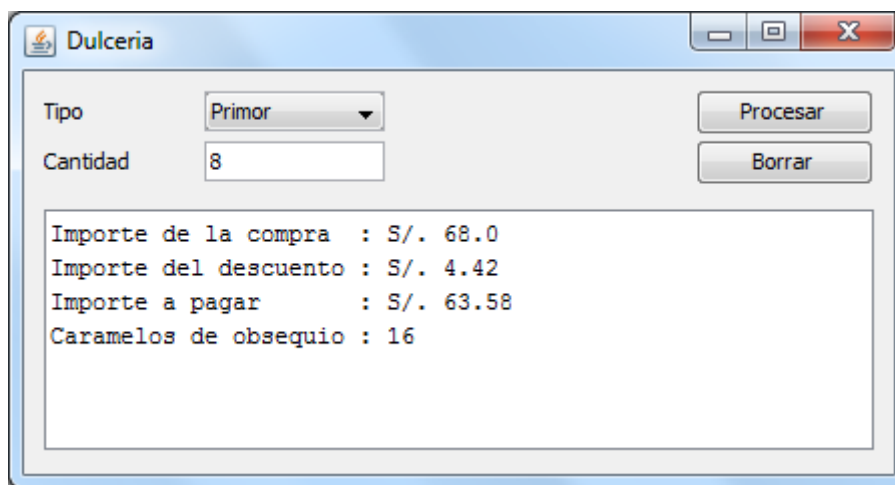
    // Calcula el importe a pagar
    imppag = impcom - impdes

    // 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, impdes, 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;
        else if (tipo == 1)
            impcom = 10.0 * cantidad;
        else if (tipo == 2)
            impcom = 7.0 * cantidad;
        else
            impcom = 12.5 * cantidad;

        // Calcula el importe del descuento
        if (cantidad < 5)
            impdes = 0.04 * impcom;
        else if (cantidad < 10)
            impdes = 0.065 * impcom;
        else if (cantidad < 15)
            impdes = 0.09 * impcom;
        else
            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 7

Una web clasifica a sus usuarios basándose en la cantidad de mensajes publicados en los foros de acuerdo con la siguiente tabla:

Número de mensajes	Clase de usuario
0 a 49	Desconocido
50 a 99	Humano
100 a 249	Diclonius Inicial
250 a 499	Diclonius Novato
500 a 999	Diclonius Experimentado
1000 a 1999	Diclonius Elite
2000 a 4999	Diclonius Definitivo
5000 a más	Diclonius Legendario

Dado el número de mensajes publicados por un usuario, diseñe un programa que determine la clasificación que le corresponde.

## Algoritmo

### Inicio

```

// Declaración de variables
entero num
cadena clase

// Entrada de datos
Leer num

// Determina la clase del usuario
si (num < 50)
    clase = "Desconocido"
sino si (num < 100)
    clase = "Humano"
sino si (num < 250)
    clase = "Diclonius Inicial"
sino si (num < 500)
    clase = "Diclonius Novato"

```

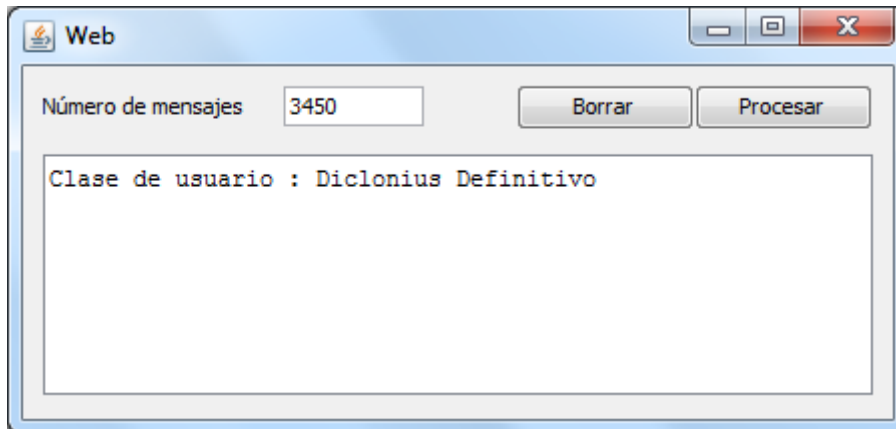
```

    sino si (num < 1000)
        clase = "Diclonius Experimentado"
    sino si (num < 2000)
        clase = "Diclonius Elite"
    sino si (num < 5000)
        clase = "Diclonius Definitivo"
    sino
        clase = "Diclonius Legendario"

    // Salida de resultados
    Imprimir clase
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 Web 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 {
                Web frame = new Web();
                frame.setVisible(true);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

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

    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);

    lblNumero = new JLabel("Número de mensajes");
    lblNumero.setBounds(10, 13, 120, 14);
    getContentPane().add(lblNumero);

    txtNumero = new JTextField();
    txtNumero.setBounds(130, 10, 70, 20);
    getContentPane().add(txtNumero);
    txtNumero.setColumns(10);

    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 num;
    String clase;

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

    // Determina la clase del usuario
    if (num < 50)
        clase = "Desconocido";
    else if (num < 100)
        clase = "Humano";
    else if (num < 250)
        clase = "Diclonius Inicial";
    else if (num < 500)
        clase = "Diclonius Novato";
    else if (num < 1000)
        clase = "Diclonius Experimentado";
    else if (num < 2000)
        clase = "Diclonius Elite";
    else if (num < 5000)
        clase = "Diclonius Definitivo";
    else
        clase = "Diclonius Legendario";

    // Salida de resultados
    txtS.setText("Clase de usuario : " + clase);
}

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