

PROGRAMACIÓN II Trabajo Práctico 4:

Programación Orientada a Objetos II

Caso Práctico Sistema de Gestión de Empleados Modelar una clase Empleado que represente a un trabajador en una empresa. Esta clase debe incluir constructores sobrecargados, métodos sobrecargados y el uso de atributos aplicando encapsulamiento y métodos estáticos para llevar control de los objetos creados.

```
1 package tp_4_poo;
2
3
4 public class Tp_4_POO {
5
6
7     public static void main(String[] args) {
8
9         Empleado e1 = new Empleado(1, "Felipe", "Gerente", 1500);
10
11         Empleado e2 = new Empleado("Carlos", "Vendedor");
12
13         e1.actualizarSalario(20);
14         e2.actualizarSalarioMonto(300);
15
16         System.out.println(e1);
17         System.out.println(e2);
18         System.out.println("Total empleados: " + Empleado.mostrarTotalEmpleados());
19     }
20
21 }
22
23 class Empleado{
24     private int id;
25     private String nombre;
26     private String puesto;
27     private double salario;
28
29     private static int totalEmpleados = 0;
30     private static int aumentoId = 2;
31
32     Empleado(int id, String nombre, String puesto, double salario) {
33         this.id = id;
34         this.nombre = nombre;
35         this.puesto = puesto;
36         this.salario = salario;
37         totalEmpleados++;
38     }
39
40     Empleado(String nombre, String puesto) {
41         this.id = siguienteId();
42         this.nombre = nombre;
43         this.puesto = puesto;
44         this.salario = 1000;
45         totalEmpleados++;
46     }
47
48     private static int siguienteId() {
49         return aumentoId++;
50     }
51
52     public void actualizarSalario(int porcentaje) {
53         this.salario = this.salario * (1 + porcentaje / 100);
54     }
55
56     public void actualizarSalarioMonto(double monto) {
57         this.salario = this.salario + monto;
58     }
59
60     public String toString() {
61         return "Empleado [id=" + id + ", nombre=" + nombre + ", puesto=" + puesto + ", salario=" + salario + "]";
62     }
63
64     public static int mostrarTotalEmpleados() {
65         return totalEmpleados;
66     }
67 }
```

```

47 |         return aumentoId++;
48 |     }
49 |
50 |     public int getId() {
51 |         return id;
52 |     }
53 |     public String getNombre() {
54 |         return nombre;
55 |     }
56 |     public void setNombre(String nombre) {
57 |         this.nombre = nombre;
58 |     }
59 |     public String getPuesto() {
60 |         return puesto;
61 |     }
62 |     public void setPuesto(String puesto) {
63 |         this.puesto = puesto;
64 |     }
65 |     public double getSalario() {
66 |         return salario;
67 |     }
68 |     public void setSalario(double salario) {
69 |         if (salario < 0) {
70 |             System.out.println("El salario no puede ser negativo");

```

```

70 |             System.out.println("El salario no puede ser negativo");
71 |             return;
72 |         }
73 |         this.salario = salario;
74 |     }
75 |
76 |     public void actualizarSalario(double porcentaje) {
77 |         double cuenta = salario += salario * (porcentaje / 100);
78 |         setSalario(cuenta);
79 |     }
80 |     public void actualizarSalarioMonto(double monto) {
81 |         double actualizando = salario = salario + monto;
82 |         setSalario(actualizando);
83 |     }
84 |
85 |     public static int mostrarTotalEmpleados() {
86 |         return totalEmpleados;
87 |     }
88 |
89 |     @Override
90 |     public String toString() {
91 |         return "ID:= " + id + ", Nombre:= " + nombre + ", Puesto:= " + puesto + ", Salario:= " + salario;
92 |     }
93 | }

```

tp_4_poo.Empleado > toString >

Output - tp_4_POO (run) x

```

run:
ID:= 1, Nombre:= Felipe, Puesto:= Gerente, Salario:= 1800.0
ID:= 2, Nombre:= Carlos, Puesto:= Vendedor, Salario:= 1300.0
Total empleados: 2
BUILD SUCCESSFUL (total time: 0 seconds)

```

Link repo: https://github.com/FacuAuciello/UTN_JAVA_2do_Cuatri