

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.
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 tp4_poo;
```

```
public class Empleado {
```

```
    // Atributos (encapsulados)
```

```
    private int id;          // Identificador único del empleado
```

```
    private String nombre;  // Nombre completo
```

```
    private String puesto;  // Cargo que desempeña
```

```
    private double salario; // Salario actual
```

```
    // Contadores
```

```
    private static int totalEmpleados = 0; // contador global
```

```
    private static int ultimoId = 0;      // id automático
```

```
    private static final double SALARIO_BASE = 100000.0; // salario por defecto
```

```
// ----- Constructores -----

// Recibe 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++;
    // Para que no se dupliquen los ID
    if (id > ultimoId) {
        ultimoId = id;
    }
}

// Solo nombre y puesto: asigna id automático y salario por defecto
public Empleado(String nombre, String puesto) {
    this.id = ++ultimoId;    // id automático
    this.nombre = nombre;
    this.puesto = puesto;
    this.salario = SALARIO_BASE;    // salario por defecto
    totalEmpleados++;
}

// ----- Sobrecargas actualizarSalario -----

// Aumenta el salario por porcentaje (10 = +10%)
public void actualizarSalario(int porcentaje) {
    if (porcentaje <= 0) {
        System.out.println("Porcentaje inválido.");
        return;
    }
}

// Aumenta el salario por un monto fijo (50000.0 = +$50.000)
public void actualizarSalario(double monto) {
    if (monto <= 0) {
        System.out.println("Monto inválido.");
        return;
    }
    this.salario = this.salario + monto;
}

// ----- Getters/Setters -----

public int getId() { return id; } // id solo lectura
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; }

// ----- toString -----
```

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

// ----- Método estático -----

public static int mostrarTotalEmpleados() {
    return totalEmpleados; // retorna cant. de empleadod
}

// -----Ejecucion main-----

public static void main(String[] args) {
    // 1) Con constructor completo
    Empleado e1 = new Empleado(100, "Ana", "Analista", 250000.0);

    // 2) Con constructor reducido (id auto + salario por defecto)
    Empleado e2 = new Empleado("Bruno", "Soporte");
    Empleado e3 = new Empleado("Carla", "Ventas");

    // 3) Aplicar sobrecargas actualizarSalario
    e2.actualizarSalario(10); // +10%
    e3.actualizarSalario(50000.0); // +$50.000

    // 4) Imprimir info
    System.out.println(e1);
    System.out.println(e2);
    System.out.println(e3);

    // 5) Mostrar total de empleados creados
    System.out.println("Total de empleados: " + Empleado.mostrarTotalEmpleados());
}
}
```