TP3: Introducción a la POO

Caso Práctico

https://github.com/MatiasBoyer/tupad_progra2

Desarrollar en Java los siguientes ejercicios aplicando los conceptos de programación orientada a objetos.

Registro de Estudiantes

a. Crear una clase Estudiante con los atributos: nombre, apellido, curso,calificación.
Métodos requeridos: mostrarInfo(), subirCalificacion(puntos), bajarCalificacion(puntos).
Tarea: Instanciar a un estudiante, mostrar su información, aumentar y disminuir calificaciones.

```java
// Estudiante.java
package com.matiasb.javatp3;

public class Estudiante {

    public String nombre;
    public String apellido;
    public String curso;
    private double calificacion;


    public void mostrarInfo()
    {
        System.out.println("-- Estudiante --");
        System.out.println("Nombre: " + this.nombre);
        System.out.println("Apellido: " + this.apellido);
        System.out.println("Curso: " + this.curso);
        System.out.println("Calificacion: " + this.calificacion);
    }

    public void subirCalificacion(double puntos)
    {
        calificacion += puntos;

        if(calificacion < 0) calificacion = 0;
    }

    public void bajarCalificacion(double puntos)
    {
        calificacion -= puntos;

        if(calificacion > 10) calificacion = 10;
    }
}
```

```java
public static void EJ1()
{
    Estudiante estudiante = new Estudiante();

    estudiante.nombre = "Matias";
    estudiante.apellido = "Boyer";
    estudiante.curso = "Progra II";
    estudiante.mostrarInfo();

    estudiante.subirCalificacion(10);
    estudiante.mostrarInfo();

    estudiante.bajarCalificacion(5);
    estudiante.mostrarInfo();
}
```

Boyer Rodríguez Matías Ruben
matiasboyer7@gmail.com
DNI 43.901.799

```
--- exec:3.1.0:exec (default-cli) @ javatp3 ---
1
Ejecutando el ejercicio 1
-- Estudiante --
Nombre: Matias
Apellido: Boyer
Curso: Progra II
Calificacion: 0.0
-- Estudiante --
Nombre: Matias
Apellido: Boyer
Curso: Progra II
Calificacion: 10.0
-- Estudiante --
Nombre: Matias
Apellido: Boyer
Curso: Progra II
Calificacion: 5.0
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  16.702 s
Finished at: 2025-09-08T20:29:10-03:00
------------------------------------------------------------------------
```

Boyer Rodríguez Matías Ruben
matiasboyer7@gmail.com
DNI 43.901.799

Registro de Mascotas

a. Crear una clase Mascota con los atributos: nombre, especie, edad.
Métodos requeridos: mostrarInfo(), cumplirAnios().
Tarea: Crear una mascota, mostrar su información, simular el paso del tiempo y verificar los cambios.

```java
package com.matiasb.javatp3;

public class Mascota {

    public String nombre;
    public String especie;
    private int edad;


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

    public void cumplirAnios()
    {
        edad += 1;
    }
}
```

```java
    public static void EJ2()
    {
        Mascota mascota = new Mascota();
        mascota.nombre = "Felipe";
        mascota.especie = "Perro";
        mascota.mostrarInfo();

        for(int i = 0; i < 5; i++)
        {
            mascota.cumplirAnios();
            mascota.mostrarInfo();
        }
    }
```

Boyer Rodríguez Matías Ruben
matiasboyer7@gmail.com
DNI 43.901.799

```
--- exec:3.1.0:exec (default-cli) @ javatp3 ---
2
Ejecutando el ejercicio 2
-- Mascota --
Nombre: Felipe
Especie: Perro
Edad: 0
-- Mascota --
Nombre: Felipe
Especie: Perro
Edad: 1
-- Mascota --
Nombre: Felipe
Especie: Perro
Edad: 2
-- Mascota --
Nombre: Felipe
Especie: Perro
Edad: 3
-- Mascota --
Nombre: Felipe
Especie: Perro
Edad: 4
-- Mascota --
Nombre: Felipe
Especie: Perro
Edad: 5
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  5.595 s
Finished at: 2025-09-08T20:43:36-03:00
------------------------------------------------------------------------
```

Boyer Rodríguez Matías Ruben
matiasboyer7@gmail.com
DNI 43.901.799

Encapsulamiento con la Clase Libro

a. Crear una clase Libro con atributos privados: titulo, autor, añoPublicacion.
Métodos requeridos: Getters para todos los atributos. Setter con validación para añoPublicacion.
Tarea: Crear un libro, intentar modificar el año con un valor inválido y luego con uno válido, mostrar la información final.

```java
package com.matiasb.javatp3;

public class Libro {

    private String titulo;
    private String autor;
    private int añoPublicacion;


    public void mostrarInfo()
    {
        System.out.println("-- Libro --");
        System.out.println("Titulo: " + this.titulo);
        System.out.println("Autor: " + this.autor);
        System.out.println("Año publicación: " + this.añoPublicacion);
    }

    public String getTitulo() {
        return titulo;
    }

    public void setTitulo(String titulo) {
        this.titulo = titulo;
    }

    public String getAutor() {
        return autor;
    }

    public void setAutor(String autor) {
        this.autor = autor;
    }

    public int getAñoPublicacion() {
        return añoPublicacion;
    }

    public void setAñoPublicacion(int añoPublicacion) {
        if(añoPublicacion < 1500 || 2026 < añoPublicacion)
        {
            System.out.println("'" + añoPublicacion + "' no está en el
rango permitido");
            return;
        }
        this.añoPublicacion = añoPublicacion;
    }
}
```

Boyer Rodríguez Matías Ruben
matiasboyer7@gmail.com
DNI 43.901.799

```java
    public static void EJ3()
    {
        Libro libro = new Libro();

        // datos correctos
        libro.setTitulo("El eternauta");
        libro.setAutor("Hector German Oesterheld");

        // tarea - intentar modificar el año con un valor inválido
        libro.setAñoPublicacion(2027);
        libro.setAñoPublicacion(1000);

        // tarea - luego con uno válido
        libro.setAñoPublicacion(1957);

        libro.mostrarInfo();
    }
```

```
--- exec:3.1.0:exec (default-cli) @ javatp3 ---
3
Ejecutando el ejercicio 3
'2027' no est en el rango permitido
'1000' no est en el rango permitido
-- Libro --
Titulo: El eternauta
Autor: Hector German Oesterheld
Ao publicacin: 1957
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  2.496 s
Finished at: 2025-09-08T20:53:59-03:00
------------------------------------------------------------------------
```

Boyer Rodríguez Matías Ruben
matiasboyer7@gmail.com
DNI 43.901.799

Gestión de Gallinas en Granja Digital

a. Crear una clase Gallina con los atributos: idGallina, edad, huevosPuestos.
Métodos requeridos: ponerHuevo(), envejecer(), mostrarEstado().
Tarea: Crear dos gallinas, simular sus acciones (envejecer y poner huevos), y mostrar su estado.

```java
package com.matiasb.javatp3;

public class Gallina {

    public int idGallina;
    private int edad;
    private int huevosPuestos;

    public void mostrarEstado()
    {
        System.out.println("-- Gallina "+idGallina+" --");
        System.out.println("Edad: " + this.edad);
        System.out.println("Huevos puestos: " + this.huevosPuestos);
    }

    public void ponerHuevo()
    {
        this.huevosPuestos += 1;
    }

    public void envejecer()
    {
        this.edad += 1;
    }

}
```

```java
    public static void EJ4()
    {
        Gallina pedro = new Gallina();
        pedro.idGallina = 1;

        Gallina enrique = new Gallina();
        enrique.idGallina = 2;

        pedro.envejecer();
        pedro.ponerHuevo();
        pedro.ponerHuevo();
        pedro.ponerHuevo();

        enrique.envejecer();
        enrique.envejecer();
        enrique.envejecer();
        enrique.ponerHuevo();

        pedro.mostrarEstado();
        enrique.mostrarEstado();
    }
```

Boyer Rodríguez Matías Ruben
matiasboyer7@gmail.com
DNI 43.901.799

```
--- exec:3.1.0:exec (default-cli) @ javatp3 ---
4
Ejecutando el ejercicio 4
-- Gallina 1 --
Edad: 1
Huevos puestos: 3
-- Gallina 2 --
Edad: 3
Huevos puestos: 1
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  2.617 s
Finished at: 2025-09-08T20:58:44-03:00
------------------------------------------------------------------------
```

Boyer Rodríguez Matías Ruben
matiasboyer7@gmail.com
DNI 43.901.799

Simulación de Nave Espacial

Crear una clase NaveEspacial con los atributos: nombre, combustible.
Métodos requeridos: despegar(), avanzar(distancia), recargarCombustible(cantidad), mostrarEstado().
Reglas: Validar que haya suficiente combustible antes de avanzar y evitar que se supere el límite al recargar.
Tarea: Crear una nave con 50 unidades de combustible, intentar avanzar sin recargar, luego recargar y avanzar correctamente. Mostrar el estado al final

```java
package com.matiasb.javatp3;

public class NaveEspacial {

    public String nombre;
    private int combustible = 50;
    private boolean despegado = false;

    public void mostrarEstado()
    {
        System.out.println("-- Nave Espacial --");
        System.out.println("Nombre: " + this.nombre);
        System.out.println("Combustible: " + this.combustible);
    }

    public void despegar()
    {
        if(this.combustible <= 50)
        {
            System.out.println("No hay combustible suficiente.");
            return;
        }

        this.combustible -= 50;
        this.despegado = true;
        System.out.println("Despegue! -50 de combustible por despegar.");
    }

    public void avanzar(int distancia)
    {
        if(!this.despegado)
        {
            System.out.println("La nave todavía no despegó");
            return;
        }

        int gasto_combustible = distancia * 2;
        if(gasto_combustible > this.combustible)
        {
            System.out.println("No hay combustible suficiente");
            return;
        }

        this.combustible -= gasto_combustible;
        System.out.println("Gaste "+gasto_combustible+" avanzando
"+distancia+" unidades.");

    }
```

Boyer Rodríguez Matías Ruben
matiasboyer7@gmail.com
DNI 43.901.799

```java
    public static void EJ5()
    {
        // creo una nave, tiene combustible 50
        NaveEspacial saturn_v = new NaveEspacial();
        saturn_v.setNombre("Saturn V");

        // intento avanzar sin despegar
        saturn_v.avanzar(1);

        // recargo 55 de combustible, supera el limite
        saturn_v.recargarCombustible(55);

        // recargo 50 de combustible, llega a 100
        saturn_v.recargarCombustible(50);

        // despego, nos gasta 50 de combustible
        saturn_v.despegar();

        // intento avanzar 26, supera el gasto de combustible
        saturn_v.avanzar(26);

        // avanzo 25, gasta 50 de combustible
        saturn_v.avanzar(25);

        // muestro el estado. debería decir 0 combustible
        saturn_v.mostrarEstado();
    }
```

```
--- exec:3.1.0:exec (default-cli) @ javatp3 ---
5
Ejecutando el ejercicio 5
La nave todav□a no despeg□
La carga de combustible supera el m□ximo
Recargo 50. El total ahora es 100
Despegue! -50 de combustible por despegar.
No hay combustible suficiente
Gaste 50 avanzando 25 unidades.
-- Nave Espacial --
Nombre: Saturn V
Combustible: 0
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  2.256 s
Finished at: 2025-09-08T21:17:59-03:00
------------------------------------------------------------------------
```

Boyer Rodríguez Matías Ruben
matiasboyer7@gmail.com
DNI 43.901.799