TRABAJO PRÁCTICO 4 – PROGRAMACIÓN ORIENTADA A OBJETOS II

Nombre: Agustina

Apellido: Cruz

Comisión: Nº12.

Materia: Programación II

Profesor: Ariel Enferrel

Año: 2025

Consignas:

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.

LINK REPOSITORIO: https://github.com/agustinamilagroscruz/UTN-TUPaD-P2/blob/main/docs/TRABAJO%20PRÁCTICO%20Nº4%20POO.pdf

RESPUESTAS:

1. Se utilizó this en los constructores para diferenciar los atributos de los parámetros recibidos.

2. Se implementaron dos constructores, uno con todos los atributos y otro con solo nombre y puesto, asignando id automático y salario por defecto.

3. Se implementaron dos métodos con el mismo nombre actualizarSalario, sobrecargados según los parámetros.

4.Se sobrescribió el método toString() para devolver la información completa del empleado.

5. Se creó un método estático que retorna el total de empleados instanciados.

6. Los atributos se declararon privados y se generaron los getters y setters.

```
package Unidad_04;

public class Empleado {

    // Atributos privados (Encapsulamiento)
    private int id;
    private String nombre;
    private String puesto;
    private double salario;

    // Atributo estático
    private static int totalEmpleados = 0;

    // Constructor con todos los atributos
    public Empleado(int id, String nombre, String puesto, double salario) {
        this.id = id;
        this.nombre = nombre;
        this.puesto = puesto;
        this.salario = salario;
        totalEmpleados++;
    }

    // Constructor sobrecargado (solo nombre y puesto)
    public Empleado(String nombre, String puesto) {
        this.id = ++totalEmpleados;
        this.nombre = nombre;
        this.puesto = puesto;
        this.salario = 50000;
    }

    // Métodos sobrecargados actualizarSalario
    public void actualizarSalario(double porcentaje) {
        this.salario += this.salario * (porcentaje / 100);
    }

    public void actualizarSalario(int cantidadFija) {
        this.salario += cantidadFija;
    }
```

```
// Método toString sobrescrito
@Override
public String toString() {
    return "Empleado [ID=" + id + ", Nombre=" + nombre +
           ", Puesto=" + puesto + ", Salario=" + salario + "]";
}

// Método estático
public static int mostrarTotalEmpleados() {
    return totalEmpleados;
}

// Getters y Setters
public int getId() { return id; }
public void setId(int id) { this.id = id; }

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

public String getPuesto() { return puesto; }
public void setPuesto(String puesto) { this.puesto = puesto; }

public double getSalario() { return salario; }
public void setSalario(double salario) { this.salario = salario; }

// Método main de prueba
public static void main(String[] args) {
    Empleado e1 = new Empleado("Ana", "Administrativa");
    Empleado e2 = new Empleado("Carlos", "Programador");
    Empleado e3 = new Empleado(10, "María", "Diseñadora", 60000);

    System.out.println(e1);
    System.out.println(e2);
    System.out.println(e3);

    e1.actualizarSalario(10);       // aumenta 10%
    e2.actualizarSalario(5000);     // aumenta cantidad fija

    System.out.println("\nDespués de actualizar salarios:");
    System.out.println(e1);
    System.out.println(e2);

    System.out.println("\nTotal de empleados: " +
```

```
Empleado.mostrarTotalEmpleados());
    }
}
```

Salida en Consola:

Empleado [ID=1, Nombre=Ana, Puesto=Administrativa, Salario=50000.0]

Empleado [ID=2, Nombre=Carlos, Puesto=Programador, Salario=50000.0]

Empleado [ID=10, Nombre=María, Puesto=Diseñadora, Salario=60000.0]

Después de actualizar salarios:

Empleado [ID=1, Nombre=Ana, Puesto=Administrativa, Salario=50010.0]

Empleado [ID=2, Nombre=Carlos, Puesto=Programador, Salario=55000.0]

Total de empleados: 3