

PROGRAMACIÓN II

Trabajo Práctico 3: Introducción a la Programación Orientada a Objetos

OBJETIVO GENERAL

Comprender los fundamentos de la Programación Orientada a Objetos, incluyendo clases, objetos, atributos y métodos, para estructurar programas de manera modular y reutilizable en Java.

MARCO TEÓRICO

Concepto	Aplicación en el proyecto
Clases y Objetos	Modelado de entidades como Estudiante, Mascota, Libro, Gallina y NaveEspacial
Atributos y Métodos	Definición de propiedades y comportamientos para cada clase
Estado e Identidad	Cada objeto conserva su propio estado (edad, calificación, combustible, etc.)
Encapsulamiento	Uso de modificadores de acceso y getters/setters para proteger datos
Modificadores de acceso	Uso de private, public y protected para controlar visibilidad
Getters y Setters	Acceso controlado a atributos privados mediante métodos
Reutilización de código	Definición de clases reutilizables en múltiples contextos

Caso Práctico

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

1. Registro de Estudiantes

- 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.

2. Registro de Mascotas

- 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.

3. Encapsulamiento con la Clase Libro

- 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.

4. Gestión de Gallinas en Granja Digital

- 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.

5. 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.

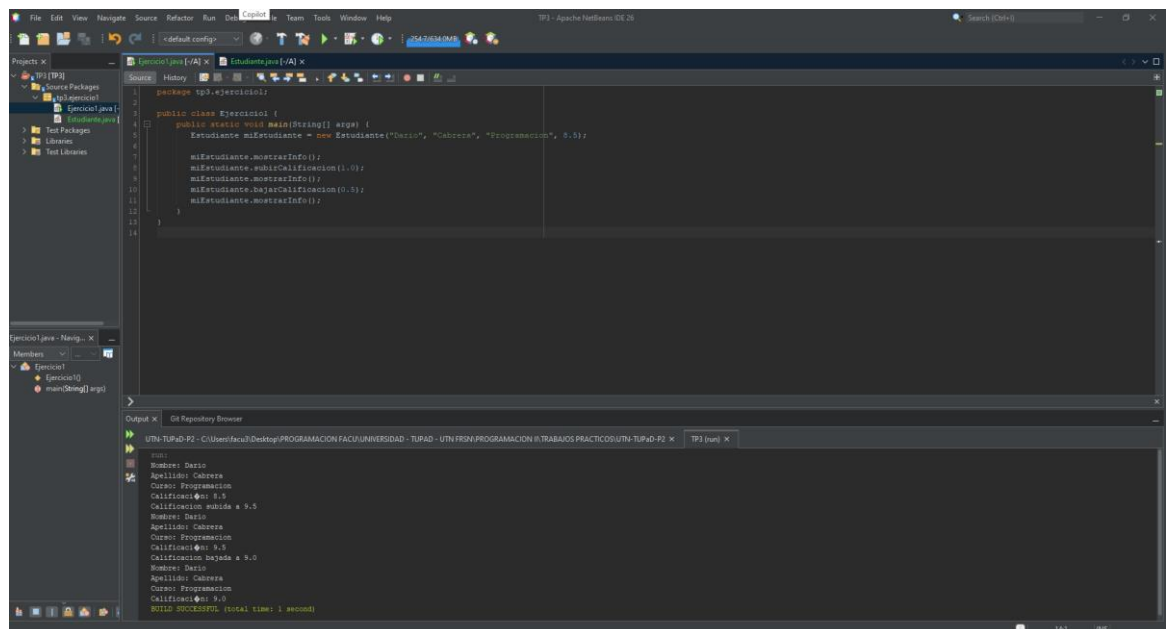
Tarea: Crear una nave con 50 unidades de combustible, intentar avanzar sin recargar, luego recargar y avanzar correctamente. Mostrar el estado al final.

CONCLUSIONES ESPERADAS

- Comprender la diferencia entre clases y objetos.
- Aplicar principios de encapsulamiento para proteger los datos.
- Usar getters y setters para gestionar atributos privados.
- Implementar métodos que definen comportamientos de los objetos.
- Manejar el estado y la identidad de los objetos correctamente.
- Aplicar buenas prácticas en la estructuración del código orientado a objetos.
- Reforzar el pensamiento modular y la reutilización del código en Java.

RESOLUCION

1.



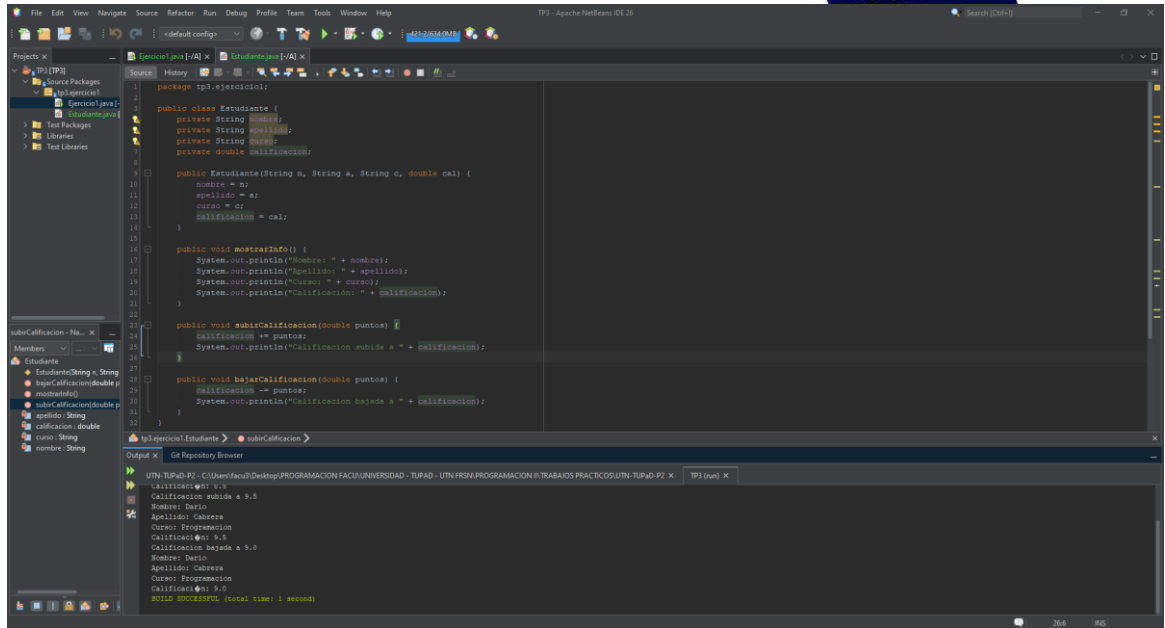
```
package tp3.ejercicio1;

public class Ejercicio1 {
    public static void main(String[] args) {
        Estudiante miEstudiante = new Estudiante("Dario", "Cabreria", "Programacion", 9.5);

        miEstudiante.mostrarInfo();
        miEstudiante.subirCalificacion(1.0);
        miEstudiante.mostrarInfo();
        miEstudiante.bajarCalificacion(0.5);
        miEstudiante.mostrarInfo();
    }
}
```

Output:

```
main:
Nombre: Dario
Apellido: Cabreria
Curso: Programacion
Calificacion: 9.5
Calificacion subida a 9.5
Nombre: Dario
Apellido: Cabreria
Curso: Programacion
Calificacion: 9.5
Calificacion bajada a 9.0
Nombre: Dario
Apellido: Cabreria
Curso: Programacion
Calificacion: 9.0
BUILD SUCCESSFUL (total time: 1 second)
```



```

package tp3.ejercicio1;

public class Estudiante {
    private String nombre;
    private String apellido;
    private String curso;
    private double calificación;

    public Estudiante(String n, String a, String c, double cal) {
        nombre = n;
        apellido = a;
        curso = c;
        calificación = cal;
    }

    public void mostrarInfo() {
        System.out.println("Nombre: " + nombre);
        System.out.println("Apellido: " + apellido);
        System.out.println("Curso: " + curso);
        System.out.println("Calificación: " + calificación);
    }

    public void subirCalificación(double puntos) {
        calificación += puntos;
        System.out.println("Calificación subida a " + calificación);
    }

    public void bajarCalificación(double puntos) {
        calificación -= puntos;
        System.out.println("Calificación bajada a " + calificación);
    }
}

```

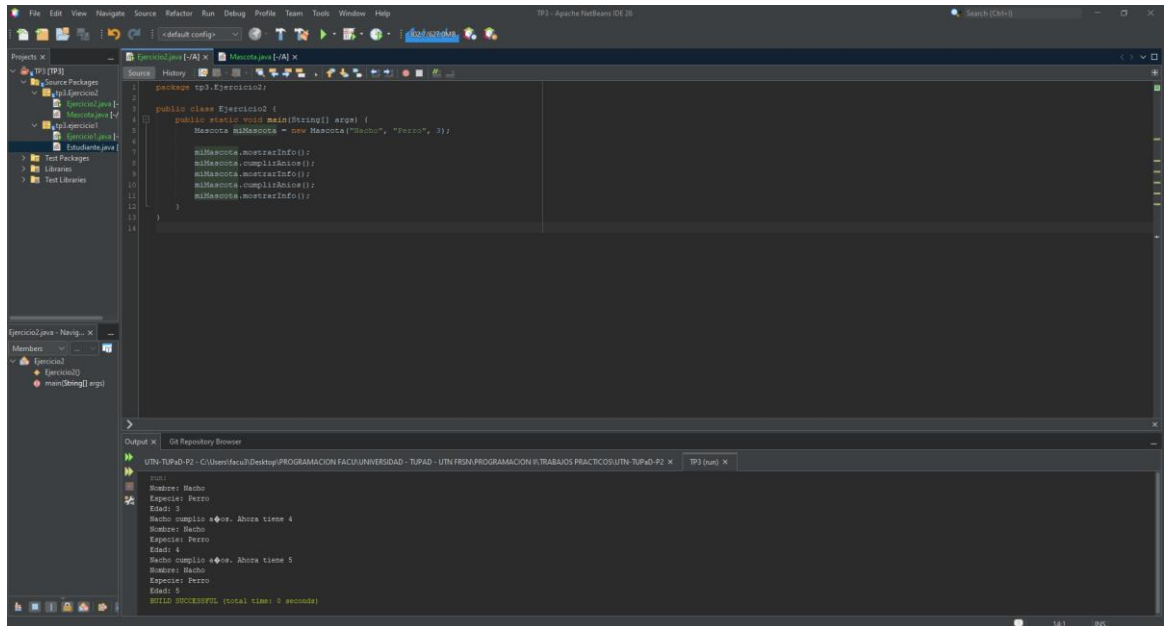
Output:

```

UTN-TUPAD-P2 - C:\Users\facu\Desktop\PROGRAMACION FACU\UNIVERSIDAD - TUPAD - UTN-FRONT\PROGRAMACION II\TRABAJO PRACTICO\UTN-TUPAD-P2 x
TP3 (run) x
Calificación: 5.5
Calificación subida a 5.5
Nombre: Darío
Apellido: Cabrera
Curso: Programación
Calificación: 5.5
Calificación bajada a 5.0
Nombre: Darío
Apellido: Cabrera
Curso: Programación
Calificación: 5.0
BUILD SUCCESSFUL (total time: 1 seconds)

```

2.



```

package tp3.ejercicio2;

public class Ejercicio2 {
    public static void main(String[] args) {
        Mascota miMascota = new Mascota("Macho", "Perro", 3);

        miMascota.mostrarInfo();
        miMascota.cumplirAños();
        miMascota.mostrarInfo();
        miMascota.cumplirAños();
        miMascota.mostrarInfo();
    }
}

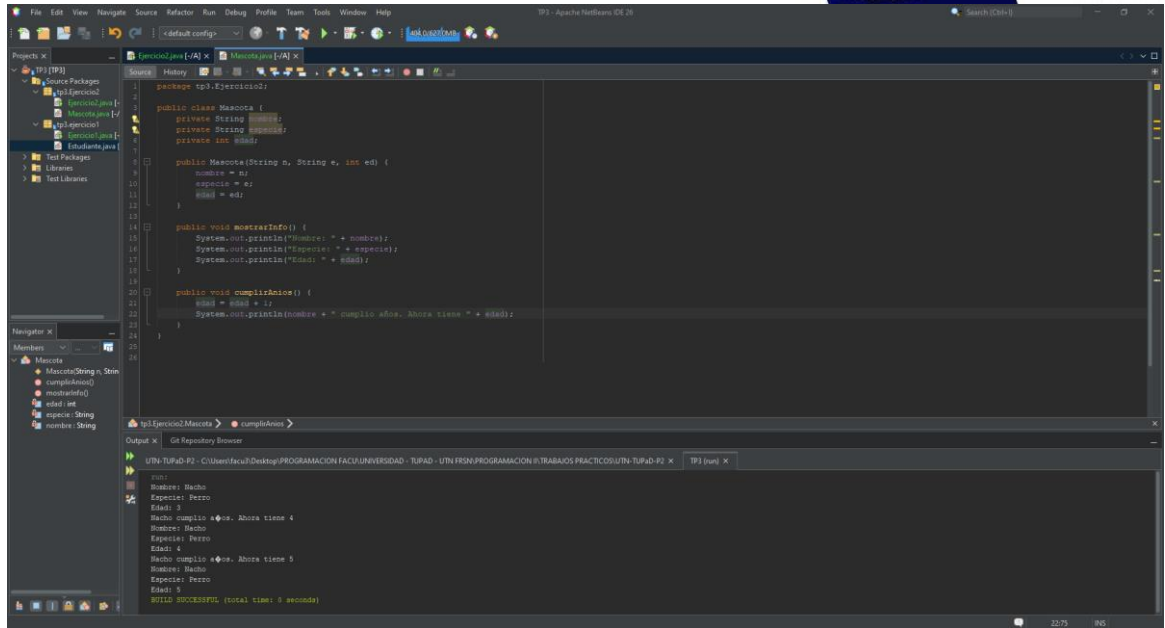
```

Output:

```

UTN-TUPAD-P2 - C:\Users\facu\Desktop\PROGRAMACION FACU\UNIVERSIDAD - TUPAD - UTN-FRONT\PROGRAMACION II\TRABAJO PRACTICO\UTN-TUPAD-P2 x
TP3 (run) x
main
Nombre: Macho
Especie: Perro
Edad: 3
Macho cumple 4 años. Ahora tiene 4
Nombre: Macho
Especie: Perro
Edad: 4
Macho cumple 5 años. Ahora tiene 5
Nombre: Macho
Especie: Perro
Edad: 5
BUILD SUCCESSFUL (total time: 0 seconds)

```



```

package tp3.Ejercicio2;

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

    public Mascota(String n, String e, int ed) {
        nombre = n;
        especie = e;
        edad = ed;
    }

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

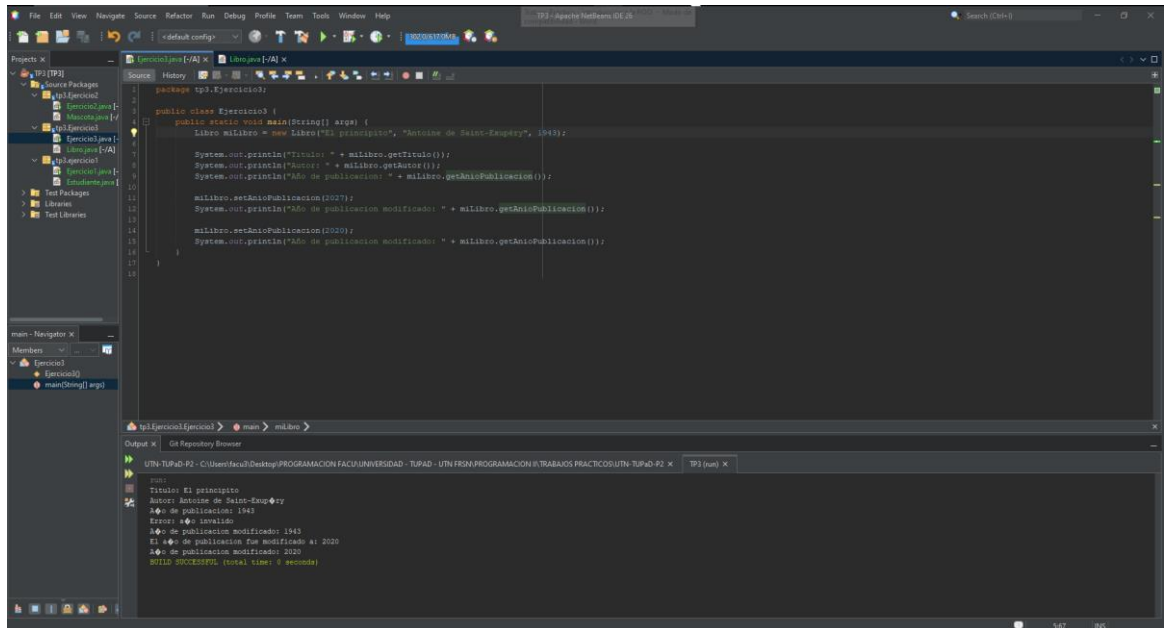
    public void cumplirAños() {
        edad = edad + 1;
        System.out.println(nombre + " cumplió años. Ahora tiene " + edad);
    }
}
    
```

Output:

```

run
Nombre: Macho
Especie: Perro
Edad: 3
Macho cumplió años. Ahora tiene 4
Nombre: Macho
Especie: Perro
Edad: 4
Macho cumplió años. Ahora tiene 5
Nombre: Macho
Especie: Perro
Edad: 5
BUILD SUCCESSFUL (total time: 0 seconds)
    
```

3.



```

package tp3.Ejercicio3;

public class Ejercicio3 {
    public static void main(String[] args) {
        Libro libro = new Libro("El principito", "Antoine de Saint-Exupéry", 1943);

        System.out.println("Titulo: " + libro.getTitulo());
        System.out.println("Autor: " + libro.getAutor());
        System.out.println("Año de publicacion: " + libro.getAñoPublicacion());

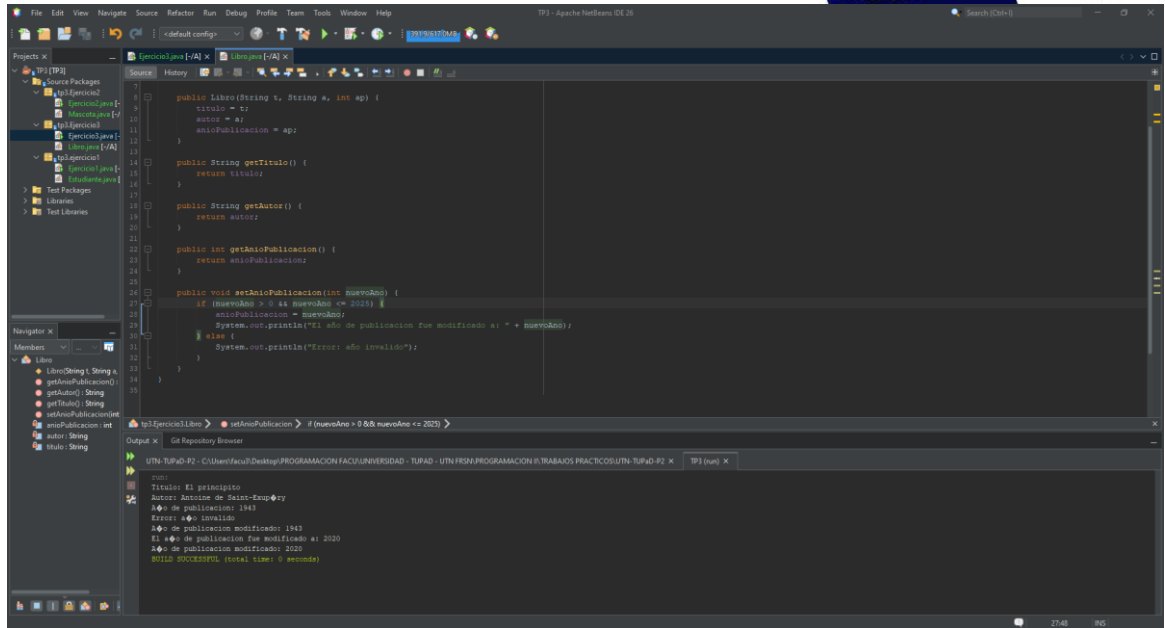
        libro.setAñoPublicacion(2027);
        System.out.println("Año de publicacion modificado: " + libro.getAñoPublicacion());

        libro.setAñoPublicacion(2020);
        System.out.println("Año de publicacion modificado: " + libro.getAñoPublicacion());
    }
}
    
```

Output:

```

main
Titulo: El principito
Autor: Antoine de Saint-Exupéry
Año de publicacion: 1943
Error: años invalido
El año de publicacion fue modificado a: 2027
El año de publicacion modificado: 2020
BUILD SUCCESSFUL (total time: 0 seconds)
    
```



```

1 public Libro(String t, String a, int ap) {
2     titulo = t;
3     autor = a;
4     anoPublicacion = ap;
5 }
6
7 public String getTitulo() {
8     return titulo;
9 }
10
11 public String getAutor() {
12     return autor;
13 }
14
15 public int getAnoPublicacion() {
16     return anoPublicacion;
17 }
18
19 public void setAnoPublicacion(int nuevoAno) {
20     if (nuevoAno > 0 && nuevoAno <= 2020) {
21         anoPublicacion = nuevoAno;
22         System.out.println("El año de publicación fue modificado a: " + nuevoAno);
23     } else {
24         System.out.println("Error: año inválido");
25     }
26 }
27
28 }
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

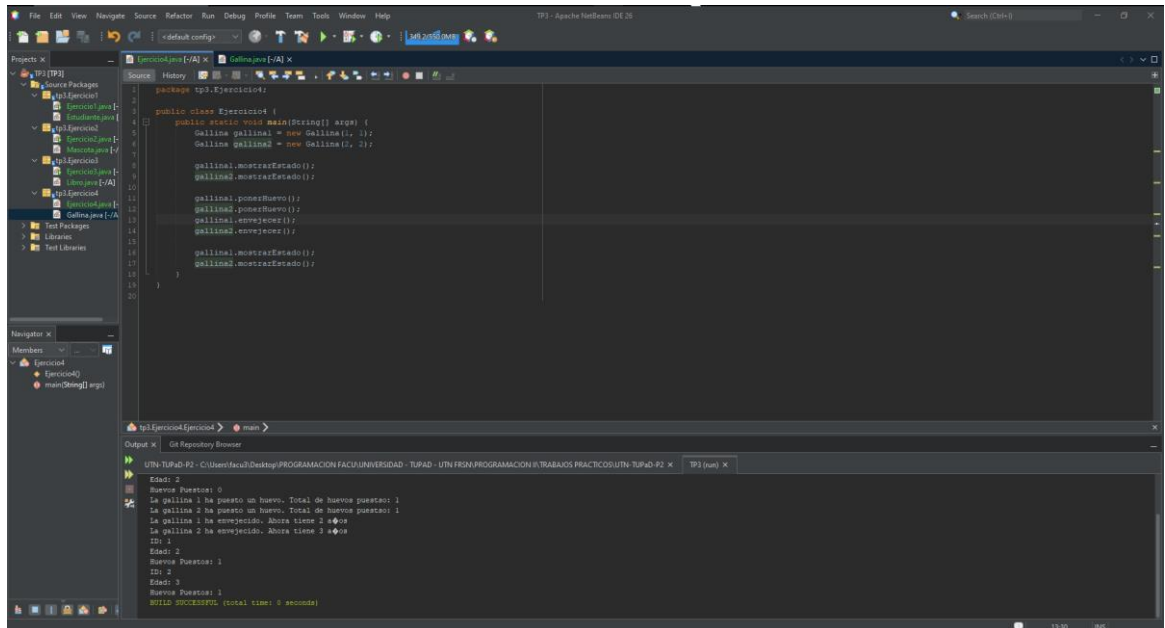
Output X | Git Repository Browser

```

UTN-TUPAD-P2 - C:\Users\facu\Desktop\PROGRAMACION FACU\UNIVERSIDAD - TUPAD - UTN-FRM\PROGRAMACION E TRABAJOS PRACTICOS\UTN-TUPAD-P2 x TP3 (run) x
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

4.



```

1 package tp3.Ejercicio04;
2
3 public class Ejercicio04 {
4     public static void main(String[] args) {
5         Gallina gallina1 = new Gallina(1, 1);
6         Gallina gallina2 = new Gallina(1, 2);
7
8         gallina1.mostrarEstado();
9         gallina2.mostrarEstado();
10
11         gallina1.ponerHuevo();
12         gallina2.ponerHuevo();
13         gallina1.envejecer();
14         gallina2.envejecer();
15
16         gallina1.mostrarEstado();
17         gallina2.mostrarEstado();
18
19     }
20 }
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

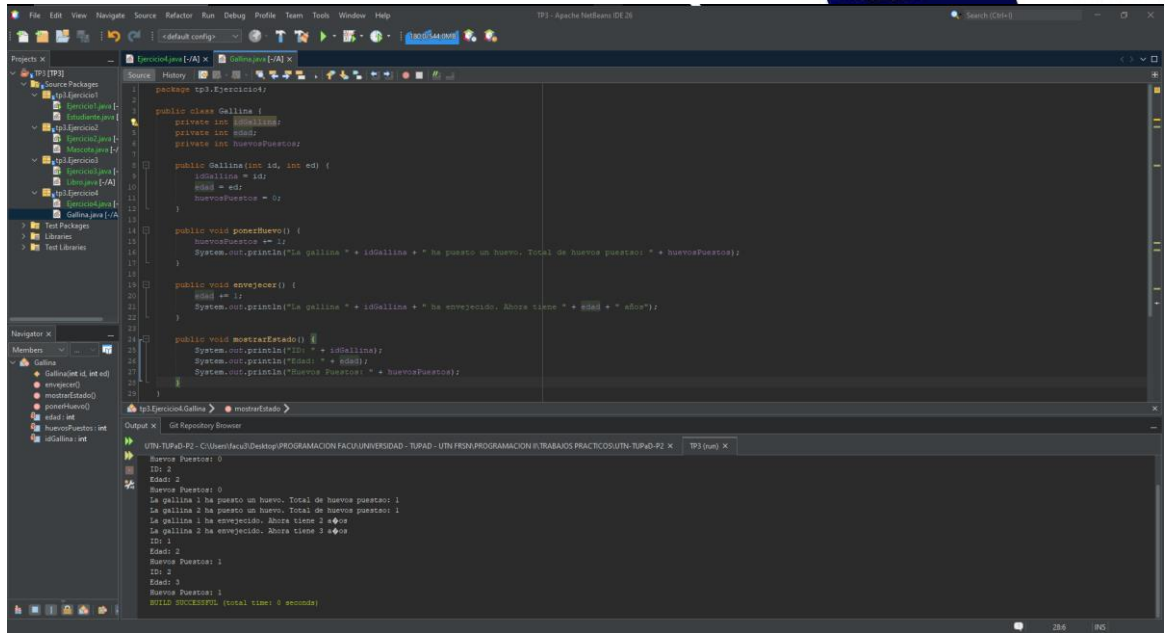
```

Output X | Git Repository Browser

```

UTN-TUPAD-P2 - C:\Users\facu\Desktop\PROGRAMACION FACU\UNIVERSIDAD - TUPAD - UTN-FRM\PROGRAMACION E TRABAJOS PRACTICOS\UTN-TUPAD-P2 x TP3 (run) x
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```



```

package tp3.Ejercicio0;

public class Gallina {
    private int idGallina;
    private int edad;
    private int huevosPuestos;

    public Gallina(int id, int ed) {
        idGallina = id;
        edad = ed;
        huevosPuestos = 0;
    }

    public void ponerHuevo() {
        huevosPuestos += 1;
        System.out.println("La gallina " + idGallina + " ha puesto un huevo. Total de huevos puestos: " + huevosPuestos);
    }

    public void envejecer() {
        edad += 1;
        System.out.println("La gallina " + idGallina + " ha envejecido. Ahora tiene " + edad + " años");
    }

    public void mostrarEstado() {
        System.out.println("ID: " + idGallina);
        System.out.println("Edad: " + edad);
        System.out.println("Huevos Puestos: " + huevosPuestos);
    }
}
    
```

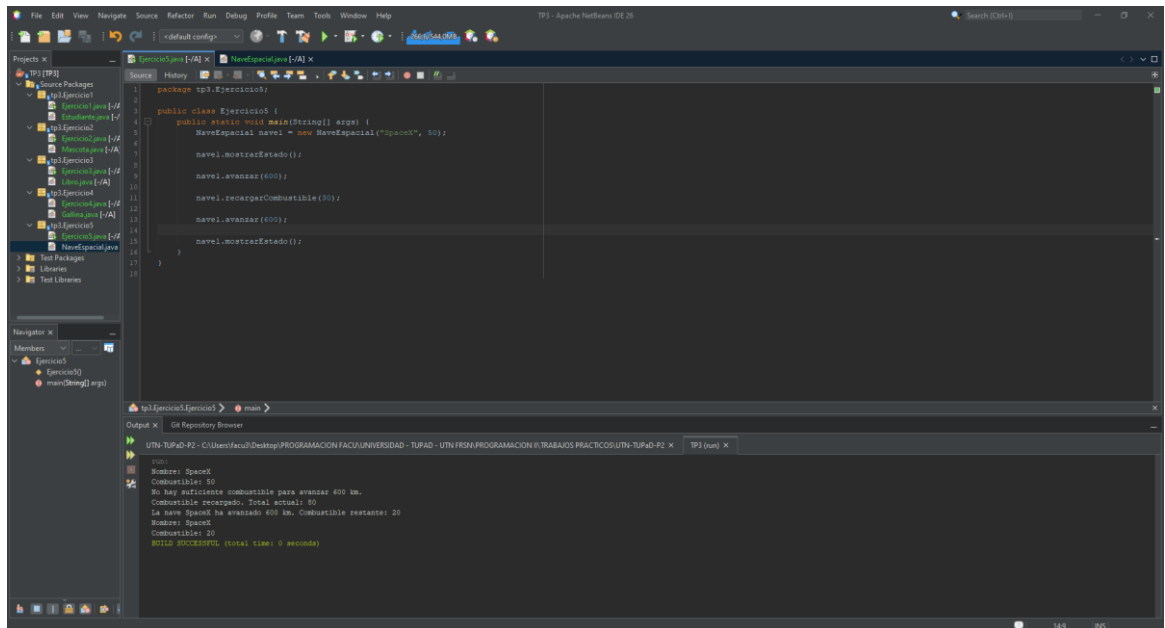
Output:

```

UTN-TUPAD-P2 - C:\Users\facu\Desktop\PROGRAMACION FACU\UNIVERSIDAD - TUPAD - UTN FRON\PROGRAMACION II\TRABAJOS PRACTICOS\UTN-TUPAD-P2 x
TP3 (run) x

Huevos Puestos: 0
ID: 1
Edad: 1
Huevos Puestos: 0
La gallina 1 ha puesto un huevo. Total de huevos puestos: 1
La gallina 1 ha puesto un huevo. Total de huevos puestos: 1
La gallina 1 ha envejecido. Ahora tiene 2 años
La gallina 2 ha envejecido. Ahora tiene 3 años
ID: 2
Huevos Puestos: 1
ID: 2
Edad: 3
Huevos Puestos: 1
BUILD SUCCESSFUL (total time: 0 seconds)
    
```

5.



```

package tp3.Ejercicio0;

public class Ejercicio0 {
    public static void main(String[] args) {
        NavEspacial nave = new NavEspacial("Spacox", 50);

        nave.mostrarEstado();

        nave.avanzar(400);

        nave.recargarCombustible(30);

        nave.avanzar(400);

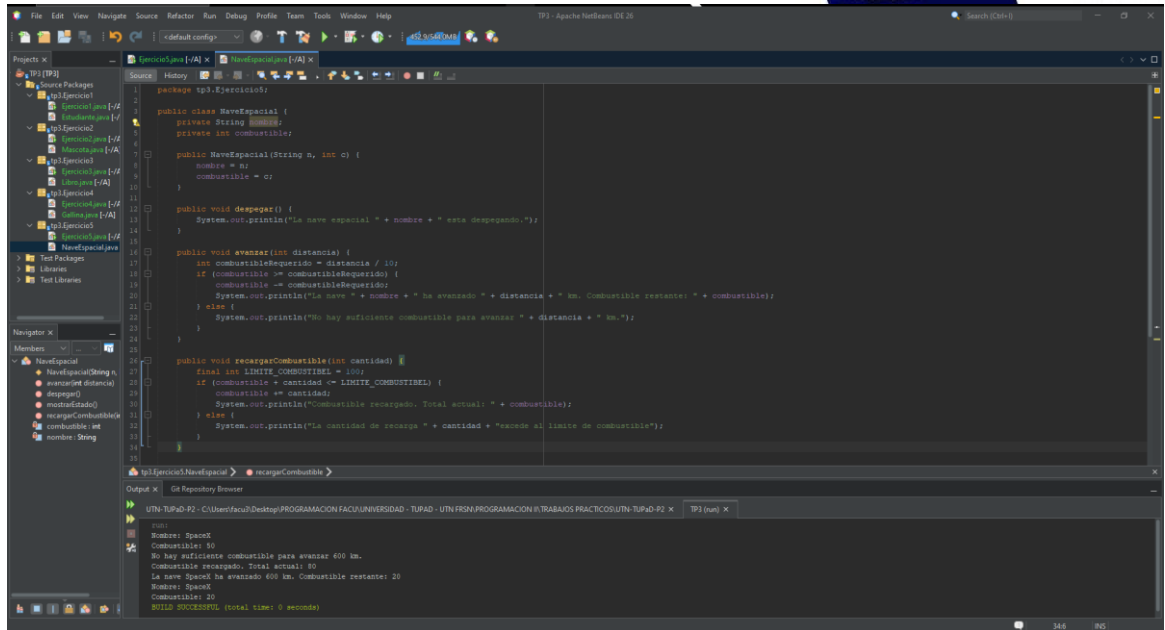
        nave.mostrarEstado();
    }
}
    
```

Output:

```

UTN-TUPAD-P2 - C:\Users\facu\Desktop\PROGRAMACION FACU\UNIVERSIDAD - TUPAD - UTN FRON\PROGRAMACION II\TRABAJOS PRACTICOS\UTN-TUPAD-P2 x
TP3 (run) x

main()
Nombre: Spacox
Combustible: 50
No hay suficiente combustible para avanzar 400 km.
Combustible recargado. Total actual: 80
La nave Spacox ha avanzado 400 km. Combustible restante: 20
Nombre: Spacox
Combustible: 20
BUILD SUCCESSFUL (total time: 0 seconds)
    
```



```
package tp3.Ejercicio0;

public class NaveEspacial {
    private String nombre;
    private int combustible;

    public NaveEspacial(String n, int c) {
        nombre = n;
        combustible = c;
    }

    public void despegar() {
        System.out.println("La nave espacial " + nombre + " se ha despegado.");
    }

    public void avanzar(int distancia) {
        int combustibleRequerido = distancia / 10;
        if (combustible >= combustibleRequerido) {
            System.out.println("La nave " + nombre + " ha avanzado " + distancia + " km. Combustible restante: " + combustible);
        } else {
            System.out.println("No hay suficiente combustible para avanzar " + distancia + " km.");
        }
    }

    public void recargarCombustible(int cantidad) {
        final int LIMITE_COMBUSTIBLE = 100;
        if (combustible + cantidad >= LIMITE_COMBUSTIBLE) {
            combustible = cantidad;
            System.out.println("Combustible recargado. Total actual: " + combustible);
        } else {
            System.out.println("La cantidad de recarga " + cantidad + " excede al límite de combustible.");
        }
    }
}
```

Output:

```
UTN-TUPAD-P2 - C:\Users\facu\Desktop\PROGRAMACION FACU\UNIVERSIDAD - TUPAD - UTN FRON\PROGRAMACION II\TRABAJO PRACTICO UTN-TUPAD-P2 x
TP3 (run) x

run
Nombre: SpaceX
Combustible: 50
No hay suficiente combustible para avanzar 600 km.
Combustible recargado. Total actual: 80
La nave SpaceX ha avanzado 600 km. Combustible restante: 20
Nombre: SpaceX
Combustible: 20
BUILD SUCCESSFUL (total time: 0 seconds)
```

Alumno: Cabrera Dario Ezequiel

DNI: 41375492

Link de repositorio: <https://github.com/Dario-Cabrera/UTN-TUPaD-P2.git>