TECNICATURA UNIVERSITARIA
EN PROGRAMACIÓN
A DISTANCIA

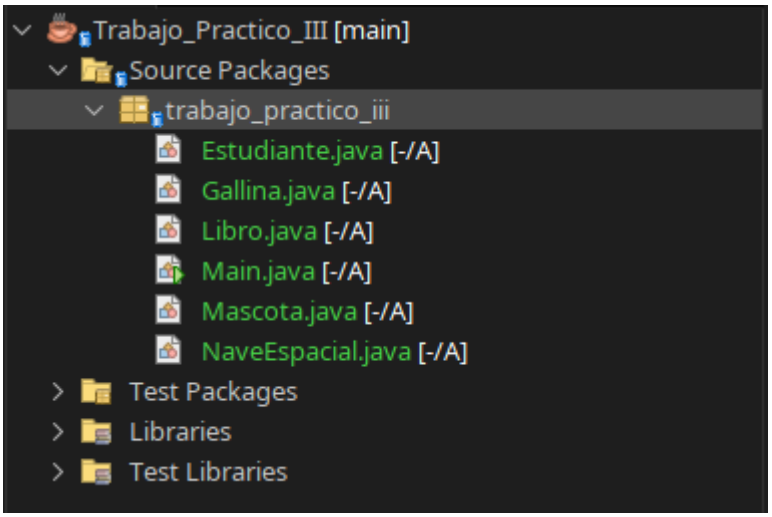
PROGRAMACIÓN II Trabajo Práctico 3: Introducción a la Programación Orientada a Objetos

Nombre: Farias, Gustavo
Comisión: M2025-13
Matrícula: 101662
Repositorio GitHub:
https://github.com/Lucenear/UTN-TUPaD-TPs/tree/main/Programacion/Programacion_II

Caso Práctico

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

Estructura del proyecto:



Codigo Main para ejecutar los casos solicitados:

```
/**
 * Nombre: Farias, Gustavo
 * Comisión: M2025-13
 * Matrícula: 101662
 */

package trabajo_practico_iii;

public class Main {
    public static void main(String[] args) {

        System.out.println("\nEjercicio 1 - Registro de estudiantes\n");
        Estudiante estudiante = new Estudiante("Gustavo", "Farias", "Programacion II", 8.5);
        estudiante.mostrarInfo();
        estudiante.subirCalificacion(1.0);
        estudiante.bajarCalificacion(0.5);
        System.out.println("Calificacion actual: " + estudiante.getCalificacion()); //uso getter
        estudiante.mostrarInfo();
        System.out.println();

        System.out.println("\nEjercicio 2 - Registro de mascotas\n");
        Mascota mascota = new Mascota("Athena", "Gato", 2);
        mascota.mostrarInfo();
        System.out.println("Edad actual: " + mascota.getEdad()); //uso getter
        mascota.cumplirAnios();
        mascota.mostrarInfo();
        System.out.println();
```

```
        System.out.println("\nEjercicio 3 - Biblioteca\n");
        Libro libro = new Libro("El jardin de los cerezos", "Anton Chejov", 1904);
        libro.mostrarInfo();
        libro.setAñoPublicacion(2030); // valor incorrecto - rechaza
        libro.setAñoPublicacion(1984); // valor ok
        System.out.println("Año actual: " + libro.getAñoPublicacion()); // uso getter
        libro.mostrarInfo();
        System.out.println();

        System.out.println("\nEjercicio 4 - El gallinero\n");
        Gallina gallina1 = new Gallina(1, 0);
        Gallina gallina2 = new Gallina(2, 2);

        gallina1.envejecer(); // Con un año puede poner huevos
        gallina1.ponerHuevo();
        gallina1.ponerHuevo();
        System.out.println("Huevos puestos por gallina " + gallina1.getIdGallina() + ": " +
gallina1.getHuevosPuestos()); // uso getter
        gallina1.mostrarEstado();

        gallina2.ponerHuevo();
        gallina2.envejecer();
        gallina2.mostrarEstado();
        System.out.println();

        System.out.println("\nEjercicio 5 - Nave Espacial\n");
        NaveEspacial nave = new NaveEspacial("Apolo XI", 50);
        nave.mostrarEstado();
        nave.avanzar(30); // Intento avanzar sin combustible suficiente
        nave.recargarCombustible(20); // Recargo
        nave.avanzar(30); // Puedo avanzar
        System.out.println("Combustible restante: " + nave.getCombustible()); // uso getter
        nave.mostrarEstado();
    }
}
```

1. 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.

```
/**
 * Nombre: Farias, Gustavo
```

```
 * Comisión: M2025-13
 * Matrícula: 101662
 */

package trabajo_practico_iii;

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("Calificacion: " + calificacion);
    }

    public void subirCalificacion(double puntos) {
        if (puntos < 0) {
            System.out.println("Error: No se puede subir una cantidad negativa de puntos");
            return;
        }
        this.calificacion += puntos;
        System.out.println("La calificacion se ha aumentado en " + puntos + " puntos");
    }

    public void bajarCalificacion(double puntos) {
        if (puntos < 0) {
            System.out.println("Eror: No se puede bajar una cantidad negativa de puntos");
            return;
        }
        if (this.calificacion - puntos < 0) {
            System.out.println("Advertencia: La calificacion no puede ser negativa. Se
establecera a 0");
            this.calificacion = 0;
        } else {
            this.calificacion -= puntos;
        }
        System.out.println("La calificacion se ha disminuido en " + puntos + " puntos");
    }

    public String getNombre() { return nombre; }
```

```
    public String getApellido() { return apellido; }
    public String getCurso() { return curso; }
    public double getCalificacion() { return calificacion; }

}
```

2. 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.

```
/**
 * Nombre: Farias, Gustavo
 * Comisión: M2025-13
 * Matrícula: 101662
 */

package trabajo_practico_iii;

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;
        if (edad < 0) {
            System.out.println("Advertencia: La edad no puede ser negativa. Se establecera en
0");
            this.edad = 0;
        } else {
            this.edad = edad;
        }
    }

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

    public void cumplirAnios() {
        this.edad++;
        System.out.println(nombre + " ha cumplido un año. Ahora tiene " + edad + " años");
```

```
    }

    public String getNombre() { return nombre; }
    public String getEspecie() { return especie; }
    public int getEdad() { return edad; }

}
```

3. 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.

```
/**
 * Nombre: Farias, Gustavo
 * Comisión: M2025-13
 * Matrícula: 101662
 */

package trabajo_practico_iii;

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) {
            System.out.println("Error: El año de publicacion debe estar entre 1 y 2025. No se realizo el cambio");
        } else {
            this.añoPublicacion = añoPublicacion;
            System.out.println("Año de publicacion actualizado correctamente a: " + añoPublicacion);
```

```
        }
    }

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

4. 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.

```
/**
 * Nombre: Farias, Gustavo
 * Comisión: M2025-13
 * Matrícula: 101662
 */

package trabajo_practico_iii;

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

    public Gallina(int idGallina, int edad) {
        this.idGallina = idGallina;
        this.edad = edad >= 0 ? edad : 0;
        this.huevosPuestos = 0;
    }

    public void ponerHuevo() {
        if (this.edad >= 1) {
            this.huevosPuestos++;
            System.out.println("La gallina " + idGallina + " puso un huevo! Total: " +
huevosPuestos);
        } else {
            System.out.println("La gallina " + idGallina + " es joven para poner huevos");
        }
    }

    public void envejecer() {
```

```
        this.edad++;
        System.out.println("La gallina " + idGallina + " ha envejecido. Ahora tiene " + edad + "
años");
    }

    public void mostrarEstado() {
        System.out.println("Estado de la gallina " + idGallina);
        System.out.println("Edad: " + edad + " años");
        System.out.println("Huevos puestos: " + huevosPuestos);
        System.out.println();
    }

    public int getIdGallina() { return idGallina; }
    public int getEdad() { return edad; }
    public int getHuevosPuestos() { return huevosPuestos; }

}
```

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.

```
/**
 * Nombre: Farias, Gustavo
 * Comisión: M2025-13
 * Matrícula: 101662
 */

package trabajo_practico_iii;

public class NaveEspacial {
    private String nombre;
    private int combustible;
    private static final int LIMITE_COMBUSTIBLE = 100;

    public NaveEspacial(String nombre, int combustibleInicial) {
        this.nombre = nombre;
        if (combustibleInicial < 0) {
            this.combustible = 0;
            System.out.println("Advertencia: El combustible inicial no puede ser negativo. Se
establece en 0");
```

```
        } else if (combustibleInicial > LIMITE_COMBUSTIBLE) {
            this.combustible = LIMITE_COMBUSTIBLE;
            System.out.println("Advertencia: El combustible supera el limite. Se establece a " +
LIMITE_COMBUSTIBLE);
        } else {
            this.combustible = combustibleInicial;
        }
    }

    public void despegar() {
        if (this.combustible > 0) {
            System.out.println(nombre + " despego exitosamente");
        } else {
            System.out.println(nombre + " no puede despegar sin combustible");
        }
    }

    public void avanzar(int distancia) {
        int consumo = distancia * 2;
        if (this.combustible >= consumo) {
            this.combustible -= consumo;
            System.out.println(nombre + " avanzo " + distancia + " unidades y consumio " +
consumo + " unidades");
        } else {
            System.out.println("Error: " + nombre + " no tiene suficiente combustible (" +
combustible + "/" + consumo + ")");
        }
    }

    public void recargarCombustible(int cantidad) {
        if (cantidad < 0) {
            System.out.println("Error: No se puede recargar una cantidad negativa");
            return;
        }
        if (this.combustible + cantidad > LIMITE_COMBUSTIBLE) {
            System.out.println("Advertencia: La recarga superaria el limite. Se recargara hasta "
+ LIMITE_COMBUSTIBLE);
            this.combustible = LIMITE_COMBUSTIBLE;
        } else {
            this.combustible += cantidad;
            System.out.println(nombre + " recargo " + cantidad + " unidades de combustible");
        }
    }

    public void mostrarEstado() {
        System.out.println("Estado de la Nave: " + nombre);
        System.out.println("Combustible: " + combustible + "/" + LIMITE_COMBUSTIBLE);
        System.out.println();
    }
```

```
    public String getNombre() { return nombre; }
    public int getCombustible() { return combustible; }
}
```