

\_1\_EntradaEjemplo1.java

```
1 package EjerciciosPrimerosPasos;
2
3 import java.util.Scanner;
4
5 public class _1_EntradaEjemplo1 {
6
7     public static void main(String[] args) {
8
9         Scanner entrada = new Scanner(System.in);
10
11         System.out.println("por favor ingrese su nombre");
12
13 //      String nombre = entrada.next(); // el método next() sólo devuelve una palabra y no
        toda la línea
14         String nombre = entrada.nextLine();
15
16         System.out.println("por favor ingrese su edad");
17
18         int edad = entrada.nextInt();
19
20         System.out.println("Hola " + nombre + " el próximo año tendrás " + (edad+1) + "
        años");
21
22     }
23 }
24
```

\_2\_EntradaEjemplo2.java

```
1 package EjerciciosPrimerosPasos;
2
3 import javax.swing.JOptionPane;
4
5 public class _2_EntradaEjemplo2 {
6
7     public static void main(String[] args) {
8
9         String nombre = JOptionPane.showInputDialog("Ingrese su nombre por favor");
10
11         int edad = Integer.parseInt(JOptionPane.showInputDialog("Ingrese su edad"));
12
13         System.out.println("Hola " + nombre + " el próximo año tendrás " + edad + " años.");
14     }
15
16 }
17
```

\_3\_EntradaNumeros.java

```
1 package EjerciciosPrimerosPasos;
2
3 import java.text.DecimalFormat;
4 import javax.swing.JOptionPane;
5
6 public class _3_EntradaNumeros {
7
8     public static void main(String[] args) {
9
10         double numero = 0.0;
11
12         while (numero < 1.0 || numero > 100.0) {
13             numero = Double.parseDouble(JOptionPane.showInputDialog("Ingrese un número del 1
14 al 100 por favor"));
15         };
16
17         DecimalFormat decimal2 = new DecimalFormat("#.##"); // Clase para convertir en
18 decimal
19         System.out.println("La raíz cuadrada de " + numero + " es: " +
20 decimal2.format(Math.sqrt(numero)));
21     }
22 }
```

\_4\_EvaluaEdad.java

```
1 package EjerciciosPrimerosPasos;
2
3 import java.util.Scanner;
4
5 public class _4_EvaluaEdad {
6
7     public static void main(String[] args) {
8
9         Scanner entrada = new Scanner(System.in);
10
11         int edad = 0;
12
13         while (edad < 1 || edad > 120) {
14             System.out.println("Ingrese su edad por favor");
15             edad = entrada.nextInt();
16         }
17
18         if (edad >= 18) {
19             if (edad <= 40) {
20                 System.out.println("Usted es Joven");
21             } else if (edad <= 64) {
22                 System.out.println("Usted es Maduro");
23             } else if (edad >= 65) {
24                 System.out.println("Usted es Mayor");
25             }
26         } else {
27             System.out.println("Usted es Adolescente");
28         }
29     }
30 }
31
```

\_5\_CalculoDeAreas.java

```
1 package EjerciciosPrimerosPasos;
2
3 import java.text.DecimalFormat;
4 import java.util.Scanner;
5
6 import javax.swing.JOptionPane;
7
8 public class _5_CalculoDeAreas {
9
10     public static void main(String[] args) {
11
12         Scanner entrada = new Scanner(System.in);
13
14         DecimalFormat decimal = new DecimalFormat("#.##");
15
16         int numero = 0;
17
18         while (numero < 1 || numero > 4) {
19
20             System.out.println("Seleccione un número por favor: \n 1 - Cuadrado \n 2 - Rectángulo \n 3 - Triángulo \n 4 - Círculo");
21
22             numero = entrada.nextInt();
23         }
24
25         if (numero == 1) {
26
27             double ladoCuadrado = Double.parseDouble(JOptionPane.showInputDialog("Ingrese el lado en cm"));
28             System.out.println("El área del cuadrado es: " + ladoCuadrado*ladoCuadrado + "cm2");
29
30         } else if (numero == 2) {
31
32             double baseRect = Double.parseDouble(JOptionPane.showInputDialog("Ingrese la base en cm"));
33             double alturaRect = Double.parseDouble(JOptionPane.showInputDialog("Ingrese la altura en cm"));
34             System.out.println("El área del rectángulo es: " + baseRect*alturaRect + "cm2");
35
36         } else if (numero == 3) {
37
38             double baseTri = Double.parseDouble(JOptionPane.showInputDialog("Ingrese la base en cm"));
39             double alturaTri = Double.parseDouble(JOptionPane.showInputDialog("Ingrese la altura en cm"));
40             System.out.println("El área del triángulo es: " + (baseTri*alturaTri)/2 + "cm2");
41
42         } else if (numero == 4) {
43
44             double radio = Double.parseDouble(JOptionPane.showInputDialog("Ingrese el radio en cm"));
45             System.out.println("El área del círculo es: " + decimal.format(Math.PI*(radio*radio)) + "cm2");
46         }
47     }
48 }
49
```

`_6_VerificacionDeEmail.java`

```
1 package EjerciciosPrimerosPasos;
2
3 import javax.swing.JOptionPane;
4
5 public class _6_VerificacionDeEmail {
6
7     public static void main(String[] args) {
8
9         String contrasenia = "bilbao";
10
11         boolean comprobar = false;
12
13         while (comprobar == false) {
14
15             String dataIn = JOptionPane.showInputDialog("Ingrese la contraseña por favor");
16
17             if (dataIn.contentEquals(contrasenia) == false) {
18
19                 System.out.println("Contraseña incorrecta, corrija e ingrese nuevamente");
20
21             } else {
22
23                 System.out.println("Contraseña correcta, bienvenido!");
24                 comprobar = true;
25             }
26         }
27     }
28 }
29
```

\_7\_AdivinaNumero.java

```
1 package EjerciciosPrimerosPasos;
2
3 import javax.swing.JOptionPane;
4
5 public class _7_AdivinaNumero {
6
7     public static void main(String[] args) {
8
9         // Random random = new Random(); // Forma más sencilla de obtener un número random
10        //
11        // int numero = random.nextInt(100);
12
13        double aleatorioDouble = Math.random() * 100 + 1;
14
15        int aleatorioInt = (int) aleatorioDouble;
16
17        int numeroIn = 0;
18
19        int intentos = 0;
20
21        while (numeroIn != aleatorioInt) {
22
23            intentos++;
24
25            numeroIn = Integer.parseInt(JOptionPane.showInputDialog("Ingrese un número entre
el 1 y el 100"));
26
27            if (numeroIn < aleatorioInt) {
28
29                System.out.println("Ingrese un número más Alto que: " + numeroIn);
30
31            } else if (numeroIn > aleatorioInt) {
32
33                System.out.println("Ingrese un número más Bajo que: " + numeroIn);
34            }
35        }
36
37        System.out.println("Has adivinado el número, felicitaciones! lo has hecho en: " +
intentos + " intentos.");
38    }
39 }
40
```

\_8\_AdivinaNumeroDowhile.java

```
1 package EjerciciosPrimerosPasos;
2
3 import java.util.Random;
4 import javax.swing.JOptionPane;
5
6 public class _8_AdivinaNumeroDowhile {
7
8     public static void main(String[] args) {
9
10         Random random = new Random();
11
12         int aleatorio = random.nextInt(100);
13
14         int nroIn = 0;
15
16         int intentos = 0;
17
18         do {
19             nroIn = Integer.parseInt(JOptionPane.showInputDialog("Ingrese un número del 1 al
20 100"));
21
22             intentos++;
23
24             if (nroIn > aleatorio) {
25                 System.out.println("Ingrese un número más Bajo que: " + nroIn);
26
27             } else if (nroIn < aleatorio) {
28                 System.out.println("Ingrese un número más Alto que: " + nroIn);
29
30             } else {
31
32                 System.out.println("Has adivinado el número, felicitaciones! lo has logrado
33 en: " + intentos + " intentos.");
34             }
35
36         } while (nroIn != aleatorio || nroIn < 1 || nroIn > 100);
37     }
38 }
39
```



\_9\_PesoIdeal.java

```
1 package EjerciciosPrimerosPasos;
2
3 import javax.swing.JOptionPane;
4
5 public class _9_PesoIdeal {
6
7     public static void main(String[] args) {
8
9         String dataIn = "";
10
11         int altura = 0;
12
13         while (!dataIn.equalsIgnoreCase("H") || !dataIn.equalsIgnoreCase("M")) {
14
15             dataIn = JOptionPane.showInputDialog("Por favor ingrese su género: H / M");
16
17             if (dataIn.equalsIgnoreCase("H")) {
18
19                 altura = Integer.parseInt(JOptionPane.showInputDialog("Ingrese su Altura en
cm (ejem: 170)"));
20
21                 System.out.println("El peso ideal para un hombre con altura " + altura + "cm
es de: " + (altura-110) + "kg");
22
23                 break;
24
25             } else if (dataIn.equalsIgnoreCase("M")) {
26
27                 altura = Integer.parseInt(JOptionPane.showInputDialog("Ingrese su Altura en
cm (ejem: 170)"));
28
29                 System.out.println("El peso ideal para una mujer con altura " + altura + "cm
es de: " + (altura-110) + "kg");
30
31                 break;
32             }
33         }
34     }
35 }
36
```

\_10\_UsoBucleFor.java

```
1 package EjerciciosPrimerosPasos;
2
3 public class _10_UsoBucleFor {
4
5     public static void main(String[] args) {
6
7         for (int i = 0; i < 10; i++) {
8
9             System.out.println("La suma progresiva de i++ es: " + i);
10        }
11
12        System.out.println("-----");
13
14        for (int i = 0; i < 20; i+=2) {
15
16            System.out.println("La suma progresiva con salto de i+=2 es: " + i);
17        }
18
19        System.out.println("-----");
20
21        for (int i = 30; i > 20; i--) {
22
23            System.out.println("La resta progresiva de i-- es: " + i);
24        }
25    }
26 }
27
```

\_11\_ComprobarDireccionEmail.java

```
1 package EjerciciosPrimerosPasos;
2
3 import javax.swing.JOptionPane;
4
5 public class _11_ComprobarDireccionEmail {
6
7     public static void main(String[] args) {
8
9         int arroba = 0;
10        boolean punto = false;
11        boolean emailTest = false;
12
13        while (emailTest == false) {
14
15            String email = JOptionPane.showInputDialog("Ingrese su email por favor");
16
17            for (int i = 0; i < email.length(); i++) {
18                if (email.charAt(i) == '@') {
19                    arroba++;
20                } else if (email.charAt(i) == '.') {
21                    punto = true;
22                }
23            }
24            if (arroba == 1 && punto) {
25                System.out.println("Email correcto, bienvenido!");
26                emailTest = true;
27                break;
28            } else {
29                System.out.println("Email incorrecto, corrija e ingrese nuevamente");
30                arroba = 0;
31            }
32        }
33    }
34 }
35
```

\_12\_Factorial.java

```
1 package EjerciciosPrimerosPasos;
2
3 import javax.swing.JOptionPane;
4
5 public class _12_Factorial {
6
7     public static void main(String[] args) {
8
9         int numero = 0;
10
11         int factorial = 1;
12
13         while (numero < 1 || numero > 10) {
14
15             numero = Integer.parseInt(JOptionPane.showInputDialog("Por favor ingrese un
16 número del 1 al 10"));
17         }
18
19         for (int i = 1; i < numero + 1; i++) {
20
21             factorial = factorial * i;
22         }
23
24         System.out.println("El factorial del número " + numero + " es: " + factorial);
25     }
26 }
```

\_13\_UsoArrays.java

```
1 package EjerciciosPrimerosPasos;
2
3 public class _13_UsoArrays {
4
5     public static void main(String[] args) {
6
7         int[] matrix1 = new int[5];
8
9         matrix1[0] = 19;
10        matrix1[1] = 25;
11        matrix1[2] = -30;
12        matrix1[3] = 68;
13        matrix1[4] = 45;
14
15        for (int i = 0; i < matrix1.length; i++) {
16
17            System.out.println("El valor del índice " + i + " es: " + matrix1[i]);
18        }
19
20        System.out.println("-----");
21
22        int[] matrix2 = {19,25,-30,68,45};
23
24        for (int i = 0; i < matrix2.length; i++) {
25
26            System.out.println("El valor del índice " + i + " es: " + matrix2[i]);
27        }
28    }
29 }
30
```

\_14\_UsoArraysII.java

```
1 package EjerciciosPrimerosPasos;
2
3 import java.util.Random;
4 import javax.swing.JOptionPane;
5
6 public class _14_UsoArraysII {
7
8     public static void main(String[] args) {
9
10         String [] paises1 = {"Colombia", "Venezuela", "Chile", "Ecuador", "Perú", "Brasil",
11 "Argentina"};
12
13         for (int i = 0; i < paises1.length; i++) {
14             System.out.println(paises1[i]);
15         }
16
17         System.out.println("_____");
18
19         System.out.println("Total: " + paises1.length + " países.");
20
21         System.out.println("-----");
22
23         String [] paises2 = new String[6];
24
25         for (int i = 0; i < paises2.length; i++) {
26             paises2 [i] = JOptionPane.showInputDialog("Por favor ingrese un país");
27
28             System.out.println(paises2[i]);
29         }
30
31         System.out.println("_____");
32
33         System.out.println("Total: " + paises2.length + " países.");
34
35         System.out.println("-----");
36
37         int [] aleatorios = new int[150];
38
39         Random random = new Random();
40
41         for (int i = 0; i < aleatorios.length; i++) {
42             aleatorios[i] = random.nextInt(98) + 1;
43
44             System.out.print(aleatorios[i] + " ");
45         }
46     }
47 }
48
49 }
50
```

\_15\_ArraysBidimensionales.java

```
1 package EjerciciosPrimerosPasos;
2
3 public class _15_ArraysBidimensionales {
4
5     public static void main(String[] args) {
6
7         int [][] matrix1 = new int[4][5];
8
9         matrix1[0][0] = 10;
10        matrix1[0][1] = 22;
11        matrix1[0][2] = 75;
12        matrix1[0][3] = 33;
13        matrix1[0][4] = 84;
14
15        matrix1[1][0] = 82;
16        matrix1[1][1] = 15;
17        matrix1[1][2] = 67;
18        matrix1[1][3] = 35;
19        matrix1[1][4] = 78;
20
21        matrix1[2][0] = 58;
22        matrix1[2][1] = 42;
23        matrix1[2][2] = 19;
24        matrix1[2][3] = 37;
25        matrix1[2][4] = 73;
26
27        matrix1[3][0] = 13;
28        matrix1[3][1] = 46;
29        matrix1[3][2] = 82;
30        matrix1[3][3] = 75;
31        matrix1[3][4] = 36;
32
33        for (int i = 0; i < matrix1.length; i++) {
34
35            for (int j = 0; j < matrix1.length + 1; j++) {
36
37                System.out.print(matrix1[i][j] + " ");
38            }
39
40            System.out.println("");
41        }
42
43        System.out.println("-----");
44
45        int [][] matrix2 = {{10,22,75,33,84}, {82,15,67,35,78}, {58,42,19,37,73},
46        {13,46,82,75,36}};
47
48        for (int i = 0; i < matrix2.length; i++) {
49
50            for (int j = 0; j < matrix2.length + 1; j++) {
51
52                System.out.print(matrix2[i][j] + " ");
53            }
54
55            System.out.println(" ");
56        }
57 }
58
```

\_16\_Array2D.java

```
1 package EjerciciosPrimerosPasos;
2
3 import java.text.DecimalFormat;
4
5 public class _16_Array2D {
6
7     public static void main(String[] args) {
8
9         double [][] matrix = new double[6][5];
10
11         double base = 10000.0;
12
13         double interes = 1.1;
14
15         DecimalFormat decimal = new DecimalFormat("#.##");
16
17         for (int i = 0; i < matrix.length; i++) {
18
19             for (int j = 0; j < matrix.length - 1; j++) {
20
21                 matrix[i][j] = base;
22
23                 base *= interes;
24
25                 System.out.print(decimal.format(matrix[i][j]) + " ");
26             }
27
28             base = 10000.0;
29
30             interes += 0.01;
31
32             System.out.println(" ");
33         }
34     }
35 }
36
```



```
1 package EjerciciosP00;
2
3 public class _1_PruebaFinal {
4
5     public static void main(String[] args) {
6
7         Empleado empleados[] = new Empleado[3];
8
9         empleados[0] = new Empleado("José de los Campos");
10        empleados[1] = new Empleado("María Arreaza");
11        empleados[2] = new Empleado("Diego de la Torre");
12
13        empleados[1].setDepartamento("Marketing");
14        empleados[2].setDepartamento("Ventas");
15
16        for (int i=0; i<empleados.length; i++) {
17
18            System.out.println(empleados[i].getEmpleado());
19        }
20    }
21 }
22
23 class Empleado {
24
25     private String nombre;
26     private String departamento;
27     private int id = 0;
28     private static int idFijo = 1;
29
30     public Empleado(String nombre) {
31
32         this.nombre = nombre;
33         departamento = "Administración";
34         id = idFijo;
35         idFijo++;
36     }
37
38     public String getEmpleado () {
39
40         return "Id: " + id + " - Empleado: " + nombre + ", Departamento: " + departamento;
41     }
42
43     public void setDepartamento (String departamento) {
44
45         this.departamento = departamento;
46     }
47 }
48 }
```

```
1 package EjerciciosP00;
2
3 class Coche {
4     private int ruedas;
5     private double largo;
6     private double ancho;
7     private int motor;
8     private double pesoPlataforma;
9     private String color;
10    private String asientosCuero;
11    private String climatizador;
12
13    public Coche(String color, String asientosCuero, String climatizador) {
14        ruedas = 4;
15        largo = 250.0;
16        ancho = 150.0;
17        motor = 3600;
18        pesoPlataforma = 500.0;
19        this.color = color;
20        this.asientosCuero = asientosCuero;
21        this.climatizador = climatizador;
22    }
23
24    public double getPesoPlataforma () {
25        return pesoPlataforma;
26    }
27
28    public String getAsientosCuero () {
29        boolean confirmarAsientos = false;
30        if (asientosCuero.equalsIgnoreCase("Si")) {
31            confirmarAsientos = true;
32        } else if (asientosCuero.equalsIgnoreCase("No")) {
33            confirmarAsientos = false;
34        }
35        if (confirmarAsientos) {
36            return "Si tiene";
37        } else {
38            return "No tiene";
39        }
40    }
41
42    public String getClimatizador () {
43        if (climatizador.equalsIgnoreCase("Si")) {
44            return "Si tiene";
45        } else {
46            return "No tiene";
47        }
48    }
49
50    public double getPesoTotal () {
51        double pesoCarroceria = 1000;
52        double pesoTotal = pesoPlataforma + pesoCarroceria;
53        if (climatizador.equalsIgnoreCase("Si")) {
54            pesoTotal += 60;
55        }
56        return pesoTotal;
57    }
58
59    public double getPrecioTotal () {
60        double precioFinal = 10000;
61        if (asientosCuero.equalsIgnoreCase("Si")) {
62            precioFinal += 600;
```

\_2\_Coche.java

```
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 +
72     "m, Motor: " + motor + "rpm, Peso Plataforma: " + pesoPlataforma + "kg, Color: " + color;
73 }
```

\_3\_UsoCoche.java

```
1 package EjerciciosP00;
2
3 import javax.swing.JOptionPane;
4
5 public class _3_UsoCoche {
6
7     public static void main(String[] args) {
8
9         String color = JOptionPane.showInputDialog("Ingrese color del coche");
10        String asientosCuero = JOptionPane.showInputDialog("Tiene asientos de cuero?
Si/No");
11        String climatizador = JOptionPane.showInputDialog("Tiene climatizador? Si/No");
12
13        Coche coche = new Coche(color, asientosCuero, climatizador);
14
15        System.out.println(coche.getCoche() + ", Asientos de Cuero: " +
coche.getAsientosCuero() + ", Climatizador: " + coche.getClimatizador() + ", Peso Total: " +
coche.getPesoTotal() + "kg, Precio Total: " + coche.getPrecioTotal() + "€.");
16
17    }
18
19 }
20
```

\_4\_Furgoneta.java

```
1 package EjerciciosP00;
2
3 class Furgoneta extends Coche {
4
5     private double capacidadCarga;
6     private int plazasExtra;
7     private String climatizadorFurgo;
8     private String asientosCueroFurgo;
9
10    public Furgoneta(String color, String asientosCuero, String climatizador, double
        capacidadCarga, int plazasExtra) {
11        super(color, asientosCuero, climatizador);
12        this.capacidadCarga = capacidadCarga;
13        this.plazasExtra = plazasExtra;
14        climatizadorFurgo = climatizador;
15        asientosCueroFurgo = asientosCuero;
16    }
17
18    public double getPesoTotalFurgo () {
19        double pesoTotalFurgo = 0;
20        double pesoCarroceríaFurgo = 2000;
21        double plazasExtraFurgo = plazasExtra * 20;
22        pesoTotalFurgo = pesoCarroceríaFurgo + plazasExtraFurgo + getPesoPlataforma();
23        if (climatizadorFurgo.equalsIgnoreCase("Si")) {
24            pesoTotalFurgo += 60.0;
25        }
26        return pesoTotalFurgo;
27    }
28
29    public double getPrecioTotalFurgo () {
30        double precioTotalFurgo = 14000;
31        double asientosExtraFurgo = plazasExtra * 200;
32        if (climatizadorFurgo.equalsIgnoreCase("Si")) {
33            precioTotalFurgo += 300;
34        }
35        if (asientosCueroFurgo.equalsIgnoreCase("Si")) {
36            precioTotalFurgo += 600;
37        }
38        return precioTotalFurgo + asientosExtraFurgo;
39    }
40
41    public String getFurgoneta() {
42        return "Capacidad de carga: " + capacidadCarga + "kg y " + plazasExtra + " plazas
        extra";
43    }
44 }
45
```

\_5\_UsoVehículo.java

```
1 package EjerciciosP00;
2
3 import javax.swing.JOptionPane;
4
5 public class _5_UsoVehículo {
6
7     public static void main(String[] args) {
8
9         String colorCoche = JOptionPane.showInputDialog("Ingrese color del coche");
10        String asientosCueroCoche = JOptionPane.showInputDialog("Coche con asientos de
cuero? Si / No");
11        String climatizadorCoche = JOptionPane.showInputDialog("Coche climatizado? Si /
No");
12
13        Coche coche = new Coche(colorCoche, asientosCueroCoche, climatizadorCoche);
14
15        String colorFurgo = JOptionPane.showInputDialog("Ingrese color de la furgoneta");
16        String asientosCueroFurgo = JOptionPane.showInputDialog("Furgo con asientos de
cuero? Si / No");
17        String climatizadorFurgo = JOptionPane.showInputDialog("Furgo climatizada? Si /
No");
18        double capacidadCargaFurgo =
Double.parseDouble(JOptionPane.showInputDialog("Capacidad de carga? (ejem: 1500)"));
19        int plazasExtraFurgo = Integer.parseInt(JOptionPane.showInputDialog("Plazas extra?
(ejem: 3)"));
20
21        Furgoneta furgoneta = new Furgoneta(colorFurgo, asientosCueroFurgo,
climatizadorFurgo, capacidadCargaFurgo, plazasExtraFurgo);
22
23        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() + "€.");
24        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());
25    }
26 }
27
```

\_6\_UsoPersona.java

```
1 package EjerciciosP00;
2
3 public class _6_UsoPersona {
4
5     public static void main(String[] args) {
6
7         Empleado2 empleado = new Empleado2("Juan Sánchez", 2500, 2000, 11, 15, "Encargado de
planta");
8         Alumno alumno = new Alumno("Mauro Rujano", "Está cursando: ", "Biología Molecular");
9
10        System.out.println("Empleado: " + empleado.getPersona() + ", " +
empleado.getEmpleado2());
11        System.out.println("Alumno: " + alumno.getPersona() + alumno.getCarrera());
12    }
13 }
14
15 abstract class Persona {
16
17     private String nombre;
18     private String descripcion;
19
20     public Persona (String nombre, String descripcion) {
21         this.nombre = nombre;
22         this.descripcion = descripcion;
23     }
24
25     public String getNombre () {
26         return nombre;
27     }
28
29     public String getDescripcion () {
30         return descripcion;
31     }
32
33     public String getPersona () {
34         return "Nombre: " + nombre + ", Descripción: " + descripcion;
35     }
36 }
37
38 class Empleado2 extends Persona {
39     private double sueldo;
40     private int id = 0;
41     private static int idFijo = 1;
42 // private Date fechaContrato;
43     private int anio;
44     private int mes;
45     private int dia;
46
47     public Empleado2 (String nombre, double sueldo, int anio, int mes, int dia, String
descripcion) {
48         super(nombre, descripcion);
49         this.sueldo = sueldo;
50         id = idFijo;
51         idFijo++;
52 // GregorianCalendar fechaCalendario = new GregorianCalendar(anio, mes, dia);
53 // fechaContrato = fechaCalendario.getTime();
54         this.anio = anio;
55         this.mes = mes;
56         this.dia = dia;
57     }
58
59     public String getEmpleado2 () {
```

\_6\_UsoPersona.java

```
60         return "Id: " + id + ", Sueldo: " + sueldo + ", Fecha Contrato: " + dia + "/" + mes
61     + "/" + anio;
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
```



\_7\_EnumUsoTallas.java

```
1 package EjerciciosP00;
2
3 import java.util.Scanner;
4
5 public class _7_EnumUsoTallas {
6
7     public enum Tallas {
8
9         PEQUEÑA("S"),
10        MEDIANA("M"),
11        GRANDE("L"),
12        EXTRAGRANDE("XL");
13
14        private String talla;
15
16        private Tallas(String t) {
17            talla = t;
18        }
19
20        public String dameTalla () {
21            return talla;
22        }
23    }
24
25    public static void main(String[] args) {
26
27        Scanner entrada = new Scanner(System.in);
28        String tallaIn = " ";
29        boolean comp = false;
30
31        // Comprobar que la data ingresada sea correcta
32        while (comp == false) {
33            System.out.println("Ingrese su talla: Pequeña, Mediana, Grande, ExtraGrande");
34            tallaIn = entrada.nextLine().toUpperCase().trim();
35            for (Tallas iTallas: Tallas.values()) { // El método values() recorre todos
los valores de Enum
36                if (tallaIn.equalsIgnoreCase(iTallas.name())) { // El método name()
retorna el nombre literar de cada variable de Enum
37                    comp = true;
38                    break;
39                }
40            }
41            if (comp == false) {
42                System.out.println("Dato incorrecto, corrija nuevamente:");
43            }
44        }
45
46        Tallas tallaInEnum = Tallas.valueOf(tallaIn); // Se crea una variable del tipo
Tallas Enum que tiene por defecto el valor String introducido por el usuario
47
48        System.out.println("Talla: " + tallaInEnum);
49
50        System.out.println("Abreviatura: " + tallaInEnum.dameTalla());
51    }
52 }
53
```

\_8\_JefesTrabajadores.java

```
1 package EjerciciosP00;
2
3 interface Jefe {
4
5     public void setDesicion(String desicion);
6 }
7
8
9 interface Trabajadores {
10
11     double salarioBase = 1500;
12
13     public void setBono(double bono);
14 }
```

\_9\_UsoEmpleado.java

```
1 package EjerciciosP00;
2
3 import java.util.Arrays;
6
7 public class _9_UsoEmpleado {
8
9     public static void main(String[] args) {
10
11         JefaturaUso jefe1 = new JefaturaUso("Miguel Santos", 3500, 2010, 10, 8);
12         jefe1.setDesicion("Vender todas las acciones con urgencia.");
13         jefe1.setBono(3500);
14
15         JefaturaUso jefe2 = new JefaturaUso("Rossana", 4200, 2006, 7, 12);
16         jefe2.setBono(3000);
17
18         EmpleadoUso empleado1 = new EmpleadoUso("Abel", 2300, 2005, 11, 25);
19         empleado1.setBono(2100);
20
21         EmpleadoUso empleadosLista[] = new EmpleadoUso [7];
22
23         empleadosLista[0] = new EmpleadoUso("Miguel", 3500, 2010, 10, 8);
24         empleadosLista[1] = new EmpleadoUso("William", 2500, 2000, 5, 1);
25         empleadosLista[2] = new EmpleadoUso("Nayla", 2800, 1998, 3, 20);
26         empleadosLista[3] = new EmpleadoUso("Diego");
27         empleadosLista[4] = new EmpleadoUso("Ángela", "Ruiz");
28         empleadosLista[5] = jefe2;
29         empleadosLista[6] = new JefaturaUso("Andrea", 5500, 2002, 4, 20);
30
31         JefaturaUso jefe3 = (JefaturaUso) empleadosLista[6];
32         jefe3.setBono(55000);
33
34         Arrays.sort(empleadosLista);
35
36         System.out.println("El jefe de Mercadeo ha tomado la decisión de: " +
37             jefe1.getDesicionJefe());
38         System.out.println("El jefe : " + jefe1.getNombre() + " tiene un bono de: " +
39             jefe1.getBonoJefe() + "€");
40         System.out.println("El empleado : " + empleado1.getNombre() + " tiene un bono de: " +
41             + empleado1.getBonoEmpleado() + "€");
42
43         for (EmpleadoUso item: empleadosLista) {
44             System.out.println("Id empleado N°" + item.getId() + " - Nombre: " +
45                 item.getNombre() + ", Salario: " + item.getSalario() + "€, Fecha de contrato: " +
46                 item.getFechaContrato());
47         }
48     }
49 }
50
51 class EmpleadoUso implements Trabajadores, Comparable {
52
53     private String nombre;
54     private double sueldo;
55     private double sueldoTotal;
56     private String apellido;
57     private double bonoEmpleado;
58     private Date fechaContrato;
59     private int id = 0;
60     private static int idFijo = 1;
61
62     public EmpleadoUso(String nombre, double sueldo, int anio, int mes, int dia) {
63         this.nombre = nombre;
64         this.sueldo = sueldo;
```

\_9\_UsoEmpleado.java

```
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
```

\_10\_PruebaTemporizador.java

```
1 package EjerciciosP00;
2
3 import java.awt.Toolkit;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6 import java.util.Date;
7 import javax.swing.JOptionPane;
8 import javax.swing.Timer;
9
10 public class _10_PruebaTemporizador {
11
12     public static void main(String[] args) {
13
14         Accion accion = new Accion();
15
16         Timer temporizador = new Timer(3000, accion);
17
18         temporizador.start();
19
20         JOptionPane.showMessageDialog(null, "Para desactivar el sonido oprima el botón OK");
21
22         System.exit(0);
23     }
24 }
25
26 class Accion implements ActionListener {
27
28     @Override
29     public void actionPerformed(ActionEvent e) {
30
31         Date hora = new Date();
32
33         System.out.println("Te muestro la hora cada cierto tiempo: " + hora);
34
35         Toolkit.getDefaultToolkit().beep();
36     }
37 }
```

\_01\_CreandoYEscribiendoEnMarcos.java

```
1 package Graficos;
2
3 import java.awt.*;
4 import javax.swing.*;
5
6 public class _01_CreandoYEscribiendoEnMarcos {
7
8     public static void main(String[] args) {
9         Marco ventana = new Marco();
10        ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
11    }
12 }
13
14 class Marco extends JFrame {
15     public Marco() {
16         Dimension pantalla = Toolkit.getDefaultToolkit().getScreenSize();
17         int ancho = (int) pantalla.getWidth();
18         int alto = (int) pantalla.getHeight();
19         setBounds(ancho/3, alto/3, ancho/3, alto/3);
20         setTitle(" Mi Ventana Java");
21         ImageIcon icono = new ImageIcon("src/Graficos/images/icon.png");
22         setIconImage(icono.getImage());
23         Lamina laminaObj = new Lamina();
24         add(laminaObj);
25         setVisible(true);
26     }
27 }
28
29 class Lamina extends JPanel {
30     public void paintComponent(Graphics g) {
31         super.paintComponent(g);
32         g.setFont(new Font("Roboto", Font.BOLD, 20));
33         g.setColor(new Color(153, 51, 255));
34         g.drawString("Título del Contenido", 100, 100);
35     }
36 }
```

```
1 package Graficos;
2
3 import java.awt.Color;
4 import java.awt.Graphics;
5 import java.awt.Graphics2D;
6 import java.awt.geom.Ellipse2D;
7 import java.awt.geom.Rectangle2D;
8 import javax.swing.JFrame;
9 import javax.swing.JPanel;
10
11 public class _02_PruebaDibujo_TrabajandoConColores {
12
13     public static void main(String[] args) {
14         Marco2 ventana = new Marco2();
15         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
16     }
17 }
18
19 class Marco2 extends JFrame {
20     public Marco2 () {
21         setSize(840, 840);
22         setTitle(" Dibujos y Colores");
23         setLocationRelativeTo(null);
24         add(new Lamina2());
25         setVisible(true);
26     }
27 }
28
29 class Lamina2 extends JPanel {
30     public Lamina2() {
31         setBackground(new Color(179, 255, 255));
32     }
33     @Override
34     protected void paintComponent(Graphics g) {
35         super.paintComponent(g);
36         g.drawRect(100, 200, 150, 100);
37         g.drawString("1 - Rectángulo Vacío Graphics", 100, 180);
38         g.fillRect(100, 400, 150, 100);
39         g.drawString("2 - Rectángulo Relleno Graphics", 100, 380);
40         Graphics2D g2D = (Graphics2D) g;
41         Rectangle2D rectangulo = new Rectangle2D.Double(500, 200, 150, 100);
42         g2D.draw(rectangulo);
43         g2D.drawString("3 - Rectángulo Graphics2D", 500, 120);
44         Ellipse2D elipse = new Ellipse2D.Double();
45         elipse setFrame(rectangulo);
46         g2D.draw(elipse);
47         g2D.drawString("4 - Elipse Graphics2D", 500, 140);
48         g2D.drawLine(500, 200, 650, 300);
49         g2D.drawString("5 - Línea Oblícu Graphics2D", 500, 370);
50         double centroX = rectangulo.getCenterX();
51         double centroY = rectangulo.getCenterY();
52         double radio = 90;
53         Ellipse2D circulo = new Ellipse2D.Double();
54         circulo.setFrameFromCenter(centroX, centroY, centroX+radio, centroY+radio);
55         g2D.draw(circulo);
56         g2D.drawString("6 - Círculo Graphics2D", 500, 390);
57         g.drawString("7 - Rectángulo Relleno Graphics", 500, 430);
58         g.setColor(new Color(255, 153, 255));
59         g.fillRect(500, 480, 200, 200);
60         g.setColor(new Color(0, 0, 0));
61         g.drawString("8 - Círculo Relleno Graphics", 500, 460);
62         g.setColor(new Color(179, 255, 179));
```



\_02\_PruebaDibujo\_TrabajandoConColores.java

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

```
1 package Graficos;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import java.awt.Font;
6 import java.awt.GraphicsEnvironment;
7 import javax.swing.JFrame;
8 import javax.swing.JLabel;
9 import javax.swing.JOptionPane;
10 import javax.swing.JPanel;
11 import javax.swing.SwingConstants;
12
13 public class _03_FuentesTipo_TrabajandoConFuentes {
14
15     public static void main(String[] args) {
16
17         Marco3 ventana = new Marco3();
18         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
19     }
20 }
21
22 class Marco3 extends JFrame {
23     public Marco3() {
24         setSize(900, 300);
25         setTitle(" Trabajando con Fuentes");
26         setLocationRelativeTo(null);
27
28         setLayout(new BorderLayout());
29         add(new JLabel(" "), BorderLayout.NORTH);
30         add(new JLabel(" "), BorderLayout.SOUTH);
31         add(new JLabel(" "), BorderLayout.WEST);
32         add(new JLabel(" "), BorderLayout.EAST);
33
34         add(new Ventana3(), BorderLayout.CENTER);
35
36         setVisible(true);
37     }
38 }
39
40 class Ventana3 extends JPanel {
41
42     String fuenteIn;
43     boolean verificador = false;
44     JLabel encabezadoText, fuenteText;
45
46     public Ventana3() {
47
48         setLayout(new BorderLayout());
49
50         fuenteIn = JOptionPane.showInputDialog("Ingrese el nombre de la fuente a
consultar").toUpperCase();
51
52         GraphicsEnvironment e = GraphicsEnvironment.getLocalGraphicsEnvironment();
53         String Fuentes[] = e.getAvailableFontFamilyNames();
54
55         for(String fuente: Fuentes) {
56             if(fuenteIn.equalsIgnoreCase(fuente)) {
57                 verificador = true;
58                 break;
59             }
60         }
61
62     }
63 }
```

```
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 }
```

\_04\_PruebaImagenes.java

```
1 package Graficos;
2
3 import java.awt.Graphics;
4 import java.awt.Image;
5 import java.io.File;
6 import javax.imageio.ImageIO;
7 import javax.swing.ImageIcon;
8 import javax.swing.JFrame;
9 import javax.swing.JPanel;
10
11 public class _04_PruebaImagenes {
12
13     public static void main(String[] args) {
14
15         Marco4 ventana = new Marco4();
16         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
17     }
18 }
19
20 class Marco4 extends JFrame {
21
22     public Marco4() {
23         setSize(600, 500);
24         setTitle(" Prueba Imágenes");
25         setLocationRelativeTo(null);
26
27         ImageIcon icono = new ImageIcon("src/Graficos/images/musicIco.png");
28         setIconImage(icono.getImage());
29
30         add(new Ventana4());
31
32         setVisible(true);
33     }
34 }
35
36 class Ventana4 extends JPanel {
37
38     private Image radioImg;
39     private Image musicImg;
40     private int anchoImg;
41     private int altoImg;
42
43     protected void paintComponent(Graphics g) {
44
45         super.paintComponent(g);
46
47         // -----MUSIC IMG-----
48
49         File musicPath = new File("src/Graficos/images/music.png");
50
51         try {
52             musicImg = ImageIO.read(musicPath);
53         } catch (Exception e) {
54             System.out.println("No File!");
55         }
56
57         g.drawImage(musicImg, 0, 0, 100, 100, null);
58
59         for (int i=0; i<600; i++) {
60             for(int j=0; j<500; j++) {
61                 g.copyArea(0, 0, 100, 100, i*100, j*100);
62             }
63         }
64     }
65 }
```

\_04\_PruebaImagenes.java

```
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 }
```

```
1 package Graficos;
2
3 import java.awt.*;
4 import java.awt.event.*;
5 import javax.swing.*;
6
7 public class _05_PruebaEventos_PruebaAcciones {
8
9     public static void main(String[] args) {
10
11         Marco5 ventana = new Marco5();
12         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
13     }
14 }
15
16 class Marco5 extends JFrame {
17
18     public Marco5() {
19
20         setSize(600, 500);
21         setTitle(" Eventos y Acciones");
22         setLocationRelativeTo(null);
23
24         add(new Ventana5());
25
26         setVisible(true);
27     }
28 }
29
30 class Ventana5 extends JPanel {
31
32     public Ventana5() {
33
34         setLayout(new GridBagLayout());
35         GridBagConstraints gbc = new GridBagConstraints();
36         gbc.insets = new Insets(0, 10, 0, 10);
37
38         AccionColor accionLima = new AccionColor("Lima", new
39 ImageIcon("src/Graficos/images/lima.png"), new Color(204, 255, 153));
40         AccionColor accionOrange = new AccionColor("Orange", new
41 ImageIcon("src/Graficos/images/orange.png"), new Color(255, 204, 102));
42         AccionColor accionGrape = new AccionColor("Grape", new
43 ImageIcon("src/Graficos/images/grape.png"), new Color(204, 204, 255));
44
45         add(new JButton(accionLima), gbc);
46         add(new JButton(accionOrange), gbc);
47         add(new JButton(accionGrape), gbc);
48
49         KeyStroke tecladoLima = KeyStroke.getKeyStroke("ctrl L");
50         KeyStroke tecladoOrange = KeyStroke.getKeyStroke("ctrl O");
51         KeyStroke tecladoGrape = KeyStroke.getKeyStroke("ctrl G");
52
53         InputMap mapaEntrada = getInputMap(WHEN_IN_FOCUSED_WINDOW);
54
55         mapaEntrada.put(tecladoLima, "eventoLima");
56         mapaEntrada.put(tecladoOrange, "eventoOrange");
57         mapaEntrada.put(tecladoGrape, "eventoGrape");
58
59         ActionMap mapaAccion = getActionMap();
60
61         mapaAccion.put("eventoLima", accionLima);
62         mapaAccion.put("eventoOrange", accionOrange);
63     }
64 }
```

```
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 }
```

```
1 package Graficos;
2
3 import java.awt.Color;
4 import java.awt.GridBagConstraints;
5 import java.awt.GridBagLayout;
6 import java.awt.event.ActionEvent;
7 import javax.swing.AbstractAction;
8 import javax.swing.ActionMap;
9 import javax.swing.ImageIcon;
10 import javax.swing.InputMap;
11 import javax.swing.JButton;
12 import javax.swing.JFrame;
13 import javax.swing.JPanel;
14 import javax.swing.KeyStroke;
15
16 public class _05_PruebaEventos_PruebaAcciones_ByPipe {
17
18     public static void main(String[] args) {
19
20         MarcoAccionesPipe ventana = new MarcoAccionesPipe();
21         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
22     }
23 }
24
25 class MarcoAccionesPipe extends JFrame {
26
27     public MarcoAccionesPipe() {
28
29         setTitle(" Eventos y Acciones by Pipe");
30         setSize(600, 400);
31         setLocationRelativeTo(null);
32
33         add(new VentanaAccionesPipe());
34
35         setVisible(true);
36     }
37 }
38
39 class VentanaAccionesPipe extends JPanel {
40
41     public VentanaAccionesPipe() {
42
43         setLayout(new GridBagLayout());
44         GridBagConstraints gbc = new GridBagConstraints();
45         gbc.insets.set(0, 0, 50, 10);
46
47         Acciones accionLima = new Acciones("Lima", new
48 ImageIcon("src/Graficos/images/lima.png"), new Color(204, 255, 153));
49         Acciones accionOrange = new Acciones("Orange", new
50 ImageIcon("src/Graficos/images/orange.png"), new Color(255, 204, 102));
51         Acciones accionGrape = new Acciones("Grape", new
52 ImageIcon("src/Graficos/images/grape.png"), new Color(204, 204, 255));
53
54         add(new JButton(accionLima), gbc);
55         add(new JButton(accionOrange), gbc);
56         add(new JButton(accionGrape), gbc);
57
58         KeyStroke tecladoLima = KeyStroke.getKeyStroke("ctrl L");
59         KeyStroke tecladoOrange = KeyStroke.getKeyStroke("ctrl O");
60         KeyStroke tecladoGrape = KeyStroke.getKeyStroke("ctrl G");
61
62         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, "Ctrl+" + 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 }
```

\_06\_EventosFocoVentanaTeclado.java

```
1 package Graficos;
2
3 import java.awt.event.KeyEvent;
4 import java.awt.event.KeyListener;
5 import java.awt.event.WindowAdapter;
6 import java.awt.event.WindowEvent;
7 import javax.swing.JFrame;
8
9 public class _06_EventosFocoVentanaTeclado {
10
11     public static void main(String[] args) {
12
13         Marco6 ventana1 = new Marco6();
14         Marco6New ventana2 = new Marco6New();
15     }
16 }
17
18 class Marco6 extends JFrame {
19
20     private String nombreVentana;
21
22     public Marco6() {
23
24         setBounds(200, 200, 500, 300);
25         setTitle(" Ventana 1ª");
26
27         nombreVentana = this.getTitle();
28
29         addWindowFocusListener(new EventosVentana(nombreVentana));
30         addWindowListener(new EventosVentana(nombreVentana));
31         addWindowStateListener(new EventosVentana(nombreVentana));
32
33         addKeyListener(new EventosTeclado());
34
35         setVisible(true);
36         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
37     }
38 }
39
40 class Marco6New extends JFrame {
41
42     private String nombreVentana;
43
44     public Marco6New() {
45
46         setBounds(900, 200, 500, 300);
47         setTitle(" Ventana 2ª");
48
49         nombreVentana = this.getTitle();
50
51         addWindowFocusListener(new EventosVentana(nombreVentana));
52         addWindowListener(new EventosVentana(nombreVentana));
53         addWindowStateListener(new EventosVentana(nombreVentana));
54
55         addKeyListener(new EventosTeclado());
56
57         setVisible(true);
58     }
59 }
60
61 class EventosVentana extends WindowAdapter {
62
```

```
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) {
```

\_06\_EventosFocoVentanaTeclado.java

```
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 }
```

\_07\_EventosRaton.java

```
1 package Graficos;
2
3 import java.awt.event.MouseAdapter;
4 import java.awt.event.MouseEvent;
5 import java.awt.event.MouseWheelEvent;
6 import javax.swing.JFrame;
7
8 public class _07_EventosRaton extends JFrame {
9
10     public static void main(String[] args) {
11
12         JFrame ventana = new JFrame();
13
14         ventana.setTitle(" Eventos del Ratón");
15         ventana.setSize(500, 400);
16         ventana.setLocationRelativeTo(null);
17
18         ventana.addMouseListener(new AccionesRaton());
19         ventana.addMouseMotionListener(new AccionesRaton());
20         ventana.addMouseWheelListener(new AccionesRaton());
21
22         ventana.setVisible(true);
23         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
24     }
25 }
26
27 class AccionesRaton extends MouseAdapter {
28
29     @Override
30     public void mouseClicked(MouseEvent e) {
31         int boton = e.getButton();
32         String click = "";
33         if (boton == 1) {
34             click = "Click Izquierdo";
35         } else if (boton == 2) {
36             click = "Rueda del Ratón";
37         } else if (boton == 3) {
38             click = "Click Derecho";
39         }
40         int x = e.getX();
41         int y = e.getY();
42         System.out.println("Se ha hecho Click con: " + click + " en la coordenada: X=" + x +
43 " Y=" + y);
44         super.mouseClicked(e);
45     }
46
47     @Override
48     public void mouseDragged(MouseEvent e) {
49         System.out.println("Se ha Arrastrado el ratón");
50         super.mouseDragged(e);
51     }
52
53     @Override
54     public void mouseEntered(MouseEvent e) {
55         System.out.println("El ratón ha Entrado en la ventana");
56         super.mouseEntered(e);
57     }
58
59     @Override
60     public void mouseExited(MouseEvent e) {
61         System.out.println("El ratón ha Salido de la ventana");
62         super.mouseExited(e);
63     }
64 }
```

\_07\_EventosRaton.java

```
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 }
```

\_08\_FocoEvento.java

```
1 package Graficos;
2 import java.awt.Color;
3 import java.awt.GridBagLayout;
4 import java.awt.GridLayout;
5 import java.awt.event.FocusEvent;
6 import java.awt.event.FocusListener;
7 import javax.swing.JFrame;
8 import javax.swing.JLabel;
9 import javax.swing.JPanel;
10 import javax.swing.JTextField;
11
12 public class _08_FocoEvento {
13
14     public static JLabel aviso;
15
16     public static void main(String[] args) {
17
18         Color fondo = new Color(230, 204, 255);
19         JTextField email;
20
21         JFrame ventana = new JFrame();
22         ventana.setTitle(" Foco Evento");
23         ventana.setSize(600, 400);
24         ventana.setLocationRelativeTo(null);
25         ventana.setLayout(new GridBagLayout());
26         ventana.getContentPane().setBackground(fondo);
27
28         JPanel lamina = new JPanel();
29         lamina.setBackground(fondo);
30
31         CampoFoco oyenteFoco = new CampoFoco();
32
33         lamina.setLayout(new GridLayout(10,1));
34
35         lamina.add(new JLabel("eMail: ")).setForeground(Color.GRAY.darker());
36         lamina.add(email = new JTextField(30));
37         email.addFocusListener(oyenteFoco);
38         lamina.add(new JLabel("")).setForeground(Color.GRAY.darker());
39         lamina.add(avisos = new JLabel("
40         lamina.add(new JLabel("")).setForeground(Color.GRAY.darker());
41         lamina.add(new JLabel("Contraseña: ")).setForeground(Color.GRAY.darker());
42         lamina.add(new JTextField(30));
43         lamina.add(new JLabel("")).setForeground(Color.GRAY.darker()); //
44         Estos labels son sólo para aumentar espacio
45         lamina.add(new JLabel("")).setForeground(Color.GRAY.darker());
46         lamina.add(new JLabel("")).setForeground(Color.GRAY.darker());
47
48         ventana.add(lamina);
49
50         ventana.setVisible(true);
51         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
52     }
53
54     static class CampoFoco implements FocusListener {
55
56         @Override
57         public void focusGained(FocusEvent e) {
58             // TODO Auto-generated method stub
59         }
60
61         @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 }
```



\_09\_VariosOyentes.java

```
1 package Graficos;
2
3 import java.awt.Color;
4 import java.awt.Font;
5 import java.awt.GridBagLayout;
6 import java.awt.event.ActionEvent;
7 import java.awt.event.ActionListener;
8 import java.util.Random;
9 import javax.swing.JButton;
10 import javax.swing.JFrame;
11 import javax.swing.JLabel;
12 import javax.swing.JPanel;
13
14 public class _09_VariosOyentes {
15
16     public static void main(String[] args) {
17         Ventana9 ventana = new Ventana9();
18     }
19 }
20
21 class Ventana9 extends JFrame {
22     public Ventana9() {
23         setSize(400, 300);
24         setTitle(" Varios Oyentes");
25         setBounds(1200, 200, 400, 300);
26         add(new Lamina9());
27         setVisible(true);
28         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
29     }
30 }
31
32 class Lamina9 extends JPanel {
33     JButton btn1, btn2;
34     public Lamina9() {
35         setBackground(new Color(255, 255, 153));
36         setLayout(new GridBagLayout());
37         add(btn1 = new JButton("Nueva Ventana"));
38         add(btn2 = new JButton("Cerrar Ventanas"));
39         btn1.addActionListener(new OyenteListener());
40     }
41
42     public class OyenteListener implements ActionListener {
43         @Override
44         public void actionPerformed(ActionEvent e) {
45             VentanaEmergente marco = new VentanaEmergente(btn2);
46         }
47     }
48
49     int numero = 1;
50
51     class VentanaEmergente extends JFrame {
52
53         public VentanaEmergente(JButton btnCerrar) {
54             setTitle("Ventana " + numero);
55             setBounds(40*numero, 40*numero, 250, 200);
56
57             add(new NombreAleatorio());
58
59             setVisible(true);
60             numero++;
61             btnCerrar.addActionListener(new OyenteCerrar());
62         }
63     }
64 }
```

```
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[] =
98     {"B","C","D","F","G","H","J","K","L","M","N","Ñ","P","Q","R","S","T","V","W","X","Y","Z","L",
99     "L","BB","RR","MM","PP","SS",};
100
101     String vocalRandom1;
102     String vocalRandom2;
103     String vocalRandom3;
104     String vocalRandom4;
105     String vocalRandom5;
106     String vocalRandom6;
107     String consonanteRandom1;
108     String consonanteRandom2;
109     String consonanteRandom3;
110     String consonanteRandom4;
111     String consonanteRandom5;
112     String consonanteRandom6;
113
114     public NombreAleatorio() {
115
116         // -----VOCALES-----
117         for(int i=0; i<vocales.length; i++) {
118             if(i == vocalNro1) {
119                 vocalRandom1 = vocales[i];
120             }
121         }
122         for(int i=0; i<vocales.length; i++) {
123             if(i == vocalNro2) {
124                 vocalRandom2 = vocales[i];
125             }
126         }
127         for(int i=0; i<vocales.length; i++) {
128             if(i == vocalNro3) {
129                 vocalRandom3 = vocales[i];
130             }
131         }
132         for(int i=0; i<vocales.length; i++) {
133             if(i == vocalNro4) {
134                 vocalRandom4 = vocales[i];
135             }
136         }
137         for(int i=0; i<vocales.length; i++) {
138             if(i == vocalNro5) {
139                 vocalRandom5 = vocales[i];
140             }
141         }
142         for(int i=0; i<vocales.length; i++) {
143             if(i == vocalNro6) {
144                 vocalRandom6 = vocales[i];
145             }
146         }
147
148         for(int i=0; i<consonantes.length; i++) {
149             if(i == consonanteNro1) {
150                 consonanteRandom1 = consonantes[i];
151             }
152         }
153         for(int i=0; i<consonantes.length; i++) {
154             if(i == consonanteNro2) {
155                 consonanteRandom2 = consonantes[i];
156             }
157         }
158         for(int i=0; i<consonantes.length; i++) {
159             if(i == consonanteNro3) {
160                 consonanteRandom3 = consonantes[i];
161             }
162         }
163         for(int i=0; i<consonantes.length; i++) {
164             if(i == consonanteNro4) {
165                 consonanteRandom4 = consonantes[i];
166             }
167         }
168         for(int i=0; i<consonantes.length; i++) {
169             if(i == consonanteNro5) {
170                 consonanteRandom5 = consonantes[i];
171             }
172         }
173         for(int i=0; i<consonantes.length; i++) {
174             if(i == consonanteNro6) {
175                 consonanteRandom6 = consonantes[i];
176             }
177         }
178
179         nombre.setText(vocalRandom1 + consonanteRandom1 + vocalRandom2 + consonanteRandom2 +
180             vocalRandom3 + consonanteRandom3 + vocalRandom4 + consonanteRandom4 +
181             vocalRandom5 + consonanteRandom5 + vocalRandom6 + consonanteRandom6);
182     }
183 }
```

```
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
```

\_09\_VariosOyentes.java

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

\_10\_Layouts.java

```
1 package Graficos;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import java.awt.FlowLayout;
6 import javax.swing.JButton;
7 import javax.swing.JFrame;
8 import javax.swing.JPanel;
9
10 public class _10_Layouts {
11
12     public static void main(String[] args) {
13
14         Ventana10 ventana = new Ventana10();
15     }
16 }
17
18 class Ventana10 extends JFrame {
19
20     public Ventana10() {
21
22         setSize(600, 500);
23         setTitle(" Layouts");
24         setLocationRelativeTo(null);
25
26         add(new Lamina10());
27
28         setVisible(true);
29         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
30     }
31 }
32
33 class Lamina10 extends JPanel {
34
35     public Lamina10() {
36         setLayout(new BorderLayout());
37         setBackground(new Color(204, 255, 102));
38
39         JPanel lamina1 = new JPanel();
40         add(lamina1, BorderLayout.NORTH);
41         lamina1.setLayout(new BorderLayout());
42
43         lamina1.add(new JButton("Botón 1"), BorderLayout.NORTH);
44         lamina1.add(new JButton("Botón 2"), BorderLayout.SOUTH);
45         lamina1.add(new JButton("Botón 3"), BorderLayout.EAST);
46         lamina1.add(new JButton("Botón 4"), BorderLayout.WEST);
47         lamina1.add(new JButton("Botón 5"), BorderLayout.CENTER);
48
49         JPanel lamina2 = new JPanel();
50         add(lamina2, BorderLayout.SOUTH);
51         lamina2.setLayout(new FlowLayout(FlowLayout.RIGHT));
52
53         lamina2.add(new JButton("Botón 1"));
54         lamina2.add(new JButton("Botón 2"));
55         lamina2.add(new JButton("Botón 3"));
56     }
57 }
```

\_11\_Calculadora.java

```
1 package Graficos;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import java.awt.Dimension;
6 import java.awt.Font;
7 import java.awt.GridLayout;
8 import java.awt.event.ActionEvent;
9 import java.awt.event.ActionListener;
10 import javax.swing.JButton;
11 import javax.swing.JFrame;
12 import javax.swing.JPanel;
13 import javax.swing.SwingConstants;
14
15 public class _11_Calculadora {
16
17     public static void main(String[] args) {
18
19         Ventana11 ventana = new Ventana11();
20     }
21 }
22
23 class Ventana11 extends JFrame {
24
25     public Ventana11() {
26         setTitle(" Calculadora");
27         setSize(300, 350);
28         setLocationRelativeTo(null);
29
30         add(new Lamina11());
31
32         setVisible(true);
33         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
34     }
35 }
36
37 class Lamina11 extends JPanel {
38
39     JPanel lamina1, lamina2;
40     JButton display, btn1, btn2, btn3, btn4, btn5, btn6, btn7, btn8, btn9, btn0, btnSum,
41         btnRes, btnMul, btnDiv, btnClr, btnEqu;
42
43     public Lamina11() {
44         setLayout(new BorderLayout());
45
46         add(lamina1 = new JPanel(), BorderLayout.NORTH);
47         lamina1.setLayout(new BorderLayout());
48         lamina1.add(display = new JButton("0"));
49         display.setHorizontalAlignment(SwingConstants.RIGHT);
50         display.setFont(new Font("Arial", Font.PLAIN, 24));
51         display.setPreferredSize(new Dimension(300, 80));
52         display.setEnabled(false);
53
54         add(lamina2 = new JPanel(), BorderLayout.CENTER);
55         lamina2.setLayout(new GridLayout(4, 4));
56         lamina2.add(btn1 = new JButton("1"));
57         lamina2.add(btn2 = new JButton("2"));
58         lamina2.add(btn3 = new JButton("3"));
59         lamina2.add(btnSum = new JButton("+"));
60         lamina2.add(btn4 = new JButton("4"));
61         lamina2.add(btn5 = new JButton("5"));
62         lamina2.add(btn6 = new JButton("6"));
```

\_11\_Calculadora.java

```
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("-")) {
```



\_11\_Calculadora.java

```
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 }
```

\_12\_PruebaTexto.java

```
1 package Graficos;
2
3 import java.awt.Color;
4 import java.awt.GridBagConstraints;
5 import java.awt.GridBagLayout;
6 import java.awt.GridLayout;
7 import java.awt.event.ActionEvent;
8 import java.awt.event.ActionListener;
9 import javax.swing.JButton;
10 import javax.swing.JFrame;
11 import javax.swing.JLabel;
12 import javax.swing.JPanel;
13 import javax.swing.JTextField;
14
15 public class _12_PruebaTexto {
16
17     public static void main(String[] args) {
18
19         Marco12 ventana = new Marco12();
20         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21     }
22 }
23
24 class Marco12 extends JFrame {
25
26     Color fondo = new Color(51, 204, 204);
27
28     public Marco12() {
29
30         setSize(600, 500);
31         setTitle(" Prueba Texto");
32         setLocationRelativeTo(null);
33         getContentPane().setBackground(fondo);
34
35         setLayout(new GridBagLayout());
36         GridBagConstraints gbc = new GridBagConstraints();
37         gbc.insets.set(0, 0, 60, 0);
38
39         add(new Ventana12(), gbc);
40
41         setVisible(true);
42     }
43
44     class Ventana12 extends JPanel {
45
46         JTextField campoTexto;
47         JLabel aviso;
48         JButton btn;
49
50         public Ventana12() {
51
52             setBackground(fondo);
53
54             setLayout(new GridLayout(7,1));
55
56             add(new JLabel("Ingrese su eMail:"));
57             add(new JLabel(""));
58             add(campoTexto = new JTextField(30));
59             add(new JLabel(""));
60             add(aviso = new JLabel(""));
61             add(new JLabel(""));
62             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 }
```

\_13\_CampoPassword.java

```
1 package Graficos;
2
3 import java.awt.Color;
4 import java.awt.GridBagLayout;
5 import java.awt.GridLayout;
6 import javax.swing.JButton;
7 import javax.swing.JFrame;
8 import javax.swing.JLabel;
9 import javax.swing.JPanel;
10 import javax.swing.JPasswordField;
11 import javax.swing.JTextField;
12 import javax.swing.event.DocumentEvent;
13 import javax.swing.event.DocumentListener;
14
15 public class _13_CampoPassword {
16
17     public static void main(String[] args) {
18
19         Marco13 ventana = new Marco13();
20         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21     }
22 }
23
24 class Marco13 extends JFrame {
25
26     public Marco13() {
27
28         setSize(550, 600);
29         setTitle(" Campo Password");
30         setLocationRelativeTo(null);
31
32         add(new Ventana13());
33
34         setVisible(true);
35     }
36 }
37
38 class Ventana13 extends JPanel {
39
40     JLabel emailTxt, passTxt, avisoMail, avisoPass;
41     JTextField emailField;
42     JPasswordField passField;
43     JButton enviarBtn;
44     JPanel lamina;
45     Color fondo = new Color(204, 255, 153);
46
47     public Ventana13() {
48
49         setBackground(fondo);
50
51         setLayout(new GridBagLayout());
52
53         add(lamina = new JPanel());
54         lamina.setBackground(fondo);
55         lamina.setLayout(new GridLayout(10,1));
56
57         lamina.add(emailTxt = new JLabel("Email:"));
58         lamina.add(emailField = new JTextField(40));
59         lamina.add(avisoMail = new JLabel(" "));
60         lamina.add(new JLabel(" "));
61         lamina.add(passTxt = new JLabel("Contraseña:"));
62         lamina.add(passField = new JPasswordField(40));
```

\_13\_CampoPassword.java

```
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    }
```

\_13\_CampoPassword.java

```
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
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
163 letras!");
164             } else {
165                 passField.setBackground(Color.WHITE);
166                 avisoPass.setText(" ");
167             }
168         }
169     }
170     @Override
171     public void removeUpdate(DocumentEvent e) {
172         char password[] = passField.getPassword();
173         for (int i = 0; i < password.length; i++) {
174             if (password.length < 6 || password.length > 12) {
175                 passField.setBackground(new Color(255, 102, 153));
176                 avisoPass.setForeground(new Color(255, 102, 153));
177                 avisoPass.setText("La contraseña debe tener mínimo 6 y máximo 12
178 letras!");
179             } else {
180                 passField.setBackground(Color.WHITE);
181                 avisoPass.setText(" ");
182             }
183         }
184     }
185 }
```

\_13\_CampoPassword.java

```
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 }
```

\_14\_EjemploArea.java

```
1 package Graficos;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import java.awt.FlowLayout;
6 import java.awt.event.ActionEvent;
7 import java.awt.event.ActionListener;
8 import javax.swing.JButton;
9 import javax.swing.JFrame;
10 import javax.swing.JLabel;
11 import javax.swing.JPanel;
12 import javax.swing.JScrollPane;
13 import javax.swing.JTextArea;
14
15 public class _14_EjemploArea {
16
17     public static void main(String[] args) {
18
19         Marco14 ventana = new Marco14();
20         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21     }
22 }
23
24 class Marco14 extends JFrame {
25
26     public Marco14() {
27
28         setSize(350, 500);
29         setTitle(" Ejemplo Área");
30         setLocationRelativeTo(null);
31         getContentPane().setBackground(new Color(153, 204, 255));
32
33         setLayout(new BorderLayout());
34
35         // -----Espacios en los Bordes-----
36         add(new JLabel(" "), BorderLayout.NORTH);
37         add(new JLabel(" "), BorderLayout.SOUTH);
38         add(new JLabel(" "), BorderLayout.EAST);
39         add(new JLabel(" "), BorderLayout.WEST);
40         // -----Espacios en los Bordes-----
41
42         add(new Ventana14(), BorderLayout.CENTER);
43
44         setVisible(true);
45     }
46 }
47
48 class Ventana14 extends JPanel {
49
50     JPanel lamina;
51     JTextArea areaIn, areaOut;
52     JLabel textoOut;
53     JButton btnEnviar;
54     JScrollPane vistaScroll;
55
56     public Ventana14() {
57
58         setBackground(new Color(153, 204, 255));
59
60         setLayout(new FlowLayout(FlowLayout.LEFT, 20,20));
61
62         add(new JLabel(" "));
```



\_14\_EjemploArea.java

```
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 }
```

\_15\_PruebaArea.java

```
1 package Graficos;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import java.awt.event.ActionEvent;
6 import java.awt.event.ActionListener;
7 import javax.swing.JButton;
8 import javax.swing.JFrame;
9 import javax.swing.JPanel;
10 import javax.swing.JScrollPane;
11 import javax.swing.JTextArea;
12
13 public class _15_PruebaArea {
14
15     public static void main(String[] args) {
16
17         Marco15 ventana = new Marco15();
18         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
19     }
20 }
21
22 class Marco15 extends JFrame {
23
24     public Marco15() {
25
26         setSize(400, 300);
27         setTitle(" Prueba Área");
28         setLocationRelativeTo(null);
29
30         add(new Ventana15());
31
32         setVisible(true);
33     }
34 }
35
36 class Ventana15 extends JPanel {
37
38     JTextArea areaTxt;
39     JScrollPane vistaScroll;
40     JPanel lamina;
41     JButton btn1, btn2;
42     String lorem = "Al contrario del pensamiento popular, el texto de Lorem Ipsum no es
43 simplemente texto aleatorio. | Al contrario del pensamiento popular, el texto de Lorem Ipsum
44 no es simplemente texto aleatorio. | Al contrario del pensamiento popular, el texto de Lorem
45 Ipsum no es simplemente texto aleatorio. | Al contrario del pensamiento popular, el texto de
46 Lorem Ipsum no es simplemente texto aleatorio. | ";
47
48     public Ventana15() {
49
50         setLayout(new BorderLayout());
51
52         add(areaTxt = new JTextArea(), BorderLayout.CENTER);
53         areaTxt.setBackground(new Color(204, 255, 255));
54         areaTxt.setForeground(new Color(153, 0, 115));
55         add(vistaScroll = new JScrollPane(areaTxt));
56
57         add(lamina = new JPanel(), BorderLayout.SOUTH);
58         lamina.setBackground(new Color(204, 204, 255));
59
60         lamina.add(btn1 = new JButton("Agregar texto"));
61         lamina.add(btn2 = new JButton("Insertar salto de línea"));
```

\_15\_PruebaArea.java

```
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 }
```

\_16\_PruebaChecks.java

```
1 package Graficos;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import java.awt.Font;
6 import java.awt.GridLayout;
7 import java.awt.event.ActionEvent;
8 import java.awt.event.ActionListener;
9 import javax.swing.JCheckBox;
10 import javax.swing.JFrame;
11 import javax.swing.JLabel;
12 import javax.swing.JPanel;
13 import javax.swing.SwingConstants;
14
15 public class _16_PruebaChecks {
16
17     public static void main(String[] args) {
18
19         Marco16 ventana = new Marco16();
20         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21     }
22 }
23
24 class Marco16 extends JFrame {
25
26     JLabel east;
27     Color fondo = new Color(179, 255, 179);
28
29     public Marco16() {
30
31         setSize(900, 400);
32         setTitle(" Prueba Checks");
33         setLocationRelativeTo(null);
34
35         setLayout(new BorderLayout());
36
37         // -----Espacios en los Bordes-----
38         add(new JLabel(" "), BorderLayout.NORTH);
39         add(new JLabel(" "), BorderLayout.SOUTH);
40         add(east = new JLabel(" ..... "), BorderLayout.EAST);
41         east.setForeground(new Color(0, 0, 0, 0));
42         add(new JLabel(" "), BorderLayout.WEST);
43         // -----Espacios en los Bordes-----
44
45         add(new Ventana16(fondo), BorderLayout.CENTER);
46
47         getContentPane().setBackground(fondo);
48
49         setVisible(true);
50     }
51 }
52
53 class Ventana16 extends JPanel {
54
55     JLabel texto1, texto2;
56     JCheckBox check1, check2;
57     JPanel lamina1, lamina2;
58
59     public Ventana16(Color fondo) {
60
61         setLayout(new BorderLayout());
62
63     }
```

\_16\_PruebaChecks.java

```
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("\nEs capaz el que piensa que es capaz.\n"));
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

```
1 package Graficos;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import java.awt.Dimension;
6 import java.awt.Font;
7 import java.awt.GraphicsEnvironment;
8 import java.awt.GridLayout;
9 import java.awt.event.ActionEvent;
10 import java.awt.event.ActionListener;
11 import javax.swing.ButtonGroup;
12 import javax.swing.JComboBox;
13 import javax.swing.JFrame;
14 import javax.swing.JLabel;
15 import javax.swing.JPanel;
16 import javax.swing.JRadioButton;
17 import javax.swing.JSlider;
18 import javax.swing.JSpinner;
19 import javax.swing.SpinnerListModel;
20 import javax.swing.SpinnerNumberModel;
21 import javax.swing.event.ChangeEvent;
22 import javax.swing.event.ChangeListener;
23
24 public class _17_SintaxisRadio_EjemploRadio_PruebaCombo_MarcoSlider_MarcoSpinner {
25
26     public static void main(String[] args) {
27
28         Marco17 ventana = new Marco17();
29         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
30     }
31 }
32
33 class Marco17 extends JFrame {
34
35     Color fondo1 = new Color(179, 255, 224);
36
37     public Marco17() {
38
39         setSize(1000, 900);
40         setTitle(" Ejercicio 3.17");
41         setLocationRelativeTo(null);
42         getContentPane().setBackground(fondo1);
43
44         setLayout(new BorderLayout());
45
46         // -----Espacios en los Bordes-----
47         add(new JLabel(" "), BorderLayout.NORTH);
48         add(new JLabel(" "), BorderLayout.SOUTH);
49         add(new JLabel(" "), BorderLayout.EAST);
50         add(new JLabel(" "), BorderLayout.WEST);
51         // -----Espacios en los Bordes-----
52
53         add(new Ventana17(fondo1), BorderLayout.CENTER);
54
55         setVisible(true);
56     }
57 }
58
59 class Ventana17 extends JPanel {
60
61     JPanel lamina1, lamina2, laminaGrid, laminaGroup1, laminaGroup2, laminaSeparator,
        laminaSpinners;
```

\_17\_SintaxisRadio\_EjemploRadio\_PruebaCombo\_MarcoSlider\_MarcoSpinner.java

```
62     JLabel texto1, texto2;
63     JRadioButton radioPequeño, radioMediano, radioGrande, radioMuyGrande, radioNegrita,
        radioCursiva, radioNormal;
64     JComboBox comboFuentes;
65     JSlider slider;
66     JSpinner spinnerFuente, spinnerTamano;
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("\nTodo es un pensamiento.\n"));
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 }
```

\_18\_MarcoMenu.java

```
1 package Graficos;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import javax.swing.JFrame;
6 import javax.swing.JLabel;
7 import javax.swing.JMenu;
8 import javax.swing.JMenuBar;
9 import javax.swing.JMenuItem;
10 import javax.swing.JPanel;
11 import javax.swing.JSeparator;
12
13 public class _18_MarcoMenu {
14
15     public static void main(String[] args) {
16
17         Marco18 ventana = new Marco18();
18         ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
19     }
20 }
21
22 class Marco18 extends JFrame {
23
24     Color fondo = new Color(230, 204, 255);
25
26     public Marco18() {
27
28         setSize(600, 500);
29         setTitle(" Marco Menú");
30         setLocationRelativeTo(null);
31         getContentPane().setBackground(fondo);
32
33         setLayout(new BorderLayout());
34
35         // -----Espacios en los Bordes-----
36         add(new JLabel(" "), BorderLayout.NORTH);
37         add(new JLabel(" "), BorderLayout.SOUTH);
38         add(new JLabel(" "), BorderLayout.EAST);
39         add(new JLabel(" "), BorderLayout.WEST);
40         // -----Espacios en los Bordes-----
41
42         add(new Ventana18(fondo), BorderLayout.CENTER);
43
44         setVisible(true);
45     }
46 }
47
48 class Ventana18 extends JPanel {
49
50     JMenuBar barra;
51     JMenu menu1, menu2, menu3, menu4, menu5;
52     JMenuItem item1, item2, item3, item4, item5, item6, item7, item8, item9, item10;
53     JSeparator separador;
54
55     public Ventana18(Color fondo) {
56
57         setBackground(fondo);
58
59         add(barra = new JMenuBar());
60
61         barra.add(menu1 = new JMenu("Inicio"));
62         barra.add(menu2 = new JMenu("Servicios"));
```

\_18\_MarcoMenu.java

```
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 }
```

\_01\_ProcesadorDeTextosFull.java

```
1 package EjerciciosBloque3;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import java.awt.Font;
6 import java.awt.event.InputEvent;
7 import java.awt.event.KeyEvent;
8 import javax.swing.ButtonGroup;
9 import javax.swing.ImageIcon;
10 import javax.swing.JButton;
11 import javax.swing.JFrame;
12 import javax.swing.JMenu;
13 import javax.swing.JMenuBar;
14 import javax.swing.JMenuItem;
15 import javax.swing.JPanel;
16 import javax.swing.JPopupMenu;
17 import javax.swing.JRadioButtonMenuItem;
18 import javax.swing.JScrollPane;
19 import javax.swing.JTextPane;
20 import javax.swing.JToolBar;
21 import javax.swing.KeyStroke;
22 import javax.swing.text.StyledEditorKit;
23
24 public class _01_ProcesadorDeTextosFull {
25
26     public static void main(String[] args) {
27
28         MarcoProcesadorTexto miMarco = new MarcoProcesadorTexto();
29     }
30 }
31
32 class MarcoProcesadorTexto extends JFrame {
33
34     public MarcoProcesadorTexto() {
35
36         setSize(700, 500);
37         setTitle("Procesador de Textos");
38         setLocationRelativeTo(null);
39
40         add(new LaminaProcesadorTexto());
41
42         setVisible(true);
43         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
44     }
45 }
46
47 class LaminaProcesadorTexto extends JPanel {
48
49     JPanel laminaSup;
50     private JMenuBar barraMenu;
51     private JMenu menuFuente, menuEstilo, menuTamano;
52     private JMenuItem fuenteArial, fuenteCourier, fuenteVerdana, estiloNegrita,
        estiloCursiva, estiloSubrayado;
53     private JRadioButtonMenuItem radio12, radio16, radio20, radio24;
54     private ButtonGroup radioGrupo = new ButtonGroup();
55     private JTextPane textPane;
56     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
```

\_01\_ProcesadorDeTextosFull.java

```
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,
138 "Arial"));
139         fuenteCourier.addActionListener(new StyledEditorKit.FontFamilyAction(null,
140 "Courier"));
141         fuenteVerdana.addActionListener(new StyledEditorKit.FontFamilyAction(null,
142 "Verdana"));
143
144         estiloNegrita.addActionListener(new StyledEditorKit.BoldAction());
145         estiloCursiva.addActionListener(new StyledEditorKit.ItalicAction());
146         estiloSubrayado.addActionListener(new StyledEditorKit.UnderlineAction());
147
148         estiloNegrita.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_N,
149 InputEvent.CTRL_DOWN_MASK));
150         estiloCursiva.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_K,
151 InputEvent.CTRL_DOWN_MASK));
152         estiloSubrayado.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_S,
153 InputEvent.CTRL_DOWN_MASK));
154
155         radio12.addActionListener(new StyledEditorKit.FontSizeAction(null, 12));
156         radio16.addActionListener(new StyledEditorKit.FontSizeAction(null, 16));
157         radio20.addActionListener(new StyledEditorKit.FontSizeAction(null, 20));
158         radio24.addActionListener(new StyledEditorKit.FontSizeAction(null, 24));
159     }
160 }
161
162 // -----CLASE
163 ITEMS-----
164 private class MenuItems {
165
```

```
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 }
```

\_02\_DisposicionesBox.java

```
1 package EjerciciosBloque3;
2
3 import java.awt.GridBagLayout;
4
5 import javax.swing.Box;
6 import javax.swing.JButton;
7 import javax.swing.JFrame;
8 import javax.swing.JLabel;
9 import javax.swing.JPanel;
10 import javax.swing.JTextField;
11
12 public class _02_DisposicionesBox {
13
14     public static void main(String[] args) {
15
16         MarcoDisposicionesBox marco = new MarcoDisposicionesBox();
17     }
18 }
19
20 class MarcoDisposicionesBox extends JFrame {
21
22     public MarcoDisposicionesBox() {
23
24         setSize(400, 600);
25         setTitle(" Disposiciones Box");
26         setLocationRelativeTo(null);
27
28         add(new LaminaDisposicionesBox());
29
30         pack();
31         setVisible(true);
32         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
33     }
34 }
35
36 class LaminaDisposicionesBox extends JPanel {
37
38     public LaminaDisposicionesBox() {
39
40         Box boxH1 = Box.createHorizontalBox();
41         Box boxH2 = Box.createHorizontalBox();
42         Box boxH3 = Box.createHorizontalBox();
43
44         boxH1.add(Box.createHorizontalStrut(50));
45         boxH1.add(new JLabel("Nombre"));
46         boxH1.add(Box.createHorizontalStrut(20));
47         boxH1.add(new JTextField(20));
48         boxH1.add(Box.createHorizontalStrut(50));
49
50         boxH2.add(Box.createHorizontalStrut(50));
51         boxH2.add(new JLabel("Email"));
52         boxH2.add(Box.createHorizontalStrut(20));
53         boxH2.add(new JTextField(20));
54         boxH2.add(Box.createHorizontalStrut(50));
55
56         boxH3.add(Box.createHorizontalStrut(50));
57         boxH3.add(new JButton("Ok"));
58         boxH3.add(Box.createGlue());
59         boxH3.add(new JButton("Cancelar"));
60         boxH3.add(Box.createHorizontalStrut(50));
61
62         Box boxV = Box.createVerticalBox();
```



\_02\_DisposicionesBox.java

```
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 }
```

\_03\_DisposicionMuelle.java

```
1 package EjerciciosBloque3;
2
3 import javax.swing.JButton;
4 import javax.swing.JFrame;
5 import javax.swing.JPanel;
6 import javax.swing.Spring;
7 import javax.swing.SpringLayout;
8
9 public class _03_DisposicionMuelle {
10
11     public static void main(String[] args) {
12
13         MarcoDisposicionMuelle marco = new MarcoDisposicionMuelle();
14     }
15 }
16
17 class MarcoDisposicionMuelle extends JFrame {
18
19     public MarcoDisposicionMuelle() {
20
21         setSize(800, 300);
22         setTitle(" Disposición Muelles");
23         setLocationRelativeTo(null);
24
25         add(new LaminaDisposicionMuelle());
26
27         setVisible(true);
28         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
29     }
30 }
31
32 class LaminaDisposicionMuelle extends JPanel {
33
34     SpringLayout disposicion;
35
36     public LaminaDisposicionMuelle() {
37
38         setLayout(disposicion = new SpringLayout());
39
40         JButton btn1 = new JButton("Botón 1");
41         JButton btn2 = new JButton("Botón 2");
42         JButton btn3 = new JButton("Botón 3");
43
44         add(btn1);
45         add(btn2);
46         add(btn3);
47
48         Spring muelle = Spring.constant(0, 0, 10);
49         Spring muelleFijo = Spring.constant(50);
50
51         disposicion.putConstraint(SpringLayout.WEST, btn1, muelle, SpringLayout.WEST, this);
52         disposicion.putConstraint(SpringLayout.WEST, btn2, muelleFijo, SpringLayout.EAST,
53             btn1);
53         disposicion.putConstraint(SpringLayout.WEST, btn3, muelleFijo, SpringLayout.EAST,
54             btn2);
54         disposicion.putConstraint(SpringLayout.EAST, this, muelle, SpringLayout.EAST, btn3);
55     }
56 }
```

\_04\_DisposicionLibre.java

```
1 package EjerciciosBloque3;
2
3 import javax.swing.JButton;
4 import javax.swing.JFrame;
5 import javax.swing.JLabel;
6 import javax.swing.JPanel;
7 import javax.swing.JTextField;
8
9 public class _04_DisposicionLibre {
10
11     public static void main(String[] args) {
12
13         MarcoDisposicionLibre marco = new MarcoDisposicionLibre();
14     }
15 }
16
17 class MarcoDisposicionLibre extends JFrame {
18
19     public MarcoDisposicionLibre() {
20
21         setTitle(" Disposición Libre");
22         setSize(400, 500);
23         setLocationRelativeTo(null);
24
25         add(new VentanaDisposicionLibre());
26
27         setVisible(true);
28         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
29     }
30 }
31
32 class VentanaDisposicionLibre extends JPanel {
33
34     public VentanaDisposicionLibre() {
35
36         setLayout(null);
37
38         JLabel txt1 = new JLabel("Nombre");
39         JLabel txt2 = new JLabel("Apellido");
40
41         JTextField field1 = new JTextField(20);
42         JTextField field2 = new JTextField(20);
43
44         JButton btn = new JButton("Enviar");
45
46         add(txt1).setBounds(50, 100, 50, 30);
47         add(field1).setBounds(150, 100, 150, 30);
48         add(txt2).setBounds(50, 200, 50, 30);
49         add(field2).setBounds(150, 200, 150, 30);
50         add(btn).setBounds(150, 300, 150, 30);
51     }
52 }
```

\_05\_PlantillasPersonalizadas.java

```
1 package EjerciciosBloque3;
2
3 import java.awt.Component;
4 import java.awt.Container;
5 import java.awt.Dimension;
6 import java.awt.LayoutManager;
7 import javax.swing.JButton;
8 import javax.swing.JFrame;
9 import javax.swing.JLabel;
10 import javax.swing.JPanel;
11 import javax.swing.JTextField;
12
13 public class _05_PlantillasPersonalizadas {
14
15     public static void main(String[] args) {
16
17         MarcoPlantillasPersonalizadas marco = new MarcoPlantillasPersonalizadas();
18     }
19 }
20
21 class MarcoPlantillasPersonalizadas extends JFrame {
22
23     public MarcoPlantillasPersonalizadas() {
24
25         setTitle(" Disposición Libre");
26         setSize(400, 400);
27         setLocationRelativeTo(null);
28
29         add(new VentanaPlantillasPersonalizadas());
30
31         setVisible(true);
32         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
33     }
34 }
35
36 class VentanaPlantillasPersonalizadas extends JPanel {
37
38     public VentanaPlantillasPersonalizadas() {
39
40         setLayout(new DistribucionPersonalizada());
41
42         JLabel txt1 = new JLabel("Nombre");
43         JLabel txt2 = new JLabel("Apellido");
44
45         JTextField field1 = new JTextField(20);
46         JTextField field2 = new JTextField(20);
47
48         JButton btn = new JButton("Enviar");
49
50         add(txt1);
51         add(field1);
52         add(txt2);
53         add(field2);
54         add(btn);
55     }
56
57     private class DistribucionPersonalizada implements LayoutManager {
58
59         @Override
60         public void layoutContainer(Container parent) {
61
62             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 }
```

```
1 package EjerciciosBloque3;
2
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5 import javax.swing.JButton;
6 import javax.swing.JFrame;
7 import javax.swing.JOptionPane;
8 import javax.swing.JPanel;
9
10 public class _06_CuadroDialogos {
11
12     public static void main(String[] args) {
13
14         MarcoCuadroDialogos marco = new MarcoCuadroDialogos();
15     }
16 }
17
18 class MarcoCuadroDialogos extends JFrame {
19
20     public MarcoCuadroDialogos() {
21
22         setTitle(" Cuadros Diálogos");
23         setSize(600, 400);
24         setLocationRelativeTo(null);
25
26         add(new VentanaCuadroDialogos());
27
28         setVisible(true);
29         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
30     }
31 }
32
33 class VentanaCuadroDialogos extends JPanel {
34
35     private JButton btn1, btn2, btn3, btn4;
36
37     public VentanaCuadroDialogos() {
38
39         add(btn1 = new JButton("Advertencia"));
40         add(btn2 = new JButton("Intro Datos"));
41         add(btn3 = new JButton("Confirmar"));
42         add(btn4 = new JButton("Selección"));
43
44         ClaseOyente oyente = new ClaseOyente();
45
46         btn1.addActionListener(oyente);
47         btn2.addActionListener(oyente);
48         btn3.addActionListener(oyente);
49         btn4.addActionListener(oyente);
50     }
51
52     private class ClaseOyente implements ActionListener {
53
54         @Override
55         public void actionPerformed(ActionEvent e) {
56
57             String tallas[] = {"S", "M", "L", "XL"};
58
59             if (e.getSource() == btn1)
60                 JOptionPane.showMessageDialog(VentanaCuadroDialogos.this, "Le advertimos que ...",
61 "Advertencia", 2);
62             if (e.getSource() == btn2)
```

\_06\_CuadroDialogos.java

```
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 }
```

\_07\_Ejercicio\_Dialogos.java

```
1 package EjerciciosBloque3;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import java.awt.Component;
6 import java.awt.Dimension;
7 import java.awt.FlowLayout;
8 import java.awt.event.ActionEvent;
9 import java.awt.event.ActionListener;
10 import java.text.DateFormat;
11 import java.text.SimpleDateFormat;
12 import java.util.Date;
13 import javax.swing.BorderFactory;
14 import javax.swing.Box;
15 import javax.swing.ButtonGroup;
16 import javax.swing.ImageIcon;
17 import javax.swing.JButton;
18 import javax.swing.JFrame;
19 import javax.swing.JLabel;
20 import javax.swing.JOptionPane;
21 import javax.swing.JPanel;
22 import javax.swing.JRadioButton;
23
24 public class _07_Ejercicio_Dialogos {
25
26     public static void main(String[] args) {
27
28         MarcoDialogos marco = new MarcoDialogos();
29     }
30 }
31
32 class MarcoDialogos extends JFrame {
33
34     public MarcoDialogos() {
35
36         setSize(700, 500);
37         setTitle(" Ejercicio Diálogos");
38         setLocationRelativeTo(null);
39
40         add(new LaminaDialogos());
41
42         pack();
43         setVisible(true);
44         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
45     }
46 }
47
48 class LaminaDialogos extends JPanel {
49
50     private JRadioButton    radioTipo1, radioTipo2, radioTipo3, radioTipo4,
51                             radioTipoMnsj1, radioTipoMnsj2, radioTipoMnsj3, radioTipoMnsj4,
52                             radioTipoMnsj5,
53                             radioMnsj1, radioMnsj2, radioMnsj3, radioMnsj4, radioMnsj5,
54                             radioConfirm1, radioConfirm2, radioConfirm3, radioConfirm4,
55                             radioOpcion1, radioOpcion2, radioOpcion3,
56                             radioEntrada1, radioEntrada2;
57     private ButtonGroup    group1, group2, group3, group4, group5, group6;
58     private Box            box1, box2, box3, box4, box5, box6,
59                             boxHorizontal1, boxHorizontal2, boxVertical;
60     private JPanel        laminaBtn;
61     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));
```

\_07\_Ejercicio\_Dialogos.java

```
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);
```

\_07\_Ejercicio\_Dialogos.java

```
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

```
1 package EjerciciosBloque4;
2
3 import java.io.File;
4
5 public class _01_AccesoFicheros {
6
7     public static void main(String[] args) {
8
9         File ruta = new File("C:\\Users\\DELL-OPTIPLEX-ENGEL\\Desktop\\Curso de Java\\
\\EjerciciosCursoJava\\src");
10
11         System.out.println("\nExiste el fichero?: " + ruta.exists());
12
13         System.out.println("\nRuta: " + ruta.getAbsolutePath());
14
15         File[] archivos = ruta.listFiles();
16
17         System.out.println("\nCantidad total de archivos en la ruta: " + archivos.length +
"\n");
18
19         for (int i = 0; i < archivos.length; i++) {
20
21             if (archivos[i].isDirectory()) {
22
23                 String[] subCarpetas = archivos[i].list();
24
25                 if (subCarpetas.length > 0) {
26                     System.out.println("Carpeta Padre: " + archivos[i].getName() + " |
Archivos:");
27                 } else {
28                     System.out.println("Carpeta Padre: " + archivos[i].getName() + " |
Archivos: No tiene.");
29                 }
30                 for (int j = 0; j < subCarpetas.length; j++) {
31                     System.out.println(subCarpetas[j]);
32                 }
33             }
34             System.out.println("\n");
35         }
36     }
37 }
38
```

\_02\_CreandoEliminandoFicherosArchivos.java

```
1 package EjerciciosBloque4;
2
3 import java.io.File;
4 import java.io.FileNotFoundException;
5 import java.io.FileReader;
6 import java.io.FileWriter;
7 import java.io.IOException;
8 import javax.swing.JOptionPane;
9
10 public class _02_CreandoEliminandoFicherosArchivos {
11
12     public static void main(String[] args) {
13
14         // -----CREA DIRECTORIO
15         NUEVO-----
16         File ruta = new
17         File("C:"+File.separator+"Users"+File.separator+"DELL-OPTIPLEX-ENGEL"+File.separator+"Deskt
18         op"+File.separator+"Curso de
19         Java"+File.separator+"EjerciciosCursoJava"+File.separator+"src"+File.separator+"servidor"+F
20         ile.separator+"Directorio Nuevo");
21
22         ruta.mkdir();
23
24         if (ruta.exists()) {
25             System.out.println("\nDirectorio Nuevo creado con éxito! =) \n");
26             JOptionPane.showMessageDialog(null, "Directorio Nuevo creado con éxito!");
27         } else {
28             System.out.println("\nDirectorio Nuevo NO existe! :( \n");
29             JOptionPane.showMessageDialog(null, "Directorio Nuevo NO existe!");
30         }
31
32         // -----CREA ARCHIVO NUEVO-----
33         File rutaArchivo = new
34         File("C:"+File.separator+"Users"+File.separator+"DELL-OPTIPLEX-ENGEL"+File.separator+"Deskt
35         op"+File.separator+"Curso de
36         Java"+File.separator+"EjerciciosCursoJava"+File.separator+"src"+File.separator+"servidor"+F
37         ile.separator+"Directorio Nuevo"+File.separator+"Archivo Nuevo.txt");
38
39         try {
40             rutaArchivo.createNewFile();
41             if (rutaArchivo.exists()) {
42                 System.out.println("\nArchivo Nuevo creado con éxito! =) \n");
43                 JOptionPane.showMessageDialog(null, "Archivo Nuevo creado con éxito!");
44             } else {
45                 System.out.println("\nEl archivo nuevo No ha podido ser creado :( \n");
46                 JOptionPane.showMessageDialog(null, "El archivo nuevo No ha podido ser
47                 creado");
48             }
49         } catch (IOException e) {
50             System.out.println("Ha surgido un error al intentar crear el archivo nuevo");
51             e.printStackTrace();
52         }
53
54         // -----INSTANCIAS-----
55         EscribirEnArchivo escribir = new EscribirEnArchivo(rutaArchivo);
56
57         try {
58             LeerDeArchivo leer = new LeerDeArchivo(rutaArchivo);
59         } catch (IOException e) {
60             e.printStackTrace();
61         }
62     }
63 }
```

## \_02\_CreandoEliminandoFicherosArchivos.java

```
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 }
```



\_03\_LeyendoEscribiendoBuffer.java

```
1 package EjerciciosBloque4;
2
3 import java.io.BufferedReader;
4 import java.io.BufferedWriter;
5 import java.io.File;
6 import java.io.FileReader;
7 import java.io.FileWriter;
8 import java.io.IOException;
9
10 public class _03_LeyendoEscribiendoBuffer {
11
12     public static void main(String[] args) {
13
14         LeerTexto leer = new LeerTexto();
15
16         EscribirTexto escribir = new EscribirTexto();
17
18         AgregarTexto agregar = new AgregarTexto();
19
20         LeerTextoBuffer leerBuffer = new LeerTextoBuffer();
21
22         EscribirTextoBuffer escribirBuffer = new EscribirTextoBuffer();
23
24         AgregarTextoBuffer agregarBuffer = new AgregarTextoBuffer();
25
26     }
27 }
28
29
30 // -----CLASES SIN BUFFER-----
31
32 // -----CLASE PARA LEER TEXTO-----
33 class LeerTexto {
34
35     public LeerTexto() {
36         File ruta = new File("C:/Users/DELL-OPTIPLEX-ENGEL/Desktop/Curso de
37         Java/EjerciciosCursoJava/src/servidor/archivo1.txt");
38         int nro = 0;
39         char letra;
40
41         try {
42             // abro flujo
43             FileReader lectura = new FileReader(ruta);
44             System.out.println("\nTEXTO LEÍDO: \n");
45
46             while (nro != -1) {
47                 nro = lectura.read();
48                 letra = (char) nro;
49                 System.out.print(letra);
50             }
51             System.out.println("");
52             // cierro flujo
53             lectura.close();
54
55         } catch (IOException e) {
56             e.printStackTrace();
57         }
58 }
59
60 // -----CLASE PARA ESCRIBIR TEXTO-----
61 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
104 // BUFFER-----
105 class LeerTextoBuffer {
106
107     public LeerTextoBuffer() {
108         File ruta = new File("C:/Users/DELL-OPTIPLEX-ENGEL/Desktop/Curso de
Java/EjerciciosCursoJava/src/servidor/archivo1_Buffer.txt");
109         int nro = 0;
110         char letra;
111
112         try {
113             // abro flujo
114             FileReader lectura = new FileReader(ruta);
115             BufferedReader buffer = new BufferedReader(lectura);
116
117             String leido = "";
```

\_03\_LeyendoEscribiendoBuffer.java

```
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 estimulum 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         }
```

\_03\_LeyendoEscribiendoBuffer.java

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

\_04\_LeyendoEscribiendoBytes.java

```
1 package EjerciciosBloque4;
2
3 import java.io.File;
4 import java.io.FileInputStream;
5 import java.io.FileNotFoundException;
6 import java.io.FileOutputStream;
7 import java.io.IOException;
8
9 public class _04_LeyendoEscribiendoBytes {
10
11     public static void main(String[] args) {
12
13         byte[] imagen = new byte[338022];
14
15         // -----LECTURA DE BYTES-----
16         File file = new File("C:/Users/DELL-OPTIPLEX-ENGEL/Desktop/Curso de
Java/EjerciciosCursoJava/src/servidor/kitty.jpg");
17
18         try {
19             // abro flujo
20             FileInputStream archivo = new FileInputStream(file);
21
22             // Usado al principio para saber la cantidad de bytes del archivo
23             int contador = 0;
24             while (archivo.read() != -1) {
25                 contador++;
26             }
27             System.out.println(contador);
28
29             try {
30                 for (int i = 0; i < imagen.length; i++) {
31                     imagen[i] = (byte) archivo.read();
32                     System.out.println(imagen[i]);
33                 }
34
35             } catch (IOException e1) {
36                 e1.printStackTrace();
37             }
38
39             // cierro flujo
40             try {
41                 archivo.close();
42             } catch (IOException e) {
43                 e.printStackTrace();
44             }
45
46         } catch (FileNotFoundException e) {
47             e.printStackTrace();
48         }
49
50         // -----ESCRITURA DE BYTES-----
51         File fileCopy = new File("C:/Users/DELL-OPTIPLEX-ENGEL/Desktop/Curso de
Java/EjerciciosCursoJava/src/servidor/kitty_copy.jpg");
52
53         try {
54             // abro flujo
55             FileOutputStream archivoOut = new FileOutputStream(fileCopy);
56
57             for (int i = 0; i < imagen.length; i++) {
58                 try {
59                     archivoOut.write(imagen[i]);
60                 } catch (IOException e) {
```

\_04\_LeyendoEscribiendoBytes.java

```
61         e.printStackTrace();
62     }
63 }
64 // cierre 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
```

\_05\_AccesoFicheros.java

```
1 package EjerciciosBloque4;
2
3 import java.io.FileInputStream;
4 import java.io.FileNotFoundException;
5 import java.io.FileOutputStream;
6 import java.io.IOException;
7 import java.io.ObjectInputStream;
8 import java.io.ObjectOutputStream;
9 import java.io.Serializable;
10 import java.util.Date;
11 import java.util.GregorianCalendar;
12
13 public class _05_AccesoFicheros {
14
15     public static void main(String[] args) {
16
17         // -----OBJETO INSTANCIAS-----
18         Jefe jefe = new Jefe("Carmen Noguera", 3500, 1998, 2, 3);
19         jefe.setIncentivo(1000);
20
21         Empleado[] empleados = new Empleado[4];
22         empleados[0] = new Empleado("José Ruco", 2500, 2005, 7, 15);
23         empleados[1] = jefe;
24         empleados[2] = new Empleado("María Patiño", 1500, 2004, 5, 20);
25         empleados[3] = new Empleado("Arcángel Molina", 1500, 2007, 3, 5);
26
27         // -----EXPORTAR OBJETO-----
28         try {
29             // abro flujo
30             FileOutputStream rutaOut = new FileOutputStream
31             ("C:/Users/DELL-OPTIPLEX-ENGEL/Desktop/Curso de
32             Java/EjerciciosCursoJava/src/servidor/data.txt");
33             ObjectOutputStream objetoOut;
34
35             try {
36                 // abro flujo
37                 objetoOut = new ObjectOutputStream(rutaOut);
38                 objetoOut.writeObject(empleados);
39
40                 // cierro flujo
41                 objetoOut.close();
42
43             } catch (IOException e) {
44                 e.printStackTrace();
45             }
46
47             // cierro flujo
48             try {
49                 rutaOut.close();
50             } catch (IOException e) {
51                 e.printStackTrace();
52             }
53
54         } catch (FileNotFoundException e) {
55             e.printStackTrace();
56         }
57
58         // -----IMPORTAR OBJETO-----
59         try {
60             // se abre flujo
61             FileInputStream rutaIn = new FileInputStream
62             ("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 }
```



```
121     public Date getFechaContrato() {
122         return fechaContrato;
123     }
124
125
126     public void subirSueldo(double porcentaje) {
127         double aumento = sueldo * porcentaje / 100;
128         sueldo += aumento;
129     }
130
131     public String dameData() {
132         return "El Nombre es: " + nombre + ", y su sueldo es de $" + sueldo + ". Fecha de
contrato: " + fechaContrato;
133     }
134 }
135
136 // -----CLASE JEFE-----
137 class Jefe extends Empleado {
138
139     // se crea la huella única del programa
140     private static final long serialVersionUID = 1L;
141
142     private double incentivo;
143
144     public Jefe(String n, double s, int anio, int mes, int dia) {
145         super(n, s, anio, mes, dia);
146         incentivo = 0;
147     }
148
149     public double getSueldo() {
150         double sueldoBase = super.getSueldo();
151         return sueldoBase + incentivo;
152     }
153
154     public void setIncentivo(double b) {
155         incentivo = b;
156     }
157
158     public String dameData() {
159         return super.toString() + ". Incentivo: " + incentivo;
160     }
161 }
```

\_06\_ProgramacionGenerica\_ArrayList.java

```
1 package EjerciciosBloque4;
2
3 import java.io.File;
4
5 public class _06_ProgramacionGenerica_ArrayList {
6
7     public static void main(String[] args) {
8
9         // -----1ERA INSTANCIA-----
10        ClaseGenerica instancia = new ClaseGenerica(6);
11        instancia.ElementoNuevo("Pedro");
12        instancia.ElementoNuevo("María");
13        instancia.ElementoNuevo("Juan");
14        instancia.ElementoNuevo("Rosita");
15        instancia.ElementoNuevo("Diego");
16        instancia.ElementoNuevo("Anna");
17
18        System.out.println(instancia.dameElemento(2));
19        System.out.println(instancia.dameElemento(4));
20
21        // -----2DA INSTANCIA-----
22        File ruta = new File("C:\\Users\\DELL-OPTIPLEX-ENGEL\\Desktop\\Curso de Java\\
23        \\EjerciciosCursoJava\\src");
24
25        ClaseGenerica instancia2 = new ClaseGenerica(1);
26        instancia2.ElementoNuevo(ruta);
27
28        System.out.println(instancia2.dameElemento(0));
29    }
30
31    // -----CLASE CON UN OBJETO GENÉRICO-----
32    class ClaseGenerica {
33
34        private Object[] array;
35        private int tamaño = 0;
36
37        public ClaseGenerica(int tamaño) {
38            array = new Object[tamaño];
39        }
40
41        public Object dameElemento(int elemento) {
42            return array[elemento];
43        }
44
45        public void ElementoNuevo(Object nuevo) {
46            array[tamaño] = nuevo;
47            tamaño++;
48        }
49    }
```

\_07\_ArrayList\_Iterator.java

```
1 package EjerciciosBloque4;
2
3 import java.util.ArrayList;
4 import java.util.Iterator;
5
6 public class _07_ArrayList_Iterator {
7
8     public static void main(String[] args) {
9
10         // -----INSTANCIAR EL ARRAYLIST-----
11         ArrayList<EmpleadoClase> misEmpleados = new ArrayList<EmpleadoClase>();
12
13         // -----ASEGURAR UN TAMAÑO DEL ARRAYLIST-----
14         misEmpleados.ensureCapacity(11);
15
16         // -----AGREGAR ELEMENTOS AL ARRAYLIST-----
17         misEmpleados.add(0, new EmpleadoClase("Diego", 38, 7800));
18         misEmpleados.add(1, new EmpleadoClase("Martha", 28, 2800));
19         misEmpleados.add(2, new EmpleadoClase("Rebecca", 32, 4800));
20         misEmpleados.add(3, new EmpleadoClase("Elain", 48, 5800));
21         misEmpleados.add(4, new EmpleadoClase("Willinton", 35, 1800));
22         misEmpleados.add(5, new EmpleadoClase("Anne", 30, 2800));
23
24         // -----RECORTAR EL ARRAYLIST-----
25         misEmpleados.trimToSize();
26
27         // -----TAMAÑO DEL ARRAYLIST-----
28         System.out.println("\nCantidad de elementos en el ArrayList: " +
29             misEmpleados.size());
30
31         // -----REEMPLAZAR UN ELEMENTO-----
32         misEmpleados.set(0, new EmpleadoClase("Juan", 42, 3500));
33
34         // -----IMPRIMIR UN ELEMENTO-----
35         System.out.println("\nEmpleado nro 3: " + misEmpleados.get(3).dameDatos() + "\n");
36
37         // -----IMPRIMIR TODOS LOS ELEMENTOS-----
38         for (int i = 0; i < misEmpleados.size(); i++) {
39             System.out.println("Empleado nro " + (i+1) + " : " +
40                 misEmpleados.get(i).dameDatos());
41         }
42
43         // -----ITERATOR-----
44         System.out.println("\nDatos obtenidos por el Iterator:");
45
46         Iterator<EmpleadoClase> iterador = misEmpleados.iterator();
47
48         while (iterador.hasNext()) {
49             // OJO: se debe crear un objeto de tipo clase para poder recorrer y evitar
50             errores
51             EmpleadoClase empleadoIterator = iterador.next();
52
53             System.out.println(empleadoIterator.dameDatos());
54
55             // -----ELIMINAR UN EMPLEADO POR SU
56             NOMBRE-----
57             if (empleadoIterator.dameNombre().equals("Willinton")) {
58                 iterador.remove();
59             }
60         }
61
62         // -----CONVERTIR ARRAYLIST EN ARRAY
```

\_07\_ArrayList\_Iterator.java

```
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 }
```

```
1 package EjerciciosBloque4;
2
3 public class _08_ClasesGenericas {
4
5     public static void main(String[] args) {
6
7         // -----INSTANCIA CLASE GENÉRICA TIPO STRING-----
8         ParejaOcho<String> empleados = new ParejaOcho<String>();
9         empleados.setVariable("Diego de la Rua");
10        System.out.println(empleados.getVariable());
11
12        // -----INSTANCIA CLASE GENÉRICA TIPO EMPLEADO-----
13        EmpleadoOcho empleado = new EmpleadoOcho("Amanda Greick", 35, 3500);
14        ParejaOcho<EmpleadoOcho> empleadoPareja = new ParejaOcho<EmpleadoOcho>();
15        empleadoPareja.setVariable(empleado);
16        ParejaOcho.imprimeVariable(empleadoPareja);
17
18        // -----INSTANCIA CLASE GENÉRICA TIPO PERSONA-----
19        Persona persona = new Persona("Alejandro Arteaza");
20        ParejaOcho<Persona> empleadoPersona = new ParejaOcho<Persona>();
21        empleadoPersona.setVariable(persona);
22        System.out.println("Genérico Tipo Persona: " + empleadoPersona.getVariable());
23    }
24 }
25
26 // -----CLASE EMPLEADO-----
27 class EmpleadoOcho {
28
29     private String nombre;
30     private int edad;
31     private double salario;
32
33     public EmpleadoOcho(String nombre, int edad, double salario) {
34         this.nombre = nombre;
35         this.edad = edad;
36         this.salario = salario;
37     }
38
39     public String dameData() {
40         return "Nombre: " + nombre + ", Edad: " + edad + "años, Salario: " + salario + "€";
41     }
42 }
43
44 // -----CLASE GENÉRICA PAREJA-----
45 class ParejaOcho<T> {
46
47     private T variable;
48
49     public ParejaOcho() {
50         variable = null;
51     }
52
53     public void setVariable(T mod) {
54         this.variable = mod;
55     }
56
57     public T getVariable() {
58         return variable;
59     }
60
61     public static void imprimeVariable(ParejaOcho<? extends EmpleadoOcho> obj) {
62         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 }
```

\_09\_MetodosGenericos.java

```
1 package EjerciciosBloque4;
2
3 import java.util.GregorianCalendar;
4
5 public class _09_MetodosGenericos {
6
7     public static void main(String[] args) {
8
9         // -----ARRAY DE
10        NOMBRES-----
11        System.out.println("ARRAY DE NOMBRES:");
12
13        String[] nombres = {"Pepito Fulano", "Marconi Lusejo", "Escarlata Dientreto",
14        "Antonie Lexpury", "Dawal Espertoo"};
15
16        ClaseMetodosGenericos metodos = new ClaseMetodosGenericos();
17
18        System.out.println(metodos.cantidadElementos(nombres));
19
20        System.out.println("Elemento menor: " + metodos.valorMenor(nombres));
21
22        // -----ARRAY DE FECHAS-----
23        System.out.println("\nARRAY DE FECHAS:");
24
25        GregorianCalendar[] fechas = new GregorianCalendar[4];
26        fechas[0] = new GregorianCalendar(2005, 3, 14);
27        fechas[1] = new GregorianCalendar(2010, 8, 20);
28        fechas[2] = new GregorianCalendar(2014, 10, 3);
29        fechas[3] = new GregorianCalendar(2003, 1, 25);
30
31        ClaseMetodosGenericos fechasM = new ClaseMetodosGenericos();
32
33        System.out.println(fechasM.cantidadElementos(fechas));
34
35        System.out.println("Elemento menor: " + fechasM.valorMenor(fechas).getTime());
36
37        // -----ARRAY DE
38        EMPLEADOS-----
39        System.out.println("\nARRAY DE EMPLEADOS:");
40
41        Empleado[] misEmpleados = new Empleado[6];
42        misEmpleados[0] = new Empleado("Asprilla", 5600, 1997, 3, 25);
43        misEmpleados[1] = new Empleado("Higuira", 5600, 1997, 3, 25);
44        misEmpleados[2] = new Empleado("Valderrama", 5600, 1997, 3, 25);
45        misEmpleados[3] = new Empleado("Rincón", 5600, 1997, 3, 25);
46        misEmpleados[4] = new Empleado("James", 5600, 1997, 3, 25);
47        misEmpleados[5] = new Empleado("Falcao", 5600, 1997, 3, 25);
48
49        ClaseMetodosGenericos jugadores = new ClaseMetodosGenericos();
50
51        System.out.println(jugadores.cantidadElementos(misEmpleados));
52    }
53 }
54
55 // -----CLASE CON MÉTODOS GENÉRICOS-----
56 class ClaseMetodosGenericos {
57
58     public static <T> String cantidadElementos(T[] obj) {
59         return "El array tiene: " + obj.length + " elementos.";
60     }
61
62     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 }
```



```
1 package EjerciciosBloque4;
2
3 public class _10_ProgramacionConcurrente_SincronizandoHilos {
4
5     public static void main(String[] args) {
6
7         ClaseHilosUno uno = new ClaseHilosUno();
8         ClaseHilosDos dos = new ClaseHilosDos(uno);
9         ClaseHilosDos tres = new ClaseHilosDos(dos);
10
11         uno.start();
12         dos.start();
13         tres.start();
14     }
15 }
16
17 // -----CLASE HILOS UNO-----
18 class ClaseHilosUno extends Thread {
19
20     public void run() {
21         for (int i = 0; i < 15; i++) {
22             System.out.println("Hilo nro: " + (i+1) + ", Hilo Id: " + getId() + ", Hilo
nombre: " + getName() + ", Hilo estado: " + getState() + ", Hilo actual: " +
Thread.currentThread());
23             try {
24                 Thread.sleep(100);
25             } catch (InterruptedException e) {
26                 e.printStackTrace();
27             }
28         }
29     }
30 }
31
32 //-----CLASE HILOS DOS-----
33 class ClaseHilosDos extends Thread {
34
35     private Thread hilo;
36
37     public ClaseHilosDos(Thread hilo) {
38         this.hilo = hilo;
39     }
40
41     public void run() {
42
43         try {
44             hilo.join();
45         } catch (InterruptedException e1) {
46             e1.printStackTrace();
47         }
48
49         for (int i = 0; i < 15; i++) {
50             System.out.println("Hilo nro: " + (i+1) + ", Hilo Id: " + getId() + ", Hilo
nombre: " + getName() + ", Hilo estado: " + getState() + ", Hilo actual: " +
Thread.currentThread());
51             try {
52                 Thread.sleep(100);
53             } catch (InterruptedException e) {
54                 e.printStackTrace();
55             }
56         }
57     }
58 }
```

\_11\_UsoThreads\_VariosBotones.java

```
1 package EjerciciosBloque4;
2
3 import java.awt.BorderLayout;
4 import java.awt.Color;
5 import java.awt.Component;
6 import java.awt.Graphics;
7 import java.awt.Graphics2D;
8 import java.awt.event.ActionEvent;
9 import java.awt.event.ActionListener;
10 import java.awt.geom.Ellipse2D;
11 import java.util.ArrayList;
12 import javax.swing.JButton;
13 import javax.swing.JFrame;
14 import javax.swing.JPanel;
15
16 import EjerciciosBloque4.MarcoPelotas.DibujarPelota;
17
18 public class _11_UsoThreads_VariosBotones {
19
20     public static void main(String[] args) {
21
22         MarcoPelotas marco = new MarcoPelotas();
23         marco.setVisible(true);
24         marco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
25     }
26 }
27
28 // -----MARCO PRINCIPAL-----
29 class MarcoPelotas extends JFrame {
30
31     private JButton btnPlay1, btnPlay2, btnPlay3, btnStop1, btnStop2, btnStop3;
32     private int marcoX, marcoY;
33     private LaminaxPelotas lamina = new LaminaxPelotas();
34     private Thread hilo1, hilo2, hilo3;
35
36     public MarcoPelotas() {
37
38         setSize(600, 400);
39         setTitle(" Hilos Pelotas");
40         setLocationRelativeTo(null);
41         setLayout(new BorderLayout());
42
43         marcoX = this.getWidth();
44         marcoY = this.getHeight() - 70;
45
46 // -----LÁMINA PELOTAS-----
47         lamina.setBackground(new Color(204, 255, 204));
48         add(lamina, BorderLayout.CENTER);
49
50 // -----LÁMINA BOTONES-----
51         JPanel laminaBotones = new JPanel();
52         laminaBotones.setBackground(new Color(204, 204, 255));
53         add(laminaBotones, BorderLayout.SOUTH);
54         laminaBotones.add(btnPlay1 = new JButton("Play 1"));
55         laminaBotones.add(btnPlay2 = new JButton("Play 2"));
56         laminaBotones.add(btnPlay3 = new JButton("Play 3"));
57         laminaBotones.add(btnStop1 = new JButton("Stop 1"));
58         laminaBotones.add(btnStop2 = new JButton("Stop 2"));
59         laminaBotones.add(btnStop3 = new JButton("Stop 3"));
60
61         btnPlay1.setBackground(new Color(255, 255, 204));
62         btnPlay2.setBackground(new Color(255, 255, 204));
```

\_11\_UsaThreads\_VariosBotones.java

```
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
140         restando posiciones
141         if ((x+xw) >= marcoX) {
142             x = marcoX - xw;
143             dx = -dx;
144         }
145         if ((y+yh) >= marcoY) {
146             y = marcoY - yh;
147             dy = -dy;
148         }
149         // si las distancias x-y de la bola son menores que el marco, entonces ir
150         sumando posiciones
151         if (x < 1) {
152             x = 1;
153             dx = -dx;
154         }
155         if (y < 1) {
156             y = 1;
157             dy = -dy;
158         }
159
160     public Ellipse2D getShape(){
161         return new Ellipse2D.Double(x,y,xw,yh);
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 {
```

\_11\_UsoThreads\_VariosBotones.java

```
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: " +
198             Thread.currentThread().isInterrupted());
199
200         // éste bucle while es el responsable del efecto de movimiento
201         while (!Thread.currentThread().isInterrupted()) {
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: " +
212             Thread.currentThread().isInterrupted());
213     }
```

\_12\_Banco\_Sin\_Sincronizar.java

```
1 package EjerciciosBloque4;
2
3 import java.util.concurrent.locks.Condition;
4 import java.util.concurrent.locks.ReentrantLock;
5
6 public class _12_Banco_Sin_Sincronizar {
7
8     public static void main(String[] args) {
9
10         BancoPipe miBanco = new BancoPipe();
11
12         for (int i=0; i<100; i++) {
13             Thread t = new Thread(miBanco);
14             t.start();
15
16             try {
17                 Thread.sleep(100);
18             } catch (InterruptedException e) {
19                 e.printStackTrace();
20             }
21         }
22     }
23 }
24
25 class BancoPipe extends Thread implements Runnable {
26
27     private double cantidadTransaccion;
28     private double saldoTotalBanco = 0;
29     private int cuentaOrigen, cuentaDestino;
30     private double[] cuentas = new double[100];
31     private ReentrantLock bloqueo = new ReentrantLock();
32     private Condition condicion = bloqueo.newCondition();
33
34     public BancoPipe() {
35
36         for (int i = 0; i < cuentas.length; i++) {
37             cuentas[i] = 2000;
38         }
39     }
40
41     public double SaldoTotalBanco() {
42         for (int i = 0; i < cuentas.length; i++) {
43             saldoTotalBanco += cuentas[i];
44         }
45         return saldoTotalBanco;
46     }
47
48     public void run() {
49
50         bloqueo.lock();
51
52         try {
53
54             Thread.sleep(100);
55
56             cantidadTransaccion = Math.random()*2000;
57
58             cuentaOrigen = (int) ((Math.random())*100);
59             cuentaDestino = (int) ((Math.random())*100);
60
61             if (cuentas[cuentaOrigen] < cantidadTransaccion) {
62
```

\_12\_Banco\_Sin\_Sincronizar.java

```
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 }
```

\_13\_Banco\_Synchronized.java

```
1 package EjerciciosBloque4;
2
3 public class _13_Banco_Synchronized {
4
5     public static void main(String[] args) {
6
7         BancoPipeSynchronized miBanco = new BancoPipeSynchronized();
8
9         for (int i=0; i<100; i++) {
10             Thread t = new Thread(miBanco);
11             t.start();
12             try {
13                 Thread.sleep(100);
14             } catch (InterruptedException e) {
15                 e.printStackTrace();
16             }
17         }
18     }
19 }
20
21 class BancoPipeSynchronized extends Thread implements Runnable {
22
23     private double cantidadTransaccion;
24     private double saldoTotalBanco = 0;
25     private int cuentaOrigen, cuentaDestino;
26     private double[] cuentas = new double[100];
27
28     public BancoPipeSynchronized() {
29
30         for (int i = 0; i < cuentas.length; i++) {
31             cuentas[i] = 2000;
32         }
33     }
34
35     public double SaldoTotalBanco() {
36         for (int i = 0; i < cuentas.length; i++) {
37             saldoTotalBanco += cuentas[i];
38         }
39         return saldoTotalBanco;
40     }
41
42     public synchronized void run() {
43
44         try {
45
46             Thread.sleep(100);
47
48             cantidadTransaccion = Math.random()*2000;
49
50             cuentaOrigen = (int) ((Math.random()*100);
51             cuentaDestino = (int) ((Math.random()*100);
52
53             if (cuentas[cuentaOrigen] < cantidadTransaccion) {
54
55                 System.out.println("\nHILO ESPERANDO.....");
56                 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);
57                 System.out.printf("\nSaldo Cuenta Origen: %1.2f€, Saldo Cuenta Destino:
%1.2f€", cuentas[cuentaOrigen], cuentas[cuentaDestino]);
58                 System.out.println("\nHilo nro: " + getName() + " " + currentThread());
59

```



\_13\_Banco\_Synchronized.java

```
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 }
```