

## PROGRAMACIÓN II

### TRABAJO PRÁCTICO 4: PROGRAMACIÓN ORIENTADA A OBJETOS II

CHRISTIAN EMMANUEL OLIVERO

<https://github.com/CHRISTIAN2320/UTN-TUPAD-Programacion2>

#### 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.*

```
public class Empleado {  
    private int id;           // Identificador único del empleado.  
    private String nombre;    // Nombre completo.  
    private String puesto;    // Cargo que desempeña.  
    private double salario;    // Salario actual.  
    private static int totalEmpleados=0; // Contador global de empleados creados.  
  
    //CONSTRUCTOR  
    public Empleado(int id, String nombre, String puesto, double salario) {  
        totalEmpleados++;  
        this.id = id;  
        this.nombre = nombre;  
        this.puesto = puesto;  
        this.salario = salario;  
    }  
    //SOBRECARGA DE CONSTRUCTOR  
    public Empleado(String nombre, String puesto) {  
        totalEmpleados++;  
        this.nombre = nombre;  
        this.puesto = puesto;  
        this.salario = 1000;  
        this.id = 1234;  
    }  
}
```

```
    public double getSalario() {  
        return salario;  
    }  
  
    // METODOS y SOBRECARGAS  
  
    public double actualizarSalario ( double cantidadAumentar){  
        return salario += cantidadAumentar;  
    }  
    public double actualizarSalario (double porcentajeAumento ){  
        return salario+= porcentajeAumento*salario/100;  
    }  
  
    // MÉTODO ESTÁTICO  
    public static int mostrarTotalEmpleados(){  
        return totalEmpleados;  
    }  
  
    @Override  
    public String toString() {  
        return "Empleado{" + "id=" + id + ", nombre=" + nombre + ", puesto=" + puesto + ", salario=" +  
salario + '}';  
    }  
}
```

```

public static void main(String[] args) {
    System.out.println("Estado Actual de los empleados: " + Empleado.mostrarTotalEmpleados());
    System.out.println("");
    Empleado e1 = new Empleado(1111, "Christian", "Programador", 1200);
    Empleado e2 = new Empleado("Emmanuel", "Analista de datos");

    System.out.println("Empleados Creados");
    System.out.println(e1); //Forma Estandar de 4 paramatros
    System.out.println(e2); // Forma Sobrecarga , con 2 param.
    System.out.println("");
    // Salarios
    System.out.println("Mostrar Salarios");
    System.out.println("Salario 1: " + e1.getSalario());
    System.out.println("Salario 2: " + e2.getSalario());

    e1.actualizarSalario(100); // Aumentar cantidad Fija
    e2.actulizarSalario(10);  // Aumentar con porcentaje

    System.out.println("");
    System.out.println("Mostrar Salarios Actualizados");
    System.out.println("Salario 1: " + e1.getSalario());
    System.out.println("Salario 2: " + e2.getSalario());

    System.out.println("");
    // Mostrar Empleados
    System.out.println("Estado actual de los empleados: " + Empleado.mostrarTotalEmpleados());
}

```

Estado Actual de los empleados: 0

Empleados Creados

Empleado{id=1111, nombre=Chrisitian, puesto=Programador, salario=1200.0}

Empleado{id=1234, nombre=Emmanuel, puesto=Analista de datos, salario=1000.0}

Mostrar Salarios

Salario 1: 1200.0

Salario 2: 1000.0

Mostrar Salarios Actualizados

Salario 1: 1300.0

Salario 2: 1100.0

Estado actual de los empleados: 2