

# TP4 Programación II

Nombre: Leandro Lopez

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.

```

public class Empleado {
    private int id;
    private String nombre;
    private String puesto;
    private double salario;
    private static int totalEmpleados;

    public Empleado(int id, String nombre, String puesto, double salario) {
        this.id = id;
        this.nombre = nombre;
        this.puesto = puesto;
        this.salario = salario;
        totalEmpleados++;
    }

    public Empleado(String nombre, String puesto) {
        this(23,nombre,puesto,500000);
    }

    public void actualizarSalario(double porcentajeaumento){
        salario+=(porcentajeaumento/100*salario);
    }

```

```

    public void actualizarSalario(int cantaumentar){
        salario+=cantaumentar;
    }

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

    public static int mostrarTotalEmpleados(){
        return totalEmpleados;
    }

    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;
    }
}

```

## 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()

```

public static void main(String[] args) {
    Empleado e1= new Empleado(1,"Leandro","ayudante",70000.0);
    Empleado e2= new Empleado(1,"Iris","encargado",99000.0);
    Empleado e3= new Empleado("Angel","gerente");

    e1.actualizarSalario(10.0);
    e2.actualizarSalario(10000);

    System.out.println("Empleados:");
    System.out.println(e1.toString());
    System.out.println(e2.toString());
    System.out.println(e3.toString());

    System.out.println("Total de empleados:"+Empleado.mostrarTotalEmpleados());
}

```

## SALIDA POR CONSOLA:

```
Run:
Empleados:
Empleado{id=1, nombre=Leandro, puesto=ayudante, salario=77000.0}
Empleado{id=2, nombre=Iris, puesto=encargado, salario=109000.0}
Empleado{id=23, nombre=Angel, puesto=gerente, salario=500000.0}
Total de empleados:3
BUILD SUCCESSFUL (total time: 0 seconds)
```