

# TP4: POO

[https://github.com/MatiasBoyer/tupad\\_progra2](https://github.com/MatiasBoyer/tupad_progra2)

## 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

## TAREAS A REALIZAR

1. Implementar la clase Empleado aplicando todos los puntos anteriores.

```
package com.matiassb.javatp4;

public class Empleado {

    // static: clase
    private static int totalEmpleados;
    public static int mostrarTotalEmpleados()
    {
        return totalEmpleados;
    }

    // instancia
    private int id;
    private String nombre;
    private String puesto;
    private double salario;

    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.id = totalEmpleados;
        this.nombre = nombre;
        this.puesto = puesto;
        this.salario = 25000;
        totalEmpleados++;
    }

    // no hay setId, los id son únicos.
    public int getId() {
        return 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;
    }

    // no hay getSalario, ya que usamos actualizarSalario
    public void actualizarSalario(int salario) {
        this.salario = salario;
    }

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

    @Override
    public String toString()
    {
        return "[Empleado] ID: "+this.id+" / Nombre: "+this.nombre+" / Puesto: "+this.puesto+" / Salario: "+this.salario+" ";
    }
}
```

Boyer Rodríguez Matías Ruben

[matiasboyer7@gmail.com](mailto:matiasboyer7@gmail.com)

DNI 43.901.799

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().

```
package com.matiassb.javatp4;

public class Javatp4 {

    public static void main(String[] args) {
        Empleado emp1 = new Empleado("Matias Boyer", "Tester");
        Empleado emp2 = new Empleado(10, "Lionel Messi", "Futbolista",
999999);
        Empleado emp3 = new Empleado(30, "Charly Garcia", "Musico",
500000);

        emp1.actualizarSalario(25.0);
        emp2.actualizarSalario(1500000);
        emp3.actualizarSalario(-50.0);

        System.out.println(emp1.toString());
        System.out.println(emp2.toString());
        System.out.println(emp3.toString());

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

```
--- exec:3.1.0:exec (default-cli) @ javatp4 ---
[Empleado] ID: 0 / Nombre: Matias Boyer / Puesto: Tester / Salario:
31250.0
[Empleado] ID: 10 / Nombre: Lionel Messi / Puesto: Futbolista / Salario:
1500000.0
[Empleado] ID: 30 / Nombre: Charly Garcia / Puesto: Musico / Salario:
250000.0
Total de empleados: 3
-----
BUILD SUCCESS
-----
Total time: 1.210 s
Finished at: 2025-09-08T22:04:38-03:00
-----
```