

PROGRAMACIÓN II Trabajo Práctico 4:

Programación Orientada a Objetos II

Alumno: Castellini Gonzalo

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.

CLASE EMPLEADO Atributos:

- int id: Identificador único del empleado.
- String nombre: Nombre completo.
- String puesto: Cargo que desempeña.
- double salario: Salario actual.
- static int totalEmpleados: Contador global de empleados creados.

REQUERIMIENTOS

1. Uso de this:

- Utilizar this en los constructores para distinguir parámetros de atributos.

2. Constructores sobrecargados:

- Uno que reciba todos los atributos como parámetros.
- Otro que reciba solo nombre y puesto, asignando un id automático y un salario por defecto.
- Ambos deben incrementar totalEmpleados.

3. Métodos sobrecargados actualizarSalario:

- Uno que reciba un porcentaje de aumento.
- Otro que reciba una cantidad fija a aumentar.

4. Método toString():

- Mostrar id, nombre, puesto y salario de forma legible.

5. Método estático mostrarTotalEmpleados():

- Retornar el total de empleados creados hasta el momento.

6. Encapsulamiento en los atributos:

- Restringir el acceso directo a los atributos de la clase.

- Crear los métodos Getters y Setters correspondientes.



```
29 package tp4.poo;
30
31
32 public class Empleado {
33     //Atributos
34     private int id;
35     private String nombre;
36     private String puesto;
37     private double salario;
38     private static int totalEmpleados; //contado de empleados-Variable de clase
39     private static int idAutoIncremental=1;
40
41
42     //Constructores
43
44     public Empleado(int id, String nombre, String puesto, double salario) {
45         this.id = id;
46         this.nombre = nombre;
47         this.puesto = puesto;
48         this.salario = salario;
49         totalEmpleados ++ ; //incrementa empleados
50     }
51
52     public Empleado(String nombre, String puesto) {
53         idAutoIncremental ++;
54         this.id = idAutoIncremental ++;
55         this.nombre = nombre;
56         this.puesto = puesto;
57         this.salario = 2500; //salario por defecto
58         totalEmpleados++;
59     }
60
61
62     //metodos
63     public double actualizarSalario(int porcentaje){
64         double actualizacion = salario*(porcentaje/100.0);
65         return salario += actualizacion;
66     }
67     public double actualizarSalario(double montoFijo){
68         return salario += montoFijo;
69     }
70
71
72     @Override
73     public String toString() {
74         return "Empleado{" +
75             "id=" + id +
76             ", nombre=" + nombre + '\'' +
77             ", puesto=" + puesto + '\'' +
```

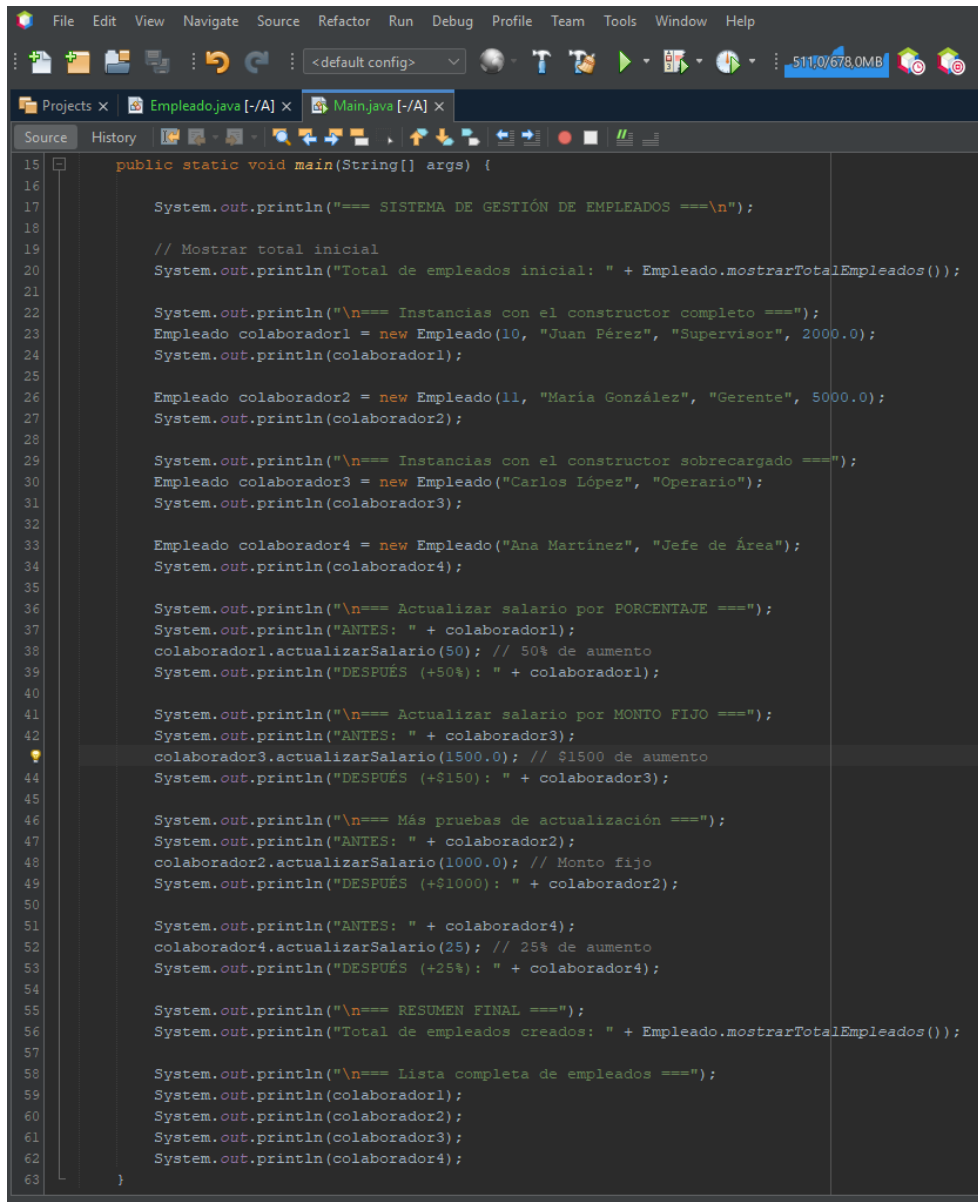
```

71
72     @Override
73     public String toString() {
74         return "Empleado{" +
75             "id=" + id +
76             ", nombre='" + nombre + '\'' +
77             ", puesto='" + puesto + '\'' +
78             ", salario=" + salario +
79             "'}";
80     }
81
82     //metodo static
83     public static int mostrarTotalEmpleados() {
84         return Empleado.totalEmpleados;
85     }
86
87     //Setters
88
89     public void setNombre(String nombre) {
90         this.nombre = nombre;
91     }
92
93     public void setPuesto(String puesto) {
94         this.puesto = puesto;
95     }
96
97     public void setSalario(double salario) {
98         this.salario = salario;
99     }
100
101
102     //Getters
103     public int getId() {
104         return id;
105     }
106
107     public double getSalario() {
108         return salario;
109     }
110
111     public String getNombre() {
112         return nombre;
113     }
114
115     public String getPuesto() {
116         return puesto;
117     }
118
119 }

```

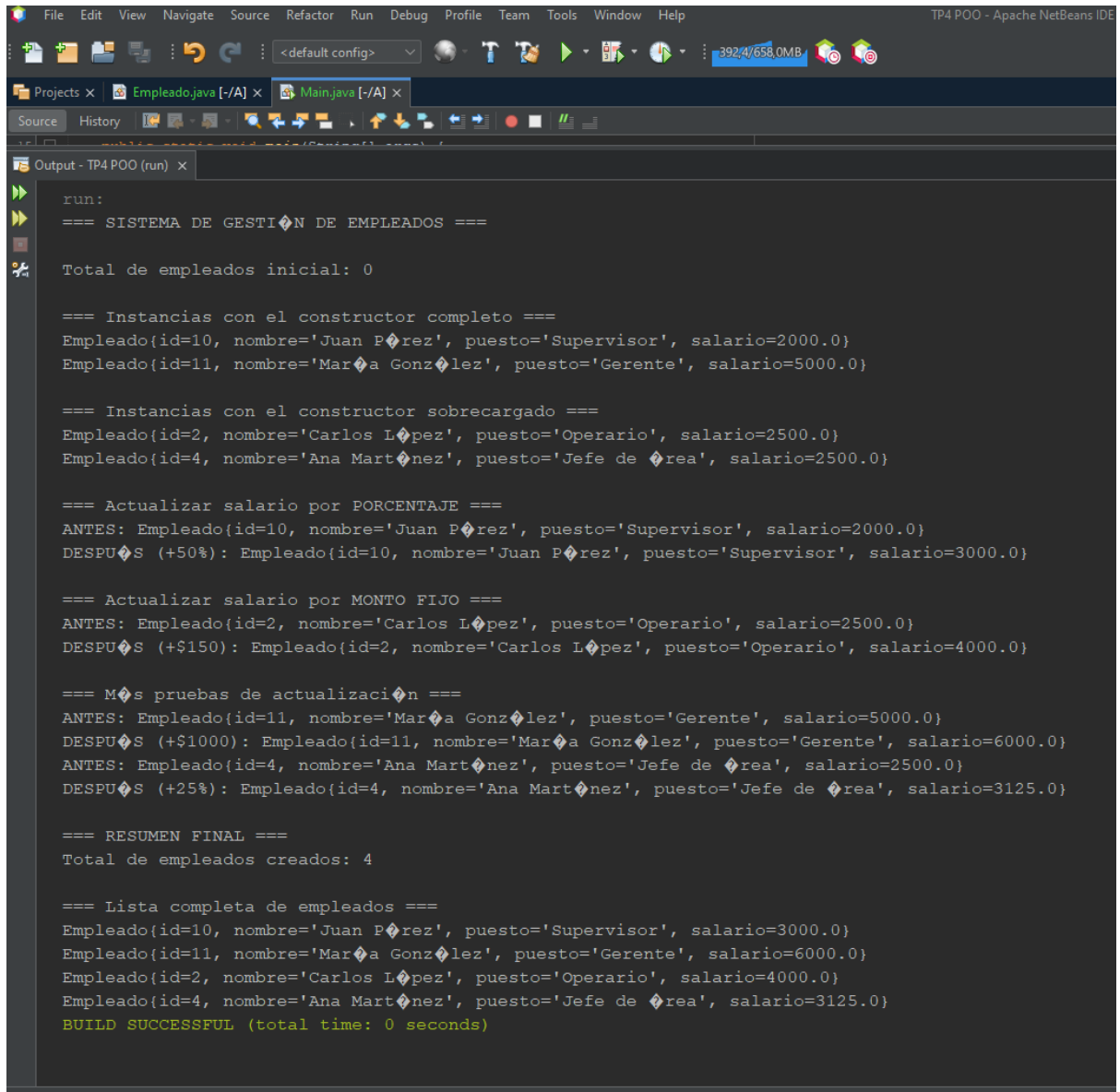
TAREAS A REALIZAR

1. Implementar la clase Empleado aplicando todos los puntos anteriores.
2. Crear una clase de prueba con método main que:
 - Instancie varios objetos usando ambos constructores.
 - Aplique los métodos actualizarSalario() sobre distintos empleados.
 - Imprima la información de cada empleado con toString().
 - Muestre el total de empleados creados con mostrarTotalEmpleados().



```
15 public static void main(String[] args) {
16
17     System.out.println("=== SISTEMA DE GESTIÓN DE EMPLEADOS ===\n");
18
19     // Mostrar total inicial
20     System.out.println("Total de empleados inicial: " + Empleado.mostrarTotalEmpleados());
21
22     System.out.println("\n=== Instancias con el constructor completo ===");
23     Empleado colaborador1 = new Empleado(10, "Juan Pérez", "Supervisor", 2000.0);
24     System.out.println(colaborador1);
25
26     Empleado colaborador2 = new Empleado(11, "María González", "Gerente", 5000.0);
27     System.out.println(colaborador2);
28
29     System.out.println("\n=== Instancias con el constructor sobrecargado ===");
30     Empleado colaborador3 = new Empleado("Carlos López", "Operario");
31     System.out.println(colaborador3);
32
33     Empleado colaborador4 = new Empleado("Ana Martínez", "Jefe de Área");
34     System.out.println(colaborador4);
35
36     System.out.println("\n=== Actualizar salario por PORCENTAJE ===");
37     System.out.println("ANTES: " + colaborador1);
38     colaborador1.actualizarSalario(50); // 50% de aumento
39     System.out.println("DESPUÉS (+50%): " + colaborador1);
40
41     System.out.println("\n=== Actualizar salario por MONTO FIJO ===");
42     System.out.println("ANTES: " + colaborador3);
43     colaborador3.actualizarSalario(1500.0); // $1500 de aumento
44     System.out.println("DESPUÉS (+$1500): " + colaborador3);
45
46     System.out.println("\n=== Más pruebas de actualización ===");
47     System.out.println("ANTES: " + colaborador2);
48     colaborador2.actualizarSalario(1000.0); // Monto fijo
49     System.out.println("DESPUÉS (+$1000): " + colaborador2);
50
51     System.out.println("ANTES: " + colaborador4);
52     colaborador4.actualizarSalario(25); // 25% de aumento
53     System.out.println("DESPUÉS (+25%): " + colaborador4);
54
55     System.out.println("\n=== RESUMEN FINAL ===");
56     System.out.println("Total de empleados creados: " + Empleado.mostrarTotalEmpleados());
57
58     System.out.println("\n=== Lista completa de empleados ===");
59     System.out.println(colaborador1);
60     System.out.println(colaborador2);
61     System.out.println(colaborador3);
62     System.out.println(colaborador4);
63 }
```

Resultados por consola



```
run:
=== SISTEMA DE GESTIÓN DE EMPLEADOS ===

Total de empleados inicial: 0

=== Instancias con el constructor completo ===
Empleado{id=10, nombre='Juan Pérez', puesto='Supervisor', salario=2000.0}
Empleado{id=11, nombre='María González', puesto='Gerente', salario=5000.0}

=== Instancias con el constructor sobrecargado ===
Empleado{id=2, nombre='Carlos López', puesto='Operario', salario=2500.0}
Empleado{id=4, nombre='Ana Martínez', puesto='Jefe de área', salario=2500.0}

=== Actualizar salario por PORCENTAJE ===
ANTES: Empleado{id=10, nombre='Juan Pérez', puesto='Supervisor', salario=2000.0}
DESPUÉS (+50%): Empleado{id=10, nombre='Juan Pérez', puesto='Supervisor', salario=3000.0}

=== Actualizar salario por MONTO FIJO ===
ANTES: Empleado{id=2, nombre='Carlos López', puesto='Operario', salario=2500.0}
DESPUÉS (+$150): Empleado{id=2, nombre='Carlos López', puesto='Operario', salario=4000.0}

=== Más pruebas de actualización ===
ANTES: Empleado{id=11, nombre='María González', puesto='Gerente', salario=5000.0}
DESPUÉS (+$1000): Empleado{id=11, nombre='María González', puesto='Gerente', salario=6000.0}
ANTES: Empleado{id=4, nombre='Ana Martínez', puesto='Jefe de área', salario=2500.0}
DESPUÉS (+25%): Empleado{id=4, nombre='Ana Martínez', puesto='Jefe de área', salario=3125.0}

=== RESUMEN FINAL ===
Total de empleados creados: 4

=== Lista completa de empleados ===
Empleado{id=10, nombre='Juan Pérez', puesto='Supervisor', salario=3000.0}
Empleado{id=11, nombre='María González', puesto='Gerente', salario=6000.0}
Empleado{id=2, nombre='Carlos López', puesto='Operario', salario=4000.0}
Empleado{id=4, nombre='Ana Martínez', puesto='Jefe de área', salario=3125.0}
BUILD SUCCESSFUL (total time: 0 seconds)
```