

Trabajo Práctico 3 - Introducción a la POO

Alumno: Bruno Pighin

Materia: Programación 2

Tecnicatura Universitaria en Programación a Distancia

Año: 2025

Introducción

En este trabajo práctico se abordan los conceptos básicos de la Programación Orientada a Objetos en Java. A partir de distintas consignas, se desarrollaron clases que representan objetos de la vida real (estudiantes, mascotas, libros, gallinas y naves espaciales), implementando atributos, métodos, encapsulamiento y validaciones.

1) Clase Estudiante

La clase Estudiante contiene los atributos nombre, apellido, curso y calificación. Incluye métodos para mostrar la información y modificar la calificación con validaciones (límite inferior 0 y superior 10).

Estudiante.java

```
package tp3poo;  
  
public class Estudiante {  
    private String nombre;  
    private String apellido;  
    private String curso;  
    private double calificacion;  
  
    public Estudiante(String nombre, String apellido, String curso, double  
calificacion) {  
        this.nombre = nombre;  
        this.apellido = apellido;
```

```

        this.curso = curso;
        this.calificacion = calificacion;
    }

    public void mostrarInfo() {
        System.out.println("Estudiante: " + nombre + " " + apellido +
            " | Curso: " + curso + " | Calificación: " + calificacion);
    }

    public void subirCalificacion(double puntos) {
        if (puntos < 0) return;
        calificacion += puntos;
        if (calificacion > 10) calificacion = 10;
    }

    public void bajarCalificacion(double puntos) {
        if (puntos < 0) return;
        calificacion -= puntos;
        if (calificacion < 0) calificacion = 0;
    }
}

```

2) Clase Mascota

La clase Mascota representa a una mascota con nombre, especie y edad. Se implementa un método cumplirAnios() que incrementa la edad en uno.

Mascota.java

```

package tp3poo;

public class Mascota {
    private String nombre;
    private String especie;

```

```
private int edad;

public Mascota(String nombre, String especie, int edad) {
    this.nombre = nombre;
    this.especie = especie;
    this.edad = edad;
}

public void mostrarInfo() {
    System.out.println("Mascota: " + nombre + " | Especie: " + especie + " |
Edad: " + edad);
}

public void cumplirAnios() {
    this.edad++;
}
}
```

3) Clase Libro (Encapsulamiento)

La clase Libro aplica el principio de encapsulamiento: los atributos son privados y se accede mediante getters y setters.

Libro.java

```
package tp3poo;

public class Libro {
    private String titulo;
    private String autor;
    private int añoPublicacion;

    public Libro(String titulo, String autor, int añoPublicacion) {
        this.titulo = titulo;
    }
}
```

```

        this.autor = autor;
        setAñoPublicacion(añoPublicacion);
    }

    public String getTitulo() { return titulo; }
    public String getAutor() { return autor; }
    public int getAñoPublicacion() { return añoPublicacion; }

    public void setAñoPublicacion(int año) {
        if (año >= 1450 && año <= 2100) {
            this.añoPublicacion = año;
        } else {
            System.out.println("Año inválido: " + año);
        }
    }

    public void mostrarInfo() {
        System.out.println("Libro: \"\" + titulo + "\" de \" + autor + \" (" +
        añoPublicacion + ")");
    }
}

```

4) Clase Gallina

La clase Gallina permite simular una granja digital. Se registran el ID de la gallina, su edad y la cantidad de huevos puestos. Incluye métodos para envejecer y poner huevos.

Gallina.java

```

package tp3poo;

public class Gallina {

```

```
private int idGallina;
private int edad;
private int huevosPuestos;

public Gallina(int idGallina, int edadInicial) {
    this.idGallina = idGallina;
    this.edad = Math.max(0, edadInicial);
    this.huevosPuestos = 0;
}

public void ponerHuevo() { huevosPuestos++; }
public void envejecer() { edad++; }

public void mostrarEstado() {
    System.out.println("Gallina #" + idGallina + " / Edad: " + edad +
        " semanas / Huevos puestos: " + huevosPuestos);
}
}
```

5) Clase NaveEspacial

La clase NaveEspacial maneja un nombre, combustible actual y combustible máximo. Incluye métodos para despegar, avanzar consumiendo combustible, recargar sin superar el máximo y mostrar el estado.

NaveEspacial.java

```
package tp3poo;

public class NaveEspacial {
    private String nombre;
    private int combustible;
    private final int MAX_COMBUSTIBLE;
```

```
public NaveEspacial(String nombre, int combustibleInicial, int
maxCombustible) {
    this.nombre = nombre;
    this.MAX_COMBUSTIBLE = Math.max(1, maxCombustible);
    this.combustible = Math.min(Math.max(0, combustibleInicial),
MAX_COMBUSTIBLE);
}

public void despegar() {
    System.out.println(nombre + " despegó.");
}

public void avanzar(int distancia) {
    if (distancia <= 0) return;
    if (combustible < distancia) {
        System.out.println("Combustible insuficiente.");
        return;
    }
    combustible -= distancia;
    System.out.println(nombre + " avanzó " + distancia + " unidades.
Combustible restante: " + combustible);
}

public void recargarCombustible(int cantidad) {
    if (cantidad <= 0) return;
    combustible += cantidad;
    if (combustible > MAX_COMBUSTIBLE) combustible =
MAX_COMBUSTIBLE;
}

public void mostrarEstado() {
    System.out.println("Nave: " + nombre + " | Combustible: " +
combustible + "/" + MAX_COMBUSTIBLE);
}
}
```

MainTP3.java

```
package tp3poo;

public class MainTP3 {
    public static void main(String[] args) {
        Estudiante e1 = new Estudiante("Bruno", "Pighin", "Programación II",
7.5);
        e1.mostrarInfo();
        e1.subirCalificacion(1.0);
        e1.bajarCalificacion(3.0);

        Mascota m1 = new Mascota("Toby", "Perro", 3);
        m1.mostrarInfo();
        m1.cumplirAnios();

        Libro l1 = new Libro("El Principito", "Antoine de Saint-Exupéry",
1943);
        l1.mostrarInfo();
        l1.setAñoPublicacion(1200);
        l1.setAñoPublicacion(2000);

        Gallina g1 = new Gallina(1, 20);
        g1.envejecer();
        g1.ponerHuevo();
        g1.mostrarEstado();

        NaveEspacial falcon = new NaveEspacial("Falcon", 50, 100);
        falcon.mostrarEstado();
        falcon.avanzar(60);
        falcon.recargarCombustible(40);
        falcon.despegar();
        falcon.avanzar(60);
        falcon.mostrarEstado();
    }
}
```
