

# Trabajo Practico N°5 de Programación II

Estudiante: Tobias Leiva

Unidad: POO

Universidad Tecnológica Nacional

## **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.

### **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

### **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.

- o Imprima la información de cada empleado con toString().
- o Muestre el total de empleados creados con mostrarTotalEmpleados().

Respuesta:

Clase Empleado:

```
public class Empleado {  
    //Atributos  
    static int contador = 1;  
    private int id;  
    private String nombre;  
    private String puesto;  
    private double salario;  
    private static int totalEmpleado;  
  
    //Constructores  
    public Empleado(int id, String nombre, String puesto, double  
salario) {  
        this.id = id;  
        this.nombre = nombre;  
        this.puesto = puesto;  
        this.salario = salario;  
        Empleado.sumEmpleados();  
    }  
  
    public Empleado(String nombre, String puesto) {  
        contador++;  
        this.id = contador;  
        this.nombre = nombre;  
        this.puesto = puesto;  
        this.salario = 1000.0;  
        Empleado.sumEmpleados();  
    }  
}
```

```

    }

    //Suma de empleados
    static public int sumEmpleados(){
        return totalEmpleado = totalEmpleado + 1;
    }

    //Mostrar la variable totalEmpleado
    static public void getTotalEmp(){
        System.out.println("Total de Empleados: " + totalEmpleado );
    }

    //Metodos sobrecargados: actualizarSalario
    public double actualizarSalario(double porcentaje){
        porcentaje = porcentaje / 100;
        return this.salario = salario +(porcentaje * salario);
    }
    public double actualizarSalario(){
        return actualizarSalario(50);
    }

    //Metodo toString()
    @Override
    public String toString() {
        return "Empleado: " + "id = " + id + ", nombre = " + nombre +
        ", puesto = " + puesto + ", salario = " + salario;
    }
}

```

Programa main:

```

public class Principal {

    public static void main(String[] args) {

```

```
//Objetos usando constructores
Empleado emp1 = new Empleado(1, "Tiziano", "Ingeniero", 3000.0);
Empleado emp2 = new Empleado("Juan", "Supervisor");
Empleado emp3 = new Empleado("Rodrigo", "Supervisor");

//Empleado 1: Actualizar Salario
emp1.actualizarSalario(10);
System.out.println(emp1);
emp1.actualizarSalario();
System.out.println(emp1);

System.out.println();

//Empleado 2: Actualizar Salario
emp2.actualizarSalario(20);
System.out.println(emp2);
emp2.actualizarSalario();
System.out.println(emp2);

System.out.println();

//Total de empleados
Empleado.getTotalEmp();

System.out.println();

//Información de los empleados con ToString()
System.out.println("Información de los Empleados");
System.out.println(emp1);
System.out.println(emp2);
System.out.println(emp3);
}
```

}