

Estructura de repetición do while

LOGRO DE LA UNIDAD DE APRENDIZAJE

Al finalizar la unidad, el alumno, mediante el uso de estructuras de repetición, implementa programas en Java que resuelven problemas concretos.

TEMARIO

1. Estructura de repetición do – while

ACTIVIDADES

Los alumnos desarrollan algoritmos que involucren el uso de estructuras de repetición.

1. Estructura de repetición do – while

La estructura **do – while** repite una acción o un conjunto de acciones mientras sea verdadera una determinada condición, para lo cual, primero, ejecuta la acción y, luego, verifica la condición. La acción puede ser una **acción simple** o una **acción compuesta** (varias acciones). En la figura 1, se muestra el diagrama de flujo de la estructura hacer-mientras.

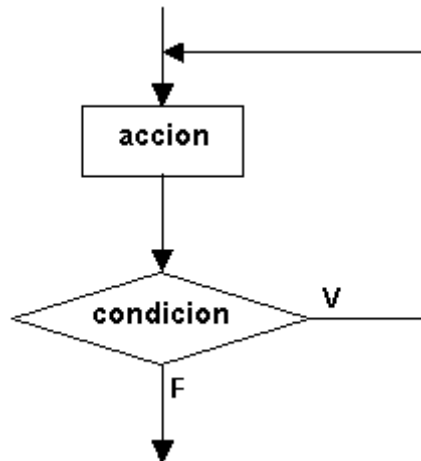


Figura 1 Diagrama de flujo de la estructura **do – while**

La sintaxis de la estructura de repetición **do – while** es la siguiente:

```
do{
    accion1;
    accion2;
    .
    .
    .
    accion3;
} while (condición);
```

2. Problemas resueltos

Problema 1

Diseñe un programa que imprima y sume la siguiente serie:

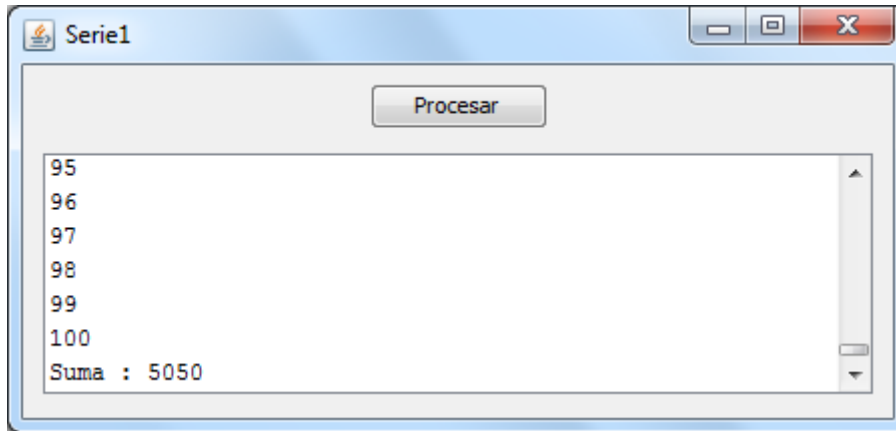
1, 2, 3, ..., 100

Solución

```
package cibertec;

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

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



```
public class Serie1 extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;
    private JButton btnProcesar;
    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 {
                    Serie1 frame = new Serie1();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    // Crea la GUI
    public Serie1() {
        setTitle("Serie1");
        setBounds(100, 100, 450, 214);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().setLayout(null);

        btnProcesar = new JButton("Procesar");
        btnProcesar.addActionListener(this);
        btnProcesar.setBounds(173, 9, 89, 23);
        getContentPane().add(btnProcesar);
    }
}
```

```

        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);
        }
    }

    // Procesa la pulsación del botón Procesar
    protected void actionPerformedBtnProcesar(ActionEvent arg0) {
        // Declara e inicializa variables
        int ter = 1, sum = 0;

        // Limpia la pantalla
        txtS.setText("");

        // Genera los términos de la serie
        do {

            // Imprime el término actual
            txtS.append(ter + "\n");

            // Suma el término actual
            sum += ter;

            // Genera el siguiente término
            ter++;

        } while (ter <= 100);

        // Imprime la suma de la serie
        txtS.append("Suma : " + sum);
    }
}

```

Problema 2

Diseña un programa que imprima y sume la siguiente serie:

3, 10, 17, 24, 31, ..., 94

Solución

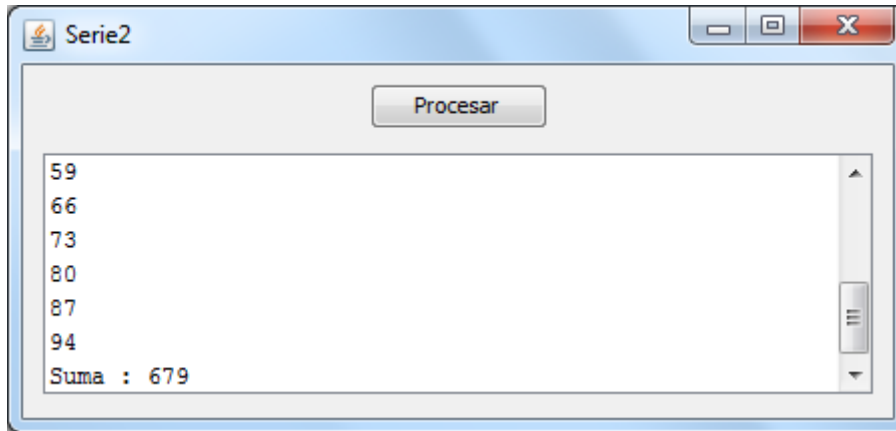
```

package cibertec;

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

```

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



```
public class Serie2 extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;
    private JButton btnProcesar;
    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 {
                    Serie2 frame = new Serie2();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    // Crea la GUI
    public Serie2() {
        setTitle("Serie2");
        setBounds(100, 100, 450, 214);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().setLayout(null);

        btnProcesar = new JButton("Procesar");
        btnProcesar.addActionListener(this);
        btnProcesar.setBounds(173, 9, 89, 23);
        getContentPane().add(btnProcesar);
    }
}
```

```

        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);
        }
    }

    // Procesa la pulsación del botón Procesar
    protected void actionPerformedBtnProcesar(ActionEvent arg0) {
        // Declara e inicializa variables
        int ter = 3, sum = 0;

        // Limpia la pantalla
        txtS.setText("");

        // Genera los términos de la serie
        do {

            // Imprime el término actual
            txtS.append(ter + "\n");

            // Suma el término actual
            sum += ter;

            // Genera el siguiente término
            ter += 7;

        } while (ter <= 94);

        // Imprime la suma de la serie
        txtS.append("Suma : " + sum);
    }
}

```

Problema 3

Diseña un programa que imprima y sume 50 términos de la siguiente serie:

5, 11, 17, 23, 29, 35, ...

Solución

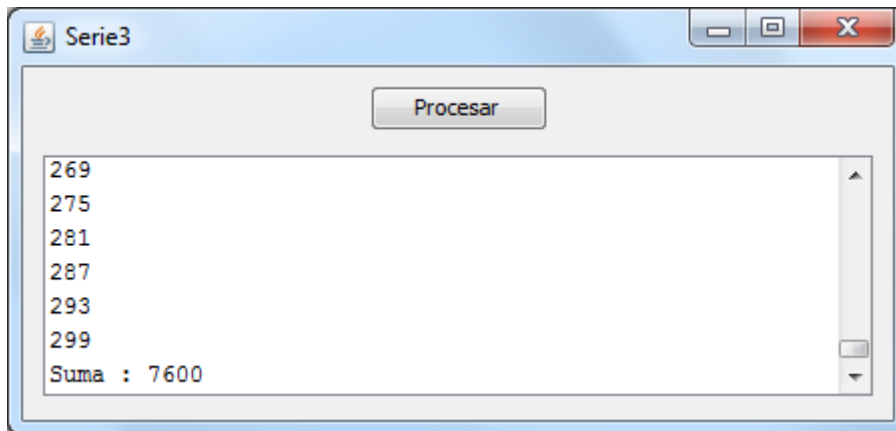
```

package cibertec;

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

```

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



```
public class Serie3 extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;
    private JButton btnProcesar;
    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 {
                    Serie3 frame = new Serie3();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    // Crea la GUI
    public Serie3() {
        setTitle("Serie3");
        setBounds(100, 100, 450, 214);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().setLayout(null);

        btnProcesar = new JButton("Procesar");
        btnProcesar.addActionListener(this);
        btnProcesar.setBounds(173, 9, 89, 23);
        getContentPane().add(btnProcesar);
    }
}
```

```

        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);
        }
    }

    // Procesa la pulsación del botón Procesar
    protected void actionPerformedBtnProcesar(ActionEvent arg0) {
        // Declara e inicializa variables
        int ter = 5, sum = 0, c = 0;

        // Limpia la pantalla
        txtS.setText("");

        // Genera los términos de la serie
        do {

            // Imprime el término actual
            txtS.append(ter + "\n");

            // Suma el término actual
            sum += ter;

            // Genera el siguiente término
            ter += 6;

            // Cuenta la cantidad de términos
            c++;

        } while (c < 50);

        // Imprime la suma de la serie
        txtS.append("Suma : " + sum);
    }
}

```

Problema 4

Diseñe un programa que imprima y sume 75 términos de la siguiente serie:

$$\frac{2}{5}, \frac{5}{9}, \frac{8}{13}, \frac{11}{17}, \dots$$

Solución

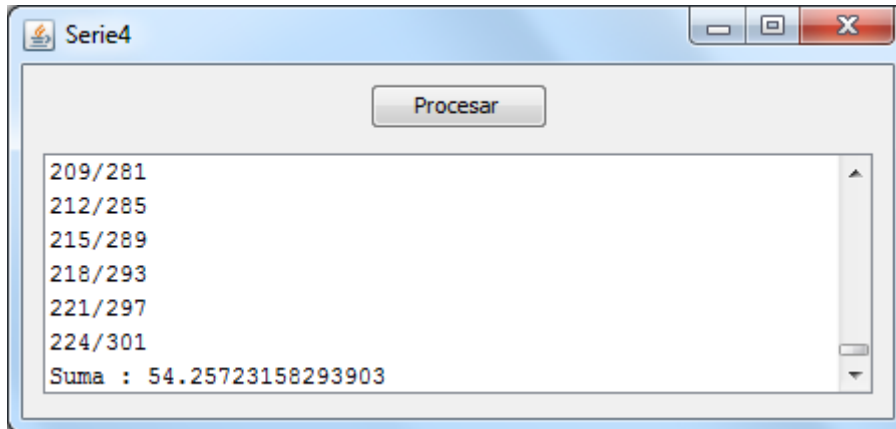
```
package cibertec;
```



```

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

```



```

public class Serie4 extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;
    private JButton btnProcesar;
    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 {
                    Serie4 frame = new Serie4();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    // Crea la GUI
    public Serie4() {
        setTitle("Serie4");
        setBounds(100, 100, 450, 214);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().setLayout(null);
    }
}

```

```

        btnProcesar = new JButton("Procesar");
        btnProcesar.addActionListener(this);
        btnProcesar.setBounds(173, 9, 89, 23);
        getContentPane().add(btnProcesar);

        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);
        }
    }

    // Procesa la pulsación del botón Procesar
    protected void actionPerformedBtnProcesar(ActionEvent arg0) {
        // Declara e inicializa variables
        int num = 2, den = 5, c = 0;
        double sum = 0;

        // Limpia la pantalla
        txtS.setText("");

        // Genera los términos de la serie
        do {

            // Imprime el término actual
            txtS.append(num + "/" + den + "\n");

            // Suma el término actual
            sum += num * 1.0 / den;

            // Genera el siguiente término
            num += 3;
            den += 4;

            // Cuenta la cantidad de términos
            c++;

        } while (c < 75);

        // Imprime la suma de la serie
        txtS.append("Suma : " + sum);
    }
}

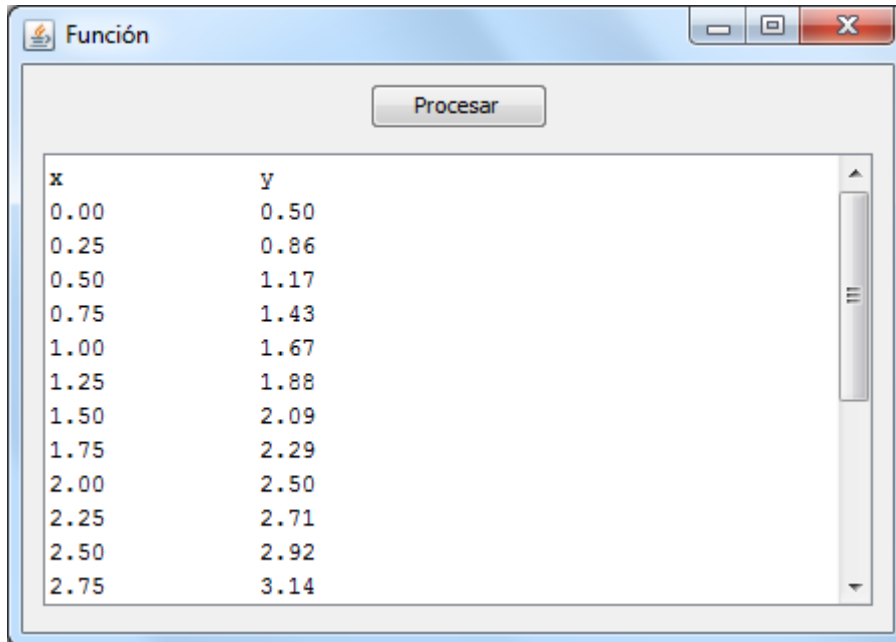
```

Problema 5

Diseña un programa que imprima una tabla de valores de x e y, para valores de x en el intervalo de 0 a 2.75 cada 0.25, siendo:

$$y = \frac{x^3 + 3x + 1}{x^2 + 2}$$

Solución



```
package cibertec;

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

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

```

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

// Crea la GUI
public Funcion() {
    setTitle("Función");
    setBounds(100, 100, 450, 321);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    btnProcesar = new JButton("Procesar");
    btnProcesar.addActionListener(this);
    btnProcesar.setBounds(173, 9, 89, 23);
    getContentPane().add(btnProcesar);

    scpScroll = new JScrollPane();
    scpScroll.setBounds(10, 44, 414, 226);
    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);
    }
}

// Procesa la pulsación del botón Procesar
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declara e inicializa variables
    double x = 0, y;

    // Imprime la cabecera
    txtS.setText(String.format(Locale.US, "%-15.2s%-15.2s\n", "x", "y"));

    // Imprime la tabla
    do {

        // Calcula el valor de y para el valor actual de x
        y = (x * x * x + 3 * x + 1) / (x * x + 2);

        // Imprime la pareja de valores x e y
        txtS.append(String.format(Locale.US, "%-15.2f%-15.2f\n", x, y));

        // Genera el siguiente valor de x
        x += 0.25;

    } while (x <= 2.75);
}

```

```

    }
}

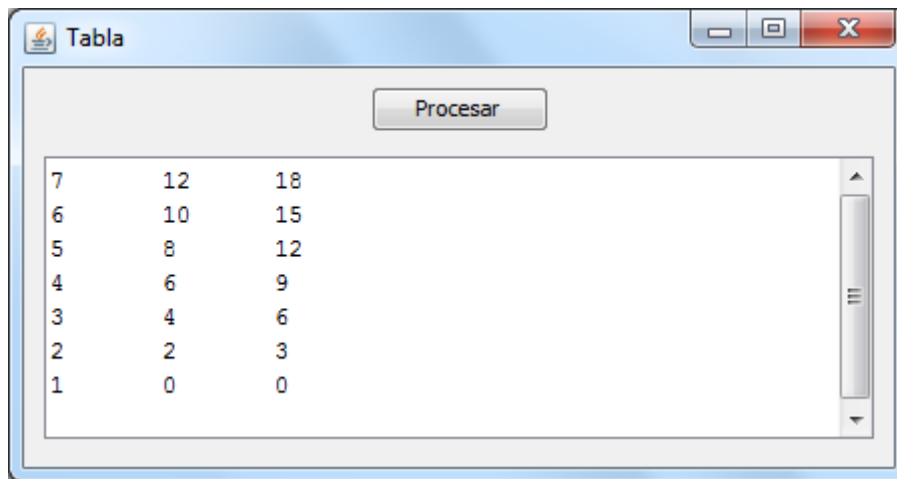
```

Problema 6

Diseñe un programa que imprima la siguiente tabla de números:

7	12	18
6	10	15
5	8	12
4	6	9
3	4	6
2	2	3
1	0	0

Solución



```

package cibertec;

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

public class Tabla extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;
    private JButton btnProcesar;
    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 {
                    Tabla frame = new Tabla();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    // Crea la GUI
    public Tabla() {
        setTitle("Tabla");
        setBounds(100, 100, 450, 237);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().setLayout(null);

        btnProcesar = new JButton("Procesar");
        btnProcesar.addActionListener(this);
        btnProcesar.setBounds(173, 9, 89, 23);
        getContentPane().add(btnProcesar);

        scpScroll = new JScrollPane();
        scpScroll.setBounds(10, 44, 414, 141);
        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);
        }
    }

    // Procesa la pulsación del botón Procesar
    protected void actionPerformedBtnProcesar(ActionEvent arg0) {
        // Declara e inicializa variables
        int a = 7, b = 12, c = 18;

        // Limpia la pantalla
        txtS.setText("");

        // Imprime la tabla
        do {

            // Imprime un elemento de cada columna
            txtS.append(a + "\t" + b + "\t" + c + "\n");

            // Genera el siguiente elemento de cada columna

```

```

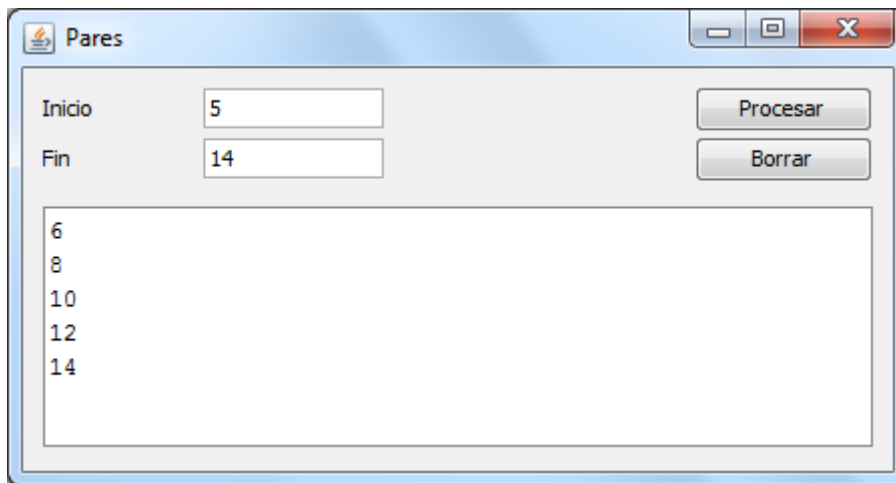
        a--;
        b -= 2;
        c -= 3;
    } while (a >= 1);
}
}

```

Problema 7

Diseñe un programa que lea los extremos de un intervalo de números enteros e imprima todos los números pares del intervalo.

Solución



```

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

    // Declaración de variables
    private JLabel lblInicio;
    private JLabel lblFin;
    private JTextField txtInicio;
    private JTextField txtFin;
    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 {
                Pares frame = new Pares();
                frame.setVisible(true);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

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

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

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

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

    txtFin = new JTextField();
    txtFin.setBounds(90, 35, 90, 20);
    getContentPane().add(txtFin);
    txtFin.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) {
        // Declara variables
        int inicio, fin, numero;

        // Ingresa los extremos del intervalo
        inicio = Integer.parseInt(txtInicio.getText());
        fin = Integer.parseInt(txtFin.getText());

        // Inicializa el número
        numero = inicio;

        // Imprime la lista de números pares
        do {
            if (numero % 2 == 0)
                txtS.append(numero + "\n");
            numero++;
        } while (numero <= fin);
    }

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

```

Problema 8

Diseñe un programa que imprima los divisores de un número natural y la cantidad de divisores encontrados.

Solución

```

package cibertec;

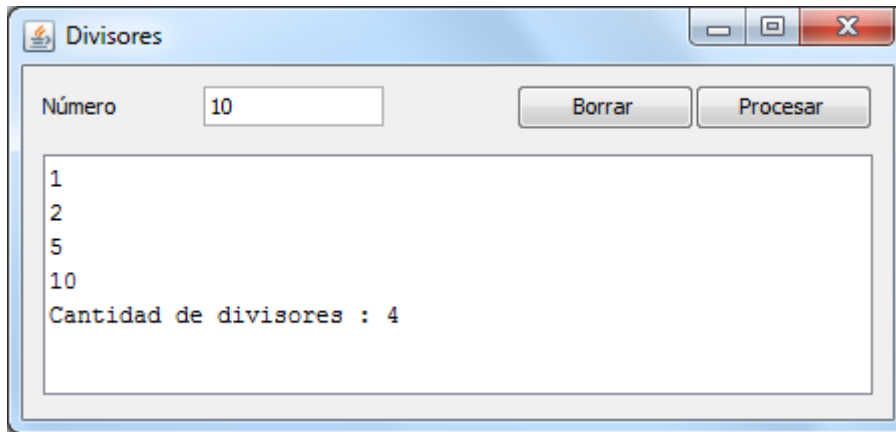
import java.awt.EventQueue;
import java.awt.Font;
import javax.swing.JFrame;

```

```

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

    // Crea la GUI
    public Divisores() {
        setTitle("Divisores");
    }
}

```

```

setBounds(100, 100, 450, 214);
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, 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) {
    // Declara e inicializa variables
    int contadiv = 0, div = 1, num;

    // Ingresa el número
    num = Integer.parseInt(txtNumero.getText());

    // Busca los divisores del número
    do {

        // Verifica si div es un divisor
        if (num % div == 0) {

            // Imprime el divisor encontrado
            txtS.append(div + "\n");

```

```

        // Cuenta el divisor encontrado
        contadiv++;
    }

    // Genera el siguiente posible divisor
    div++;

} while (div <= num);

// Imprime la cantidad de divisores encontrados
txtS.append("Cantidad de divisores : " + contadiv);
}

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

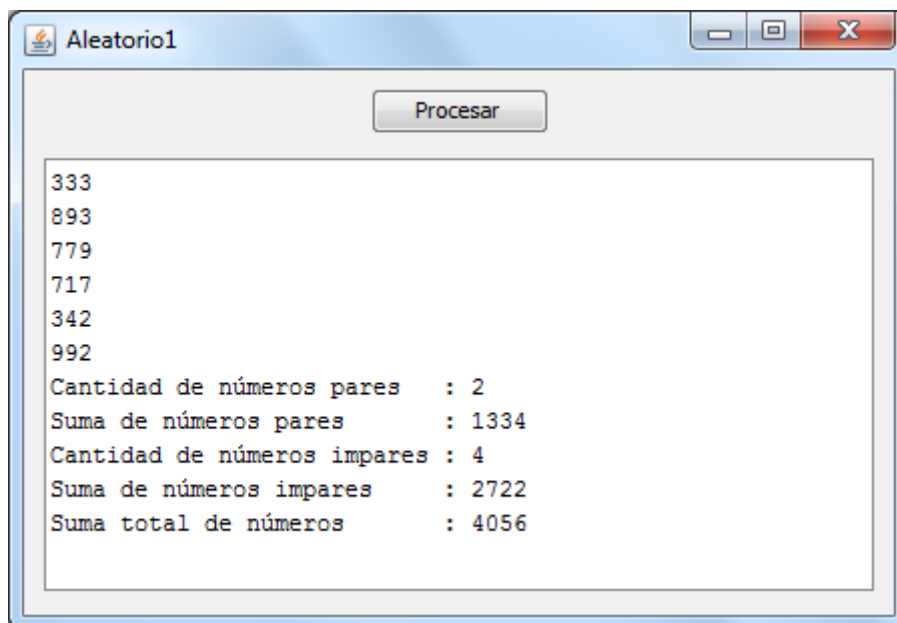
```

Problema 9

Diseña un programa que genere números aleatorios enteros del intervalo 100 a 999 hasta obtener un número par mayor o igual a 500. Imprima los números generados y determine:

- La suma de los números generados
- La cantidad de números pares generados
- La cantidad de números impares generados
- La suma de los números pares generados
- La suma de los números impares generados

Solución



```
package cibertec;
```

```

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

public class Aleatorio1 extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;
    private JButton btnProcesar;
    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 {
                    Aleatorio1 frame = new Aleatorio1();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    // Crea la GUI
    public Aleatorio1() {
        setTitle("Aleatorio1");
        setBounds(100, 100, 450, 310);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().setLayout(null);

        btnProcesar = new JButton("Procesar");
        btnProcesar.addActionListener(this);
        btnProcesar.setBounds(173, 9, 89, 23);
        getContentPane().add(btnProcesar);

        scpScroll = new JScrollPane();
        scpScroll.setBounds(10, 44, 414, 216);
        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);
    }
}

// Procesa la pulsación del botón Procesar
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declara e inicializa variables
    int sumtot = 0, canpar = 0, sumpar = 0, canimp = 0, sumimp = 0, n;

    // Limpia la pantalla
    txtS.setText("");

    // Genera los números aleatorios
    do {
        // Genera un número aleatorio de 100 a 999
        n = (int) ((999 - 100 + 1) * Math.random() + 100);

        // Imprime el número generado
        txtS.append(n + "\n");

        // Suma el número generado
        sumtot += n;

        // Cuenta y suma el número según sea par o impar
        if (n % 2 == 0) {
            canpar++;
            sumpar += n;
        }
        else {
            canimp++;
            sumimp += n;
        }
    } while (!(n % 2 == 0 && n >= 500));

    // Imprime el reporte solicitado
    txtS.append("Cantidad de números pares : " + canpar + "\n");
    txtS.append("Suma de números pares : " + sumpar + "\n");
    txtS.append("Cantidad de números impares : " + canimp + "\n");
    txtS.append("Suma de números impares : " + sumimp + "\n");
    txtS.append("Suma total de números : " + sumtot);
}
}

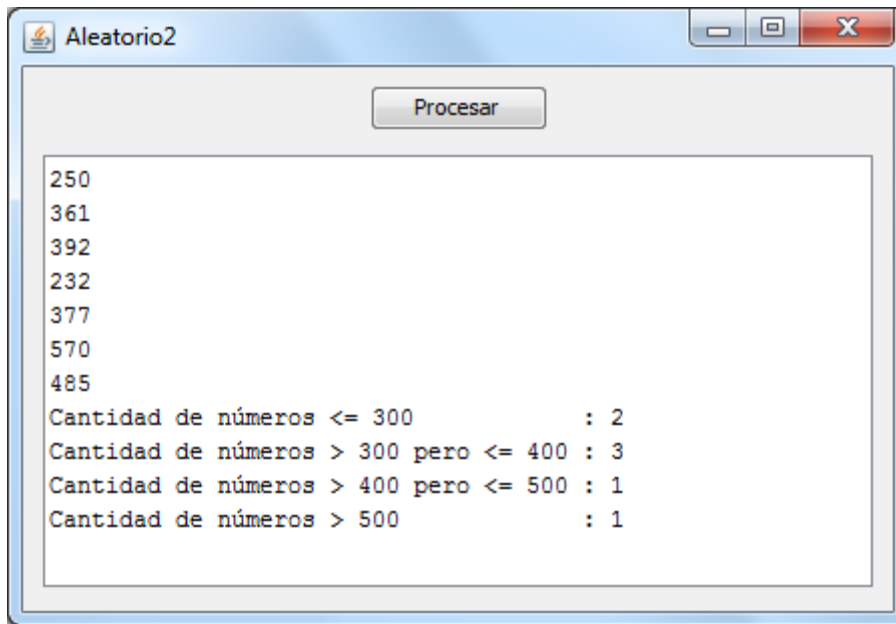
```

Problema 10

Diseñe un programa que genere números aleatorios enteros del intervalo 200 a 600 hasta obtener un número impar mayor de 400 pero menor de 500. Imprima los números generados y determine:

- La cantidad de números generados ≤ 300
- La cantidad de números generados > 300 pero ≤ 400
- La cantidad de números generados > 400 pero ≤ 500
- La cantidad de números generados > 500

Solución



```
package cibertec;

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

public class Aleatorio2 extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;
    private JButton btnProcesar;
    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 {
                    Aleatorio2 frame = new Aleatorio2();
                    frame.setVisible(true);
                }
                catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

```

    }
    });
}

// Crea la GUI
public Aleatorio2() {
    setTitle("Aleatorio2");
    setBounds(100, 100, 450, 310);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    btnProcesar = new JButton("Procesar");
    btnProcesar.addActionListener(this);
    btnProcesar.setBounds(173, 9, 89, 23);
    getContentPane().add(btnProcesar);

    scpScroll = new JScrollPane();
    scpScroll.setBounds(10, 44, 414, 216);
    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);
    }
}

// Procesa la pulsación del botón Procesar
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declara e inicializa variables
    int c1 = 0, c2 = 0, c3 = 0, c4 = 0, n;

    // Limpia la pantalla
    txtS.setText("");

    // Genera los números aleatorios
    do {
        // Genera un número aleatorio de 200 a 600
        n = (int) ((600 - 200 + 1) * Math.random() + 200);

        // Imprime el número generado
        txtS.append(n + "\n");

        // Cuenta el número generado
        if (n <= 300)
            c1++;
        else if (n <= 400)
            c2++;
        else if (n <= 500)
            c3++;
        else
            c4++;
    } while (!(n % 2 != 0 && n > 400 && n < 500));
}

```



```

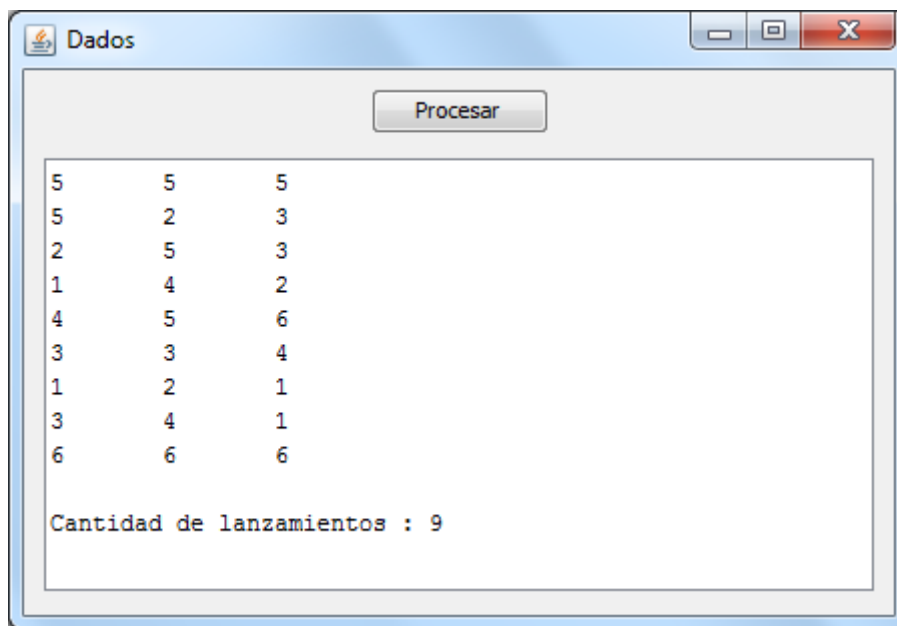
// Imprime el reporte solicitado
txtS.append("Cantidad de números <= 300           : " + c1 + "\n");
txtS.append("Cantidad de números > 300 pero <= 400 : " + c2 + "\n");
txtS.append("Cantidad de números > 400 pero <= 500 : " + c3 + "\n");
txtS.append("Cantidad de números > 500           : " + c4);
    }
}

```

Problema 11

Diseñe un programa que lance tres dados aleatoriamente hasta obtener 6 en los tres dados. Imprima los puntajes obtenidos en cada dado y la cantidad de lanzamientos efectuados.

Solución



```

package cibertec;

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

public class Datos extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;
    private JButton btnProcesar;
    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 {
                Dados frame = new Dados();
                frame.setVisible(true);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

// Crea la GUI
public Dados() {
    setTitle("Dados");
    setBounds(100, 100, 450, 310);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    btnProcesar = new JButton("Procesar");
    btnProcesar.addActionListener(this);
    btnProcesar.setBounds(173, 9, 89, 23);
    getContentPane().add(btnProcesar);

    scpScroll = new JScrollPane();
    scpScroll.setBounds(10, 44, 414, 216);
    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);
    }
}

// Procesa la pulsación del botón Procesar
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declara e inicializa variables
    int c = 0, dado1, dado2, dado3;

    // Limpia la pantalla
    txtS.setText("");

    // Simula el lanzamiento de los tres dados

```

```

do {
    // Genera los puntajes de los tres dados
    dado1 = (int) (6 * Math.random() + 1);
    dado2 = (int) (6 * Math.random() + 1);
    dado3 = (int) (6 * Math.random() + 1);

    // Imprime los puntajes de los dados
    txtS.append(dado1 + "\t" + dado2 + "\t" + dado3 + "\n");

    // Cuenta el lanzamiento efectuado
    c++;
} while (!(dado1 == 6 && dado2 == 6 && dado3 == 6));

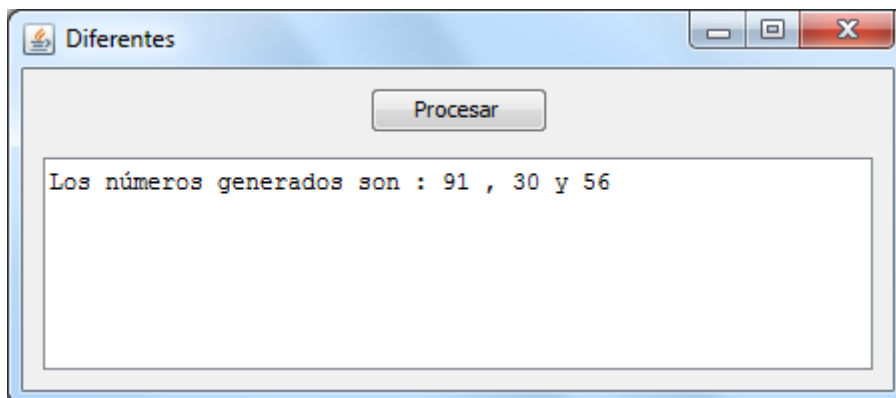
// Imprime la cantidad de lanzamientos
txtS.append("\nCantidad de lanzamientos : " + c);
}
}

```

Problema 12

Diseñe un programa que genere tres números aleatorios enteros diferentes de dos cifras.

Solución



```

package cibertec;

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

public class Diferentes extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;
    private JButton btnProcesar;
    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 {
                Diferentes frame = new Diferentes();
                frame.setVisible(true);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

// Crea la GUI
public Diferentes() {
    setTitle("Diferentes");
    setBounds(100, 100, 450, 198);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    btnProcesar = new JButton("Procesar");
    btnProcesar.addActionListener(this);
    btnProcesar.setBounds(173, 9, 89, 23);
    getContentPane().add(btnProcesar);

    scpScroll = new JScrollPane();
    scpScroll.setBounds(10, 44, 414, 106);
    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);
    }
}

// Procesa la pulsación del botón Procesar
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declara e inicializa variables
    int n1, n2, n3;

    // Limpia la pantalla
    txtS.setText("");

    // Genera los números aleatorios
    do {
        n1 = aleatorio();

```

```

        n2 = aleatorio();
        n3 = aleatorio();
    } while (n1 == n2 || n2 == n3 || n1 == n3);

    // Imprime el reporte solicitado
    txtS.setText("Los números generados son : ");
    txtS.append (n1 + " , " + n2 + " y " + n3);
}

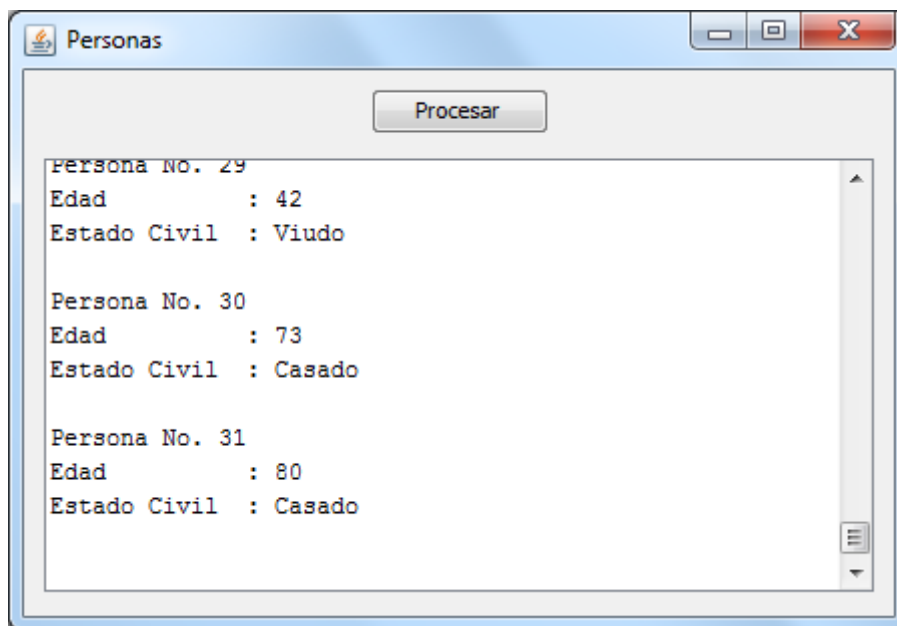
// Genera y retorna un número aleatorio de dos cifras
int aleatorio(){
    return (int) ((99 - 10 + 1) * Math.random() + 10);
}
}

```

Problema 13

Diseñe un programa que genere aleatoriamente la edad y el estado civil de un conjunto de personas hasta obtener una edad igual a 80. La edad será obtenida del intervalo 25 a 80. El estado civil será soltero, casado, viudo o divorciado.

Solución



```

package cibertec;

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

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

```

```

private JButton btnProcesar;
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 {
                Personas frame = new Personas();
                frame.setVisible(true);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

// Crea la GUI
public Personas() {
    setTitle("Personas");
    setBounds(100, 100, 450, 310);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    btnProcesar = new JButton("Procesar");
    btnProcesar.addActionListener(this);
    btnProcesar.setBounds(173, 9, 89, 23);
    getContentPane().add(btnProcesar);

    scpScroll = new JScrollPane();
    scpScroll.setBounds(10, 44, 414, 216);
    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);
    }
}

// Procesa la pulsación del botón Procesar
protected void actionPerformedBtnProcesar(ActionEvent arg0) {
    // Declara e inicializa variables
    int numero = 1; // Número de persona
    int edad;      // Edad de la persona

```

```

String estado; // Estado civil de la persona

// Limpia la pantalla
txtS.setText("");

// Repite hasta que la edad sea 80
do {
    // Genera los datos de la persona
    edad = generaEdad();
    estado = generaEstadoCivil();

    // Imprime los datos de la persona
    txtS.append("Persona No. " + number + "\n");
    txtS.append("Edad      : " + edad + "\n");
    txtS.append("Estado Civil : " + estado + "\n\n");

    // Incrementa el número de persona
    number++;
} while (edad != 80);
}

// Genera y retorna la edad de una persona del intervalo 25 a 80
int generaEdad() {
    return (int) ((80 - 25 + 1) * Math.random() + 25);
}

// Genera y retorna el estado civil de una persona
String generaEstadoCivil() {

    // Genera un número aleatorio del intervalo 1 a 4
    int e = (int) ((4 - 1 + 1) * Math.random() + 1);

    // Obtiene y retorna el estado civil
    switch (e) {
        case 1:
            return "Soltero";
        case 2:
            return "Casado";
        case 3:
            return "Viudo";
        default:
            return "Divorciado";
    }
}
}

```