

**UNIDAD DE
APRENDIZAJE****6****SEMANA****12**

Estructura de repetición **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. Introducción
2. Estructura de repetición while
3. Problemas resueltos

ACTIVIDADES

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

1. Introducción

Se denominan **estructuras repetitivas** o **estructuras de repetición** a aquellas estructuras que permiten repetir instrucciones. A las estructuras repetitivas se conocen también como **estructuras iterativas** o **bucles**, a las instrucciones a repetir se conocen como el **cuerpo del bucle** y al hecho de repetir la secuencia de instrucciones se denomina **iteración**. En el caso del lenguaje Java, tenemos tres tipos de estructuras repetitivas: las estructuras **while**, **do...while** y **for**.

2. Estructura de repetición while

La estructura **while** repite una acción o un conjunto de acciones mientras sea verdadera una determinada condición, para lo cual, primero, verifica la condición y, luego, ejecuta la acción. La acción puede ser una **acción simple** o una **acción compuesta** (bloque de acciones encerradas entre llaves). En la Figura 1, se muestra el diagrama de flujo de la estructura **while**.

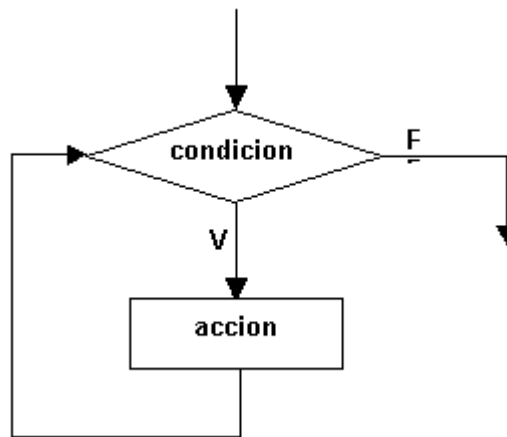


Figura 1 Diagrama de flujo de la estructura **while**

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

Para una sola acción por repetir:

```
while (condicion)
    accion;
```

Para más de una acción por repetir:

```
while (condicion) {
    accion1;
    accion2;
    .
    .
    accion3;
}
```

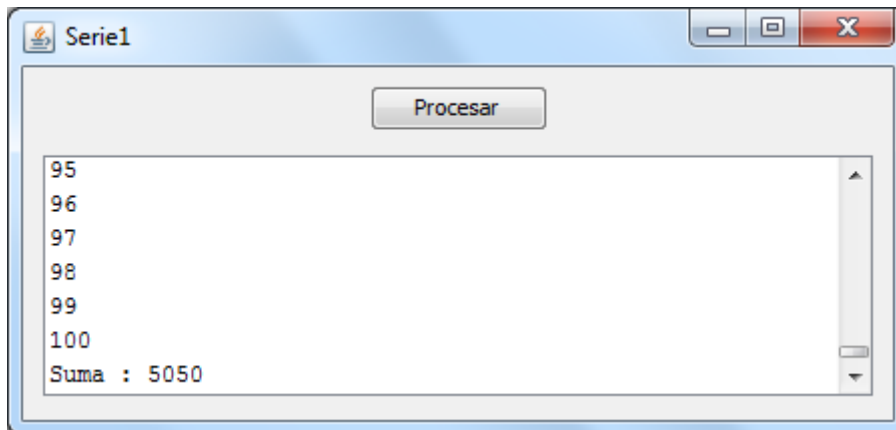
3. Problemas resueltos

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;

    // Genera los términos de la serie
    while (ter <= 100) {

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

        // Suma el término actual
        sum += ter;

        // Pasa al siguiente término
        ter++;
    }

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

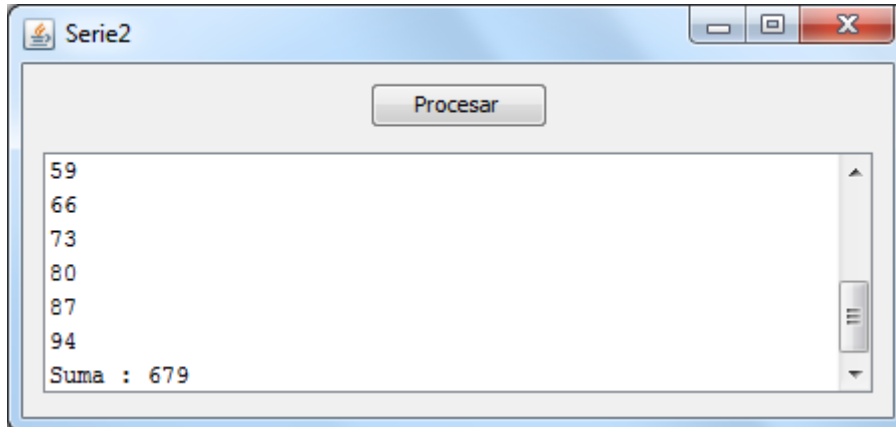
```

Problema 2

Diseñe 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;

    // Genera los términos de la serie
    while (ter <= 94) {

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

        // Suma el término actual
        sum += ter;

        // Pasa al siguiente término
        ter += 7;
    }

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

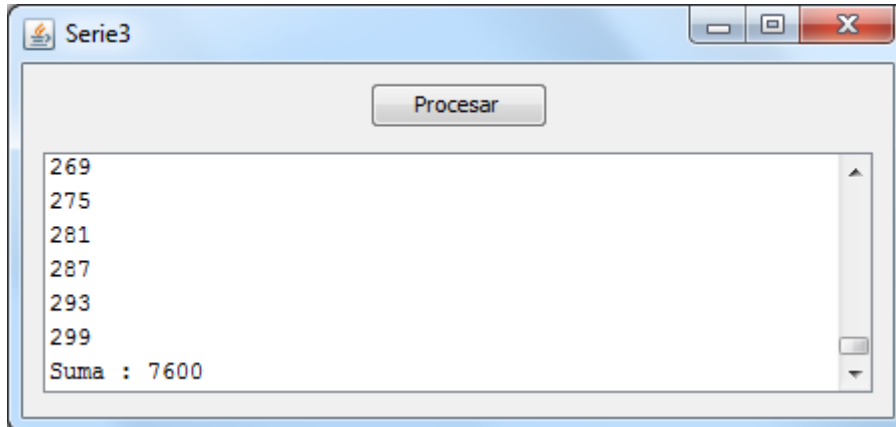
```

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;

    // Genera los términos de la serie
    while (c < 50) {

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

        // Suma el término actual
        sum += ter;

        // Pasa al siguiente término
        ter += 6;

        // Cuenta la cantidad de términos
        c++;
    }

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

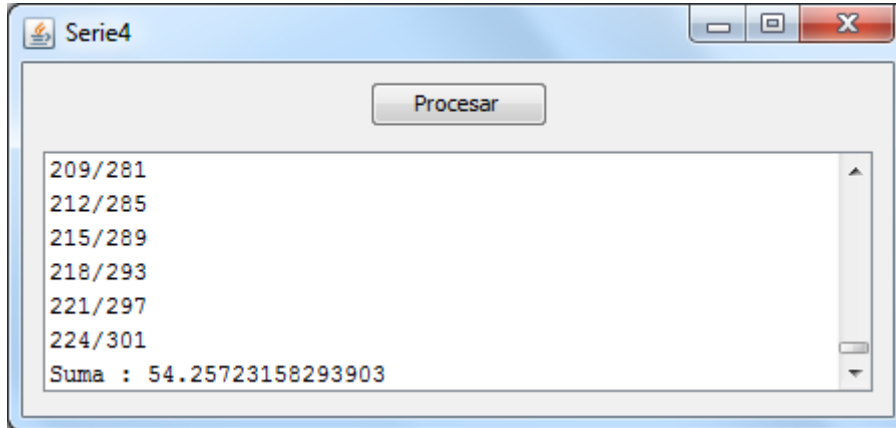
```


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;

    // Genera los términos de la serie
    while (c < 75) {

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

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

        // Pasa al siguiente término
        num += 3;
        den += 4;

        // Cuenta la cantidad de términos
        c++;
    }

    // Imprime la suma de la serie

```

```

        txtS.append("Suma : " + sum);
    }
}

```

Problema 5

Diseñe 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

| x | y |
|------|------|
| 0.00 | 0.50 |
| 0.25 | 0.86 |
| 0.50 | 1.17 |
| 0.75 | 1.43 |
| 1.00 | 1.67 |
| 1.25 | 1.88 |
| 1.50 | 2.09 |
| 1.75 | 2.29 |
| 2.00 | 2.50 |
| 2.25 | 2.71 |
| 2.50 | 2.92 |
| 2.75 | 3.14 |

```

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.append(String.format(Locale.US, "%-15.2s%-15.2s\n", "x", "y"));

    // Imprime la tabla
    while (x <= 2.75) {

        // 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
        txt5.append(String.format(Locale.US, "%-15.2f%-15.2f\n", x, y));

        // Incrementa x en 0.25
        x += 0.25;
    }
}
}

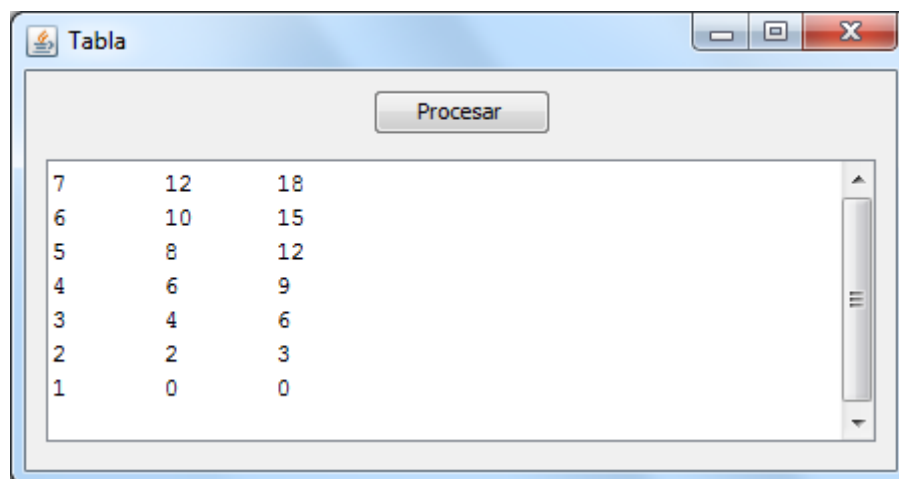
```

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;

    // Imprime la tabla
    while (a >= 1) {

```

```

        // Imprime un elemento de cada columna
        txt5.append(a + "\t" + b + "\t" + c + "\n");

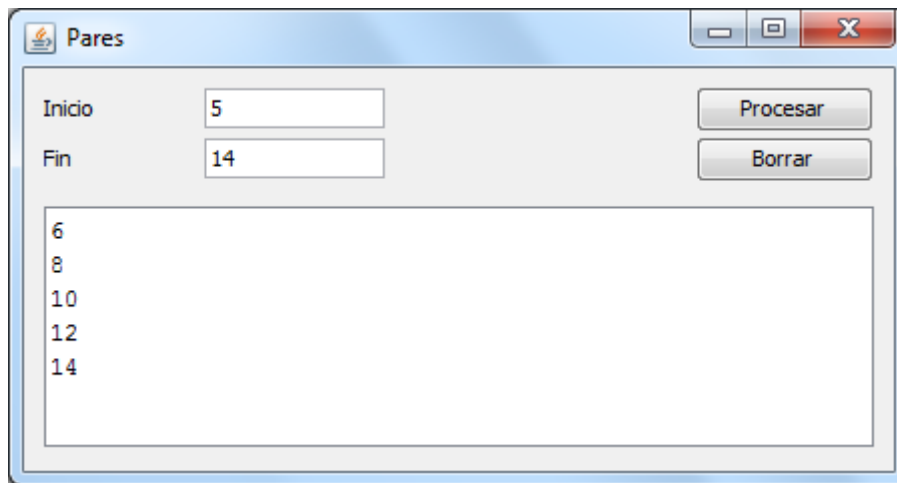
        // Pasa al siguiente elemento de cada columna
        a--;
        b -= 2;
        c -= 3;
    }
}
}

```

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());

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

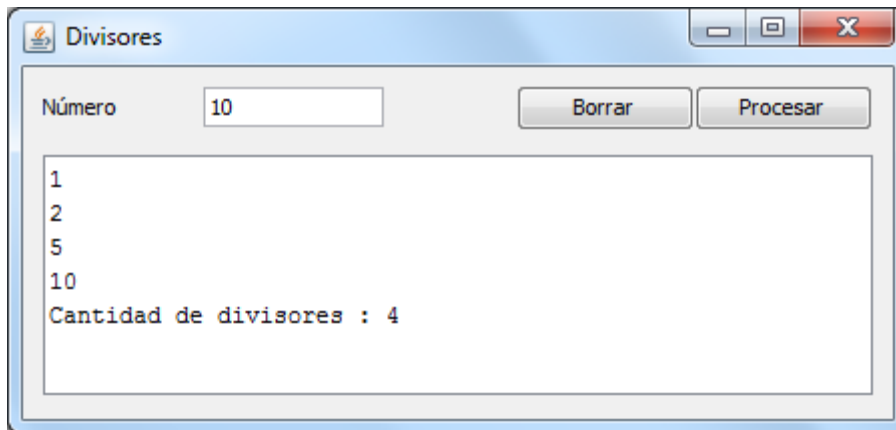
    // Procesa la pulsación del botón Borrar
    protected void actionPerformedBtnBorrar(ActionEvent arg0) {
        txtInicio.setText("");
        txtFin.setText("");
        txtS.setText("");
        txtInicio.requestFocus();
    }
}

```

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 variables
    int contadiv = 0, divisor = 1, numero;

    // Ingresar el número
    numero = Integer.parseInt(txtNumero.getText());

    // Imprime los divisores del número

```

```
// Determina la cantidad de divisores del número
while (divisor <= numero) {
    if (numero % divisor == 0) {
        txtS.append(divisor + "\n");
        contadiv++;
    }
    divisor++;
}

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

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