Resolución Práctico 4: Orientación a Objetos

Resolución: Kata - Gestión de Empleados

Código Java:

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

    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;

    }

    // Constructor con nombre y puesto (salario por defecto)
    public Empleado(String nombre, String puesto) {
        totalEmpleados++;
```

```
        this.id = totalEmpleados;
        this.nombre = nombre;
        this.puesto = puesto;
        this.salario = 1000.0; // Salario por defecto
    }

    // Método para actualizar salario con un porcentaje
    public void actualizarSalario(double porcentaje) {
        this.salario += this.salario * (porcentaje / 100);
    }

    // Método para actualizar salario con cantidad fija
    public void actualizarSalario(int cantidad) {
        this.salario += cantidad;
    }

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

    // Método estático para mostrar el total de empleados
    public static int mostrarTotalEmpleados() {
        return totalEmpleados;
    }
}
```

Ejemplo de uso:

```
public class TestEmpleado {
   public static void main(String[] args) {
      Empleado emp1 = new Empleado(1, "Juan Pérez", "Desarrollador",
2000.0);
      Empleado emp2 = new Empleado("María López", "Analista");
      emp1.actualizarSalario(10);
      emp2.actualizarSalario(200);
      System.out.println(emp1);
      System.out.println(emp2);
      System.out.println("Total de empleados: " +
Empleado.mostrarTotalEmpleados());
   }
}
```

Resolución: Kata - Gestión de Productos

Código Java:

```
public class Producto {
    private int id;
    private String nombre;
    private String categoria;
    private double precio;
    private int cantidad;

    private static int totalProductos = 0;

    // Constructor con todos los atributos
    public Producto(int id, String nombre, String categoria, double precio, int cantidad) {
        this.id = id;
        this.nombre = nombre;
        this.categoria = categoria;
        this.precio = precio;
        this.cantidad = cantidad;
        totalProductos++;
    }

    // Constructor con nombre y categoria (precio y cantidad por defecto)
    public Producto(String nombre, String categoria) {
        this.id = totalProductos + 1;
```

```
        this.nombre = nombre;
        this.categoria = categoria;
        this.precio = 0.0;
        this.cantidad = 0;
        totalProductos++;
    }

    // Método para actualizar precio con un porcentaje
    public void actualizarPrecio(double porcentaje) {
        this.precio += this.precio * (porcentaje / 100);
    }

    // Método para actualizar precio con cantidad fija
    public void actualizarPrecio(int cantidadFija) {
        this.precio += cantidadFija;
    }

    @Override
    public String toString() {
        return "Producto [id=" + id + ", nombre=" + nombre + ", categoria=" +
categoria + ", precio=" + precio + ", cantidad=" + cantidad + "]";
    }

    // Método estático para mostrar el total de productos
    public static int mostrarTotalProductos() {
        return totalProductos;
    }
```

```
}
```

Ejemplo de uso:

```
public class TestProducto {
    public static void main(String[] args) {
        Producto prod1 = new Producto(1, "Laptop", "Electrónica", 1500.0, 10);
        Producto prod2 = new Producto("Smartphone", "Electrónica");
        prod1.actualizarPrecio(5);
        prod2.actualizarPrecio(100);
        System.out.println(prod1);
        System.out.println(prod2);
        System.out.println("Total de productos: " +
Producto.mostrarTotalProductos());
    }
}
```