

Trabajo Práctico 3

Alumna: Diana Falla

Github: <https://github.com/Diana-Falla/Programacion2-TP3>

1. Registro de Estudiantes:

```
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("Nombre: " + nombre + " " + apellido);  
        System.out.println("Curso: " + curso);  
        System.out.println("Calificación: " + calificacion);  
        System.out.println("-----");  
    }  
  
    public void subirCalificacion(double puntos) {  
        calificacion += puntos;  
        System.out.println("La calificación subió " + puntos + " puntos. Nueva calificación: " + calificacion);  
    }  
  
    public void bajarCalificacion(double puntos) {  
        calificacion -= puntos;  
        if (calificacion < 0) {  
            calificacion = 0; |  
        }  
        System.out.println("La calificación bajó " + puntos + " puntos. Nueva calificación: " + calificacion);  
    }  
}
```

Clase Principal para probar el Código :

```
public class TP3DianaFalla {  
  
    public static void main(String[] args) {  
        Estudiante estudiante1 = new Estudiante("Diana", "Falla", "Programacion", 85);  
  
        estudiante1.mostrarInfo();  
  
        estudiante1.subirCalificacion(10);  
  
        estudiante1.bajarCalificacion(20);  
  
        estudiante1.mostrarInfo();  
    }  
}
```

```
run:
Nombre: Diana Falla
Curso: Programacion
Calificacion: 85.0
-----
La calificacion subio 10.0 puntos. Nueva calificación: 95.0
La calificacion bajo 20.0 puntos. Nueva calificación: 75.0
Nombre: Diana Falla
Curso: Programacion
Calificacion: 75.0
-----
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Registro de Mascotas

```
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("Nombre: " + nombre);
        System.out.println("Especie: " + especie);
        System.out.println("Edad: " + edad + " años");
        System.out.println("-----");
    }

    public void cumplirAños() {
        edad++;
        System.out.println(nombre + " cumplio años, Ahora tiene " + edad + " años.");
    }

}
```

Clase Principal :

```
//-----Ejercicio 2-----  
// Crear una mascota  
Mascota mascota1 = new Mascota("Firulaís", "Perro", 3);  
  
mascota1.mostrarInfo();  
  
mascota1.cumplirAnios();  
mascota1.cumplirAnios();  
  
mascota1.mostrarInfo();
```

Resultado consola:

```
Nombre: Firulaís  
Especie: Perro  
Edad: 3 años  
-----  
Firulaís cumplió a 3 años, Ahora tiene 4 años.  
Firulaís cumplió a 4 años, Ahora tiene 5 años.  
Nombre: Firulaís  
Especie: Perro  
Edad: 5 años  
-----  
BUILD SUCCESSFUL (total time: 0 seconds)
```

3. Encapsulamiento con la Clase Libro:

```
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 añoPublicacion) {
        if (añoPublicacion > 0 && añoPublicacion <= 2025) {
            this.añoPublicacion = añoPublicacion;
        } else {
            System.out.println("Año invalido: " + añoPublicacion + ". No se aplico el cambio.");
        }
    }

    public void mostrarInfo() {
        System.out.println("Titulo: " + titulo);
        System.out.println("Autor: " + autor);
        System.out.println("Año de publicacion: " + añoPublicacion);
        System.out.println("-----");
    }
}
```

Clase Principal:

```
//-----Ejercicio 3-----

    Libro libro1 = new Libro("Cien años de soledad", "Gabriel García Marquez", 1967);

    libro1.mostrarInfo();

    libro1.setAñoPublicacion(3025);

    libro1.setAñoPublicacion(1982);

    libro1.mostrarInfo();
```

Resultado Consola:

```
-----  
Titulo: Cien anios de soledad  
Autor: Gabriel Garcia Marquez  
Anio de publicacion: 1967  
-----  
Anio invalido: 3025. No se aplico el cambio.  
Titulo: Cien anios de soledad  
Autor: Gabriel Garcia Marquez  
Anio de publicacion: 1982  
-----  
BUILD SUCCESSFUL (total time: 0 seconds)
```

4. Gestión de Gallinas en Granja Digital

```
public class Gallina {  
    private int idGallina;  
    private int edad;  
    private int huevosPuestos;  
  
    public Gallina(int idGallina, int edadInicial) {  
        this.idGallina = idGallina;  
        this.edad = edadInicial;  
        this.huevosPuestos = 0;  
    }  
  
    public void ponerHuevo() {  
        huevosPuestos++;  
        System.out.println("Gallina " + idGallina + " puso un huevo. Total: " + huevosPuestos);  
    }  
  
    public void envejecer() {  
        edad++;  
        System.out.println("Gallina " + idGallina + " cumplio un anio. Edad actual: " + edad);  
    }  
  
    public void mostrarEstado() {  
        System.out.println("Gallina " + idGallina + ": Edad = " + edad + " anios, Huevos puestos = " + huevosPuestos);  
        System.out.println("-----");  
    }  
}
```

Clase Principal:

```
//-----Ejercicio 4-----  
  
    // Creamos dos gallinas  
    Gallina g1 = new Gallina(1, 2);  
    Gallina g2 = new Gallina(2, 1);  
  
    g1.ponerHuevo();  
    g1.ponerHuevo();  
    g1.envejecer();  
  
    g2.ponerHuevo();  
    g2.envejecer();  
    g2.ponerHuevo();  
  
    g1.mostrarEstado();  
    g2.mostrarEstado();  
}  
}
```

Resultado Consola:

```
-----  
Gallina 1 puso un huevo. Total: 1  
Gallina 1 puso un huevo. Total: 2  
Gallina 1 cumplio un anio. Edad actual: 3  
Gallina 2 puso un huevo. Total: 1  
Gallina 2 cumplio un anio. Edad actual: 2  
Gallina 2 puso un huevo. Total: 2  
Gallina 1: Edad = 3 anios, Huevos puestos = 2  
-----  
Gallina 2: Edad = 2 anios, Huevos puestos = 2  
-----  
BUILD SUCCESSFUL (total time: 0 seconds)
```

4. Simulación de Nave Espacial

```
public class NaveEspacial {
    private String nombre;
    private int combustible;
    private final int CAPACIDAD_MAX = 100;

    public NaveEspacial(String nombre, int combustibleInicial) {
        this.nombre = nombre;
        if (combustibleInicial > CAPACIDAD_MAX) {
            this.combustible = CAPACIDAD_MAX;
        } else {
            this.combustible = combustibleInicial;
        }
    }

    public void despegar() {
        if (combustible >= 10) {
            combustible -= 10;
            System.out.println(nombre + " despegó. Combustible restante: " + combustible);
        } else {
            System.out.println("No hay suficiente combustible para despegar.");
        }
    }

    public void avanzar(int distancia) {
        int consumo = distancia * 2;
        if (combustible >= consumo) {
            combustible -= consumo;
            System.out.println(nombre + " avanzó " + distancia + " km. Combustible restante: " + combustible);
        } else {
            System.out.println("Combustible insuficiente para avanzar " + distancia + " km.");
        }
    }

    public void recargarCombustible(int cantidad) {
        if (combustible + cantidad > CAPACIDAD_MAX) {
            combustible = CAPACIDAD_MAX;
            System.out.println("Se recargó al máximo. Combustible actual: " + combustible);
        } else {
            combustible += cantidad;
            System.out.println("Se recargó " + cantidad + " unidades. Combustible actual: " + combustible);
        }
    }

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

Clase Principal:

```
//-----Ejercicio 5-----  
// Creamos una nave con 50 unidades de combustible  
NaveEspacial navel = new NaveEspacial("Apollo", 50);  
  
navel.despegar();  
navel.avanzar(30);  
  
// Recargar combustible  
navel.recargarCombustible(40);  
  
// Avanzar correctamente  
navel.avanzar(20);  
  
// Mostrar estado final  
navel.mostrarEstado();  
}  
}
```

Resultado Consola:

```
-----  
Apollo despegó. Combustible restante: 40  
Combustible insuficiente para avanzar 30 km.  
Se recargó 40 unidades. Combustible actual: 80  
Apollo avanzó 20 km. Combustible restante: 40  
Nave: Apollo  
Combustible: 40/100  
-----  
BUILD SUCCESSFUL (total time: 0 seconds)
```