```java
package EjerciciosPrimerosPasos;

import java.util.Scanner;

public class _1_EntradaEjemplo1 {

    public static void main(String[] args) {

        Scanner entrada = new Scanner(System.in);

        System.out.println("por favor ingrese su nombre");

//      String nombre = entrada.next();  // el método next() sólo devuelve una palabra y no toda la línea
        String nombre = entrada.nextLine();

        System.out.println("por favor ingrese su edad");

        int edad = entrada.nextInt();

        System.out.println("Hola " + nombre + " el próximo año tendrás " + (edad+1) + " años");

    }
}

```

_2_EntradaEjemplo2.java

```java
package EjerciciosPrimerosPasos;

import javax.swing.JOptionPane;

public class _2_EntradaEjemplo2 {

    public static void main(String[] args) {

        String nombre = JOptionPane.showInputDialog("Ingrese su nombre por favor");

        int edad = Integer.parseInt(JOptionPane.showInputDialog("Ingrese su edad"));

        System.out.println("Hola " + nombre + " el próximo año tendrás " + edad + " años.");
    }

}

```

```java
package EjerciciosPrimerosPasos;

import java.text.DecimalFormat;
import javax.swing.JOptionPane;

public class _3_EntradaNumeros {

    public static void main(String[] args) {

        double numero = 0.0;

        while (numero < 1.0 || numero > 100.0) {
            numero = Double.parseDouble(JOptionPane.showInputDialog("Ingrese un número del 1 al 100 por favor"));
            };

        DecimalFormat decimal2 = new DecimalFormat("#.##");  // Clase para convertir en decimal

        System.out.println("La raíz cuadrada de " + numero + " es: " + decimal2.format(Math.sqrt(numero)));
    }
}

```

```java
package EjerciciosPrimerosPasos;

import java.util.Scanner;

public class _4_EvaluaEdad {

    public static void main(String[] args) {

        Scanner entrada = new Scanner(System.in);

        int edad = 0;

        while (edad < 1 || edad > 120) {
            System.out.println("Ingrese su edad por favor");
            edad = entrada.nextInt();
        }

        if (edad >= 18) {
            if (edad <= 40) {
                System.out.println("Usted es Joven");
            } else if (edad <= 64) {
                System.out.println("Usted es Maduro");
            } else if (edad >= 65) {
                System.out.println("Usted es Mayor");
            }
        } else {
            System.out.println("Usted es Adolescente");
        }
    }
}

```

```java
package EjerciciosPrimerosPasos;

import java.text.DecimalFormat;
import java.util.Scanner;

import javax.swing.JOptionPane;

public class _5_CalculoDeAreas {

    public static void main(String[] args) {

        Scanner entrada = new Scanner(System.in);

        DecimalFormat decimal = new DecimalFormat("#.##");

        int numero = 0;

        while (numero < 1 || numero > 4) {

            System.out.println("Seleccione un número por favor: \n 1 - Cuadrado \n 2 - Rectángulo \n 3 - Triángulo \n 4 - Círculo");

            numero = entrada.nextInt();
        }

        if (numero == 1) {

            double ladoCuadrado = Double.parseDouble(JOptionPane.showInputDialog("Ingrese el lado en cm"));
            System.out.println("El área del cuadrado es: " + ladoCuadrado*ladoCuadrado + "cm2");

        } else if (numero == 2) {

            double baseRect = Double.parseDouble(JOptionPane.showInputDialog("Ingrese la base en cm"));
            double alturaRect = Double.parseDouble(JOptionPane.showInputDialog("Ingrese la altura en cm"));
            System.out.println("El área del rectángulo es: " + baseRect*alturaRect + "cm2");

        } else if (numero == 3) {

            double baseTri = Double.parseDouble(JOptionPane.showInputDialog("Ingrese la base en cm"));
            double alturaTri = Double.parseDouble(JOptionPane.showInputDialog("Ingrese la altura en cm"));
            System.out.println("El área del triángulo es: " + (baseTri*alturaTri)/2 + "cm2");

        } else if (numero == 4) {

            double radio = Double.parseDouble(JOptionPane.showInputDialog("Ingrese el radio en cm"));
            System.out.println("El área del círculo es: " + decimal.format(Math.PI*(radio*radio)) + "cm2");
        }
    }
}

```

_6_VerificacionDeEmail.java

```java
package EjerciciosPrimerosPasos;

import javax.swing.JOptionPane;

public class _6_VerificacionDeEmail {

    public static void main(String[] args) {

        String contrasenia = "bilbao";

        boolean comprobar = false;

        while (comprobar == false) {

            String dataIn = JOptionPane.showInputDialog("Ingrese la contraseña por favor");

            if (dataIn.contentEquals(contrasenia) == false) {

                System.out.println("Contraseña incorrecta, corrija e ingrese nuevamente");

            } else {

                System.out.println("Contraseña correcta, bienvenido!");
                comprobar = true;
            }
        }
    }
}

```

```java
package EjerciciosPrimerosPasos;

import javax.swing.JOptionPane;

public class _7_AdivinaNumero {

    public static void main(String[] args) {

//      Random random = new Random();     // Forma más sencilla d eobtener un número random
//
//      int numero = random.nextInt(100);

        double aleatorioDouble = Math.random() * 100 + 1;

        int aleatorioInt = (int) aleatorioDouble;

        int numeroIn = 0;

        int intentos = 0;

        while (numeroIn != aleatorioInt) {

            intentos++;

            numeroIn = Integer.parseInt(JOptionPane.showInputDialog("Ingrese un número entre el 1 y el 100"));

            if (numeroIn < aleatorioInt) {

                System.out.println("Ingrese un número más Alto que: " + numeroIn);

            } else if (numeroIn > aleatorioInt) {

                System.out.println("Ingrese un número más Bajo que: " + numeroIn);
            }
        }

        System.out.println("Has adivinado el número, felicitaciones! lo has hecho en: " + intentos + " intentos.");
    }
}

```

```java
package EjerciciosPrimerosPasos;

import java.util.Random;
import javax.swing.JOptionPane;

public class _8_AdivinaNumeroDoWhile {

    public static void main(String[] args) {

        Random random = new Random();

        int aleatorio = random.nextInt(100);

        int nroIn = 0;

        int intentos = 0;

        do {
            nroIn = Integer.parseInt(JOptionPane.showInputDialog("Ingrese un número del 1 al 100"));

            intentos++;

            if (nroIn > aleatorio) {

                System.out.println("Ingrese un número más Bajo que: " + nroIn);

            } else if (nroIn < aleatorio) {

                System.out.println("Ingrese un número más Alto que: " + nroIn);

            } else {

                System.out.println("Has adivinado el número, felicitaciones! lo has logrado en: " + intentos + " intentos.");
            }

        } while (nroIn != aleatorio || nroIn < 1 || nroIn > 100);
    }
}

```

```java
package EjerciciosPrimerosPasos;

import javax.swing.JOptionPane;

public class _9_PesoIdeal {

    public static void main(String[] args) {

        String dataIn = "";

        int altura = 0;

        while (!dataIn.equalsIgnoreCase("H") || !dataIn.equalsIgnoreCase("M")) {

            dataIn = JOptionPane.showInputDialog("Por favor ingrese su género: H / M");

            if (dataIn.equalsIgnoreCase("H")) {

                altura = Integer.parseInt(JOptionPane.showInputDialog("Ingrese su Altura en cm (ejem: 170)"));

                System.out.println("El peso ideal para un hombre con altura " + altura + "cm es de: " + (altura-110) + "kg");

                break;

            } else if (dataIn.equalsIgnoreCase("M")) {

                altura = Integer.parseInt(JOptionPane.showInputDialog("Ingrese su Altura en cm (ejem: 170)"));

                System.out.println("El peso ideal para una mujer con altura " + altura + "cm es de: " + (altura-110) + "kg");

                break;
            }
        }
    }
}

```

```java
package EjerciciosPrimerosPasos;

public class _10_UsoBucleFor {

    public static void main(String[] args) {

        for (int i = 0; i < 10; i++) {

            System.out.println("La suma progresiva de i++ es: " + i);
        }

        System.out.println("-----------------------------------------------");

        for (int i = 0; i < 20; i+=2) {

            System.out.println("La suma progresiva con salto de i+=2 es: " + i);
        }

        System.out.println("-----------------------------------------------");

        for (int i = 30; i > 20; i--) {

            System.out.println("La resta progresiva de i-- es: " + i);
        }
    }
}

```

```java
package EjerciciosPrimerosPasos;

import javax.swing.JOptionPane;

public class _11_ComprobarDireccionEmail {

    public static void main(String[] args) {

        int arroba = 0;
        boolean punto = false;
        boolean emailTest = false;

        while (emailTest == false) {

            String email = JOptionPane.showInputDialog("Ingrese su email por favor");

            for (int i = 0; i < email.length(); i++) {
                if (email.charAt(i) == '@') {
                    arroba++;
                } else if (email.charAt(i) == '.') {
                    punto = true;
                }
            }
            if (arroba == 1 && punto) {
                System.out.println("Email correcto, bienvenido!");
                emailTest = true;
                break;
            } else {
                System.out.println("Email incorrecto, corrija e ingrese nuevamente");
                arroba = 0;
            }
        }
    }
}

```

```java
package EjerciciosPrimerosPasos;

import javax.swing.JOptionPane;

public class _12_Factorial {

    public static void main(String[] args) {

        int numero = 0;

        int factorial = 1;

        while (numero < 1 || numero > 10) {

            numero = Integer.parseInt(JOptionPane.showInputDialog("Por favor ingrese un número del 1 al 10"));
        }

        for (int i = 1; i < numero + 1; i++) {

            factorial = factorial * i;
        }

        System.out.println("El factorial del número " + numero + " es: " + factorial);
    }
}

```

```java
package EjerciciosPrimerosPasos;

public class _13_UsoArrays {

    public static void main(String[] args) {

        int[] matrix1 = new int[5];

        matrix1[0] = 19;
        matrix1[1] = 25;
        matrix1[2] = -30;
        matrix1[3] = 68;
        matrix1[4] = 45;

        for (int i = 0; i < matrix1.length; i++) {

            System.out.println("El valor del índice " + i + " es: " + matrix1[i]);
        }

        System.out.println("------------------------------");

        int[] matrix2 = {19,25,-30,68,45};

        for (int i = 0; i < matrix2.length; i++) {

            System.out.println("El valor del índice " + i + " es: " + matrix2[i]);
        }
    }
}

```

```java
package EjerciciosPrimerosPasos;

import java.util.Random;
import javax.swing.JOptionPane;

public class _14_UsoArraysII {

    public static void main(String[] args) {

        String [] paises1 = {"Colombia", "Venezuela", "Chile", "Eucador", "Perú", "Brasil", "Argentina"};

        for (int i = 0; i < paises1.length; i++) {

            System.out.println(paises1[i]);
        }

        System.out.println("_____________________");

        System.out.println("Total: " + paises1.length + " países.");

        System.out.println("---------------------");

        String [] paises2 = new String[6];

        for (int i = 0; i < paises2.length; i++) {

            paises2 [i] = JOptionPane.showInputDialog("Por favor ingrese un país");

            System.out.println(paises2[i]);
        }

        System.out.println("_____________________");

        System.out.println("Total: " + paises2.length + " países.");

        System.out.println("---------------------");

        int [] aleatorios = new int[150];

        Random random = new Random();

        for (int i = 0; i < aleatorios.length; i++) {

            aleatorios[i] = random.nextInt(98) + 1;

            System.out.print(aleatorios[i] + " ");
        }
    }
}

```

```java
package EjerciciosPrimerosPasos;

public class _15_ArraysBidimensionales {

    public static void main(String[] args) {

        int [][] matrix1 = new int[4][5];

        matrix1[0][0] = 10;
        matrix1[0][1] = 22;
        matrix1[0][2] = 75;
        matrix1[0][3] = 33;
        matrix1[0][4] = 84;

        matrix1[1][0] = 82;
        matrix1[1][1] = 15;
        matrix1[1][2] = 67;
        matrix1[1][3] = 35;
        matrix1[1][4] = 78;

        matrix1[2][0] = 58;
        matrix1[2][1] = 42;
        matrix1[2][2] = 19;
        matrix1[2][3] = 37;
        matrix1[2][4] = 73;

        matrix1[3][0] = 13;
        matrix1[3][1] = 46;
        matrix1[3][2] = 82;
        matrix1[3][3] = 75;
        matrix1[3][4] = 36;

        for (int i = 0; i < matrix1.length; i++) {

            for (int j = 0; j < matrix1.length + 1; j++) {

                System.out.print(matrix1[i][j] + " ");
            }

            System.out.println("");
        }

        System.out.println("--------------");

        int [][] matrix2 = {{10,22,75,33,84}, {82,15,67,35,78}, {58,42,19,37,73},
  {13,46,82,75,36}};

        for (int i = 0; i < matrix2.length; i++) {

            for (int j = 0; j < matrix2.length + 1; j++) {

                System.out.print(matrix2[i][j] + " ");
            }

            System.out.println(" ");
        }
    }
}

```

```java
package EjerciciosPrimerosPasos;

import java.text.DecimalFormat;

public class _16_Array2D {

    public static void main(String[] args) {

        double [][] matrix = new double[6][5];

        double base = 10000.0;

        double interes = 1.1;

        DecimalFormat decimal = new DecimalFormat("#.##");

        for (int i = 0; i < matrix.length; i++) {

            for (int j = 0; j < matrix.length - 1; j++) {

                matrix[i][j] = base;

                base *= interes;

                System.out.print(decimal.format(matrix[i][j]) + " ");
            }

            base = 10000.0;

            interes += 0.01;

            System.out.println(" ");
        }
    }
}

```

```java
package EjerciciosPOO;

public class _1_PruebaFinal {

    public static void main(String[] args) {

        Empleado empleados[] = new Empleado[3];

        empleados[0] = new Empleado("José de los Campos");
        empleados[1] = new Empleado("María Arreaza");
        empleados[2] = new Empleado("Diego de la Torre");

        empleados[1].setDepartamento("Marketing");
        empleados[2].setDepartamento("Ventas");

        for (int i=0; i<empleados.length; i++) {

            System.out.println(empleados[i].getEmpleado());
        }
    }
}

class Empleado {

    private String nombre;
    private String departamento;
    private int id = 0;
    private static int idFijo = 1;

    public Empleado(String nombre) {

        this.nombre = nombre;
        departamento = "Administración";
        id = idFijo;
        idFijo++;
    }

    public String getEmpleado () {

        return "Id: " + id + " - Empleado: " + nombre + ", Departamento: " + departamento;
    }

    public void setDepartamento (String departamento) {

        this.departamento = departamento;

    }
}
```

```java
package EjerciciosPOO;

class Coche {
    private int ruedas;
    private double largo;
    private double ancho;
    private int motor;
    private double pesoPlataforma;
    private String color;
    private String asientosCuero;
    private String climatizador;

    public Coche(String color, String asientosCuero, String climatizador) {
        ruedas = 4;
        largo = 250.0;
        ancho = 150.0;
        motor = 3600;
        pesoPlataforma = 500.0;
        this.color = color;
        this.asientosCuero = asientosCuero;
        this.climatizador = climatizador;
    }

    public double getPesoPlataforma () {
        return pesoPlataforma;
    }

    public String getAsientosCuero () {
        boolean confirmarAsientos = false;
        if (asientosCuero.equalsIgnoreCase("Si")) {
            confirmarAsientos = true;
        } else if (asientosCuero.equalsIgnoreCase("No")) {
            confirmarAsientos = false;
        }
        if (confirmarAsientos) {
            return "Si tiene";
        } else {
            return "No tiene";
        }
    }

    public String getClimatizador () {
        if (climatizador.equalsIgnoreCase("Si")) {
            return "Si tiene";
        } else {
            return "No tiene";
        }
    }

    public double getPesoTotal () {
        double pesoCarroceria = 1000;
        double pesoTotal = pesoPlataforma + pesoCarroceria;
        if (climatizador.equalsIgnoreCase("Si")) {
            pesoTotal += 60;
        }
        return pesoTotal;
    }

    public double getPrecioTotal () {
        double precioFinal = 10000;
        if (asientosCuero.equalsIgnoreCase("Si")) {
            precioFinal += 600;
```

```
63          }
64          if (climatizador.equalsIgnoreCase("Si")) {
65              precioFinal += 900;
66          }
67          return precioFinal;
68      }
69
70      public String getCoche () {
71          return "Ruedas: " + ruedas + "ud, Largo: " + largo/100 + "m, Ancho: " + ancho/100 +
    "m, Motor: " + motor + "rpm, Peso Plataforma: " + pesoPlataforma + "kg, Color: " + color;
72      }
73  }
```

_3_UsoCoche.java

```java
package EjerciciosPOO;

import javax.swing.JOptionPane;

public class _3_UsoCoche {

    public static void main(String[] args) {

        String color = JOptionPane.showInputDialog("Ingrese color del coche");
        String asientosCuero = JOptionPane.showInputDialog("Tiene asientos de cuero? Si/No");
        String climatizador = JOptionPane.showInputDialog("Tiene climatizador? Si/No");

        Coche coche = new Coche(color, asientosCuero, climatizador);

        System.out.println(coche.getCoche() + ", Asientos de Cuero: " + coche.getAsientosCuero() + ", Climatizador: " + coche.getClimatizador() + ", Peso Total: " + coche.getPesoTotal() + "kg, Precio Total: " + coche.getPrecioTotal() + "€.");

    }

}

```

```java
package EjerciciosPOO;

class Furgoneta extends Coche {

    private double capacidadCarga;
    private int plazasExtra;
    private String climatizadorFurgo;
    private String asientosCueroFurgo;

    public Furgoneta(String color, String asientosCuero, String climatizador, double capacidadCarga, int plazasExtra) {
        super(color, asientosCuero, climatizador);
        this.capacidadCarga = capacidadCarga;
        this.plazasExtra = plazasExtra;
        climatizadorFurgo = climatizador;
        asientosCueroFurgo = asientosCuero;
    }

    public double getPesoTotalFurgo () {
        double pesoTotalFurgo = 0;
        double pesoCarroceríaFurgo = 2000;
        double plazasExtraFurgo = plazasExtra * 20;
        pesoTotalFurgo = pesoCarroceríaFurgo + plazasExtraFurgo + getPesoPlataforma();
        if (climatizadorFurgo.equalsIgnoreCase("Si")) {
            pesoTotalFurgo += 60.0;
        }
        return pesoTotalFurgo;
    }

    public double getPrecioTotalFurgo () {
        double precioTotalFurgo = 14000;
        double asientosExtraFurgo = plazasExtra * 200;
        if (climatizadorFurgo.equalsIgnoreCase("Si")) {
            precioTotalFurgo += 300;
        }
        if (asientosCueroFurgo.equalsIgnoreCase("Si")) {
            precioTotalFurgo += 600;
        }
        return precioTotalFurgo + asientosExtraFurgo;
    }

    public String getFurgoneta() {
        return "Capacidad de carga: " + capacidadCarga + "kg y " + plazasExtra + " plazas extra";
    }
}

```

_5_UsoVehículo.java

```java
package EjerciciosPOO;

import javax.swing.JOptionPane;

public class _5_UsoVehículo {

    public static void main(String[] args) {

        String colorCoche = JOptionPane.showInputDialog("Ingrese color del coche");
        String asientosCueroCoche = JOptionPane.showInputDialog("Coche con asientos de cuero? Si / No");
        String climatizadorCoche = JOptionPane.showInputDialog("Coche climatizado? Si / No");

        Coche coche = new Coche(colorCoche, asientosCueroCoche, climatizadorCoche);

        String colorFurgo = JOptionPane.showInputDialog("Ingrese color de la furgoneta");
        String asientosCueroFurgo = JOptionPane.showInputDialog("Furgo con asientos de cuero? Si / No");
        String climatizadorFurgo = JOptionPane.showInputDialog("Furgo climatizada? Si / No");
        double capacidadCargaFurgo = Double.parseDouble(JOptionPane.showInputDialog("Capacidad de carga? (ejem: 1500"));
        int plazasExtraFurgo = Integer.parseInt(JOptionPane.showInputDialog("Plazas extra? (ejem: 3)"));

        Furgoneta furgoneta = new Furgoneta(colorFurgo, asientosCueroFurgo, climatizadorFurgo, capacidadCargaFurgo, plazasExtraFurgo);

        System.out.println("El Coche tiene: " + coche.getCoche() + ", Asientos de Cuero: " + coche.getAsientosCuero() + ", Climatizador: " + coche.getClimatizador() + ", Peso Total: " + coche.getPesoTotal() + "kg, Precio Total: " + coche.getPrecioTotal() + "€.");
        System.out.println("La Furgoneta tiene: " + furgoneta.getCoche() + ", Asientos de Cuero: " + furgoneta.getAsientosCuero() + ", Climatizador: " + furgoneta.getClimatizador() + ", Peso Total de la Furgoneta: " + furgoneta.getPesoTotalFurgo() + ", " + furgoneta.getFurgoneta() + ", Precio Total: " + furgoneta.getPrecioTotalFurgo());
    }
}

```

```java
package EjerciciosPOO;

public class _6_UsoPersona {

    public static void main(String[] args) {

        Empleado2 empleado = new Empleado2("Juan Sánchez", 2500, 2000, 11, 15, "Encargado de planta");
        Alumno alumno = new Alumno("Mauro Rujano", "Está cursando: ", "Biología Molecular");

        System.out.println("Empleado: " + empleado.getPersona() + ", " + empleado.getEmpleado2());
        System.out.println("Alumno: " + alumno.getPersona() + alumno.getCarrera());
    }
}

abstract class Persona {

    private String nombre;
    private String descripcion;

    public Persona (String nombre, String descripcion) {
        this.nombre = nombre;
        this.descripcion = descripcion;
    }

    public String getNombre () {
        return nombre;
    }

    public String getDescripcion () {
        return descripcion;
    }

    public String getPersona () {
        return "Nombre: " + nombre + ", Descripción: " + descripcion;
    }
}

class Empleado2 extends Persona {
    private double sueldo;
    private int id = 0;
    private static int idFijo = 1;
//  private Date fechaContrato;
    private int anio;
    private int mes;
    private int dia;

    public Empleado2 (String nombre, double sueldo, int anio, int mes, int dia, String descripcion) {
        super(nombre, descripcion);
        this.sueldo = sueldo;
        id = idFijo;
        idFijo++;
//      GregorianCalendar fechaCalendario = new GregorianCalendar(anio, mes, dia);
//      fechaContrato = fechaCalendario.getTime();
        this.anio = anio;
        this.mes = mes;
        this.dia = dia;
    }

    public String getEmpleado2 () {
```

```
60          return "Id: " + id + ", Sueldo; " + sueldo + ", Fecha Contrato: " + dia + "/" + mes
  + "/" + anio;
61      }
62  }
63
64  class Alumno extends Persona {
65      private String carrera;
66
67      public Alumno (String nombre, String descripcion, String carrera) {
68          super(nombre, descripcion);
69          this.carrera = carrera;
70      }
71
72      public String getCarrera () {
73          return carrera;
74      }
75  }
76
```

```java
package EjerciciosPOO;

import java.util.Scanner;

public class _7_EnumUsoTallas {

    public enum Tallas {

        PEQUEÑA("S"),
        MEDIANA("M"),
        GRANDE("L"),
        EXTRAGRANDE("XL");

        private String talla;

        private Tallas(String t) {
            talla = t;
        }

        public String dameTalla () {
            return talla;
        }
    }

    public static void main(String[] args) {

        Scanner entrada = new Scanner(System.in);
        String tallaIn = " ";
        boolean comp = false;

        // Comprobar que la data ingresada sea correcta
        while (comp == false) {
            System.out.println("Ingrese su talla: Pequeña, Mediana, Grande, ExtraGrande");
            tallaIn = entrada.nextLine().toUpperCase().trim();
            for (Tallas iTallas: Tallas.values()) {     // El método values() recorre todos los valores de Enum
                if (tallaIn.equalsIgnoreCase(iTallas.name())) {     // El método name() retorna el nombre literar de cada variable de Enum
                    comp = true;
                    break;
                }
            }
            if (comp == false) {
                System.out.println("Dato incorrecto, corrija nuevamente:");
            }
        }

        Tallas tallaInEnum = Tallas.valueOf(tallaIn);   // Se crea una variable del tipo Tallas Enum que tiene por defecto el valor String introducido por el susuario

        System.out.println("Talla: " + tallaInEnum);

        System.out.println("Abreviatura: " + tallaInEnum.dameTalla());
    }
}

```

```java
package EjerciciosPOO;

interface Jefe {

    public void setDesicion(String desicion);
}


interface Trabajadores {

    double salarioBase = 1500;

    public void setBono(double bono);
}
```

```java
package EjerciciosPOO;

import java.util.Arrays;

public class _9_UsoEmpleado {

    public static void main(String[] args) {

        JefaturaUso jefe1 = new JefaturaUso("Miguel Santos", 3500, 2010, 10, 8);
        jefe1.setDesicion("Vender todas las acciones con urgencia.");
        jefe1.setBono(3500);

        JefaturaUso jefe2 = new JefaturaUso("Rossana", 4200, 2006, 7, 12);
        jefe2.setBono(3000);

        EmpleadoUso empleado1 = new EmpleadoUso("Abel", 2300, 2005, 11, 25);
        empleado1.setBono(2100);

        EmpleadoUso empleadosLista[] = new EmpleadoUso [7];

        empleadosLista[0] = new EmpleadoUso("Miguel", 3500, 2010, 10, 8);
        empleadosLista[1] = new EmpleadoUso("William", 2500, 2000, 5, 1);
        empleadosLista[2] = new EmpleadoUso("Nayla", 2800, 1998, 3, 20);
        empleadosLista[3] = new EmpleadoUso("Diego");
        empleadosLista[4] = new EmpleadoUso("Ángela", "Ruiz");
        empleadosLista[5] = jefe2;
        empleadosLista[6] = new JefaturaUso("Andrea", 5500, 2002, 4, 20);

        JefaturaUso jefe3 = (JefaturaUso) empleadosLista[6];
        jefe3.setBono(55000);

        Arrays.sort(empleadosLista);

        System.out.println("El jefe de Mercadeo ha tomado la desición de: " +
jefe1.getDesicionJefe());
        System.out.println("El jefe : " + jefe1.getNombre() + " tiene un bono de: " +
jefe1.getBonoJefe() + "€");
        System.out.println("El empleado : " + empleado1.getNombre() + " tiene un bono de: "
+ empleado1.getBonoEmpleado() + "€");

        for (EmpleadoUso item: empleadosLista) {
            System.out.println("Id empleado Nº" + item.getId() + " - Nombre: " +
item.getNombre() + ", Salario: " + item.getSalario() + "€, Fecha de contrato: " +
item.getFechaContrato());
        }
    }
}

class EmpleadoUso implements Trabajadores, Comparable {

    private String nombre;
    private double sueldo;
    private double sueldoTotal;
    private String apellido;
    private double bonoEmpleado;
    private Date fechaContrato;
    private int id = 0;
    private static int idFijo = 1;

    public EmpleadoUso(String nombre, double sueldo, int anio, int mes, int dia) {
        this.nombre = nombre;
        this.sueldo = sueldo;
```

```
60          GregorianCalendar fechaGregorian = new GregorianCalendar(anio,mes-1,dia);
61          fechaContrato = fechaGregorian.getTime();
62      }
63
64      public EmpleadoUso(String nombre, String apellido) {
65          this.nombre = nombre;
66          this.apellido = apellido;
67      }
68
69      public EmpleadoUso(String nombre) {
70          this(nombre,3000,2000,1,15);
71      }
72
73      public String getNombre() {
74          return nombre;
75      }
76
77      public int getId() {
78          id = idFijo;
79          idFijo++;
80          return id;
81      }
82
83      public double getSalario() {
84          if(sueldo > 0) {
85              sueldoTotal = sueldo;
86          } else {
87              sueldoTotal = salarioBase;
88          }
89          return sueldoTotal;
90      }
91
92      public Date getFechaContrato() {
93          return fechaContrato;
94      }
95
96      @Override
97      public void setBono(double bono) {
98          this.bonoEmpleado = bono;
99      }
100
101     public double getBonoEmpleado() {
102         return bonoEmpleado;
103     }
104
105     @Override
106     public int compareTo(Object objeto) {
107         EmpleadoUso objetoNew = (EmpleadoUso) objeto;
108         if (this.sueldoTotal < objetoNew.sueldoTotal) {
109             return -1;
110         }
111         if (this.sueldoTotal > objetoNew.sueldoTotal) {
112             return 1;
113         }
114         return 0;
115     }
116 }
117
118 class JefaturaUso extends EmpleadoUso implements Jefe {
119
120     private double sueldo;
121     private double bonoJefe;
```

```
122     private String desicion;
123 
124     public JefaturaUso(String nombre, double sueldo, int anio, int mes, int dia) {
125         super(nombre, sueldo, anio, mes, dia);
126         this.sueldo = sueldo;
127     }
128 
129     @Override
130     public void setBono(double bono) {
131         this.bonoJefe = bono;
132     }
133 
134     public double getBonoJefe() {
135         return bonoJefe;
136     }
137 
138     public double getSalarioTotalJefe() {
139         double salarioTotalJefe = sueldo + bonoJefe;
140         return salarioTotalJefe;
141     }
142 
143     @Override
144     public void setDesicion(String desicion) {
145         this.desicion = desicion;
146     }
147 
148     public String getDesicionJefe() {
149         return desicion;
150     }
151 }
152 
```

```java
package EjerciciosPOO;

import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Date;
import javax.swing.JOptionPane;
import javax.swing.Timer;

public class _10_PruebaTemporizador {

    public static void main(String[] args) {

        Accion accion = new Accion();

        Timer temporizador = new Timer(3000, accion);

        temporizador.start();

        JOptionPane.showMessageDialog(null, "Para desactivar el sonido oprima el botón OK");

        System.exit(0);
    }
}

class Accion implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent e) {

        Date hora = new Date();

        System.out.println("Te muestro la hora cada cierto tiempo: " + hora);

        Toolkit.getDefaultToolkit().beep();
    }
}
```

_01_CreandoYEscribiendoEnMarcos.java

```java
package Graficos;

import java.awt.*;
import javax.swing.*;

public class _01_CreandoYEscribiendoEnMarcos {

    public static void main(String[] args) {
        Marco ventana = new Marco();
        ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class Marco extends JFrame {
    public Marco() {
        Dimension pantalla = Toolkit.getDefaultToolkit().getScreenSize();
        int ancho = (int) pantalla.getWidth();
        int alto = (int) pantalla.getHeight();
        setBounds(ancho/3, alto/3, ancho/3, alto/3);
        setTitle("  Mi Ventana Java");
        ImageIcon icono = new ImageIcon("src/Graficos/images/icon.png");
        setIconImage(icono.getImage());
        Lamina laminaObj = new Lamina();
        add(laminaObj);
        setVisible(true);
    }
}

class Lamina extends JPanel {
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.setFont(new Font("Roboto", Font.BOLD, 20));
        g.setColor(new Color(153, 51, 255));
        g.drawString("Título del Contenido", 100, 100);
    }
}
```

```java
package Graficos;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Ellipse2D;
import java.awt.geom.Rectangle2D;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class _02_PruebaDibujo_TrabajandoConColores {

    public static void main(String[] args) {
        Marco2 ventana = new Marco2();
        ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class Marco2 extends JFrame {
    public Marco2 () {
        setSize(840, 840);
        setTitle("  Dibujos y Colores");
        setLocationRelativeTo(null);
        add(new Lamina2());
        setVisible(true);
    }
}

class Lamina2 extends JPanel {
    public Lamina2() {
        setBackground(new Color(179, 255, 255));
    }
    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.drawRect(100, 200, 150, 100);
        g.drawString("1 - Rectángulo Vacío Graphics", 100, 180);
        g.fillRect(100, 400, 150, 100);
        g.drawString("2 - Rectángulo Relleno Graphics", 100, 380);
        Graphics2D g2D = (Graphics2D) g;
        Rectangle2D rectangulo = new Rectangle2D.Double(500, 200, 150, 100);
        g2D.draw(rectangulo);
        g2D.drawString("3 - Rectángulo Graphics2D", 500, 120);
        Ellipse2D elipse = new Ellipse2D.Double();
        elipse.setFrame(rectangulo);
        g2D.draw(elipse);
        g2D.drawString("4 - Elipse Graphics2D", 500, 140);
        g2D.drawLine(500, 200, 650, 300);
        g2D.drawString("5 - Línea Oblícua Graphics2D", 500, 370);
        double centroX = rectangulo.getCenterX();
        double centroY = rectangulo.getCenterY();
        double radio = 90;
        Ellipse2D circulo = new Ellipse2D.Double();
        circulo.setFrameFromCenter(centroX, centroY, centroX+radio, centroY+radio);
        g2D.draw(circulo);
        g2D.drawString("6 - Círculo Graphics2D", 500, 390);
        g.drawString("7 - Rectángulo Relleno Graphics", 500, 430);
        g.setColor(new Color(255, 153, 255));
        g.fillRect(500, 480, 200, 200);
        g.setColor(new Color(0, 0, 0));
        g.drawString("8 - Círculo Relleno Graphics", 500, 460);
        g.setColor(new Color(179, 255, 179));
```

```
63          g.fillOval(500, 480, 200, 200);
64      }
65  }
```

```java
package Graficos;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Font;
import java.awt.GraphicsEnvironment;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.SwingConstants;

public class _03_FuentesTipo_TrabajandoConFuentes {

    public static void main(String[] args) {

        Marco3 ventana = new Marco3();
        ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class Marco3 extends JFrame {
    public Marco3() {
        setSize(900, 300);
        setTitle(" Trabajando con Fuentes");
        setLocationRelativeTo(null);

        setLayout(new BorderLayout());
        add(new JLabel("             "), BorderLayout.NORTH);
        add(new JLabel("             "), BorderLayout.SOUTH);
        add(new JLabel("             "), BorderLayout.WEST);
        add(new JLabel("             "), BorderLayout.EAST);

        add(new Ventana3(), BorderLayout.CENTER);

        setVisible(true);
    }
}

class Ventana3 extends JPanel {

    String fuenteIn;
    boolean verificador = false;
    JLabel encabezadoText, fuenteText;

    public Ventana3() {

        setLayout(new BorderLayout());

        fuenteIn = JOptionPane.showInputDialog("Ingrese el nombre de la fuente a consultar").toUpperCase();

        GraphicsEnvironment e = GraphicsEnvironment.getLocalGraphicsEnvironment();
        String Fuentes[] = e.getAvailableFontFamilyNames();

        for(String fuente: Fuentes) {
            if(fuenteIn.equalsIgnoreCase(fuente)) {
                verificador = true;
                break;
            }
        }

```

```
62          if(verificador) {
63              add(encabezadoText = new JLabel("El tipo de Fuente:  " + fuenteIn + "  Sí se
  encuentra instalada en el sistema."), JLabel.CENTER);
64              encabezadoText.setFont(new Font(fuenteIn, Font.BOLD, 22));
65              encabezadoText.setForeground(new Color(0, 153, 51));
66              encabezadoText.setHorizontalAlignment(SwingConstants.CENTER);
67              setBackground(new Color(204, 255, 221));
68
69              System.out.println("El tipo de Fuente: " + fuenteIn + " Sí se encuentra
  instalada en el sistema.");
70
71          } else {
72              add(encabezadoText = new JLabel("El tipo de Fuente:  " + fuenteIn + "  No está
  instalada en el sistema."), JLabel.CENTER);
73              encabezadoText.setFont(new Font("Arial", Font.BOLD, 22));
74              encabezadoText.setForeground(new Color(153, 0, 61));
75              encabezadoText.setHorizontalAlignment(SwingConstants.CENTER);
76              setBackground(new Color(255, 204, 224));
77
78              System.out.println("El tipo de Fuente:  " + fuenteIn + "  No está instalada en
  el sistema.");
79          }
80      }
81  }
```

```java
package Graficos;

import java.awt.Graphics;
import java.awt.Image;
import java.io.File;
import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class _04_PruebaImagenes {

    public static void main(String[] args) {

        Marco4 ventana = new Marco4();
        ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class Marco4 extends JFrame {

    public Marco4() {
        setSize(600, 500);
        setTitle(" Prueba Imágenes");
        setLocationRelativeTo(null);

        ImageIcon icono = new ImageIcon("src/Graficos/images/musicIco.png");
        setIconImage(icono.getImage());

        add(new Ventana4());

        setVisible(true);
    }
}

class Ventana4 extends JPanel {

    private Image radioImg;
    private Image musicImg;
    private int anchoImg;
    private int altoImg;

    protected void paintComponent(Graphics g) {

        super.paintComponent(g);

        // --------------------MUSIC IMG-------------------------

        File musicPath = new File("src/Graficos/images/music.png");

        try {
            musicImg = ImageIO.read(musicPath);
        } catch (Exception e) {
            System.out.println("No File!");
        }

        g.drawImage(musicImg, 0, 0, 100, 100, null);

        for (int i=0; i<600; i++) {
            for(int j=0; j<500; j++) {
                g.copyArea(0, 0, 100, 100, i*100, j*100);
            }
```

```
63          }
64
65          // --------------------RADIO IMG-------------------------
66
67          File radioPath = new File("src/Graficos/images/radio.png");
68
69          try {
70              radioImg = ImageIO.read(radioPath);
71          } catch (Exception e) {
72              System.out.println("No File!");
73          }
74
75          anchoImg = radioImg.getWidth(null);
76          altoImg = radioImg.getHeight(null);
77
78          int x = (this.getWidth() - anchoImg/2) / 2;
79          int y = (this.getHeight() - altoImg/2) / 2;
80
81          g.drawImage(radioImg, x, y, anchoImg/2, altoImg/2, null);
82      }
83  }
```

```java
package Graficos;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class _05_PruebaEventos_PruebaAcciones {

    public static void main(String[] args) {

        Marco5 ventana = new Marco5();
        ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class Marco5 extends JFrame {

    public Marco5() {

        setSize(600, 500);
        setTitle(" Eventos y Acciones");
        setLocationRelativeTo(null);

        add(new Ventana5());

        setVisible(true);
    }
}

class Ventana5 extends JPanel {

    public Ventana5() {

        setLayout(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(0, 10, 0, 10);

        AccionColor accionLima = new AccionColor("Lima", new ImageIcon("src/Graficos/images/lima.png"), new Color(204, 255, 153));
        AccionColor accionOrange = new AccionColor("Orange", new ImageIcon("src/Graficos/images/orange.png"), new Color(255, 204, 102));
        AccionColor accionGrape = new AccionColor("Grape", new ImageIcon("src/Graficos/images/grape.png"), new Color(204, 204, 255));

        add(new JButton(accionLima), gbc);
        add(new JButton(accionOrange), gbc);
        add(new JButton(accionGrape), gbc);

        KeyStroke tecladoLima = KeyStroke.getKeyStroke("ctrl L");
        KeyStroke tecladoOrange = KeyStroke.getKeyStroke("ctrl O");
        KeyStroke tecladoGrape = KeyStroke.getKeyStroke("ctrl G");

        InputMap mapaEntrada = getInputMap(WHEN_IN_FOCUSED_WINDOW);

        mapaEntrada.put(tecladoLima, "eventoLima");
        mapaEntrada.put(tecladoOrange, "eventoOrange");
        mapaEntrada.put(tecladoGrape, "eventoGrape");

        ActionMap mapaAccion = getActionMap();

        mapaAccion.put("eventoLima", accionLima);
        mapaAccion.put("eventoOrange", accionOrange);
```

```
60          mapaAccion.put("eventoGrape", accionGrape);
61      }
62
63      private class AccionColor extends AbstractAction {
64
65          public AccionColor(String nombre, Icon icono, Color color_boton) {
66
67              putValue(Action.NAME, nombre);
68              putValue(Action.SMALL_ICON, icono);
69              putValue(Action.SHORT_DESCRIPTION, "(Ctrl+" + nombre.charAt(0) + ") " + "Cambia
    el fondo a color " + nombre);
70              putValue("color_de_fondo", color_boton);
71          }
72
73          public void actionPerformed(ActionEvent e) {
74              Color c = (Color) getValue("color_de_fondo");
75              setBackground(c);
76          }
77      }
78  }
```

```java
package Graficos;

import java.awt.Color;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.event.ActionEvent;
import javax.swing.AbstractAction;
import javax.swing.ActionMap;
import javax.swing.ImageIcon;
import javax.swing.InputMap;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.KeyStroke;

public class _05_PruebaEventos_PruebaAcciones_ByPipe {

    public static void main(String[] args) {

        MarcoAccionesPipe ventana = new MarcoAccionesPipe();
        ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class MarcoAccionesPipe extends JFrame {

    public MarcoAccionesPipe() {

        setTitle("  Eventos y Acciones by Pipe");
        setSize(600, 400);
        setLocationRelativeTo(null);

        add(new VentanaAccionesPipe());

        setVisible(true);
    }
}

class VentanaAccionesPipe extends JPanel {

    public VentanaAccionesPipe() {

        setLayout(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets.set(0, 0, 50, 10);

        Acciones accionLima = new Acciones("Lima", new
ImageIcon("src/Graficos/images/lima.png"), new Color(204, 255, 153));
        Acciones accionOrange = new Acciones("Orange", new
ImageIcon("src/Graficos/images/orange.png"), new Color(255, 204, 102));
        Acciones accionGrape = new Acciones("Grape", new
ImageIcon("src/Graficos/images/grape.png"), new Color(204, 204, 255));

        add(new JButton(accionLima), gbc);
        add(new JButton(accionOrange), gbc);
        add(new JButton(accionGrape), gbc);

        KeyStroke tecladoLima = KeyStroke.getKeyStroke("ctrl L");
        KeyStroke tecladoOrange = KeyStroke.getKeyStroke("ctrl O");
        KeyStroke tecladoGrape = KeyStroke.getKeyStroke("ctrl G");

        InputMap mapaEntrada = getInputMap(WHEN_IN_FOCUSED_WINDOW);
```

```
60
61          mapaEntrada.put(tecladoLima, "tecladoLima");
62          mapaEntrada.put(tecladoOrange, "tecladoOrange");
63          mapaEntrada.put(tecladoGrape, "tecladoGrape");
64
65          ActionMap mapaAccion = getActionMap();
66
67          mapaAccion.put("tecladoLima", accionLima);
68          mapaAccion.put("tecladoOrange", accionOrange);
69          mapaAccion.put("tecladoGrape", accionGrape);
70      }
71
72      class Acciones extends AbstractAction {
73
74          public Acciones(String nombre, ImageIcon icon, Color color) {
75              putValue(NAME, nombre);
76              putValue(SMALL_ICON, icon);
77              putValue("dameColor", color);
78              putValue(SHORT_DESCRIPTION, "Ctr+" + nombre.charAt(0) + "  Cambia el fondo a
    color: " + nombre);
79          }
80
81          @Override
82          public void actionPerformed(ActionEvent e) {
83              Color color = (Color) getValue("dameColor");
84              setBackground(color);
85          }
86      }
87  }
```

```java
package Graficos;

import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import javax.swing.JFrame;

public class _06_EventosFocoVentanaTeclado {

    public static void main(String[] args) {

        Marco6 ventana1 = new Marco6();
        Marco6New ventana2 = new Marco6New();
    }
}

class Marco6 extends JFrame {

    private String nombreVentana;

    public Marco6() {

        setBounds(200, 200, 500, 300);
        setTitle(" Ventana 1º");

        nombreVentana = this.getTitle();

        addWindowFocusListener(new EventosVentana(nombreVentana));
        addWindowListener(new EventosVentana(nombreVentana));
        addWindowStateListener(new EventosVentana(nombreVentana));

        addKeyListener(new EventosTeclado());

        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class Marco6New extends JFrame  {

    private String nombreVentana;

    public Marco6New() {

        setBounds(900, 200, 500, 300);
        setTitle(" Ventana 2º");

        nombreVentana = this.getTitle();

        addWindowFocusListener(new EventosVentana(nombreVentana));
        addWindowListener(new EventosVentana(nombreVentana));
        addWindowStateListener(new EventosVentana(nombreVentana));

        addKeyListener(new EventosTeclado());

        setVisible(true);
    }
}

class EventosVentana extends WindowAdapter {

```

```
63      String nombre;
64
65      public EventosVentana(String nombre) {
66          this.nombre = nombre;
67      }
68
69      @Override
70      public void windowActivated(WindowEvent e) {
71          System.out.println("La Ventana se ha Activado " + nombre);
72          super.windowActivated(e);
73      }
74
75      @Override
76      public void windowDeactivated(WindowEvent e) {
77          System.out.println("La Ventana se ha Desactivado" + nombre);
78          super.windowDeactivated(e);
79      }
80
81      @Override
82      public void windowOpened(WindowEvent e) {
83          System.out.println("La Ventana se ha Abierto" + nombre);
84          super.windowOpened(e);
85      }
86
87      @Override
88      public void windowClosing(WindowEvent e) {
89          System.out.println("La Ventana se está Cerrando" + nombre);
90          super.windowClosing(e);
91      }
92
93      @Override
94      public void windowClosed(WindowEvent e) {
95          System.out.println("La Ventana se ha Cerrado" + nombre);
96          super.windowClosed(e);
97      }
98
99      @Override
100     public void windowIconified(WindowEvent e) {
101         System.out.println("La Ventana se ha Minimizado" + nombre);
102         super.windowIconified(e);
103     }
104
105     @Override
106     public void windowDeiconified(WindowEvent e) {
107         System.out.println("La Ventana se ha Maximizado" + nombre);
108         super.windowDeiconified(e);
109     }
110
111     @Override
112     public void windowGainedFocus(WindowEvent e) {
113         System.out.println("La Ventana ha Ganado el Foco" + nombre);
114         super.windowGainedFocus(e);
115     }
116
117     @Override
118     public void windowLostFocus(WindowEvent e) {
119         System.out.println("La Ventana ha Perdido el Foco" + nombre);
120         super.windowLostFocus(e);
121     }
122
123     @Override
124     public void windowStateChanged(WindowEvent e) {
```

```
125         System.out.println("La Ventana ha Cambiado de Estado" + nombre);
126         super.windowStateChanged(e);
127     }
128 }
129 
130 
131 class EventosTeclado implements KeyListener {
132 
133     @Override
134     public void keyTyped(KeyEvent e) {
135         System.out.println("Se ha Tecleado la tecla: " + e.getKeyChar());
136     }
137 
138     @Override
139     public void keyPressed(KeyEvent e) {
140         System.out.println("Se ha Presionado la tecla: " + e.getKeyChar());
141     }
142 
143     @Override
144     public void keyReleased(KeyEvent e) {
145         System.out.println("Se ha Levantado la tecla: " + e.getKeyChar());
146     }
147 }
```

```java
package Graficos;

import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.MouseWheelEvent;
import javax.swing.JFrame;

public class _07_EventosRaton extends JFrame {

    public static void main(String[] args) {

        JFrame ventana = new JFrame();

        ventana.setTitle(" Eventos del Ratón");
        ventana.setSize(500, 400);
        ventana.setLocationRelativeTo(null);

        ventana.addMouseListener(new AccionesRaton());
        ventana.addMouseMotionListener(new AccionesRaton());
        ventana.addMouseWheelListener(new AccionesRaton());

        ventana.setVisible(true);
        ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class AccionesRaton extends MouseAdapter {

    @Override
    public void mouseClicked(MouseEvent e) {
        int boton = e.getButton();
        String click = "";
        if  (boton == 1) {
            click = "Click Izquierdo";
        } else if  (boton == 2) {
            click = "Rueda del Ratón";
        } else if  (boton == 3) {
            click = "Click Derecho";
        }
        int x = e.getX();
        int y = e.getY();
        System.out.println("Se ha hecho Click con: " + click + " en la coordenada: X=" + x + " Y=" + y);
        super.mouseClicked(e);
    }

    @Override
    public void mouseDragged(MouseEvent e) {
        System.out.println("Se ha Arrastrado el ratón");
        super.mouseDragged(e);
    }

    @Override
    public void mouseEntered(MouseEvent e) {
        System.out.println("El ratón ha Entrado en la ventana");
        super.mouseEntered(e);
    }

    @Override
    public void mouseExited(MouseEvent e) {
        System.out.println("El ratón ha Salido de la ventana");
        super.mouseExited(e);
```

```
62      }
63
64      @Override
65      public void mouseMoved(MouseEvent e) {
66  //      System.out.println("Se ha Movido el ratón");
67          super.mouseMoved(e);
68      }
69
70      @Override
71      public void mousePressed(MouseEvent e) {
72          System.out.println("Se ha Presionado el ratón");
73          super.mousePressed(e);
74      }
75
76      @Override
77      public void mouseReleased(MouseEvent e) {
78          System.out.println("Se ha Soltado la tecla del ratón");
79          super.mouseReleased(e);
80      }
81
82      @Override
83      public void mouseWheelMoved(MouseWheelEvent e) {
84          System.out.println("Se ha Movido la Rueda del ratón");
85          super.mouseWheelMoved(e);
86      }
87  }
```

```java
package Graficos;
import java.awt.Color;
import java.awt.GridBagLayout;
import java.awt.GridLayout;
import java.awt.event.FocusEvent;
import java.awt.event.FocusListener;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class _08_FocoEvento {

    public static JLabel aviso;

    public static void main(String[] args) {

        Color fondo = new Color(230, 204, 255);
        JTextField email;

        JFrame ventana = new JFrame();
        ventana.setTitle(" Foco Evento");
        ventana.setSize(600, 400);
        ventana.setLocationRelativeTo(null);
        ventana.setLayout(new GridBagLayout());
        ventana.getContentPane().setBackground(fondo);

        JPanel lamina = new JPanel();
        lamina.setBackground(fondo);

        CampoFoco oyenteFoco = new CampoFoco();

        lamina.setLayout(new GridLayout(10,1));

        lamina.add(new JLabel("eMail: ")).setForeground(Color.GRAY.darker());
        lamina.add(email = new JTextField(30));
        email.addFocusListener(oyenteFoco);
        lamina.add(new JLabel("                ")).setForeground(Color.GRAY.darker());
        lamina.add(aviso = new JLabel("
"
)).setForeground(Color.RED.darker());
        lamina.add(new JLabel("                ")).setForeground(Color.GRAY.darker());
        lamina.add(new JLabel("Contraseña: ")).setForeground(Color.GRAY.darker());
        lamina.add(new JTextField(30));
        lamina.add(new JLabel("                ")).setForeground(Color.GRAY.darker());  //
Éstos labels son sólo para aumentar espacio
        lamina.add(new JLabel("                ")).setForeground(Color.GRAY.darker());
        lamina.add(new JLabel("                ")).setForeground(Color.GRAY.darker());

        ventana.add(lamina);

        ventana.setVisible(true);
        ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    static class CampoFoco implements FocusListener {

        @Override
        public void focusGained(FocusEvent e) {
            // TODO Auto-generated method stub
        }

        @Override
```

```
61          public void focusLost(FocusEvent e) {
62              JTextField emailObj = (JTextField) e.getSource();
63              String emailTxt = emailObj.getText();
64              int arroba = 0;
65
66              boolean punto = false;
67
68              for (int i=0; i<emailTxt.length(); i++) {
69                  if (emailTxt.charAt(i) == '@') {
70                      arroba++;
71                  }
72                  if (emailTxt.charAt(i) == '.') {
73                      punto = true;
74                  }
75              }
76
77              if (arroba == 1 && punto) {
78                  aviso.setText(" ");
79                  System.out.println("Email correcto");
80              } else {
81                  aviso.setText("eMail Incorrecto, corrija e intente nuevamente");
82                  System.out.println("Email Incorrecto");
83                  arroba = 0;
84              }
85          }
86      }
87  }
```

```java
package Graficos;

import java.awt.Color;
import java.awt.Font;
import java.awt.GridBagLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Random;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class _09_VariosOyentes {

    public static void main(String[] args) {
        Ventana9 ventana = new Ventana9();
    }
}

class Ventana9 extends JFrame {
    public Ventana9() {
        setSize(400, 300);
        setTitle(" Varios Oyentes");
        setBounds(1200, 200, 400, 300);
        add(new Lamina9());
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class Lamina9 extends JPanel {
    JButton btn1, btn2;
    public Lamina9() {
        setBackground(new Color(255, 255, 153));
        setLayout(new GridBagLayout());
        add(btn1 = new JButton("Nueva Ventana"));
        add(btn2 = new JButton("Cerrar Ventanas"));
        btn1.addActionListener(new OyenteListener());
    }

    public class OyenteListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            VentanaEmergente marco = new VentanaEmergente(btn2);
        }
    }

    int numero = 1;

    class VentanaEmergente extends JFrame {

        public VentanaEmergente(JButton btnCerrar) {
            setTitle("Ventana " + numero);
            setBounds(40*numero, 40*numero, 250, 200);

            add(new NombreAleatorio());

            setVisible(true);
            numero++;
            btnCerrar.addActionListener(new OyenteCerrar());
        }
```

```
63
64          class OyenteCerrar implements ActionListener {
65              @Override
66              public void actionPerformed(ActionEvent e) {
67                  dispose();
68                  numero = 1;
69              }
70          }
71      }
72  }
73
74  class NombreAleatorio extends JPanel {
75
76      JLabel nombre;
77
78      Random random = new Random();
79      int aleatorio1 = random.nextInt(100)+155;
80      int aleatorio2 = random.nextInt(100)+155;
81      int aleatorio3 = random.nextInt(100)+155;
82
83      int vocalNro1 = random.nextInt(9);
84      int vocalNro2 = random.nextInt(9);
85      int vocalNro3 = random.nextInt(9);
86      int vocalNro4 = random.nextInt(9);
87      int vocalNro5 = random.nextInt(9);
88      int vocalNro6 = random.nextInt(9);
89      int consonanteNro1 = random.nextInt(27);
90      int consonanteNro2 = random.nextInt(27);
91      int consonanteNro3 = random.nextInt(27);
92      int consonanteNro4 = random.nextInt(27);
93      int consonanteNro5 = random.nextInt(27);
94      int consonanteNro6 = random.nextInt(27);
95
96      String vocales[] = {"A","E","I","O","U","AA","EE","II","OO","UU"};
97      String consonantes[] =
    {"B","C","D","F","G","H","J","K","L","M","N","Ñ","P","Q","R","S","T","V","W","X","Y","Z","L
    L","BB","RR","MM","PP","SS",};
98
99      String vocalRandom1;
100     String vocalRandom2;
101     String vocalRandom3;
102     String vocalRandom4;
103     String vocalRandom5;
104     String vocalRandom6;
105     String consonanteRandom1;
106     String consonanteRandom2;
107     String consonanteRandom3;
108     String consonanteRandom4;
109     String consonanteRandom5;
110     String consonanteRandom6;
111
112     public NombreAleatorio() {
113
114         // -----------VOCALES-----------------------
115         for(int i=0; i<vocales.length; i++) {
116             if(i == vocalNro1) {
117                 vocalRandom1 = vocales[i];
118             }
119         }
120         for(int i=0; i<vocales.length; i++) {
121             if(i == vocalNro2) {
122                 vocalRandom2 = vocales[i];
```

```
123             }
124         }
125         for(int i=0; i<vocales.length; i++) {
126             if(i == vocalNro3) {
127                 vocalRandom3 = vocales[i];
128             }
129         }
130         for(int i=0; i<vocales.length; i++) {
131             if(i == vocalNro4) {
132                 vocalRandom4 = vocales[i];
133             }
134         }
135         for(int i=0; i<vocales.length; i++) {
136             if(i == vocalNro5) {
137                 vocalRandom5 = vocales[i];
138             }
139         }
140         for(int i=0; i<vocales.length; i++) {
141             if(i == vocalNro6) {
142                 vocalRandom6 = vocales[i];
143             }
144         }
145         // -----------CONSONANTES----------------------
146         for(int i=0; i<consonantes.length; i++) {
147             if(i == consonanteNro1) {
148                 consonanteRandom1 = consonantes[i];
149             }
150         }
151         for(int i=0; i<consonantes.length; i++) {
152             if(i == consonanteNro2) {
153                 consonanteRandom2 = consonantes[i];
154             }
155         }
156         for(int i=0; i<consonantes.length; i++) {
157             if(i == consonanteNro3) {
158                 consonanteRandom3 = consonantes[i];
159             }
160         }
161         for(int i=0; i<consonantes.length; i++) {
162             if(i == consonanteNro4) {
163                 consonanteRandom4 = consonantes[i];
164             }
165         }
166         for(int i=0; i<consonantes.length; i++) {
167             if(i == consonanteNro5) {
168                 consonanteRandom5 = consonantes[i];
169             }
170         }
171         for(int i=0; i<consonantes.length; i++) {
172             if(i == consonanteNro6) {
173                 consonanteRandom6 = consonantes[i];
174             }
175         }
176 
177         setBackground(new Color(aleatorio1, aleatorio2, aleatorio3));
178         System.out.println("El color RGB de la ventana nueva es: " + aleatorio1 + " " +
    aleatorio2 + " " + aleatorio3);
179         System.out.println("Letras aleatorias: " + vocalRandom1 + consonanteRandom1 +
    vocalRandom2 + consonanteRandom2 + vocalRandom3);
180 
181         setLayout(new GridBagLayout());
182 
```

```
183         add(nombre = new JLabel(vocalRandom1 + consonanteRandom1 + vocalRandom2 +
    consonanteRandom2 + vocalRandom3));
184         nombre.setFont(new Font("Roboto", Font.PLAIN, 24));
185     }
186 }
```

```java
package Graficos;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class _10_Layouts {

    public static void main(String[] args) {

        Ventana10 ventana = new Ventana10();
    }
}

class Ventana10 extends JFrame {

    public Ventana10() {

        setSize(600, 500);
        setTitle("  Layouts");
        setLocationRelativeTo(null);

        add(new Lamina10());

        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class Lamina10 extends JPanel {

    public Lamina10() {
        setLayout(new BorderLayout());
        setBackground(new Color(204, 255, 102));

        JPanel lamina1 = new JPanel();
        add(lamina1, BorderLayout.NORTH);
        lamina1.setLayout(new BorderLayout());

        lamina1.add(new JButton("Botón 1"), BorderLayout.NORTH);
        lamina1.add(new JButton("Botón 2"), BorderLayout.SOUTH);
        lamina1.add(new JButton("Botón 3"), BorderLayout.EAST);
        lamina1.add(new JButton("Botón 4"), BorderLayout.WEST);
        lamina1.add(new JButton("Botón 5"), BorderLayout.CENTER);

        JPanel lamina2 = new JPanel();
        add(lamina2, BorderLayout.SOUTH);
        lamina2.setLayout(new FlowLayout(FlowLayout.RIGHT));

        lamina2.add(new JButton("Botón 1"));
        lamina2.add(new JButton("Botón 2"));
        lamina2.add(new JButton("Botón 3"));
    }
}
```

```java
package Graficos;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.SwingConstants;

public class _11_Calculadora {

    public static void main(String[] args) {

        Ventana11 ventana = new Ventana11();
    }
}

class Ventana11 extends JFrame {

    public Ventana11() {
        setTitle(" Calculadora");
        setSize(300, 350);
        setLocationRelativeTo(null);

        add(new Lamina11());

        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class Lamina11 extends JPanel {

    JPanel lamina1, lamina2;
    JButton display, btn1, btn2, btn3, btn4, btn5, btn6, btn7, btn8, btn9, btn0, btnSum, btnRes, btnMul, btnDiv, btnClr, btnEqu;

    public Lamina11() {
        setLayout(new BorderLayout());

        add(lamina1 = new JPanel(), BorderLayout.NORTH);
        lamina1.setLayout(new BorderLayout());
        lamina1.add(display = new JButton("0"));
        display.setHorizontalAlignment(SwingConstants.RIGHT);
        display.setFont(new Font("Arial", Font.PLAIN, 24));
        display.setPreferredSize(new Dimension(300,80));
        display.setEnabled(false);

        add(lamina2 = new JPanel(), BorderLayout.CENTER);
        lamina2.setLayout(new GridLayout(4,4));
        lamina2.add(btn1 = new JButton("1"));
        lamina2.add(btn2 = new JButton("2"));
        lamina2.add(btn3 = new JButton("3"));
        lamina2.add(btnSum = new JButton("+"));
        lamina2.add(btn4 = new JButton("4"));
        lamina2.add(btn5 = new JButton("5"));
        lamina2.add(btn6 = new JButton("6"));
```

```
62          lamina2.add(btnRes = new JButton("-"));
63          lamina2.add(btn7 = new JButton("7"));
64          lamina2.add(btn8 = new JButton("8"));
65          lamina2.add(btn9 = new JButton("9"));
66          lamina2.add(btnMul = new JButton("*"));
67          lamina2.add(btnClr = new JButton("clr"));
68          lamina2.add(btn0 = new JButton("0"));
69          lamina2.add(btnEqu = new JButton("="));
70          lamina2.add(btnDiv = new JButton("/"));
71
72          OyenteNumeros oyenteNro = new OyenteNumeros();
73
74          btn1.addActionListener(oyenteNro);
75          btn2.addActionListener(oyenteNro);
76          btn3.addActionListener(oyenteNro);
77          btn4.addActionListener(oyenteNro);
78          btn5.addActionListener(oyenteNro);
79          btn6.addActionListener(oyenteNro);
80          btn7.addActionListener(oyenteNro);
81          btn8.addActionListener(oyenteNro);
82          btn9.addActionListener(oyenteNro);
83          btn0.addActionListener(oyenteNro);
84
85          OyenteOperandos oyenteOperando = new OyenteOperandos();
86
87          btnSum.addActionListener(oyenteOperando);
88          btnRes.addActionListener(oyenteOperando);
89          btnMul.addActionListener(oyenteOperando);
90          btnDiv.addActionListener(oyenteOperando);
91          btnClr.addActionListener(oyenteOperando);
92          btnEqu.addActionListener(oyenteOperando);
93
94          btn1.setBackground(new Color(230, 255, 204));
95          btn2.setBackground(new Color(230, 255, 204));
96          btn3.setBackground(new Color(230, 255, 204));
97          btn4.setBackground(new Color(230, 255, 204));
98          btn5.setBackground(new Color(230, 255, 204));
99          btn6.setBackground(new Color(230, 255, 204));
100         btn7.setBackground(new Color(230, 255, 204));
101         btn8.setBackground(new Color(230, 255, 204));
102         btn9.setBackground(new Color(230, 255, 204));
103         btn0.setBackground(new Color(230, 255, 204));
104
105         btnSum.setBackground(new Color(230, 204, 255));
106         btnRes.setBackground(new Color(230, 204, 255));
107         btnMul.setBackground(new Color(230, 204, 255));
108         btnDiv.setBackground(new Color(230, 204, 255));
109         btnClr.setBackground(new Color(230, 204, 255));
110         btnEqu.setBackground(new Color(230, 204, 255));
111     }
112
113     class OyenteNumeros implements ActionListener {
114         @Override
115         public void actionPerformed(ActionEvent e) {
116
117             JButton btnNro = (JButton) e.getSource();
118             double nro = Double.parseDouble(btnNro.getText());
119
120             if(display.getText() == "0") {
121                 display.setText("");
122                 display.setText(btnNro.getText());
123             } else {
```

```
124                 display.setText(display.getText() + btnNro.getText());
125             }
126         }
127     }
128
129     class OyenteOperandos implements ActionListener {
130
131         double resultado = 0;
132         String simbolo = "";
133
134         @Override
135         public void actionPerformed(ActionEvent e) {
136
137             JButton operandoBtn = (JButton) e.getSource();
138             String operando = operandoBtn.getText();
139
140             if(operando.equals("+")) {
141                 resultado = resultado + Double.parseDouble(display.getText());
142                 display.setText("0");
143                 simbolo = "+";
144             }
145             if(operando.equals("-")) {
146                 if(resultado == 0) {
147                     resultado = Double.parseDouble(display.getText());
148                     display.setText("0");
149                     simbolo = "-";
150                 } else {
151                     resultado = resultado - Double.parseDouble(display.getText());
152                     display.setText("0");
153                     simbolo = "-";
154                 }
155             }
156             if(operando.equals("*")) {
157                 if(resultado == 0) {
158                     resultado = (resultado+1) * Double.parseDouble(display.getText());
159                     display.setText("0");
160                     simbolo = "*";
161                 } else {
162                     resultado = resultado * Double.parseDouble(display.getText());
163                     display.setText("0");
164                     simbolo = "*";
165                 }
166             }
167             if(operando.equals("/")) {
168                 if(resultado == 0) {
169                     resultado = Double.parseDouble(display.getText());
170                     display.setText("0");
171                     simbolo = "/";
172                 } else {
173                     resultado = resultado / Double.parseDouble(display.getText());
174                     display.setText("0");
175                     simbolo = "/";
176                 }
177             }
178
179             if(operando.equals("=")) {
180                 if(simbolo.equals("+")) {
181                     resultado = resultado + Double.parseDouble(display.getText());
182                     display.setText(String.valueOf(resultado));
183                     resultado = 0;
184                 }
185                 if(simbolo.equals("-")) {
```

```
186                     resultado = resultado - Double.parseDouble(display.getText());
187                     display.setText(String.valueOf(resultado));
188                     resultado = 0;
189                 }
190                 if(simbolo.equals("*")) {
191                     resultado = resultado * Double.parseDouble(display.getText());
192                     display.setText(String.valueOf(resultado));
193                     resultado = 0;
194                 }
195                 if(simbolo.equals("/")) {
196                     resultado = resultado / Double.parseDouble(display.getText());
197                     display.setText(String.valueOf(resultado));
198                     resultado = 0;
199                 }
200             }
201 
202             if(operando.equals("clr")) {
203                 resultado = 0;
204                 display.setText("0");
205                 simbolo = " ";
206             }
207         }
208     }
209 }
```

```java
package Graficos;

import java.awt.Color;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class _12_PruebaTexto {

    public static void main(String[] args) {

        Marco12 ventana = new Marco12();
        ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class Marco12 extends JFrame {

    Color fondo = new Color(51, 204, 204);

    public Marco12() {

        setSize(600, 500);
        setTitle(" Prueba Texto");
        setLocationRelativeTo(null);
        getContentPane().setBackground(fondo);

        setLayout(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets.set(0, 0, 60, 0);

        add(new Ventana12(), gbc);

        setVisible(true);
    }

    class Ventana12 extends JPanel {

        JTextField campoTexto;
        JLabel aviso;
        JButton btn;

        public Ventana12() {

            setBackground(fondo);

            setLayout(new GridLayout(7,1));

            add(new JLabel("Ingrese su eMail:"));
            add(new JLabel(""));
            add(campoTexto = new JTextField(30));
            add(new JLabel(""));
            add(aviso = new JLabel(""));
            add(new JLabel(""));
            add(btn = new JButton("Enviar"));
```

```
63
64              btn.addActionListener(new ClaseOyente());
65          }
66
67          class ClaseOyente implements ActionListener {
68
69              @Override
70              public void actionPerformed(ActionEvent e) {
71
72                  int arroba = 0;
73                  boolean punto = false;
74
75                  for (int i = 0; i < campoTexto.getText().length(); i++) {
76                      if(campoTexto.getText().charAt(i) == '@') {
77                          arroba++;
78                      }
79                      if(campoTexto.getText().charAt(i) == '.') {
80                          punto = true;
81                      }
82                  }
83                  if(arroba == 1 && punto) {
84                      aviso.setForeground(new Color(102, 153, 0));
85                      aviso.setText("eMail correcto, Bienvenido!");
86                      arroba = 0;
87                  } else {
88                      aviso.setForeground(new Color(255, 0, 102));
89                      aviso.setText("eMail incorrecto, corrija e intente nuevamente");
90                      arroba = 0;
91                  }
92              }
93          }
94      }
95  }
```

```java
package Graficos;

import java.awt.Color;
import java.awt.GridBagLayout;
import java.awt.GridLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JTextField;
import javax.swing.event.DocumentEvent;
import javax.swing.event.DocumentListener;

public class _13_CampoPassword {

    public static void main(String[] args) {

        Marco13 ventana = new Marco13();
        ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class Marco13 extends JFrame {

    public Marco13() {

        setSize(550, 600);
        setTitle(" Campo Password");
        setLocationRelativeTo(null);

        add(new Ventana13());

        setVisible(true);
    }
}

class Ventana13 extends JPanel {

    JLabel emailTxt, passTxt, avisoMail, avisoPass;
    JTextField emailField;
    JPasswordField passField;
    JButton enviarBtn;
    JPanel lamina;
    Color fondo = new Color(204, 255, 153);

    public Ventana13() {

        setBackground(fondo);

        setLayout(new GridBagLayout());

        add(lamina = new JPanel());
        lamina.setBackground(fondo);
        lamina.setLayout(new GridLayout(10,1));

        lamina.add(emailTxt = new JLabel("Email:"));
        lamina.add(emailField = new JTextField(40));
        lamina.add(avisoMail = new JLabel(" "));
        lamina.add(new JLabel(" "));
        lamina.add(passTxt = new JLabel("Contraseña:"));
        lamina.add(passField = new JPasswordField(40));
```

```
63          lamina.add(avisoPass = new JLabel(" "));
64          lamina.add(new JLabel(" "));
65          lamina.add(enviarBtn = new JButton("Enviar"));
66
67          emailField.getDocument().addDocumentListener(new OyenteEmail());
68          passField.getDocument().addDocumentListener(new OyentePassword());
69      }
70
71      // ----------------------OYENTE EMAIL-----------------------------------------
72      class OyenteEmail implements DocumentListener {
73          @Override
74          public void insertUpdate(DocumentEvent e) {
75              String email = emailField.getText();
76              int arroba = 0;
77              boolean punto = false;
78
79              for (int i = 0; i < email.length(); i++) {
80                  if (email.charAt(i) == '@') {
81                      arroba++;
82                  }
83                  if (email.charAt(i) == '.') {
84                      punto = true;
85                  }
86              }
87              if (email.length() > 8) {
88                  if (arroba == 1 && punto) {
89                      emailField.setBackground(Color.WHITE);
90                      avisoMail.setText(" ");
91                      arroba = 0;
92                  } else {
93                      emailField.setBackground(new Color(255, 102, 153));
94                      avisoMail.setForeground(new Color(255, 102, 153));
95                      avisoMail.setText("eMail incorrecto, corrija!");
96                  }
97              }
98          }
99          @Override
100         public void removeUpdate(DocumentEvent e) {
101             String email = emailField.getText();
102             int arroba = 0;
103             boolean punto = false;
104
105             for (int i = 0; i < email.length(); i++) {
106                 if (email.charAt(i) == '@') {
107                     arroba++;
108                 }
109                 if (email.charAt(i) == '.') {
110                     punto = true;
111                 }
112             }
113             if (email.length() > 8) {
114                 if (arroba == 1 && punto) {
115                     emailField.setBackground(Color.WHITE);
116                     avisoMail.setText(" ");
117                     arroba = 0;
118                 } else {
119                     emailField.setBackground(new Color(255, 102, 153));
120                     avisoMail.setForeground(new Color(255, 102, 153));
121                     avisoMail.setText("eMail incorrecto, corrija!");
122                 }
123             }
124         }
```

```
125         @Override
126         public void changedUpdate(DocumentEvent e) {
127             String email = emailField.getText();
128             int arroba = 0;
129             boolean punto = false;
130 
131             for (int i = 0; i < email.length(); i++) {
132                 if (email.charAt(i) == '@') {
133                     arroba++;
134                 }
135                 if (email.charAt(i) == '.') {
136                     punto = true;
137                 }
138             }
139             if (email.length() > 8) {
140                 if (arroba == 1 && punto) {
141                     emailField.setBackground(Color.WHITE);
142                     avisoMail.setText(" ");
143                     arroba = 0;
144                 } else {
145                     emailField.setBackground(new Color(255, 102, 153));
146                     avisoMail.setForeground(new Color(255, 102, 153));
147                     avisoMail.setText("eMail incorrecto, corrija!");
148                 }
149             }
150         }
151     }
152 
153     // ----------------------OYENTE
    PASSWORDFIELD-----------------------------------------
154     class OyentePassword implements DocumentListener {
155         @Override
156         public void insertUpdate(DocumentEvent e) {
157             char password[] = passField.getPassword();
158             for (int i = 0; i < password.length; i++) {
159                 if(password.length < 6 || password.length > 12) {
160                     passField.setBackground(new Color(255, 102, 153));
161                     avisoPass.setForeground(new Color(255, 102, 153));
162                     avisoPass.setText("La contraseña debe tener mínimo 6 y máximo 12
    letras!");
163                 } else {
164                     passField.setBackground(Color.WHITE);
165                     avisoPass.setText(" ");
166                 }
167             }
168         }
169         @Override
170         public void removeUpdate(DocumentEvent e) {
171             char password[] = passField.getPassword();
172             for (int i = 0; i < password.length; i++) {
173                 if(password.length < 6 || password.length > 12) {
174                     passField.setBackground(new Color(255, 102, 153));
175                     avisoPass.setForeground(new Color(255, 102, 153));
176                     avisoPass.setText("La contraseña debe tener mínimo 6 y máximo 12
    letras!");
177                 } else {
178                     passField.setBackground(Color.WHITE);
179                     avisoPass.setText(" ");
180                 }
181             }
182         }
183         @Override
```

```
184         public void changedUpdate(DocumentEvent e) {
185             char password[] = passField.getPassword();
186             for (int i = 0; i < password.length; i++) {
187                 if(password.length < 6 || password.length > 12) {
188                     passField.setBackground(new Color(255, 102, 153));
189                     avisoPass.setForeground(new Color(255, 102, 153));
190                     avisoPass.setText("La contraseña debe tener mínimo 6 y máximo 12
    letras!");
191                 } else {
192                     passField.setBackground(Color.WHITE);
193                     avisoPass.setText(" ");
194                 }
195             }
196         }
197     }
198 }
```

```java
package Graficos;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

public class _14_EjemploArea {

    public static void main(String[] args) {

        Marco14 ventana = new Marco14();
        ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class Marco14 extends JFrame {

    public Marco14() {

        setSize(350, 500);
        setTitle(" Ejemplo Área");
        setLocationRelativeTo(null);
        getContentPane().setBackground(new Color(153, 204, 255));

        setLayout(new BorderLayout());

        // -----------Espacios en los Bordes-----------------
        add(new JLabel("         "), BorderLayout.NORTH);
        add(new JLabel("         "), BorderLayout.SOUTH);
        add(new JLabel("         "), BorderLayout.EAST);
        add(new JLabel("         "), BorderLayout.WEST);
        // -----------Espacios en los Bordes-----------------

        add(new Ventana14(), BorderLayout.CENTER);

        setVisible(true);
    }
}

class Ventana14 extends JPanel {

    JPanel lamina;
    JTextArea areaIn, areaOut;
    JLabel textoOut;
    JButton btnEnviar;
    JScrollPane vistaScroll;

    public Ventana14() {

        setBackground(new Color(153, 204, 255));

        setLayout(new FlowLayout(FlowLayout.LEFT, 20,20));

        add(new JLabel(" "));
```

```
63          add(new JLabel(" "));
64          add(areaIn = new JTextArea(8,20));
65          areaIn.setText(lorem);
66          areaIn.setLineWrap(true);
67          add(vistaScroll = new JScrollPane(areaIn));
68          add(btnEnviar = new JButton("Enviar"));
69          add(new JLabel(" "));
70          add(new JLabel(" "));
71          add(areaOut = new JTextArea(8,20));
72          areaOut.setLineWrap(true);
73          areaOut.setEnabled(false);
74          areaOut.setDisabledTextColor(Color.DARK_GRAY);
75          areaOut.setBackground(new Color(179, 204, 230));
76          add(vistaScroll = new JScrollPane(areaOut));
77          add(textoOut = new JLabel(" "));
78
79          btnEnviar.addActionListener(new ClaseOyente());
80      }
81
82      class ClaseOyente implements ActionListener {
83
84          @Override
85          public void actionPerformed(ActionEvent e) {
86
87              String entrada = areaIn.getText();
88              areaOut.setText(entrada);
89
90              textoOut.setText("Data enviada!");
91          }
92      }
93
94      String lorem = "There are many variations of passages of Lorem Ipsum available, but the
    majority have suffered alteration in some form, by injected humour, or randomised words
    which don't look even slightly believable. If you are going to use a passage of Lorem Ipsum,
    you need to be sure there isn't anything embarrassing hidden in the middle of text. All the
    Lorem Ipsum generators on the Internet tend to repeat predefined chunks as necessary, making
    this the first true generator on the Internet. It uses a dictionary of over 200 Latin words,
    combined with a handful of model sentence structures, to generate Lorem Ipsum which looks
    reasonable. The generated Lorem Ipsum is therefore always free from repetition, injected
    humour, or non-characteristic words etc.";
95  }
```

```java
package Graficos;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

public class _15_PruebaArea {

    public static void main(String[] args) {

        Marco15 ventana = new Marco15();
        ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class Marco15 extends JFrame {

    public Marco15() {

        setSize(400, 300);
        setTitle(" Prueba Área");
        setLocationRelativeTo(null);

        add(new Ventana15());

        setVisible(true);
    }
}

class Ventana15 extends JPanel {

    JTextArea areaTxt;
    JScrollPane vistaScroll;
    JPanel lamina;
    JButton btn1, btn2;
    String lorem = "Al contrario del pensamiento popular, el texto de Lorem Ipsum no es simplemente texto aleatorio. | Al contrario del pensamiento popular, el texto de Lorem Ipsum no es simplemente texto aleatorio. | Al contrario del pensamiento popular, el texto de Lorem Ipsum no es simplemente texto aleatorio. | Al contrario del pensamiento popular, el texto de Lorem Ipsum no es simplemente texto aleatorio. | ";

    public Ventana15() {

        setLayout(new BorderLayout());

        add(areaTxt = new JTextArea(), BorderLayout.CENTER);
        areaTxt.setBackground(new Color(204, 255, 255));
        areaTxt.setForeground(new Color(153, 0, 115));
        add(vistaScroll = new JScrollPane(areaTxt));

        add(lamina = new JPanel(), BorderLayout.SOUTH);
        lamina.setBackground(new Color(204, 204, 255));

        lamina.add(btn1 = new JButton("Agregar texto"));
        lamina.add(btn2 = new JButton("Insertar salto de línea"));

```

```
59          btn1.setBackground(new Color(204, 204, 255));
60          btn2.setBackground(new Color(204, 204, 255));
61
62          btn1.addActionListener(new ClaseOyente());
63          btn2.addActionListener(new ClaseOyente());
64      }
65
66      class ClaseOyente implements ActionListener {
67
68          @Override
69          public void actionPerformed(ActionEvent e) {
70
71              if (e.getSource() == btn1) {
72                  areaTxt.append(lorem);
73              }
74              if (e.getSource() == btn2) {
75                  if (areaTxt.getLineWrap() == false) {
76                      areaTxt.setLineWrap(true);
77                      btn2.setText("Quitar salto de línea");
78                      btn2.setBackground(new Color(255, 153, 204));
79                  } else {
80                      areaTxt.setLineWrap(false);
81                      btn2.setText("Insertar salto de línea");
82                      btn2.setBackground(new Color(153, 255, 204));
83                  }
84              }
85          }
86      }
87  }
```

```java
package Graficos;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JCheckBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.SwingConstants;

public class _16_PruebaChecks {

    public static void main(String[] args) {

        Marco16 ventana = new Marco16();
        ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class Marco16 extends JFrame {

    JLabel east;
    Color fondo = new Color(179, 255, 179);

    public Marco16() {

        setSize(900, 400);
        setTitle(" Prueba Checks");
        setLocationRelativeTo(null);

        setLayout(new BorderLayout());

        // -----------Espacios en los Bordes-----------------
        add(new JLabel("         "), BorderLayout.NORTH);
        add(new JLabel("         "), BorderLayout.SOUTH);
        add(east = new JLabel("   ......................      "), BorderLayout.EAST);
        east.setForeground(new Color(0, 0, 0, 0));
        add(new JLabel("         "), BorderLayout.WEST);
        // -----------Espacios en los Bordes-----------------

        add(new Ventana16(fondo), BorderLayout.CENTER);

        getContentPane().setBackground(fondo);

        setVisible(true);
    }
}

class Ventana16 extends JPanel {

    JLabel texto1, texto2;
    JCheckBox check1, check2;
    JPanel lamina1, lamina2;

    public Ventana16(Color fondo) {

        setLayout(new BorderLayout());

```

```
63          add(lamina1 = new JPanel(), BorderLayout.CENTER);
64          lamina1.setLayout(new GridLayout(7,1));
65          lamina1.setBackground(fondo);
66
67          lamina1.add(new JLabel("    "));
68          lamina1.add(new JLabel("    "));
69          lamina1.add(new JLabel("    "));
70          lamina1.add(texto1 = new JLabel("\"Es capaz el que piensa que es capaz.\""));
71          texto1.setFont(new Font("Verdana", Font.PLAIN, 34));
72          texto1.setHorizontalAlignment(SwingConstants.RIGHT);
73          lamina1.add(new JLabel("    "));
74          lamina1.add(texto2 = new JLabel("Buda"));
75          texto2.setFont(new Font("Verdana", Font.PLAIN, 24));
76          texto2.setHorizontalAlignment(SwingConstants.RIGHT);
77
78          add(lamina2 = new JPanel(), BorderLayout.SOUTH);
79          lamina2.setBackground(fondo);
80
81          lamina2.add(check1 = new JCheckBox("Negrita", false));
82          check1.setFont(new Font("Verdana", Font.PLAIN, 18));
83          check1.setBackground(fondo);
84          lamina2.add(check2 = new JCheckBox("Cursiva", false));
85          check2.setFont(new Font("Verdana", Font.PLAIN, 18));
86          check2.setBackground(fondo);
87
88          check1.addActionListener(new ClaseOyente());
89          check2.addActionListener(new ClaseOyente());
90      }
91
92      class ClaseOyente implements ActionListener {
93
94          @Override
95          public void actionPerformed(ActionEvent e) {
96
97              int constante = 0;
98
99              if (check1.isSelected() && !check2.isSelected()) {
100                 constante = 1;
101             } else if (!check1.isSelected() && check2.isSelected()) {
102                 constante = 2;
103             } else if (check1.isSelected() && check2.isSelected()) {
104                 constante = 3;
105             } else {
106                 constante = 0;
107             }
108
109             texto1.setFont(new Font("Verdana", constante, 34));
110         }
111     }
112 }
```

_17_SintaxisRadio_EjemploRadio_PruebaCombo_MarcoSlider_MarcoSpinner.java

```java
package Graficos;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.GraphicsEnvironment;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.ButtonGroup;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JRadioButton;
import javax.swing.JSlider;
import javax.swing.JSpinner;
import javax.swing.SpinnerListModel;
import javax.swing.SpinnerNumberModel;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;

public class _17_SintaxisRadio_EjemploRadio_PruebaCombo_MarcoSlider_MarcoSpinner {

    public static void main(String[] args) {

        Marco17 ventana = new Marco17();
        ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class Marco17 extends JFrame {

    Color fondo1 = new Color(179, 255, 224);

    public Marco17() {

        setSize(1000, 900);
        setTitle(" Ejercicio 3.17");
        setLocationRelativeTo(null);
        getContentPane().setBackground(fondo1);

        setLayout(new BorderLayout());

        // -----------Espacios en los Bordes-----------------
        add(new JLabel("         "), BorderLayout.NORTH);
        add(new JLabel("         "), BorderLayout.SOUTH);
        add(new JLabel("         "), BorderLayout.EAST);
        add(new JLabel("         "), BorderLayout.WEST);
        // -----------Espacios en los Bordes-----------------

        add(new Ventana17(fondo1), BorderLayout.CENTER);

        setVisible(true);
    }
}

class Ventana17 extends JPanel {

    JPanel lamina1, lamina2, laminaGrid, laminaGroup1, laminaGroup2, laminaSeparator,
  laminaSpinners;
```

```
62      JLabel texto1, texto2;
63      JRadioButton radioPequeño, radioMediano, radioGrande, radioMuyGrande, radioNegrita,
    radioCursiva, radioNormal;
64      JComboBox comboFuentes;
65      JSlider slider;
66      JSpinner spinnerFuente, spinnerTamanio;
67      String fuentes[] =
    GraphicsEnvironment.getLocalGraphicsEnvironment().getAvailableFontFamilyNames();
68      String fuenteTipo = "Arial";
69      int fuenteFormat = 0;
70      int fuenteSize = 30;
71
72      public Ventana17(Color fondo1) {
73
74          setBackground(fondo1);
75          add(laminaGrid = new JPanel());
76          laminaGrid.setLayout(new GridLayout(2,1));
77
78          // -------------LÁMINA 1-----------------------------------------
79          laminaGrid.add(lamina1 = new JPanel());
80          lamina1.setPreferredSize(new Dimension(800,200));
81          lamina1.setLayout(new GridLayout(2,1));
82          lamina1.setBackground(fondo1);
83
84          // -------------TEXTO PRINCIPAL-----------------------------------------
85          lamina1.add(texto1 = new JLabel("\"Todo es un pensamiento.\""));
86          texto1.setFont(new Font(fuenteTipo, fuenteFormat, fuenteSize));
87          texto1.setHorizontalAlignment(JLabel.CENTER);
88          lamina1.add(texto2 = new JLabel("Buda                "));
89          texto2.setFont(new Font("Arial", Font.PLAIN, 22));
90          texto2.setHorizontalAlignment(JLabel.RIGHT);
91
92          // -------------LÁMINA 2-----------------------------------------
93          laminaGrid.add(lamina2 = new JPanel());
94          lamina2.setLayout(new GridLayout(9,1));
95
96          // -------------GRUPO 1 RADIOS-----------------------------------------
97          lamina2.add(laminaGroup1 = new JPanel());
98          laminaGroup1.add(radioPequeño = new JRadioButton("Pequeña", false));
99          laminaGroup1.add(radioMediano = new JRadioButton("Mediana", true));
100         laminaGroup1.add(radioGrande = new JRadioButton("Grande", false));
101         laminaGroup1.add(radioMuyGrande = new JRadioButton("Muy Grande", false));
102
103         ButtonGroup grupo1 = new ButtonGroup();
104         grupo1.add(radioPequeño);
105         grupo1.add(radioMediano);
106         grupo1.add(radioGrande);
107         grupo1.add(radioMuyGrande);
108
109         lamina2.add(laminaSeparator = new JPanel());
110         laminaSeparator.add(new JLabel("                  "));
111         laminaSeparator.add(new JLabel("                  "));
112
113         // -------------GRUPO 2 RADIOS-----------------------------------------
114         lamina2.add(laminaGroup2 = new JPanel());
115         laminaGroup2.add(radioNegrita = new JRadioButton("Negrita", false));
116         laminaGroup2.add(radioCursiva = new JRadioButton("Cursiva", false));
117         laminaGroup2.add(radioNormal = new JRadioButton("Normal", true));
118
119         ButtonGroup grupo2 = new ButtonGroup();
120         grupo2.add(radioNegrita);
121         grupo2.add(radioCursiva);
```

```
122         grupo2.add(radioNormal);
123 
124         lamina2.add(laminaSeparator = new JPanel());
125         laminaSeparator.add(new JLabel("                 "));
126 
127         // -------------SPINNERS-----------------------------------------
128         lamina2.add(laminaSpinners = new JPanel());
129         laminaSpinners.add(spinnerFuente = new JSpinner(new SpinnerListModel(fuentes)));
130         spinnerFuente.setPreferredSize(new Dimension(150,30));
131 
132         laminaSpinners.add(spinnerTamanio = new JSpinner(new SpinnerNumberModel(30, 10, 60,
    2)));
133         spinnerTamanio.setPreferredSize(new Dimension(50,30));
134 
135         lamina2.add(laminaSeparator = new JPanel());
136         laminaSeparator.add(new JLabel("                 "));
137 
138         // -------------SLIDER-------------------------------------------
139         lamina2.add(slider = new JSlider(10, 60, 30));
140         slider.setMajorTickSpacing(5);
141         slider.setMinorTickSpacing(1);
142         slider.setPaintLabels(true);
143         slider.setPaintTicks(true);
144 
145         lamina2.add(laminaSeparator = new JPanel());
146         laminaSeparator.add(new JLabel("                 "));
147 
148         // -------------COMBOBOX-----------------------------------------
149         lamina2.add(comboFuentes = new JComboBox());
150         for (int i = 0; i < fuentes.length; i++) {
151             comboFuentes.addItem(fuentes[i]);
152         }
153 
154         // -------------OYENTES------------------------------------------
155         ActionOyente oyenteAction = new ActionOyente();
156 
157         radioPequeño.addActionListener(oyenteAction);
158         radioMediano.addActionListener(oyenteAction);
159         radioGrande.addActionListener(oyenteAction);
160         radioMuyGrande.addActionListener(oyenteAction);
161         radioNegrita.addActionListener(oyenteAction);
162         radioCursiva.addActionListener(oyenteAction);
163         radioNormal.addActionListener(oyenteAction);
164         comboFuentes.addActionListener(oyenteAction);
165 
166         ChangeOyente oyenteChange = new ChangeOyente();
167 
168         slider.addChangeListener(oyenteChange);
169         spinnerFuente.addChangeListener(oyenteChange);
170         spinnerTamanio.addChangeListener(oyenteChange);
171     }
172 
173     // -------------OYENTES ACTION-----------------------------------------
174     class ActionOyente implements ActionListener {
175 
176         public ActionOyente() {
177 
178         }
179         @Override
180         public void actionPerformed(ActionEvent e) {
181 
182             if (e.getSource() == comboFuentes) fuenteTipo =
```

```
    (String)comboFuentes.getSelectedItem();
183
184             if (radioNegrita.isSelected()) fuenteFormat = 1;
185             if (radioCursiva.isSelected()) fuenteFormat = 2;
186             if (radioNormal.isSelected()) fuenteFormat = 0;
187
188             if (radioPequeño.isSelected()) fuenteSize = 20;
189             if (radioMediano.isSelected()) fuenteSize = 30;
190             if (radioGrande.isSelected()) fuenteSize = 40;
191             if (radioMuyGrande.isSelected()) fuenteSize = 50;
192
193             texto1.setFont(new Font(fuenteTipo, fuenteFormat, fuenteSize));
194         }
195     }
196
197     // -------------OYENTES CHANGE-----------------------------------------
198     class ChangeOyente implements ChangeListener {
199
200         @Override
201         public void stateChanged(ChangeEvent e) {
202
203             if (e.getSource() == slider) fuenteSize = (int) slider.getValue();
204             if (e.getSource() == spinnerTamanio) fuenteSize = (int)
    spinnerTamanio.getValue();
205             if (e.getSource() == spinnerFuente) fuenteTipo = (String)
    spinnerFuente.getValue();
206
207             texto1.setFont(new Font(fuenteTipo, fuenteFormat, fuenteSize));
208         }
209     }
210 }
```

```java
package Graficos;

import java.awt.BorderLayout;
import java.awt.Color;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JPanel;
import javax.swing.JSeparator;

public class _18_MarcoMenu {

    public static void main(String[] args) {

        Marco18 ventana = new Marco18();
        ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class Marco18 extends JFrame {

    Color fondo = new Color(230, 204, 255);

    public Marco18() {

        setSize(600, 500);
        setTitle(" Marco Menú");
        setLocationRelativeTo(null);
        getContentPane().setBackground(fondo);

        setLayout(new BorderLayout());

        // -----------Espacios en los Bordes-----------------
        add(new JLabel("         "), BorderLayout.NORTH);
        add(new JLabel("         "), BorderLayout.SOUTH);
        add(new JLabel("         "), BorderLayout.EAST);
        add(new JLabel("         "), BorderLayout.WEST);
        // -----------Espacios en los Bordes-----------------

        add(new Ventana18(fondo), BorderLayout.CENTER);

        setVisible(true);
    }
}

class Ventana18 extends JPanel {

    JMenuBar barra;
    JMenu menu1, menu2, menu3, menu4, menu5;
    JMenuItem item1, item2, item3, item4, item5, item6, item7, item8, item9, item10;
    JSeparator separador;

    public Ventana18(Color fondo) {

        setBackground(fondo);

        add(barra = new JMenuBar());

        barra.add(menu1 = new JMenu("Inicio"));
        barra.add(menu2 = new JMenu("Servicios"));
```

```
63          barra.add(menu3 = new JMenu("Galería"));
64          barra.add(menu4 = new JMenu("Contacto"));
65
66              menu1.add(item1 = new JMenuItem("Ofertas"));
67              menu1.add(item2 = new JMenuItem("Productos"));
68              menu1.addSeparator();
69              menu1.add(menu5 = new JMenu("Novedades"));
70                  menu5.add(item10 = new JMenuItem("Año 2020"));
71
72              menu2.add(item4 = new JMenuItem("Básicos"));
73              menu2.add(item5 = new JMenuItem("Exclusivos"));
74              menu2.add(item6 = new JMenuItem("Outlet"));
75
76              menu3.add(item7 = new JMenuItem("Clientes"));
77              menu3.add(item8 = new JMenuItem("Eventos"));
78
79              menu4.add(item9 = new JMenuItem("Reservas"));
80      }
81  }
```

```java
package EjerciciosBloque3;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Font;
import java.awt.event.InputEvent;
import java.awt.event.KeyEvent;
import javax.swing.ButtonGroup;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JPanel;
import javax.swing.JPopupMenu;
import javax.swing.JRadioButtonMenuItem;
import javax.swing.JScrollPane;
import javax.swing.JTextPane;
import javax.swing.JToolBar;
import javax.swing.KeyStroke;
import javax.swing.text.StyledEditorKit;

public class _01_ProcesadorDeTextosFull {

    public static void main(String[] args) {

        MarcoProcesadorTexto miMarco = new MarcoProcesadorTexto();
    }
}

class MarcoProcesadorTexto extends JFrame {

    public MarcoProcesadorTexto() {

        setSize(700, 500);
        setTitle("Procesador de Textos");
        setLocationRelativeTo(null);

        add(new LaminaProcesadorTexto());

        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class LaminaProcesadorTexto extends JPanel {

    JPanel laminaSup;
    private JMenuBar barraMenu;
    private JMenu menuFuente, menuEstilo, menuTamanio;
    private JMenuItem fuenteArial, fuenteCourier, fuenteVerdana, estiloNegrita,
  estiloCursiva, estiloSubrayado;
    private JRadioButtonMenuItem radio12, radio16, radio20, radio24;
    private ButtonGroup radioGrupo = new ButtonGroup();
    private JTextPane textPane;
    private String lorem = "At vero eos et accusamus \nEt iusto odio dignissimos ducimus
  qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas
  molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui
  officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum
  facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio
  cumque nihil impedit quo minus id quod maxime placeat facere possimus, omnis voluptas
```

```
    assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis
    aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non
    recusandae. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis
    voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat. \n \n
    At vero eos et accusamus \nEt iusto odio dignissimos ducimus qui blanditiis praesentium
    voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati
    cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id
    est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam
    libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod
    maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus.
    Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet
    ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum hic
    tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut
    perferendis doloribus asperiores repellat. \n \n";
57      private JScrollPane scrollPanel;
58      private JPopupMenu popupMenu;
59      private JToolBar toolBar;
60      private JButton toolBold, toolItalic, toolUnderline, toolBlue, toolGreen, toolRed,
    toolLeft, toolCenter, toolRight;
61
62      public LaminaProcesadorTexto() {
63
64          setLayout(new BorderLayout());
65
66          add(laminaSup = new JPanel(), BorderLayout.NORTH);
67
68          // ------------------------BARRA MENU---------------------------------
69          laminaSup.add(barraMenu = new JMenuBar());
70          barraMenu.add(menuFuente = new JMenu("Fuente"));
71          barraMenu.add(menuEstilo = new JMenu("Estilo"));
72          barraMenu.add(menuTamanio = new JMenu("Tamaño"));
73
74          MenuItems items = new MenuItems();
75
76          // ------------------------ITEMS FUENTE---------------------------------
77          menuFuente.add(items.fuenteArial());
78          menuFuente.add(items.fuenteCourier());
79          menuFuente.add(items.fuenteVerdana());
80
81          // ------------------------ITEMS ESTILO---------------------------------
82          menuEstilo.add(items.estiloNegrita());
83          menuEstilo.add(items.estiloCursiva());
84          menuEstilo.add(items.estiloSubrayado());
85
86          // ------------------------ITEMS TAMAÑO---------------------------------
87          menuTamanio.add(items.radio12());
88          menuTamanio.add(items.radio16());
89          menuTamanio.add(items.radio20());
90          menuTamanio.add(items.radio24());
91          radioGrupo.add(radio12);
92          radioGrupo.add(radio16);
93          radioGrupo.add(radio20);
94          radioGrupo.add(radio24);
95
96          // ------------------------TEXT PANE---------------------------------
97          add(textPane = new JTextPane());
98          textPane.setText(lorem);
99          textPane.setFont(new Font("Times New Roman", Font.PLAIN, 16));
100         add(scrollPanel = new JScrollPane(textPane));
101
102         ClaseOyentes oyentes = new ClaseOyentes();
103
```

```
104         // ------------------------POPUP MENU---------------------------------
105         popupMenu = new JPopupMenu();
106 
107         popupMenu.add(items.fuenteArial());
108         popupMenu.add(items.fuenteCourier());
109         popupMenu.add(items.fuenteVerdana());
110 
111         popupMenu.addSeparator();
112         popupMenu.add(items.estiloNegrita());
113         popupMenu.add(items.estiloCursiva());
114         popupMenu.add(items.estiloSubrayado());
115 
116         popupMenu.addSeparator();
117         popupMenu.add(items.radio12());
118         popupMenu.add(items.radio16());
119         popupMenu.add(items.radio20());
120         popupMenu.add(items.radio24());
121         radioGrupo.add(radio12);
122         radioGrupo.add(radio16);
123         radioGrupo.add(radio20);
124         radioGrupo.add(radio24);
125 
126         textPane.setComponentPopupMenu(popupMenu);
127 
128         add(BarraDeHerramientas(), BorderLayout.WEST);
129 
130         ClaseOyentes oyentes2 = new ClaseOyentes();
131     }
132 
133     // ------------------------CLASE OYENTES-----------------------------------
134     private class ClaseOyentes {
135 
136         public ClaseOyentes () {
137             fuenteArial.addActionListener(new StyledEditorKit.FontFamilyAction(null,
    "Arial"));
138             fuenteCourier.addActionListener(new StyledEditorKit.FontFamilyAction(null,
    "Courier"));
139             fuenteVerdana.addActionListener(new StyledEditorKit.FontFamilyAction(null,
    "Verdana"));
140 
141             estiloNegrita.addActionListener(new StyledEditorKit.BoldAction());
142             estiloCursiva.addActionListener(new StyledEditorKit.ItalicAction());
143             estiloSubrayado.addActionListener(new StyledEditorKit.UnderlineAction());
144 
145             estiloNegrita.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_N,
    InputEvent.CTRL_DOWN_MASK));
146             estiloCursiva.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_K,
    InputEvent.CTRL_DOWN_MASK));
147             estiloSubrayado.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_S,
    InputEvent.CTRL_DOWN_MASK));
148 
149             radio12.addActionListener(new StyledEditorKit.FontSizeAction(null, 12));
150             radio16.addActionListener(new StyledEditorKit.FontSizeAction(null, 16));
151             radio20.addActionListener(new StyledEditorKit.FontSizeAction(null, 20));
152             radio24.addActionListener(new StyledEditorKit.FontSizeAction(null, 24));
153         }
154     }
155 
156     // ---------------------------------------------CLASE
    ITEMS-------------------------------------------
157     private class MenuItems {
158 
```

```
159         public JMenuItem fuenteArial() {return fuenteArial = new JMenuItem("Arial", new
    ImageIcon("src/images/tipo.png"));}
160         public JMenuItem fuenteCourier() {return fuenteCourier = new JMenuItem("Courier",
    new ImageIcon("src/images/tipo.png"));}
161         public JMenuItem fuenteVerdana() {return fuenteVerdana = new JMenuItem("Verdana",
    new ImageIcon("src/images/tipo.png"));}
162
163         public JMenuItem estiloNegrita() {return estiloNegrita = new JMenuItem("Negrita",
    new ImageIcon("src/images/style.png"));}
164         public JMenuItem estiloCursiva() {return estiloCursiva = new JMenuItem("Cursiva",
    new ImageIcon("src/images/style.png"));}
165         public JMenuItem estiloSubrayado() {return estiloSubrayado = new
    JMenuItem("Subrayado", new ImageIcon("src/images/style.png"));}
166
167         public JRadioButtonMenuItem radio12() {return radio12 = new
    JRadioButtonMenuItem("12", new ImageIcon("src/images/fontSize.png"));}
168         public JRadioButtonMenuItem radio16() {return radio16 = new
    JRadioButtonMenuItem("16", new ImageIcon("src/images/fontSize.png"));}
169         public JRadioButtonMenuItem radio20() {return radio20 = new
    JRadioButtonMenuItem("20", new ImageIcon("src/images/fontSize.png"));}
170         public JRadioButtonMenuItem radio24() {return radio24 = new
    JRadioButtonMenuItem("24", new ImageIcon("src/images/fontSize.png"));}
171     }
172
173     //
    ----------------------------------------------TOOLBAR-----------------------------------------
    -------
174     private JToolBar BarraDeHerramientas() {
175
176         toolBar = new JToolBar("ToolBar");
177         toolBar.setOrientation(1);
178
179         toolBar.addSeparator();
180         toolBar.add(toolBold = new JButton(new ImageIcon("src/images/bold.png")));
181         toolBar.add(toolItalic = new JButton(new ImageIcon("src/images/italic.png")));
182         toolBar.add(toolUnderline = new JButton(new
    ImageIcon("src/images/underline.png")));
183         toolBar.addSeparator();
184         toolBar.add(toolBlue = new JButton(new ImageIcon("src/images/azul.png")));
185         toolBar.add(toolGreen = new JButton(new ImageIcon("src/images/verde.png")));
186         toolBar.add(toolRed = new JButton(new ImageIcon("src/images/rojo.png")));
187         toolBar.addSeparator();
188         toolBar.add(toolLeft = new JButton(new ImageIcon("src/images/izqda.png")));
189         toolBar.add(toolCenter = new JButton(new ImageIcon("src/images/centro.png")));
190         toolBar.add(toolRight = new JButton(new ImageIcon("src/images/dcha.png")));
191
192         toolBold.addActionListener(new StyledEditorKit.BoldAction());
193         toolItalic.addActionListener(new StyledEditorKit.ItalicAction());
194         toolUnderline.addActionListener(new StyledEditorKit.UnderlineAction());
195
196         toolBlue.addActionListener(new StyledEditorKit.ForegroundAction(null, Color.BLUE));
197         toolGreen.addActionListener(new StyledEditorKit.ForegroundAction(null,
    Color.GREEN.darker()));
198         toolRed.addActionListener(new StyledEditorKit.ForegroundAction(null, Color.RED));
199
200         toolLeft.addActionListener(new StyledEditorKit.AlignmentAction(null, 0));
201         toolCenter.addActionListener(new StyledEditorKit.AlignmentAction(null, 1));
202         toolRight.addActionListener(new StyledEditorKit.AlignmentAction(null, 2));
203
204         return toolBar;
205     }
206 }
```

```java
package EjerciciosBloque3;

import java.awt.GridBagLayout;

import javax.swing.Box;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class _02_DisposicionesBox {

    public static void main(String[] args) {

        MarcoDisposicionesBox marco = new MarcoDisposicionesBox();
    }
}

class MarcoDisposicionesBox extends JFrame {

    public MarcoDisposicionesBox() {

        setSize(400, 600);
        setTitle("  Disposiciones Box");
        setLocationRelativeTo(null);

        add(new LaminaDisposicionesBox());

        pack();
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class LaminaDisposicionesBox extends JPanel {

    public LaminaDisposicionesBox() {

        Box boxH1 = Box.createHorizontalBox();
        Box boxH2 = Box.createHorizontalBox();
        Box boxH3 = Box.createHorizontalBox();

        boxH1.add(Box.createHorizontalStrut(50));
        boxH1.add(new JLabel("Nombre"));
        boxH1.add(Box.createHorizontalStrut(20));
        boxH1.add(new JTextField(20));
        boxH1.add(Box.createHorizontalStrut(50));

        boxH2.add(Box.createHorizontalStrut(50));
        boxH2.add(new JLabel("Email"));
        boxH2.add(Box.createHorizontalStrut(20));
        boxH2.add(new JTextField(20));
        boxH2.add(Box.createHorizontalStrut(50));

        boxH3.add(Box.createHorizontalStrut(50));
        boxH3.add(new JButton("Ok"));
        boxH3.add(Box.createGlue());
        boxH3.add(new JButton("Cancelar"));
        boxH3.add(Box.createHorizontalStrut(50));

        Box boxV = Box.createVerticalBox();
```

```
63
64          boxV.add(Box.createVerticalStrut(80));
65          boxV.add(boxH1);
66          boxV.add(Box.createVerticalStrut(30));
67          boxV.add(boxH2);
68          boxV.add(Box.createVerticalStrut(30));
69          boxV.add(boxH3);
70          boxV.add(Box.createVerticalStrut(100));
71
72          add(boxV);
73      }
74  }
```

```java
package EjerciciosBloque3;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.Spring;
import javax.swing.SpringLayout;

public class _03_DisposicionMuelle {

    public static void main(String[] args) {

        MarcoDisposicionMuelle marco = new MarcoDisposicionMuelle();
    }
}

class MarcoDisposicionMuelle extends JFrame {

    public MarcoDisposicionMuelle() {

        setSize(800, 300);
        setTitle("  Disposición Muelles");
        setLocationRelativeTo(null);

        add(new LaminaDisposicionMuelle());

        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class LaminaDisposicionMuelle extends JPanel {

    SpringLayout disposicion;

    public LaminaDisposicionMuelle() {

        setLayout(disposicion =new SpringLayout());

        JButton btn1 = new JButton("Botón 1");
        JButton btn2 = new JButton("Botón 2");
        JButton btn3 = new JButton("Botón 3");

        add(btn1);
        add(btn2);
        add(btn3);

        Spring muelle = Spring.constant(0, 0, 10);
        Spring muelleFijo = Spring.constant(50);

        disposicion.putConstraint(SpringLayout.WEST, btn1, muelle, SpringLayout.WEST, this);
        disposicion.putConstraint(SpringLayout.WEST, btn2, muelleFijo, SpringLayout.EAST,
  btn1);
        disposicion.putConstraint(SpringLayout.WEST, btn3, muelleFijo, SpringLayout.EAST,
  btn2);
        disposicion.putConstraint(SpringLayout.EAST, this, muelle, SpringLayout.EAST, btn3);
    }
}
```

```java
package EjerciciosBloque3;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class _04_DisposicionLibre {

    public static void main(String[] args) {

        MarcoDisposicionLibre marco = new MarcoDisposicionLibre();
    }
}

class MarcoDisposicionLibre extends JFrame {

    public MarcoDisposicionLibre() {

        setTitle(" Disposición Libre");
        setSize(400, 500);
        setLocationRelativeTo(null);

        add(new VentanaDisposicionLibre());

        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class VentanaDisposicionLibre extends JPanel {

    public VentanaDisposicionLibre() {

        setLayout(null);

        JLabel txt1 = new JLabel("Nombre");
        JLabel txt2 = new JLabel("Apellido");

        JTextField field1 = new JTextField(20);
        JTextField field2 = new JTextField(20);

        JButton btn = new JButton("Enviar");

        add(txt1).setBounds(50, 100, 50, 30);
        add(field1).setBounds(150, 100, 150, 30);
        add(txt2).setBounds(50, 200, 50, 30);
        add(field2).setBounds(150, 200, 150, 30);
        add(btn).setBounds(150, 300, 150, 30);
    }
}
```

```java
package EjerciciosBloque3;

import java.awt.Component;
import java.awt.Container;
import java.awt.Dimension;
import java.awt.LayoutManager;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class _05_PlantillasPersonalizadas {

    public static void main(String[] args) {

        MarcoPlantillasPersonalizadas marco = new MarcoPlantillasPersonalizadas();
    }
}

class MarcoPlantillasPersonalizadas extends JFrame {

    public MarcoPlantillasPersonalizadas() {

        setTitle(" Disposición Libre");
        setSize(400, 400);
        setLocationRelativeTo(null);

        add(new VentanaPlantillasPersonalizadas());

        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class VentanaPlantillasPersonalizadas extends JPanel {

    public VentanaPlantillasPersonalizadas() {

        setLayout(new DistribucionPersonalizada());

        JLabel txt1 = new JLabel("Nombre");
        JLabel txt2 = new JLabel("Apellido");

        JTextField field1 = new JTextField(20);
        JTextField field2 = new JTextField(20);

        JButton btn = new JButton("Enviar");

        add(txt1);
        add(field1);
        add(txt2);
        add(field2);
        add(btn);
    }

    private class DistribucionPersonalizada implements LayoutManager {

        @Override
        public void layoutContainer(Container parent) {

            int x = parent.getWidth() / 2;
```

```
63              int y = 50;
64
65              for (int i=0; i<parent.getComponentCount(); i++) {
66                  parent.getComponent(i).setBounds(x-100, y, 100, 30);
67                  x += 100;
68                  if ((i+1) % 2 == 0) {
69                      x = parent.getWidth() / 2;
70                      y += 100;
71                  }
72              }
73          }
74
75          @Override
76          public void addLayoutComponent(String name, Component comp) {
77              // TODO Auto-generated method stub
78          }
79
80          @Override
81          public void removeLayoutComponent(Component comp) {
82              // TODO Auto-generated method stub
83          }
84
85          @Override
86          public Dimension preferredLayoutSize(Container parent) {
87              // TODO Auto-generated method stub
88              return null;
89          }
90
91          @Override
92          public Dimension minimumLayoutSize(Container parent) {
93              // TODO Auto-generated method stub
94              return null;
95          }
96      }
97  }
```

```java
package EjerciciosBloque3;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;

public class _06_CuadroDialogos {

    public static void main(String[] args) {

        MarcoCuadroDialogos marco = new MarcoCuadroDialogos();
    }
}

class MarcoCuadroDialogos extends JFrame {

    public MarcoCuadroDialogos() {

        setTitle(" Cuadros Diálogos");
        setSize(600, 400);
        setLocationRelativeTo(null);

        add(new VentanaCuadroDialogos());

        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class VentanaCuadroDialogos extends JPanel {

    private JButton btn1, btn2, btn3, btn4;

    public VentanaCuadroDialogos() {

        add(btn1 = new JButton("Advertencia"));
        add(btn2 = new JButton("Intro Datos"));
        add(btn3 = new JButton("Confirmar"));
        add(btn4 = new JButton("Selección"));

        ClaseOyente oyente = new ClaseOyente();

        btn1.addActionListener(oyente);
        btn2.addActionListener(oyente);
        btn3.addActionListener(oyente);
        btn4.addActionListener(oyente);
    }

    private class ClaseOyente implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent e) {

            String tallas[] = {"S","M","L","XL"};

            if (e.getSource() == btn1)
JOptionPane.showMessageDialog(VentanaCuadroDialogos.this, "Le advertimos que ...",
"Advertencia", 2);
            if (e.getSource() == btn2)
```

```
   JOptionPane.showInputDialog(VentanaCuadroDialogos.this, "Introduzca sus datos");
61             if (e.getSource() == btn3)
   JOptionPane.showConfirmDialog(VentanaCuadroDialogos.this, "Elija una opción", "Confirmar",
   1);
62             if (e.getSource() == btn4)
   JOptionPane.showOptionDialog(VentanaCuadroDialogos.this, "Seleccione su talla", "Opciones",
   1, 1, null, tallas, 0);
63         }
64     }
65 }
```

```java
package EjerciciosBloque3;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Component;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.BorderFactory;
import javax.swing.Box;
import javax.swing.ButtonGroup;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JRadioButton;

public class _07_Ejercicio_Dialogos {

    public static void main(String[] args) {

        MarcoDialogos marco = new MarcoDialogos();
    }
}

class MarcoDialogos extends JFrame {

    public MarcoDialogos() {

        setSize(700, 500);
        setTitle("  Ejercicio Diálogos");
        setLocationRelativeTo(null);

        add(new LaminaDialogos());

        pack();
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

class LaminaDialogos extends JPanel {

    private JRadioButton    radioTipo1, radioTipo2, radioTipo3, radioTipo4,
                            radioTipoMnsj1, radioTipoMnsj2, radioTipoMnsj3, radioTipoMnsj4,
  radioTipoMnsj5,
                            radioMnsj1, radioMnsj2, radioMnsj3, radioMnsj4, radioMnsj5,
                            radioConfirm1, radioConfirm2, radioConfirm3, radioConfirm4,
                            radioOpcion1, radioOpcion2, radioOpcion3,
                            radioEntrada1, radioEntrada2;
    private ButtonGroup     group1, group2, group3, group4, group5, group6;
    private Box             box1, box2, box3, box4, box5, box6,
                            boxHorizontal1, boxHorizontal2, boxVertical;
    private JPanel          laminaBtn;
    private JButton         btn;

```

```
62      public LaminaDialogos() {
63
64          setLayout(new BorderLayout());
65
66          // ------------------------BOX 1-----------------------------------
67          box1 = Box.createVerticalBox();
68          box1.setBorder(BorderFactory.createTitledBorder("Tipo"));
69          box1.setPreferredSize(new Dimension(200,200));
70          box1.setMinimumSize(new Dimension(200,200));
71          box1.setMaximumSize(new Dimension(200,200));
72          box1.add(radioTipo1 = new JRadioButton("Mensaje", true));
73          box1.add(radioTipo2 = new JRadioButton("Confirmar", false));
74          box1.add(radioTipo3 = new JRadioButton("Opción", false));
75          box1.add(radioTipo4 = new JRadioButton("Entrada", false));
76          group1 = new ButtonGroup();
77          group1.add(radioTipo1);
78          group1.add(radioTipo2);
79          group1.add(radioTipo3);
80          group1.add(radioTipo4);
81
82          // ------------------------BOX 2-----------------------------------
83          box2 = Box.createVerticalBox();
84          box2.setBorder(BorderFactory.createTitledBorder("Tipo Mensaje"));
85          box2.setPreferredSize(new Dimension(200,200));
86          box2.setMinimumSize(new Dimension(200,200));
87          box2.setMaximumSize(new Dimension(200,200));
88          box2.add(radioTipoMnsj1 = new JRadioButton("ERROR_MESSAGE", true));
89          box2.add(radioTipoMnsj2 = new JRadioButton("INFORMATION_MESSAGE", false));
90          box2.add(radioTipoMnsj3 = new JRadioButton("WARNING_MESSAGE", false));
91          box2.add(radioTipoMnsj4 = new JRadioButton("QUESTION_MESSAGE", false));
92          box2.add(radioTipoMnsj5 = new JRadioButton("PLAIN_MESSAGE", false));
93          group2 = new ButtonGroup();
94          group2.add(radioTipoMnsj1);
95          group2.add(radioTipoMnsj2);
96          group2.add(radioTipoMnsj3);
97          group2.add(radioTipoMnsj4);
98          group2.add(radioTipoMnsj5);
99
100         // ------------------------BOX 3-----------------------------------
101         box3 = Box.createVerticalBox();
102         box3.setBorder(BorderFactory.createTitledBorder("Mensaje"));
103         box3.setPreferredSize(new Dimension(200,200));
104         box3.setMinimumSize(new Dimension(200,200));
105         box3.setMaximumSize(new Dimension(200,200));
106         box3.add(radioMnsj1 = new JRadioButton("Cadena", true));
107         box3.add(radioMnsj2 = new JRadioButton("Icono", false));
108         box3.add(radioMnsj3 = new JRadioButton("Componente", false));
109         box3.add(radioMnsj4 = new JRadioButton("Otros", false));
110         box3.add(radioMnsj5 = new JRadioButton("Objetc[]", false));
111         group3 = new ButtonGroup();
112         group3.add(radioMnsj1);
113         group3.add(radioMnsj2);
114         group3.add(radioMnsj3);
115         group3.add(radioMnsj4);
116         group3.add(radioMnsj5);
117
118         // ------------------------BOX 4-----------------------------------
119         box4 = Box.createVerticalBox();
120         box4.setBorder(BorderFactory.createTitledBorder("Confirmar"));
121         box4.setPreferredSize(new Dimension(200,200));
122         box4.setMinimumSize(new Dimension(200,200));
123         box4.setMaximumSize(new Dimension(200,200));
```

```
124         box4.add(radioConfirm1 = new JRadioButton("DEFAULT_OPTION", true));
125         box4.add(radioConfirm2 = new JRadioButton("YES_NO_OPTION", false));
126         box4.add(radioConfirm3 = new JRadioButton("YES_NO_CANCEL_OPTION", false));
127         box4.add(radioConfirm4 = new JRadioButton("OK_CANCEL_OPTION", false));
128         group4 = new ButtonGroup();
129         group4.add(radioConfirm1);
130         group4.add(radioConfirm2);
131         group4.add(radioConfirm3);
132         group4.add(radioConfirm4);
133 
134         // ------------------------BOX 5----------------------------------
135         box5 = Box.createVerticalBox();
136         box5.setBorder(BorderFactory.createTitledBorder("Opción"));
137         box5.setPreferredSize(new Dimension(200,200));
138         box5.setMinimumSize(new Dimension(200,200));
139         box5.setMaximumSize(new Dimension(200,200));
140         box5.add(radioOpcion1 = new JRadioButton("String[]", true));
141         box5.add(radioOpcion2 = new JRadioButton("Icon[]", false));
142         box5.add(radioOpcion3 = new JRadioButton("Object[]", false));
143         group5 = new ButtonGroup();
144         group5.add(radioOpcion1);
145         group5.add(radioOpcion2);
146         group5.add(radioOpcion3);
147 
148         // ------------------------BOX 6----------------------------------
149         box6 = Box.createVerticalBox();
150         box6.setBorder(BorderFactory.createTitledBorder("Entrada"));
151         box6.setPreferredSize(new Dimension(200,200));
152         box6.setMinimumSize(new Dimension(200,200));
153         box6.setMaximumSize(new Dimension(200,200));
154         box6.add(radioEntrada1 = new JRadioButton("Campo de texto", true));
155         box6.add(radioEntrada2 = new JRadioButton("Combo", false));
156         group6 = new ButtonGroup();
157         group6.add(radioEntrada1);
158         group6.add(radioEntrada2);
159 
160         // -----------------HORIZONTAL BOXES-------------------------------
161         boxHorizontal1 = Box.createHorizontalBox();
162         boxHorizontal1.add(box1);
163         boxHorizontal1.add(box2);
164         boxHorizontal1.add(box3);
165 
166         boxHorizontal2 = Box.createHorizontalBox();
167         boxHorizontal2.add(box4);
168         boxHorizontal2.add(box5);
169         boxHorizontal2.add(box6);
170 
171         // ------------------VERTICAL BOX----------------------------------
172         boxVertical = Box.createVerticalBox();
173         boxVertical.add(boxHorizontal1);
174         boxVertical.add(boxHorizontal2);
175         add(boxVertical, BorderLayout.CENTER);
176 
177         // -----------------BOTÓN MOSTRAR----------------------------------
178         laminaBtn = new JPanel();
179         laminaBtn.setLayout(new FlowLayout());
180         laminaBtn.add(btn = new JButton("Mostrar"));
181         add(laminaBtn, BorderLayout.SOUTH);
182 
183         btn.addActionListener(new AccionOyentes());
184     }
185 
```

```
186     private class AccionOyentes implements ActionListener {
187 
188         Component componente = LaminaDialogos.this;
189         Date fecha = new Date();
190         Object mensaje = "Mensaje de la Ventana";
191         String titulo = "Título de la Ventana";
192         int mnsjTipo = 0;
193         int opcionTipo = 0;
194         ImageIcon icono = new ImageIcon("src/images/icono.png");
195         ImageIcon iconoObj1 = new ImageIcon("src/images/Obj (1).png");
196         ImageIcon iconoObj2 = new ImageIcon("src/images/Obj (2).png");
197         ImageIcon iconoObj3 = new ImageIcon("src/images/Obj (3).png");
198         JButton btnObj1 = new JButton("Vídeo Cámara");
199         JButton btnObj2 = new JButton("Foto Cámara");
200         JButton btnObj3 = new JButton("Smart Phone");
201         Object opciones[];
202         Object opciones1[] = {"Vídeo", "Foto", "Móvil"};
203         Object opciones2[] = {new ImageIcon("src/images/Obj (1).png"), new
    ImageIcon("src/images/Obj (2).png"), new ImageIcon("src/images/Obj (3).png")};
204         Object opciones3[] = {new ImageIcon("src/images/Obj (1).png"), "Vídeo", new
    ImageIcon("src/images/Obj (2).png"), "Foto", new ImageIcon("src/images/Obj (3).png"),
    "Móvil",};
205         Object seleccion[] = {"Vídeo", "Foto", "Móvil"};
206 
207         @Override
208         public void actionPerformed(ActionEvent e) {
209 
210             JPanel lamina = new JPanel();
211             lamina.setPreferredSize(new Dimension(100,50));
212             lamina.setBackground(Color.CYAN);
213             lamina.add(new JLabel("Componente"), JLabel.CENTER);
214 
215             DateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy");
216 
217             Object objetos[] = {iconoObj1, btnObj1, iconoObj2, btnObj2, iconoObj3,
    btnObj3};
218 
219             if (radioTipoMnsj1.isSelected()) mnsjTipo = 0;
220             else if (radioTipoMnsj2.isSelected()) mnsjTipo = 1;
221             else if (radioTipoMnsj3.isSelected()) mnsjTipo = 2;
222             else if (radioTipoMnsj4.isSelected()) mnsjTipo = 3;
223             else if (radioTipoMnsj5.isSelected()) mnsjTipo = JOptionPane.PLAIN_MESSAGE;
224 
225             if (radioMnsj1.isSelected()) mensaje = mensaje;
226             else if (radioMnsj2.isSelected()) mensaje = (Object) icono;
227             else if (radioMnsj3.isSelected()) mensaje = lamina;
228             else if (radioMnsj4.isSelected()) mensaje = (Object) "Fecha:
    "+dateFormat.format(fecha);
229             else if (radioMnsj5.isSelected()) mensaje = objetos;
230 
231             if (radioConfirm1.isSelected()) opcionTipo = JOptionPane.DEFAULT_OPTION;
232             else if (radioConfirm2.isSelected()) opcionTipo = JOptionPane.YES_NO_OPTION;
233             else if (radioConfirm3.isSelected()) opcionTipo =
    JOptionPane.YES_NO_CANCEL_OPTION;
234             else if (radioConfirm4.isSelected()) opcionTipo = JOptionPane.OK_CANCEL_OPTION;
235 
236             if (radioOpcion1.isSelected()) opciones = opciones1;
237             else if (radioOpcion2.isSelected()) opciones = opciones2;
238             else if (radioOpcion3.isSelected()) opciones = opciones3;
239 
240             if (radioTipo1.isSelected()) JOptionPane.showMessageDialog(componente, mensaje,
    titulo, mnsjTipo, null);
```

```
241             if (radioTipo2.isSelected()) JOptionPane.showConfirmDialog(componente, mensaje,
    titulo, opcionTipo, mnsjTipo, null);
242             if (radioTipo3.isSelected()) JOptionPane.showOptionDialog(componente, mensaje,
    titulo, opcionTipo, mnsjTipo, null, opciones, 0);
243             if (radioTipo4.isSelected()) {
244                 if (radioEntrada1.isSelected()) JOptionPane.showInputDialog(componente,
    mensaje, titulo, mnsjTipo);
245                 else if (radioEntrada2.isSelected())
    JOptionPane.showInputDialog(componente, mensaje, titulo, mnsjTipo, null, seleccion, 0);
246             }
247         }
248     }
249 }
```

_01_AccesoFicheros.java

```java
package EjerciciosBloque4;

import java.io.File;

public class _01_AccesoFicheros {

    public static void main(String[] args) {

        File ruta = new File("C:\\Users\\DELL-OPTIPLEX-ENGEL\\Desktop\\Curso de Java\\EjerciciosCursoJava\\src");

        System.out.println("\nExiste el fichero?: " + ruta.exists());

        System.out.println("\nRuta: " + ruta.getAbsolutePath());

        File[] archivos = ruta.listFiles();

        System.out.println("\nCantidad total de archivos en la ruta: " + archivos.length + "\n");

        for (int i = 0; i < archivos.length; i++) {

            if (archivos[i].isDirectory()) {

                String[] subCarpetas = archivos[i].list();

                if (subCarpetas.length > 0) {
                    System.out.println("Carpeta Padre: " + archivos[i].getName() + " | Archivos:");
                } else {
                    System.out.println("Carpeta Padre: " + archivos[i].getName() + " | Archivos: No tiene.");
                }
                for (int j = 0; j < subCarpetas.length; j++) {
                    System.out.println(subCarpetas[j]);
                }
            }
            System.out.println("\n");
        }
    }
}

```

```java
package EjerciciosBloque4;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.JOptionPane;

public class _02_CreandoEliminandoFicherosArchivos {

    public static void main(String[] args) {

        // ---------------------------CREA DIRECTORIO NUEVO---------------------------------
        File ruta = new File("C:"+File.separator+"Users"+File.separator+"DELL-OPTIPLEX-ENGEL"+File.separator+"Desktop"+File.separator+"Curso de Java"+File.separator+"EjerciciosCursoJava"+File.separator+"src"+File.separator+"servidor"+File.separator+"Directorio Nuevo");

        ruta.mkdir();

        if (ruta.exists()) {
            System.out.println("\nDirectorio Nuevo creado con éxito! =) \n");
            JOptionPane.showMessageDialog(null, "Directorio Nuevo creado con éxito!");
        } else {
            System.out.println("\nDirectorio Nuevo NO exite! :( \n");
            JOptionPane.showMessageDialog(null, "Directorio Nuevo NO exite!");
        }

        // ---------------------------CREA ARCHIVO NUEVO---------------------------------
        File rutaArchivo = new File("C:"+File.separator+"Users"+File.separator+"DELL-OPTIPLEX-ENGEL"+File.separator+"Desktop"+File.separator+"Curso de Java"+File.separator+"EjerciciosCursoJava"+File.separator+"src"+File.separator+"servidor"+File.separator+"Directorio Nuevo"+File.separator+"Archivo Nuevo.txt");

        try {
            rutaArchivo.createNewFile();
            if (rutaArchivo.exists()) {
                System.out.println("\nArchivo Nuevo creado con éxito! =) \n");
                JOptionPane.showMessageDialog(null, "Archivo Nuevo creado con éxito!");
            } else {
                System.out.println("\nEl archivo nuevo No ha podido ser creado :( \n");
                JOptionPane.showMessageDialog(null, "El archivo nuevo No ha podido ser creado");
            }
        } catch (IOException e) {
            System.out.println("Ha surgido un error al intentar crear el archivo nuevo");
            e.printStackTrace();
        }

        // ---------------------------INSTANCIAS---------------------------------
        EscribirEnArchivo escribir = new EscribirEnArchivo(rutaArchivo);

        try {
            LeerDeArchivo leer = new LeerDeArchivo(rutaArchivo);
        } catch (IOException e) {
            e.printStackTrace();
        }

```

```
53          // --------------------------ELIMINAR ARCHIVO---------------------------------
54          int confirmar = JOptionPane.showConfirmDialog(null, "¿Deseas eliminar el
    archivo?");
55
56          if (confirmar == 0) {
57              rutaArchivo.delete();
58          } else {
59              JOptionPane.showMessageDialog(null, "Archivo conservado");
60              System.out.println("\n \nArchivo conservado");
61              return;
62          }
63
64          // --------------------------REVISAR SI EXISTE EL
    ARCHIVO-----------------------------------
65          if (!rutaArchivo.exists()) {
66              JOptionPane.showMessageDialog(null, "Archivo eliminado con éxito");
67              System.out.println("\n \nArchivo eliminado con éxito");
68          }
69      }
70  }
71
72  // --------------------------CLASE PARA ESCRIBIR EN ARCHIVO
    NUEVO----------------------------------
73  class EscribirEnArchivo {
74      File ruta;
75      String parrafo = "Lorem ipsum dolor sit amet consectetur adipiscing elit vel sociosqu,
    vitae id pretium ultricies volutpat semper sodales pulvinar et metus, praesent fames nam
    porta justo dui sem donec. Facilisi magnis leo dictum a lacinia rutrum per faucibus, hac
    auctor nisl tempor elementum quis non molestie pellentesque, risus ultricies placerat
    potenti porta pulvinar donec. Tincidunt ultrices nulla laoreet nostra quam luctus pharetra
    dictumst donec, vulputate lacinia urna torquent accumsan sodales eu phasellus per, varius
    eleifend condimentum ligula interdum tristique magna natoque.";
76
77      public EscribirEnArchivo(File ruta) {
78          this.ruta = ruta;
79
80          try {
81              // abro flujo
82              FileWriter escribir = new FileWriter(ruta);
83              escribir.write(parrafo);
84              // cierro flujo
85              escribir.close();
86              JOptionPane.showMessageDialog(null, "Texto escrito con éxito");
87              System.out.println("TEXTO ESCRITO CON ÉXITO");
88
89          } catch (IOException e1) {
90              e1.printStackTrace();
91          }
92      }
93  }
94
95  class LeerDeArchivo {
96      File ruta;
97
98      public LeerDeArchivo(File ruta) throws IOException {
99          this.ruta = ruta;
100         try {
101             // abro flujo
102             FileReader leer = new FileReader(ruta);
103             int nro = 0;
104             JOptionPane.showMessageDialog(null, "Texto leído con éxito");
105             System.out.println("\nTEXTO LEÍDO:");
```

```
106
107             while (nro != -1) {
108                 nro = leer.read();
109                 char letra = (char) nro;
110
111                 if (nro != -1) {
112                     System.out.print(letra);
113                 }
114             }
115             // cierro flujo
116             leer.close();
117
118         } catch (FileNotFoundException e) {
119             e.printStackTrace();
120         }
121     }
122 }
```

```java
package EjerciciosBloque4;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class _03_LeyendoEscribiendoBuffer {

    public static void main(String[] args) {

        LeerTexto leer = new LeerTexto();

        EscribirTexto escribir = new EscribirTexto();

        AgregarTexto agregar = new AgregarTexto();

        LeerTextoBuffer leerBuffer = new LeerTextoBuffer();

        EscribirTextoBuffer escribirBuffer = new EscribirTextoBuffer();

        AgregarTextoBuffer agregarBuffer = new AgregarTextoBuffer();

    }
}


// -------------------------CLASES SIN BUFFER-----------------------------------

// ----------------------CLASE PARA LEER TEXTO-----------------------------------
class LeerTexto {

    public LeerTexto() {
        File ruta = new File("C:/Users/DELL-OPTIPLEX-ENGEL/Desktop/Curso de Java/EjerciciosCursoJava/src/servidor/archivo1.txt");
        int nro = 0;
        char letra;

        try {
            // abro flujo
            FileReader lectura = new FileReader(ruta);
            System.out.println("\nTEXTO LEÍDO: \n");

            while (nro != -1) {
                nro = lectura.read();
                letra = (char) nro;
                System.out.print(letra);
            }
            System.out.println("");
            // cierro flujo
            lectura.close();

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

// ---------------------CLASE PARA ESCRIBIR TEXTO--------------------------------
class EscribirTexto {
```

```
62
63      public EscribirTexto() {
64          File ruta = new File("C:/Users/DELL-OPTIPLEX-ENGEL/Desktop/Curso de
    Java/EjerciciosCursoJava/src/servidor/archivo2.txt");
65          String texto = "Nor again is there anyone who loves or pursues or desires to obtain
    pain of itself.";
66
67          try {
68              // abro flujo
69              FileWriter escritura = new FileWriter(ruta);
70              escritura.write(texto);
71              // cierro flujo
72              escritura.close();
73
74          } catch (IOException e) {
75              e.printStackTrace();
76          }
77      }
78  }
79
80  // ----------------------CLASE PARA AGREGAR TEXTO---------------------------------
81  class AgregarTexto {
82
83      public AgregarTexto() {
84          File ruta = new File("C:/Users/DELL-OPTIPLEX-ENGEL/Desktop/Curso de
    Java/EjerciciosCursoJava/src/servidor/archivo2.txt");
85          String texto = "\n \nQui dolorem ipsum, quia dolor sit amet consectetur adipisci
    velit, sed quia non numquam eius modi tempora incidunt, ut labore et dolore magnam aliquam
    quaerat voluptatem.";
86
87          try {
88              // abro flujo
89              FileWriter escritura = new FileWriter(ruta, true);
90              escritura.write(texto);
91              // cierro flujo
92              escritura.close();
93
94          } catch (IOException e) {
95              e.printStackTrace();
96          }
97      }
98  }
99
100
101 // -------------------------CLASES CON BUFFER-----------------------------------
102
103 // --------------------CLASE PARA LEER TEXTO CON
    BUFFER-------------------------------------
104 class LeerTextoBuffer {
105
106     public LeerTextoBuffer() {
107         File ruta = new File("C:/Users/DELL-OPTIPLEX-ENGEL/Desktop/Curso de
    Java/EjerciciosCursoJava/src/servidor/archivo1_Buffer.txt");
108         int nro = 0;
109         char letra;
110
111         try {
112             // abro flujo
113             FileReader lectura = new FileReader(ruta);
114             BufferedReader buffer = new BufferedReader(lectura);
115
116             String leido = "";
```

```
117             System.out.println("\n \nTEXTO LEÍDO BUFFER: \n");
118 
119             while (leido != null) {
120                 leido = buffer.readLine();
121                 if (leido != null) System.out.println(leido);
122             }
123             // cierro flujo
124             lectura.close();
125             buffer.close();
126 
127         } catch (IOException e) {
128             e.printStackTrace();
129         }
130     }
131 }
132 
133 // ------------------CLASE PARA ESCRIBIR TEXTO CON BUFFER---------------------------------
134 class EscribirTextoBuffer {
135 
136     public EscribirTextoBuffer() {
137         File ruta = new File("C:/Users/DELL-OPTIPLEX-ENGEL/Desktop/Curso de
    Java/EjerciciosCursoJava/src/servidor/archivo2_Buffer.txt");
138         String texto = "Argumus again is there estratus who lorus or pursues or desires to
    obtain pain of itself.";
139 
140         try {
141             // abro flujo
142             FileWriter escritura = new FileWriter(ruta);
143             BufferedWriter buffer = new BufferedWriter(escritura);
144 
145             buffer.write(texto);
146 
147             // cierro flujo
148 //          escritura.close();  // éste flujo no se puede cerrar para permitir el buffer
149             buffer.close();
150 
151         } catch (IOException e) {
152             e.printStackTrace();
153         }
154     }
155 }
156 
157 // -------------------CLASE PARA AGREGAR TEXTO CON BUFFER---------------------------------
158 class AgregarTextoBuffer {
159 
160     public AgregarTextoBuffer() {
161         File ruta = new File("C:/Users/DELL-OPTIPLEX-ENGEL/Desktop/Curso de
    Java/EjerciciosCursoJava/src/servidor/archivo2_Buffer.txt");
162         String texto = "\n \nAmet estimulatum adipisci verdinum, sed quia non numquam eius
    modi tempora incidunt, ut labore et dolore magnam aliquam quaerat esporiatum.";
163 
164         try {
165             // abro flujo
166             FileWriter escritura = new FileWriter(ruta, true);
167             BufferedWriter buffer = new BufferedWriter(escritura);
168 
169             buffer.write(texto);
170 
171             // cierro flujo
172 //          escritura.close();  // éste flujo no se puede cerrar para permitir el buffer
173             buffer.close();
174 
```

```
175         } catch (IOException e) {
176             e.printStackTrace();
177         }
178     }
179 }
```

```java
package EjerciciosBloque4;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;

public class _04_LeyendoEscribiendoBytes {

    public static void main(String[] args) {

        byte[] imagen = new byte[338022];

        // ----------------------------LECTURA DE BYTES-------------------------------
        File file = new File("C:/Users/DELL-OPTIPLEX-ENGEL/Desktop/Curso de Java/EjerciciosCursoJava/src/servidor/kitty.jpg");

        try {
            // abro flujo
            FileInputStream archivo = new FileInputStream(file);

            // Usado al principio para saber la cantidad de bytes del archivo
//          int contador = 0;
//              while (archivo.read() != -1) {
//                  contador++;
//              }
//              System.out.println(contador);

            try {
                for (int i = 0; i < imagen.length; i++) {
                    imagen[i] = (byte) archivo.read();
                    System.out.println(imagen[i]);
                }

            } catch (IOException e1) {
                e1.printStackTrace();
            }

            // ciero flujo
            try {
                archivo.close();
            } catch (IOException e) {
                e.printStackTrace();
            }

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

        // ----------------------------ESCRITURA DE BYTES-------------------------------
        File fileCopy = new File("C:/Users/DELL-OPTIPLEX-ENGEL/Desktop/Curso de Java/EjerciciosCursoJava/src/servidor/kitty_copy.jpg");

        try {
            // abro flujo
            FileOutputStream archivoOut = new FileOutputStream(fileCopy);

            for (int i = 0; i < imagen.length; i++) {
                try {
                    archivoOut.write(imagen[i]);
                } catch (IOException e) {
```

```
61                      e.printStackTrace();
62                  }
63              }
64              // cierro flujo
65              try {
66                  archivoOut.close();
67              } catch (IOException e) {
68                  e.printStackTrace();
69              }
70
71          } catch (FileNotFoundException e) {
72              e.printStackTrace();
73          }
74      }
75  }
76
```

```java
package EjerciciosBloque4;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.Date;
import java.util.GregorianCalendar;

public class _05_AccesoFicheros {

    public static void main(String[] args) {

        // ----------------------OBJETO INSTANCIAS------------------------------
        Jefe jefe = new Jefe("Carmen Noguera", 3500, 1998, 2, 3);
        jefe.setIncentivo(1000);

        Empleado[] empleados = new Empleado[4];
        empleados[0] = new Empleado("José Ruco", 2500, 2005, 7, 15);
        empleados[1] = jefe;
        empleados[2] = new Empleado("María Patiño", 1500, 2004, 5, 20);
        empleados[3] = new Empleado("Arcángel Molina", 1500, 2007, 3, 5);

        // ----------------------EXPORTAR OBJETO------------------------------
        try {
            // abro flujo
            FileOutputStream  rutaOut = new FileOutputStream
("C:/Users/DELL-OPTIPLEX-ENGEL/Desktop/Curso de
Java/EjerciciosCursoJava/src/servidor/data.txt");
            ObjectOutputStream objetoOut;

            try {
                // abro flujo
                objetoOut = new ObjectOutputStream(rutaOut);
                objetoOut.writeObject(empleados);

                // cierro flujo
                objetoOut.close();

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

            // cierro flujo
            try {
                rutaOut.close();
            } catch (IOException e) {
                e.printStackTrace();
            }

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

        // ----------------------IMPORTAR OBJETO------------------------------
        try {
            // se abre flujo
            FileInputStream rutaIn = new FileInputStream
("C:/Users/DELL-OPTIPLEX-ENGEL/Desktop/Curso de
```

```
Java/EjerciciosCursoJava/src/servidor/data.txt");
60
61              try {
62                  // se abre flujo
63                  ObjectInputStream objetoIn = new ObjectInputStream(rutaIn);
64
65                  try {
66                      Empleado[] empleadosIn = (Empleado[]) objetoIn.readObject();
67
68                      for (int i = 0; i < empleadosIn.length; i++) {
69                          System.out.println(empleadosIn[i].dameData());
70                      }
71
72                  } catch (ClassNotFoundException e) {
73                      e.printStackTrace();
74                  }
75
76                  // se cierra flujo
77                  objetoIn.close();
78
79              } catch (IOException e) {
80                  e.printStackTrace();
81              }
82
83              // se cierra flujo
84              try {
85                  rutaIn.close();
86              } catch (IOException e) {
87                  e.printStackTrace();
88              }
89
90          } catch (FileNotFoundException e) {
91              e.printStackTrace();
92          }
93      }
94  }
95
96  //---------------------------CLASE EMPLEADO--------------------------------------
97  class Empleado implements Serializable  {
98
99      // se crea la huella única del programa
100     private static final long serialVersionUID = 1L;
101
102     private String nombre;
103     private double sueldo;
104     private Date fechaContrato;
105
106     public Empleado(String n, double s, int anio, int mes, int dia) {
107         nombre = n;
108         sueldo = s;
109         GregorianCalendar calendario = new GregorianCalendar(anio, mes-1, dia);
110         fechaContrato = calendario.getTime();
111     }
112
113     public String getNombre() {
114         return nombre;
115     }
116
117     public double getSueldo() {
118         return sueldo;
119     }
120
```

```java
    public Date getFechaContrato() {
        return fechaContrato;
    }


    public void subirSueldo(double porcentaje) {
        double aumento = sueldo * porcentaje / 100;
        sueldo += aumento;
    }

    public String dameData() {
        return "El Nombre es: " + nombre + ", y su sueldo es de $" + sueldo + ". Fecha de contrato: " + fechaContrato;
    }
}

 // ---------------------------CLASE JEFE---------------------------------------
class Jefe extends Empleado {

    // se crea la huella única del programa
    private static final long serialVersionUID = 1L;

    private double incentivo;

    public Jefe(String n, double s, int anio, int mes, int dia) {
        super(n, s, anio, mes, dia);
        incentivo = 0;
    }

    public double getSueldo() {
        double sueldoBase = super.getSueldo();
        return sueldoBase + incentivo;
    }

    public void setIncentivo(double b) {
        incentivo = b;
    }

    public String dameData() {
        return super.toString() + ". Incentivo: " + incentivo;
    }
}
```

```java
package EjerciciosBloque4;

import java.io.File;

public class _06_ProgramacionGenerica_ArrayList {

    public static void main(String[] args) {

        // ----------------1ERA INSTANCIA-------------------
        ClaseGenerica instancia = new ClaseGenerica(6);
        instancia.ElementoNuevo("Pedro");
        instancia.ElementoNuevo("María");
        instancia.ElementoNuevo("Juan");
        instancia.ElementoNuevo("Rosita");
        instancia.ElementoNuevo("Diego");
        instancia.ElementoNuevo("Anna");

        System.out.println(instancia.dameElemento(2));
        System.out.println(instancia.dameElemento(4));

        // ----------------2DA INSTANCIA-------------------
        File ruta = new File("C:\\Users\\DELL-OPTIPLEX-ENGEL\\Desktop\\Curso de Java\
\EjerciciosCursoJava\\src");

        ClaseGenerica instancia2 = new ClaseGenerica(1);
        instancia2.ElementoNuevo(ruta);

        System.out.println(instancia2.dameElemento(0));
    }
}

// ----------------------CLASE CON UN OBJETO GENÉRICO----------------------
class ClaseGenerica {

    private Object[] array;
    private int tamanio = 0;

    public ClaseGenerica(int tamanio) {
        array = new Object[tamanio];
    }

    public Object dameElemento(int elemento) {
        return array[elemento];
    }

    public void ElementoNuevo(Object nuevo) {
        array[tamanio] = nuevo;
        tamanio++;
    }
}
```

```java
package EjerciciosBloque4;

import java.util.ArrayList;
import java.util.Iterator;

public class _07_ArrayList_Iterator {

    public static void main(String[] args) {

        // -------------------INSTANCIAR EL ARRAYLIST-----------------------------------
        ArrayList<EmpleadoClase> misEmpleados = new ArrayList<EmpleadoClase>();

        // ---------------ASEGURAR UN TAMAÑO DEL ARRAYLIST--------------------------------
        misEmpleados.ensureCapacity(11);

        // ----------------AGREGAR ELEMENTOS AL ARRAYLIST---------------------------------
        misEmpleados.add(0, new EmpleadoClase("Diego", 38, 7800));
        misEmpleados.add(1, new EmpleadoClase("Martha", 28, 2800));
        misEmpleados.add(2, new EmpleadoClase("Rebecca", 32, 4800));
        misEmpleados.add(3, new EmpleadoClase("Elain", 48, 5800));
        misEmpleados.add(4, new EmpleadoClase("Willinton", 35, 1800));
        misEmpleados.add(5, new EmpleadoClase("Anne", 30, 2800));

        // ---------------------RECORTAR EL ARRAYLIST-------------------------------------
        misEmpleados.trimToSize();

        // ---------------------TAMAÑO DEL ARRAYLIST--------------------------------------
        System.out.println("\nCantidad de elementos en el ArrayList: " +
misEmpleados.size());

        // ---------------------REEMPLAZAR UN ELEMENTO------------------------------------
        misEmpleados.set(0, new EmpleadoClase("Juan", 42, 3500));

        // ---------------------IMPRIMIR UN ELEMENTO--------------------------------------
        System.out.println("\nEmpleado nro 3: " + misEmpleados.get(3).dameDatos() + "\n");

        // -----------------IMPRIMIR TODOS LOS ELEMENTOS--------------------------------------
        for (int i = 0; i < misEmpleados.size(); i++) {
            System.out.println("Empleado nro " + (i+1) + " : " +
misEmpleados.get(i).dameDatos());
        }

        // -------------------------------ITERATOR-----------------------------------------
        System.out.println("\nDatos obtenidos por el Iterator:");

        Iterator<EmpleadoClase> iterador = misEmpleados.iterator();

        while (iterador.hasNext()) {
            // OJO: se debe crear un objeto de tipo clase para poder recorrer y evitar
errores
            EmpleadoClase empleadoIterator = iterador.next();

            System.out.println(empleadoIterator.dameDatos());

            // -----------------ELIMINAR UN EMPLEADO POR SU
NOMBRE------------------------------
            if (empleadoIterator.dameNombre().equals("Willinton")) {
                iterador.remove();
            }
        }

        // --------------CONVERTIR ARRAYLIST EN ARRAY
```

```
  NORMAL---------------------------------
59          EmpleadoClase[] nuevoArray = new EmpleadoClase[misEmpleados.size()];
60          misEmpleados.toArray(nuevoArray);
61
62          // ----------------------IMPRIMIR ARRAY NORMAL-----------------------------------
63          System.out.println("\nArrayList convertido a Array Normal:");
64
65          for (int i = 0; i < nuevoArray.length; i++) {
66              System.out.println("Empleado: " + nuevoArray[i].dameDatos());
67          }
68      }
69  }
70
71  // --------------------CLASE EMPLEADO---------------------
72  class EmpleadoClase {
73
74      private String nombre;
75      private int edad;
76      private double salario;
77
78      public EmpleadoClase(String nombre, int edad, double salario) {
79          this.nombre = nombre;
80          this.edad = edad;
81          this.salario = salario;
82      }
83
84      public String dameNombre() {
85          return nombre;
86      }
87
88      public String dameDatos() {
89          return "Nombre: " + nombre + ", Edad: " + edad + ", Salario: " + salario + "€";
90      }
91  }
```

```java
package EjerciciosBloque4;

public class _08_ClasesGenericas {

    public static void main(String[] args) {

        // --------------INSTANCIA CLASE GENÉRICA TIPO STRING----------------
        ParejaOcho<String> empleados = new ParejaOcho<String>();
        empleados.setVariable("Diego de la Rua");
        System.out.println(empleados.getVariable());

        // --------------INSTANCIA CLASE GENÉRICA TIPO EMPLEADO----------------
        EmpleadoOcho empleado = new EmpleadoOcho("Amanda Greick", 35, 3500);
        ParejaOcho<EmpleadoOcho> empleadoPareja = new ParejaOcho<EmpleadoOcho>();
        empleadoPareja.setVariable(empleado);
        ParejaOcho.imprimeVariable(empleadoPareja);

        // -----------INSTANCIA CLASE GENÉRICA TIPO PERSONA-------------------
        Persona persona = new Persona("Alejandro Arteaza");
        ParejaOcho<Persona> empleadoPersona = new ParejaOcho<Persona>();
        empleadoPersona.setVariable(persona);
        System.out.println("Genérico Tipo Persona: " + empleadoPersona.getVariable());
    }
}

// ---------------------------CLASE EMPLEADO-------------------------------
class EmpleadoOcho {

    private String nombre;
    private int edad;
    private double salario;

    public EmpleadoOcho(String nombre, int edad, double salario) {
        this.nombre = nombre;
        this.edad = edad;
        this.salario = salario;
    }

    public String dameData() {
        return "Nombre: " + nombre + ", Edad: " + edad + "años, Salario: " + salario + "€";
    }
}

// -------------------------CLASE GENÉRICA PAREJA-----------------------------
class ParejaOcho<T>  {

    private T variable;

    public ParejaOcho() {
        variable = null;
    }

    public void setVariable(T mod) {
        this.variable = mod;
    }

    public T getVariable() {
        return variable;
    }

    public static void imprimeVariable(ParejaOcho<? extends EmpleadoOcho> obj) {
        EmpleadoOcho variable = obj.getVariable();
```

```
63          System.out.print("Genérico Tipo Empleado: ");
64          System.out.println(variable);
65      }
66  }
67
68  // -------------------------CLASE PERSONA------------------------------
69  class Persona  {
70
71      private String nombre;
72
73      public Persona(String nombre) {
74          this.nombre = nombre;
75      }
76
77      public String getNombre() {
78          return nombre;
79      }
80
81  }
```

```java
package EjerciciosBloque4;

import java.util.GregorianCalendar;

public class _09_MetodosGenericos {

    public static void main(String[] args) {

        // ----------------------------ARRAY DE NOMBRES--------------------------------------
        System.out.println("ARRAY DE NOMBRES:");

        String[] nombres = {"Pepito Fulano", "Marconi Lusejo", "Escarlata Dientreto", "Antonie Lexpury", "Dawal Espertoo"};

        ClaseMetodosGenericos metodos = new ClaseMetodosGenericos();

        System.out.println(metodos.cantidadElementos(nombres));

        System.out.println("Elemento menor: " + metodos.valorMenor(nombres));

        // ----------------------------ARRAY DE FECHAS-----------------------------------------
        System.out.println("\nARRAY DE FECHAS:");

        GregorianCalendar[] fechas = new GregorianCalendar[4];
        fechas[0] = new GregorianCalendar(2005, 3, 14);
        fechas[1] = new GregorianCalendar(2010, 8, 20);
        fechas[2] = new GregorianCalendar(2014, 10, 3);
        fechas[3] = new GregorianCalendar(2003, 1, 25);

        ClaseMetodosGenericos fechasM = new ClaseMetodosGenericos();

        System.out.println(fechasM.cantidadElementos(fechas));

        System.out.println("Elemento menor: " + fechasM.valorMenor(fechas).getTime());

        // ----------------------------ARRAY DE EMPLEADOS--------------------------------------
        System.out.println("\nARRAY DE EMPLEADOS:");

        Empleado[] misEmpleados = new Empleado[6];
        misEmpleados[0] = new Empleado("Asprilla", 5600, 1997, 3, 25);
        misEmpleados[1] = new Empleado("Higuita", 5600, 1997, 3, 25);
        misEmpleados[2] = new Empleado("Valderrama", 5600, 1997, 3, 25);
        misEmpleados[3] = new Empleado("Rincón", 5600, 1997, 3, 25);
        misEmpleados[4] = new Empleado("James", 5600, 1997, 3, 25);
        misEmpleados[5] = new Empleado("Falcao", 5600, 1997, 3, 25);

        ClaseMetodosGenericos jugadores = new ClaseMetodosGenericos();

        System.out.println(jugadores.cantidadElementos(misEmpleados));
    }
}

// ------------------------CLASE CON MÉTODOS GENÉRICOS---------------------------------
class ClaseMetodosGenericos {

    public static <T> String cantidadElementos(T[] obj) {
        return "El array tiene: " +  obj.length + " elementos.";
    }

    public static <T extends Comparable> T valorMenor(T[] array) {
```

```
60          T menor = array[0];
61          for (int i = 0; i < array.length; i++) {
62              if (menor.compareTo(array[i]) > 0) {
63                  menor = array[i];
64              }
65          }
66          return menor;
67      }
68  }
```

```java
package EjerciciosBloque4;

public class _10_ProgramacionConcurrente_SincronizandoHilos {

    public static void main(String[] args) {

        ClaseHilosUno uno = new ClaseHilosUno();
        ClaseHilosDos dos = new ClaseHilosDos(uno);
        ClaseHilosDos tres = new ClaseHilosDos(dos);

        uno.start();
        dos.start();
        tres.start();
    }
}

// ---------------------------CLASE HILOS UNO---------------------------------------
class ClaseHilosUno extends Thread {

    public void run() {
        for (int i = 0; i < 15; i++) {
            System.out.println("Hilo nro: " + (i+1) + ", Hilo Id: " + getId() + ", Hilo nombre: " + getName() + ", Hilo estado: " + getState() + ", Hilo actual: " + Thread.currentThread());
            try {
                Thread.sleep(100);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

//---------------------------CLASE HILOS DOS---------------------------------------
class ClaseHilosDos extends Thread {

    private Thread hilo;

    public ClaseHilosDos(Thread hilo) {
        this.hilo = hilo;
    }

    public void run() {

        try {
            hilo.join();
        } catch (InterruptedException e1) {
            e1.printStackTrace();
        }

        for (int i = 0; i < 15; i++) {
            System.out.println("Hilo nro: " + (i+1) + ", Hilo Id: " + getId() + ", Hilo nombre: " + getName() + ", Hilo estado: " + getState() + ", Hilo actual: " + Thread.currentThread());
            try {
                Thread.sleep(100);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```java
package EjerciciosBloque4;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Component;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.geom.Ellipse2D;
import java.util.ArrayList;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;

import EjerciciosBloque4.MarcoPelotas.DibujarPelota;

public class _11_UsoThreads_VariosBotones {

    public static void main(String[] args) {

        MarcoPelotas marco = new MarcoPelotas();
        marco.setVisible(true);
        marco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

// --------------------------MARCO PRINCIPAL-------------------------------------
class MarcoPelotas extends JFrame {

    private JButton btnPlay1, btnPlay2, btnPlay3, btnStop1, btnStop2, btnStop3;
    private int marcoX, marcoY;
    private LaminaxPelotas lamina = new LaminaxPelotas();
    private Thread hilo1, hilo2, hilo3;

    public MarcoPelotas() {

        setSize(600, 400);
        setTitle(" Hilos Pelotas");
        setLocationRelativeTo(null);
        setLayout(new BorderLayout());

        marcoX = this.getWidth();
        marcoY = this.getHeight() - 70;

//      // ----------------------LÁMINA PELOTAS---------------------------------
        lamina.setBackground(new Color(204, 255, 204));
        add(lamina, BorderLayout.CENTER);

        // ----------------------LÁMINA BOTONES---------------------------------
        JPanel laminaBotones = new JPanel();
        laminaBotones.setBackground(new Color(204, 204, 255));
        add(laminaBotones, BorderLayout.SOUTH);
        laminaBotones.add(btnPlay1 = new JButton("Play 1"));
        laminaBotones.add(btnPlay2 = new JButton("Play 2"));
        laminaBotones.add(btnPlay3 = new JButton("Play 3"));
        laminaBotones.add(btnStop1 = new JButton("Stop 1"));
        laminaBotones.add(btnStop2 = new JButton("Stop 2"));
        laminaBotones.add(btnStop3 = new JButton("Stop 3"));

        btnPlay1.setBackground(new Color(255, 255, 204));
        btnPlay2.setBackground(new Color(255, 255, 204));
```

```
63          btnPlay3.setBackground(new Color(255, 255, 204));
64          btnStop1.setBackground(new Color(255, 204, 255));
65          btnStop2.setBackground(new Color(255, 204, 255));
66          btnStop3.setBackground(new Color(255, 204, 255));
67
68          // ----------------------OYENTES ACCIONES--------------------------------
69          AccionesBotones oyente = new AccionesBotones();
70          btnPlay1.addActionListener(oyente);
71          btnPlay2.addActionListener(oyente);
72          btnPlay3.addActionListener(oyente);
73          btnStop1.addActionListener(oyente);
74          btnStop2.addActionListener(oyente);
75          btnStop3.addActionListener(oyente);
76      }
77
78      // ----------------------------CLASE ACCIONES-----------------------------------
79      class AccionesBotones implements ActionListener {
80          @Override
81          public void actionPerformed(ActionEvent e) {
82
83              if (e.getSource() == btnPlay1) LanzarHilos(e);
84              if (e.getSource() == btnPlay2) LanzarHilos(e);
85              if (e.getSource() == btnPlay3) LanzarHilos(e);
86
87              if (e.getSource() == btnStop1) DetenerHilos(e);
88              if (e.getSource() == btnStop2) DetenerHilos(e);
89              if (e.getSource() == btnStop3) DetenerHilos(e);
90          }
91      }
92
93      // -------------------------MÉTODO LANZAR HILOS-----------------------------------
94      public void LanzarHilos(ActionEvent e) {
95
96          DibujarPelota pelotaNew = new DibujarPelota();
97          lamina.addjuntar(pelotaNew);
98
99          Runnable runner = new ClaseRunnable(pelotaNew, lamina);
100
101         if (e.getSource().equals(btnPlay1)) {
102             hilo1 = new Thread(runner);
103             hilo1.start();
104         } else if (e.getSource().equals(btnPlay2)) {
105             hilo2 = new Thread(runner);
106             hilo2.start();
107         } else if (e.getSource().equals(btnPlay3)) {
108             hilo3 = new Thread(runner);
109             hilo3.start();
110         }
111     }
112
113     // -------------------------MÉTODO DETENER HILOS-----------------------------------
114     public void DetenerHilos(ActionEvent e) {
115
116         if (e.getSource().equals(btnStop1)) {
117             hilo1.interrupt();
118         } else if (e.getSource().equals(btnStop2)) {
119             hilo2.interrupt();
120         } else if (e.getSource().equals(btnStop3)) {
121             hilo3.interrupt();
122         }
123     }
124
```

```
125     // ------------------------CLASE DIBUJAR PELOTA---------------------------------
126     class DibujarPelota extends JPanel {
127         private int x, y, xw, yh, dx, dy;
128 
129         public DibujarPelota() {
130             dx = 1;
131             dy = 1;
132             xw = 35;
133             yh = 35;
134         }
135 
136         public void pintarPelotas() {
137             x += dx;
138             y += dy;
139             // si las distancias x-y de la bola son mayores que el marco, entonces ir
    restando posiciones
140             if ((x+xw) >= marcoX) {
141                 x = marcoX - xw;
142                 dx = -dx;
143             }
144             if ((y+yh) >= marcoY) {
145                 y = marcoY - yh;
146                 dy = -dy;
147             }
148             // si las distancias x-y de la bola son menores que el marco, entonces ir
    sumando posiciones
149             if (x < 1) {
150                 x = 1;
151                 dx = -dx;
152             }
153             if (y < 1) {
154                 y = 1;
155                 dy = -dy;
156             }
157         }
158 
159         public Ellipse2D getShape(){
160             return new Ellipse2D.Double(x,y,xw,yh);
161         }
162     }
163 
164     // ------------------------CLASE MOVER PELOTAS---------------------------------
165     class LaminaxPelotas extends JPanel {
166 
167         ArrayList<DibujarPelota> pelotas = new ArrayList<MarcoPelotas.DibujarPelota>();
168 
169         public void addjuntar(DibujarPelota b) {
170             pelotas.add(b);
171         }
172 
173         public void paintComponent(Graphics g) {
174             super.paintComponent(g);
175             Graphics2D g2 = (Graphics2D) g;
176                 g2.setColor(new Color(255, 0, 102));
177             for(DibujarPelota p: pelotas)
178                 g2.fill(p.getShape());
179         }
180     }
181 }
182 
183 // ------------------------CLASE RUNNABLE---------------------------------------
184 class ClaseRunnable implements Runnable {
```

```
185
186     private DibujarPelota unaPelota;
187     private Component unComponente;
188
189     public ClaseRunnable(DibujarPelota unaPelota, Component unComponente) {
190         this.unaPelota = unaPelota;
191         this.unComponente = unComponente;
192     }
193
194     @Override
195     public void run() {
196
197         System.out.println("Estado del hilo al iniciar: " +
    Thread.currentThread().isInterrupted());
198
199         // éste bucle while es el responsable del efecto de movimiento
200         while (!Thread.currentThread().isInterrupted()) {
201
202             unaPelota.pintarPelotas();
203             unComponente.paint(unComponente.getGraphics());
204
205             try {
206                 Thread.sleep(1);
207             } catch (InterruptedException e) {
208                 e.printStackTrace();
209             }
210         }
211         System.out.println("Estado del hilo al finalizar: " +
    Thread.currentThread().isInterrupted());
212     }
213 }
```

```java
package EjerciciosBloque4;

import java.util.concurrent.locks.Condition;
import java.util.concurrent.locks.ReentrantLock;

public class _12_Banco_Sin_Sincronizar {

    public static void main(String[] args) {

        BancoPipe miBanco = new BancoPipe();

        for (int i=0; i<100; i++) {
            Thread t = new Thread(miBanco);
            t.start();

            try {
                Thread.sleep(100);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

class BancoPipe extends Thread implements Runnable {

    private double cantidadTransaccion;
    private double saldoTotalBanco = 0;
    private int cuentaOrigen, cuentaDestino;
    private double[] cuentas = new double[100];
    private ReentrantLock bloqueo = new ReentrantLock();
    private Condition condicion = bloqueo.newCondition();

    public BancoPipe() {

        for (int i = 0; i < cuentas.length; i++) {
            cuentas[i] = 2000;
        }
    }

    public double SaldoTotalBanco() {
        for (int i = 0; i < cuentas.length; i++) {
            saldoTotalBanco += cuentas[i];
        }
        return saldoTotalBanco;
    }

    public void run() {

        bloqueo.lock();

        try {

            Thread.sleep(100);

            cantidadTransaccion = Math.random()*2000;

            cuentaOrigen = (int) ((Math.random())*100);
            cuentaDestino = (int) ((Math.random())*100);

            if (cuentas[cuentaOrigen] < cantidadTransaccion) {

```

```
63                System.out.println("\nHILO ESPERANDO.....................");
64                System.out.printf("Transferencia a la espera de saldo suficiente! | Monto
  solicitado: %1.2f€, Cuenta Origen nro: %d, Cuenta Destino nro: %d", cantidadTransaccion,
  cuentaOrigen, cuentaDestino);
65                System.out.printf("\nSaldo Cuenta Origen: %1.2f€, Saldo Cuenta Destino:
  %1.2f€", cuentas[cuentaOrigen], cuentas[cuentaDestino]);
66                System.out.println("\nHilo nro: " + getName() + " " + currentThread());
67
68                condicion.await();
69
70                if (cuentas[cuentaOrigen] >= cantidadTransaccion) {
71                    System.out.print("\nHILO LIBERADO!!!!!!!!!!!!!!!!!!!!!!!!");
72                }
73
74            } else {
75
76                cuentas[cuentaOrigen] -= cantidadTransaccion;
77                cuentas[cuentaDestino] += cantidadTransaccion;
78
79                System.out.printf("\nTransferencia exitosa! | Monto transferido: %1.2f€,
  Cuenta Origen nro: %d, Cuenta Destino nro: %d, Saldo Total Banco: %1.2f€",
  cantidadTransaccion, cuentaOrigen, cuentaDestino, SaldoTotalBanco());
80
81                System.out.printf("\nSaldo Nuevo Cuenta Origen: %1.2f€, Saldo Nuevo Cuenta
  Destino: %1.2f€", cuentas[cuentaOrigen], cuentas[cuentaDestino]);
82                System.out.println("\nHilo nro: " + getName() + " " + currentThread());
83
84                saldoTotalBanco = 0;
85            }
86
87            condicion.signalAll();
88
89        }
90        catch (InterruptedException e) {
91            e.printStackTrace();
92
93        } finally {
94            bloqueo.unlock();
95        }
96    }
97 }
```

```java
package EjerciciosBloque4;

public class _13_Banco_Synchronized {

    public static void main(String[] args) {

        BancoPipeSynchronized miBanco = new BancoPipeSynchronized();

        for (int i=0; i<100; i++) {
            Thread t = new Thread(miBanco);
            t.start();
            try {
                Thread.sleep(100);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

class BancoPipeSynchronized extends Thread implements Runnable {

    private double cantidadTransaccion;
    private double saldoTotalBanco = 0;
    private int cuentaOrigen, cuentaDestino;
    private double[] cuentas = new double[100];

    public BancoPipeSynchronized() {

        for (int i = 0; i < cuentas.length; i++) {
            cuentas[i] = 2000;
        }
    }

    public double SaldoTotalBanco() {
        for (int i = 0; i < cuentas.length; i++) {
            saldoTotalBanco += cuentas[i];
        }
        return saldoTotalBanco;
    }

    public synchronized void run() {

        try {

            Thread.sleep(100);

            cantidadTransaccion = Math.random()*2000;

            cuentaOrigen = (int) ((Math.random())*100);
            cuentaDestino = (int) ((Math.random())*100);

            if (cuentas[cuentaOrigen] < cantidadTransaccion) {

                System.out.println("\nHILO ESPERANDO.....................");
                System.out.printf("Transferencia a la espera de saldo suficiente! | Monto solicitado: %1.2f€, Cuenta Origen nro: %d, Cuenta Destino nro: %d", cantidadTransaccion, cuentaOrigen, cuentaDestino);
                System.out.printf("\nSaldo Cuenta Origen: %1.2f€, Saldo Cuenta Destino: %1.2f€", cuentas[cuentaOrigen], cuentas[cuentaDestino]);
                System.out.println("\nHilo nro: " + getName() + " " + currentThread());

```

```
60                  wait();
61
62                  if (cuentas[cuentaOrigen] >= cantidadTransaccion) {
63                      System.out.print("\nHILO LIBERADO!!!!!!!!!!!!!!!!!!!!!!!!!");
64
65                      cuentas[cuentaOrigen] -= cantidadTransaccion;
66                      cuentas[cuentaDestino] += cantidadTransaccion;
67
68                      System.out.printf("\nTransferencia exitosa! | Monto transferido: %1.2f€,
  Cuenta Origen nro: %d, Cuenta Destino nro: %d, Saldo Total Banco: %1.2f€",
  cantidadTransaccion, cuentaOrigen, cuentaDestino, SaldoTotalBanco());
69
70                      System.out.printf("\nSaldo Nuevo Cuenta Origen: %1.2f€, Saldo Nuevo
  Cuenta Destino: %1.2f€", cuentas[cuentaOrigen], cuentas[cuentaDestino]);
71                      System.out.println("\nHilo nro: " + getName() + " " + currentThread());
72
73                      saldoTotalBanco = 0;
74                  }
75              } else {
76
77                  cuentas[cuentaOrigen] -= cantidadTransaccion;
78                  cuentas[cuentaDestino] += cantidadTransaccion;
79
80                  System.out.printf("\nTransferencia exitosa! | Monto transferido: %1.2f€,
  Cuenta Origen nro: %d, Cuenta Destino nro: %d, Saldo Total Banco: %1.2f€",
  cantidadTransaccion, cuentaOrigen, cuentaDestino, SaldoTotalBanco());
81
82                  System.out.printf("\nSaldo Nuevo Cuenta Origen: %1.2f€, Saldo Nuevo Cuenta
  Destino: %1.2f€", cuentas[cuentaOrigen], cuentas[cuentaDestino]);
83                  System.out.println("\nHilo nro: " + getName() + " " + currentThread());
84
85                  saldoTotalBanco = 0;
86              }
87
88              notifyAll();
89          }
90          catch (InterruptedException e) {
91              e.printStackTrace();
92          }
93      }
94  }
```

```java
package EjerciciosBloque5;

public class _01_HashCode_Equals {

    public static void main(String[] args) {

        Libro libro1 = new Libro("El Mendigo", "Arturo Prieto", 2005, 1001);
        Libro libro2 = new Libro("El Mendigo", "Arturo Prieto", 2005, 1001);

        if (libro1.equals(libro2)) {
            System.out.println("Los dos libros son Iguales");
        } else {
            System.out.println("Los dos libros son Diferentes");
        }

        System.out.println("HasCode Libro 1: " + libro1.hashCode() + ", HasCode Libro 2: " + libro2.hashCode());
    }
}

class Libro {
    private String titulo, autor;
    private int anio, ISBN;

    public Libro(String titulo, String autor, int anio, int ISBN) {
        this.titulo = titulo;
        this.autor = autor;
        this.anio = anio;
        this.ISBN = ISBN;
    }

    public String getData() {
        return "Título del libro: " + titulo + ", Autor: " + autor + ", Año: " + anio + ", ISBN: " + ISBN;
    }

//  public boolean equals(Libro obj) {
//      if (this.ISBN == obj.ISBN) {
//          return true;
//      } else {
//          return false;
//      }
//  }

    @Override
    public int hashCode() {
        final int prime = 31;
        int result = 1;
        result = prime * result + ISBN;
        return result;
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        Libro other = (Libro) obj;
        if (ISBN != other.ISBN)
```

```
61              return false;
62          return true;
63      }
64  }
```

```java
package EjerciciosBloque5;

import java.util.HashSet;
import java.util.Iterator;

public class _02_Set_HashSet {

    public static void main(String[] args) {

        Cliente cliente1 = new Cliente("Adrián López", 1001, 35000);
        Cliente cliente2 = new Cliente("Julián Andrade", 1002, 15000);
        Cliente cliente3 = new Cliente("Karina Parada", 1003, 25000);
        Cliente cliente4 = new Cliente("Martha Ramírez", 1004, 45000);
        Cliente cliente5 = new Cliente("Eustoquio Rolón", 1005, 38000);
        Cliente cliente6 = new Cliente("José Contreras", 1006, 23000);

        // ------------------------CREAR LA COLECCIÓN-----------------------------------
        HashSet<Cliente> clientes = new HashSet<Cliente>();

        clientes.add(cliente1);
        clientes.add(cliente2);
        clientes.add(cliente3);
        clientes.add(cliente4);
        clientes.add(cliente5);
        clientes.add(cliente6);

        for (Cliente cliente : clientes) {
            System.out.println("Nombre del cliente: " + cliente.getNombre() + ", Nro de Cuenta: " + cliente.getNroCuenta() + ", Saldo: " + cliente.getSaldo());
        }

        // ---------------------AGREGAR CLIENTE DUPLICADO---------------------------------
        System.out.println("\nColección con cliente duplicado:");

        Cliente cliente7 = new Cliente("Karina Parada", 1003, 25000);

        clientes.add(cliente7);

        for (Cliente cliente : clientes) {
            System.out.println("Nombre del cliente: " + cliente.getNombre() + ", Nro de Cuenta: " + cliente.getNroCuenta() + ", Saldo: " + cliente.getSaldo());
        }

        // --------------------IMPRIMIR NOMBRE CON ITERATOR-------------------------------
        System.out.println("\nNombres impresos con iterator:");

        Iterator<Cliente> iterador = clientes.iterator();

        while (iterador.hasNext()) {
            System.out.println("Cliente: " +  iterador.next().getNombre());
        }

        // --------------------ELIMINAR UN CLIENTE CON ITERATOR--------------------------------
        System.out.println("\nCliente Eustoquio eliminado con iterator:");

        iterador = clientes.iterator();
        while (iterador.hasNext()) {

            if (iterador.next().getNombre().equals("Eustoquio Rolón")) {
                iterador.remove();
            }
```

```java
        }

        iterador = clientes.iterator();
        while (iterador.hasNext()) {
            System.out.println("Cliente: " +  iterador.next().getNombre());
        }
    }
}

// --------------------CLASE CLIENTE--------------------------------
class Cliente {

    private String nombre;
    private int nroCuenta;
    private double saldo;

    public Cliente(String nombre, int nroCuenta, double saldo) {
        this.nombre = nombre;
        this.nroCuenta = nroCuenta;
        this.saldo = saldo;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public void setNroCuenta(int nroCuenta) {
        this.nroCuenta = nroCuenta;
    }

    public void setSaldo(double saldo) {
        this.saldo = saldo;
    }

    public String getNombre() {
        return nombre;
    }

    public int getNroCuenta() {
        return nroCuenta;
    }

    public double getSaldo() {
        return saldo;
    }

    @Override
    public int hashCode() {
        final int prime = 31;
        int result = 1;
        result = prime * result + nroCuenta;
        return result;
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
```

```
122         Cliente other = (Cliente) obj;
123         if (nroCuenta != other.nroCuenta)
124             return false;
125         return true;
126     }
127 }
```

```java
package EjerciciosBloque5;

import java.util.LinkedList;
import java.util.ListIterator;

public class _03_LinkedList {

    public static void main(String[] args) {

        LinkedList<String> nombres = new LinkedList<String>();

        nombres.add("Dayana");
        nombres.add("Alejandra");
        nombres.add("Rebecca");
        nombres.add("Carolina");
        nombres.add("Andrea");
        nombres.add("Luisana");
        nombres.add("Mariana");

        System.out.println("\nTamaño de la colección: la colección tiene " + nombres.size()
  + " elementos:\n");

        for (String nombre : nombres) {
            System.out.println(nombre);
        }

        System.out.println("\nAgregar elemento en la posición 2 con un ListIterator:");

        ListIterator<String> iterador = nombres.listIterator();

        iterador.next();
        iterador.add("Roxana");

        for (String nombre : nombres) {
            System.out.println(nombre);
        }
    }
}
```

```java
package EjerciciosBloque5;

import java.util.LinkedList;
import java.util.ListIterator;

public class _04_LinkedList_Enlazada {

    public static void main(String[] args) {

        LinkedList<String> paises = new LinkedList<String>();

        paises.add("Argentina");
        paises.add("Perú");
        paises.add("Paraguay");
        paises.add("Colombia");
        paises.add("Venezuela");
        paises.add("Ecuador");
        paises.add("Chile");

        LinkedList<String> capitales = new LinkedList<String>();

        capitales.add("Buenos Aires");
        capitales.add("Lima");
        capitales.add("Asunción");
        capitales.add("Bogotá");
        capitales.add("Caracas");
        capitales.add("Quito");
        capitales.add("Santiago");

        // -----------------AGREGAR CAPITALES A LA COLECCIÓN PAÍSES----------------
        System.out.println("\nCapitales agregadas a colección países:");
        ListIterator<String> iteradorCapitales = capitales.listIterator();
        ListIterator<String> iteradorPaises = paises.listIterator();

        while (iteradorPaises.hasNext()) {
            if (iteradorCapitales.hasNext()) {
            iteradorPaises.next();
            iteradorPaises.add(iteradorCapitales.next());
            }
        }

        for (String data : paises) {
            System.out.println(data);
        }

        // --------------------ELIMINAR CAPITALES PARES----------------------
        System.out.println("\nCapitales restantes:");
        iteradorCapitales = capitales.listIterator();

        while (iteradorCapitales.hasNext()) {
            iteradorCapitales.next();
            if (iteradorCapitales.hasNext()) {
                iteradorCapitales.next();
                iteradorCapitales.remove();
            }
        }

        for (String capital : capitales) {
            System.out.println(capital);
        }

        // --------------ELIMINAR CAPITALES RESTANTES DE LA COLECCIÓN
```

```
  PAÍSES-----------------
        System.out.println("\nColección paises sin capitales restantes:");

        paises.removeAll(capitales);

        for (String pais : paises) {
            System.out.println(pais);
        }
    }
}
```

```java
package EjerciciosBloque5;

import java.util.Comparator;
import java.util.TreeSet;

public class _05_TreeSet {

    public static void main(String[] args) {

        TreeSet<String> nombres = new TreeSet<String>();

        nombres.add("María");
        nombres.add("Fernanda");
        nombres.add("Bertha");
        nombres.add("Danna");
        nombres.add("Xiomara");
        nombres.add("Daniela");

        System.out.println(nombres);

        // -------------------COLECCIÓN ARTÍCULOS-------------------------
        System.out.println("\nColección de artículos ordenados por el número");

        Articulo articulo1 = new Articulo(1001, "Ventilador");
        Articulo articulo2 = new Articulo(1002, "Aspirador");
        Articulo articulo3 = new Articulo(1003, "Televisor");
        Articulo articulo4 = new Articulo(1004, "Radio");
        Articulo articulo5 = new Articulo(1005, "Calentador");
        Articulo articulo6 = new Articulo(1006, "Batidor");

        TreeSet<Articulo> articulos = new TreeSet<Articulo>();

        articulos.add(articulo4);
        articulos.add(articulo2);
        articulos.add(articulo3);
        articulos.add(articulo6);
        articulos.add(articulo1);
        articulos.add(articulo5);

        for (Articulo articulo : articulos) {
            System.out.println(articulo.getDescripcion());
        }

        // ----------------NUEVA INSTANCIA ARTÍCULO SIN PARÁMETROS-------------------------
        System.out.println("\nArtículos ordenados por la descripción:");

        Articulo articuloComp = new Articulo();

        TreeSet<Articulo> articulos2 = new TreeSet<Articulo>(articuloComp);

        articulos2.add(articulo6);
        articulos2.add(articulo5);
        articulos2.add(articulo3);
        articulos2.add(articulo1);
        articulos2.add(articulo2);
        articulos2.add(articulo4);

        for (Articulo articulo : articulos2) {
            System.out.println(articulo.getDescripcion());
        }

        // ----------------------------NUEVA COLECCIÓN----------------------------------
```

```java
        System.out.println("\nArtículos con clase interna ordenados por la descripción:");

        TreeSet<Articulo> articulos3 = new TreeSet<Articulo>(new Comparator<Articulo>() {
                @Override
                public int compare(Articulo o1, Articulo o2) {
                    return o1.getSoloDescripcion().compareTo(o2.getSoloDescripcion());
                }
            }
            );

        articulos3.add(articulo2);
        articulos3.add(articulo4);
        articulos3.add(articulo3);
        articulos3.add(articulo1);
        articulos3.add(articulo6);
        articulos3.add(articulo5);

        for (Articulo articulo : articulos3) {
            System.out.println(articulo.getDescripcion());
        }
    }
}

// -------------------CLASE ARTÍCULO-------------------------
class Articulo implements Comparable<Articulo>, Comparator<Articulo> {
    private int nro;
    private String descripcion;

    public Articulo(int nro, String descripcion) {
        this.nro = nro;
        this.descripcion = descripcion;
    }

    public Articulo() {

    }

    public String getDescripcion() {
        return "Número del artículo: " + nro + ", Descripción: " + descripcion;
    }

    public String getSoloDescripcion() {
        return  descripcion;
    }

    // ----------------MÉTODO DE COMPARABLE--------------------
    @Override
    public int compareTo(Articulo obj) {
        return nro - obj.nro;
    }

    // ----------------MÉTODO DE COMPARATOR--------------------
    @Override
    public int compare(Articulo obj1, Articulo obj2) {
        return obj1.descripcion.compareTo(obj2.descripcion);
    }
}
```

```java
package EjerciciosBloque5;

import java.util.HashMap;
import java.util.Map;

public class _06_Map_HashMap {

    public static void main(String[] args) {

        // ---------------------CREAR COLECCIÓN----------------------------
        System.out.println("\nImprimir la colección:");
        HashMap<String, Empleado> empleados = new HashMap<String, Empleado>();

        empleados.put("101", new Empleado("Ricardo"));
        empleados.put("102", new Empleado("María"));
        empleados.put("103", new Empleado("Annabel"));
        empleados.put("104", new Empleado("Andrés"));
        empleados.put("105", new Empleado("Diego"));
        empleados.put("106", new Empleado("Laura"));

        System.out.println(empleados);

        // ---------------------ELIMINAR UN EMPLEADO----------------------------
        System.out.println("\nEmpleado 104 eliminado:");

        empleados.remove("104");

        System.out.println(empleados);

        // ---------------------SUSTITUIR UN EMPLEADO----------------------------
        System.out.println("\nEmpleado 102 sustituido por Juana:");

        empleados.replace("102", new Empleado("Juana"));

        System.out.println(empleados);

        // ------------------IMPRIMIR COLECCIÓN CON ENTRYSET-------------------------
        System.out.println("\nColección impresa con entrySet():");

        System.out.println(empleados.entrySet());

        // ----------------IMPRIMIR COLECCIÓN CON FOREACH Y ENTRY-----------------------
        System.out.println("\nColección impresa con foreach y Map.Entry:");

        for (Map.Entry<String, Empleado> empleado : empleados.entrySet()) {
            String clave = empleado.getKey();
            Empleado data = empleado.getValue();
            System.out.println("Clave= " +  clave + " | Data= " + data);
        }
    }
}

// -----------------CLASE EMPLEADO-------------------------
class Empleado {

    private String nombre;
    private double sueldo;

    public Empleado(String nombre) {

        this.nombre = nombre;
        sueldo = 2000;
```

```
63      }
64
65      public String toString() {
66          return " Empleado: " + nombre + ", Sueldo: " + sueldo;
67      }
68  }
```

```java
package EjerciciosBloque5;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.net.InetAddress;
import java.net.ServerSocket;
import java.net.Socket;
import java.net.UnknownHostException;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;

public class _07_Sockets_Cliente {

    public static void main(String[] args) {

        MarcoChat marco = new MarcoChat();
    }
}
// ----------------------MARCO VENTANA UI--------------------------
class MarcoChat extends JFrame {

    private int screenX, screenY, xCliente, yCliente, xMarco, yMarco;

    public MarcoChat() {
        xMarco = 300;
        yMarco = 400;
        Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
        screenX = (int) screenSize.getWidth();
        screenY = (int) screenSize.getHeight();
        xCliente = (screenX/2) + xMarco;
        yCliente = (screenY/2) - 100;
        setBounds(xCliente, yCliente, xMarco, yMarco);

        setLayout(new BorderLayout());
        add(new LaminaEspacio(), BorderLayout.NORTH);
        add(new LaminaChat(), BorderLayout.CENTER);

        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

//----------------------LÁMINA ESPACIO VACÍO-------------------------
class LaminaEspacio extends JPanel {
    private JLabel espacio;
    public LaminaEspacio() {
        setBackground(new Color(204, 255, 204));
```

```
63          add(espacio = new JLabel("    "));
64      }
65  }
66
67  //----------------------LÁMINA VENTANA UI--------------------------
68  class LaminaChat extends JPanel {
69
70      private String nickUsuario;
71      private JLabel nickTitulo, nick, contactosTitulo;
72      private JComboBox contactos;
73      private JTextArea chat;
74      private JScrollPane vistaScroll;
75      private JTextField entrada;
76      private JButton btnEnviar;
77      private String myIp;
78
79      public LaminaChat() {
80
81          RecibirData dataIn = new RecibirData();
82
83          //----------------------UI--------------------------
84          setBackground(new Color(204, 255, 204));
85          add(nickTitulo = new JLabel("nick: "));
86          nickUsuario = JOptionPane.showInputDialog("Ingrese su nick");
87          add(nick = new JLabel(nickUsuario.toUpperCase()));
88          add(contactosTitulo = new JLabel("  |  contactos: "));
89          add(contactos = new JComboBox());
90          add(chat = new JTextArea(14,20));
91          chat.setLineWrap(true);
92          chat.setEnabled(false);
93          add(vistaScroll = new JScrollPane(chat));
94          add(entrada = new JTextField(20));
95          add(btnEnviar = new JButton("Enviar"));
96
97          //----------------------MYIP--------------------------
98          try {
99              myIp = InetAddress.getLocalHost().getHostAddress();
100         } catch (UnknownHostException e) {
101             System.out.println(e.getMessage());
102             e.printStackTrace();
103         }
104
105         //----------------------OYENTES--------------------------
106         entrada.addActionListener(new AccionEnviarData());
107         btnEnviar.addActionListener(new AccionEnviarData());
108
109         //----------------------DATA--------------------------
110         SenialConexion signal = new SenialConexion();
111     }
112
113     //-------------------AVISAR AL SERVIDOR DE USUARIO CONECTADO----------------------
114     class SenialConexion implements Runnable {
115
116         private String nickSignal, mensajeSignal, ipSignal;
117         private Thread hilo;
118
119         public SenialConexion() {
120             nickSignal = nickUsuario;
121             mensajeSignal = "onlinex";
122             ipSignal = myIp;
123
124             hilo = new Thread(this);
```

```
125             hilo.start();
126         }
127 
128         @Override
129         public void run() {
130 
131             PaqueteData paqueteSignal = new PaqueteData(mensajeSignal, nickSignal,
    ipSignal);
132 
133             try {
134                 // se abre ecnhufe
135                 Socket enchufeSignal = new Socket(ipSignal, 9999);
136                 // se abre flujo
137                 ObjectOutputStream flujoSignal = new
    ObjectOutputStream(enchufeSignal.getOutputStream());
138                 // se monta data
139                 flujoSignal.writeObject(paqueteSignal);
140 
141                 // se cierran enchufes y flujos
142                 enchufeSignal.close();
143                 flujoSignal.close();
144 
145             } catch (UnknownHostException e) {
146                 // TODO Auto-generated catch block
147                 e.printStackTrace();
148             } catch (IOException e) {
149                 // TODO Auto-generated catch block
150                 e.printStackTrace();
151             }
152         }
153     }
154 
155     //----------------------ACCIÓN ENVIAR MENSAJES-------------------------
156     public class AccionEnviarData implements ActionListener {
157         @Override
158         public void actionPerformed(ActionEvent e) {
159 
160             if (e.getSource() == btnEnviar || e.getSource() == entrada) {
161 
162                 String mensaje = entrada.getText();
163                 String nick = nickUsuario;
164                 String contacto = myIp;
165 
166                 try {
167                     // ase abre enchufe
168                     Socket enchufe = new Socket(myIp, 9999);
169                     // se abre flujo
170                     ObjectOutputStream flujoOut = new
    ObjectOutputStream(enchufe.getOutputStream());
171                     // se instancia la data a enviar
172                     PaqueteData paqueteOut = new PaqueteData(mensaje, nick, contacto);
173                     // se monta la data al flujo
174                     flujoOut.writeObject(paqueteOut);
175 
176                     // se cierran enchufes y flujos
177                     enchufe.close();
178                     flujoOut.close();
179 
180                 } catch (UnknownHostException e1) {
181                     System.out.println(e1.getMessage());
182                     e1.printStackTrace();
183                 } catch (IOException e1) {
```

```
184                     System.out.println(e1.getMessage());
185                     e1.printStackTrace();
186                 }
187 
188                 //----------------------------ACCIÓN LOCAL-----------------------------
189                 chat.append("\n" + nickUsuario.toUpperCase() + ": " + entrada.getText());
190                 entrada.setText("");
191                 entrada.requestFocus();
192 
193                 System.out.println("msj enviado | my ip: " + myIp);
194             }
195         }
196     }
197 
198     //----------------------RECIBIR MENSAJES-------------------------
199     class RecibirData implements Runnable {
200 
201         private String nickIn, mensajeIn, ipIn;
202         private Thread hilo;
203         private PaqueteData paqueteIn;
204 
205         public RecibirData() {
206             PaqueteData paqueteIn = new PaqueteData(mensajeIn, nickIn, ipIn);
207             hilo = new Thread(this);
208             hilo.start();
209         }
210 
211         @Override
212         public void run() {
213 
214             try {
215                 ServerSocket enchufeServer = new ServerSocket(9090);
216 
217                 while (true) {
218 
219                     // se abre enchufe
220                     Socket enchufeIn = enchufeServer.accept();
221                     // se abre flujo
222                     ObjectInputStream flujoIn = new
    ObjectInputStream(enchufeIn.getInputStream());
223 
224                     paqueteIn = (PaqueteData) flujoIn.readObject();
225                     nickIn = paqueteIn.getNick();
226                     mensajeIn = paqueteIn.getMensaje();
227                     ipIn = paqueteIn.getIp();
228 
229                     System.out.println("Nick recibido: " +  paqueteIn.getNick());
230                     System.out.println("Mensaje recibido: " +  paqueteIn.getMensaje());
231 
232                     // se cierran enchufes y flujos
233                     enchufeIn.close();
234                     flujoIn.close();
235 
236                     if (!mensajeIn.equals("onlinex")) {
237                         System.out.println("Entró a mensaje recibido diferente de
    onlinex");
238                         //----------------------AGREGAR DATA AL
    CHAT------------------------
239                         chat.append("\nRecibidox: " + nickIn + ": " + mensajeIn);
240 
241                     } else {
242                         contactos.addItem(ipIn);
```

```
243                         System.out.println("Ip en bucle else: " + ipIn);
244                     }
245                 }
246 
247             } catch (IOException e) {
248                 System.out.println("Mensaje Error: " + e.getMessage());
249                 e.printStackTrace();
250             } catch (ClassNotFoundException e) {
251                 System.out.println("Mensaje Error: " + e.getMessage());
252                 e.printStackTrace();
253             }
254         }
255     }
256 }
257 //----------------------CLASE PAQUETE DATA-------------------------
258 class PaqueteData implements Serializable {
259     private String mensaje, nick, ip;
260 
261     public PaqueteData(String mensaje, String nick, String ip) {
262         this.mensaje = mensaje;
263         this.nick = nick;
264         this.ip = ip;
265     }
266 
267     public String getMensaje() {
268         return mensaje;
269     }
270 
271     public String getNick() {
272         return nick;
273     }
274 
275     public String getIp() {
276         return ip;
277     }
278 
279     public void setMensaje(String mensaje) {
280         this.mensaje = mensaje;
281     }
282 
283     public void setNick(String nick) {
284         this.nick = nick;
285     }
286 
287     public void setIp(String ip) {
288         this.ip = ip;
289     }
290 }
```

```java
package EjerciciosBloque5;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Toolkit;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.ArrayList;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

public class _07_Sockets_Servidor {

    public static void main(String[] args) {

        MarcoServidor marco = new MarcoServidor();
    }
}
// ----------------------MARCO VENTANA UI--------------------------
class MarcoServidor extends JFrame {

    private int screenX, screenY, xCliente, yCliente, xMarco, yMarco;

    public MarcoServidor() {
        xMarco = 300;
        yMarco = 400;
        Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
        screenX = (int) screenSize.getWidth();
        screenY = (int) screenSize.getHeight();
        xCliente = (screenX/2) + xMarco;
        yCliente = (screenY/2) - (yMarco + 100);
        setBounds(xCliente, yCliente, xMarco, yMarco);

        setLayout(new BorderLayout());
        add(new LaminaEspacioS(), BorderLayout.NORTH);
        add(new LaminaServidor(), BorderLayout.CENTER);

        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

//----------------------LÁMINA ESPACIO VACÍO--------------------------
class LaminaEspacioS extends JPanel {
    private JLabel espacio;
    public LaminaEspacioS() {
        setBackground(new Color(204, 255, 204));
        add(espacio = new JLabel("    "));
    }
}

//----------------------LÁMINA VENTANA UI--------------------------
class LaminaServidor extends JPanel {

    private JLabel titulo;
```

```
63      private JTextArea chat;
64      private JScrollPane vistaScroll;
65
66      public LaminaServidor() {
67          //----------------------UI-------------------------
68          setBackground(new Color(204, 255, 204));
69          add(titulo = new JLabel(":::SERVIDOR:::"));
70          add(chat = new JTextArea(18,26));
71          chat.setLineWrap(true);
72          chat.setEnabled(false);
73          add(vistaScroll = new JScrollPane(chat));
74
75          RecibirData recibirData = new RecibirData();
76      }
77
78      //----------------------RECIBIR DATA-------------------------
79      public class RecibirData implements Runnable {
80
81          private String mensaje, nick, ip;
82          private Thread hilo = new Thread(this);  // NOTA: el hilo siempre debe apuntar al
    entorno ejem: this
83          private ArrayList<PaqueteData> listaUsuarios;
84
85          public RecibirData() {
86              listaUsuarios = new ArrayList<PaqueteData>();
87              hilo.start();
88          }
89
90          @Override
91          public void run() {
92
93              try {
94                  System.out.println("Entró al Run");
95                  // se crea enchufe servidor
96                  ServerSocket enchufeServer = new ServerSocket(9999);
97                  //
98                  PaqueteData paqueteIn = new PaqueteData(mensaje, nick, ip);
99
100                 while (true) {
101
102                     // se abre el enchufe servidor
103                     Socket enchufeIn = enchufeServer.accept();
104                     //
105                     ObjectInputStream flujoIn = new
    ObjectInputStream(enchufeIn.getInputStream());
106                     //
107                     paqueteIn = (PaqueteData) flujoIn.readObject();
108                     //
109                     mensaje = paqueteIn.getMensaje();
110                     nick = paqueteIn.getNick();
111                     ip = paqueteIn.getIp();
112
113                     // se cierran enchufes y flujos
114 //                  enchufeServer.close();  // NOTA: el enchufe servidor NO se debe cerrar
    nunca!!!
115                     enchufeIn.close();
116                     flujoIn.close();
117
118                     if (!mensaje.equals("onlinex")) {
119                         //----------------------------ACCIÓN
    LOCAL-----------------------------
120                         chat.append("\n" + nick.toUpperCase() + ": " + mensaje);
```

```
121
122                         //----------------------------ENVIAR
    DATA-----------------------------
123                         // se abre enchufe
124                         Socket enchufeOut = new Socket(ip, 9090);
125                         // se abre flujo
126                         ObjectOutputStream flujoOut = new
    ObjectOutputStream(enchufeOut.getOutputStream());
127                         flujoOut.writeObject(paqueteIn);
128
129                         // se cierran enchufes y flujos
130                         enchufeOut.close();
131                         flujoOut.close();
132
133                     } else {
134                         //-------------ALMACENAR Y ENVIAR SEÑAL USUARIO
    CONECTADO-----------------
135                         listaUsuarios.add(paqueteIn);
136
137                         // se abre enchufe
138                         Socket enchufeOut = new Socket(ip, 9090);
139                         // se abre flujo
140                         ObjectOutputStream flujoOut = new
    ObjectOutputStream(enchufeOut.getOutputStream());
141                         flujoOut.writeObject(paqueteIn);
142
143                         // se cierran enchufes y flujos
144                         enchufeOut.close();
145                         flujoOut.close();
146
147                     }
148                 }
149
150             } catch (IOException e) {
151                 System.out.println(e.getMessage());
152                 e.printStackTrace();
153             } catch (ClassNotFoundException e) {
154                 System.out.println(e.getMessage());
155                 e.printStackTrace();
156             }
157         }
158     }
159 }
```