

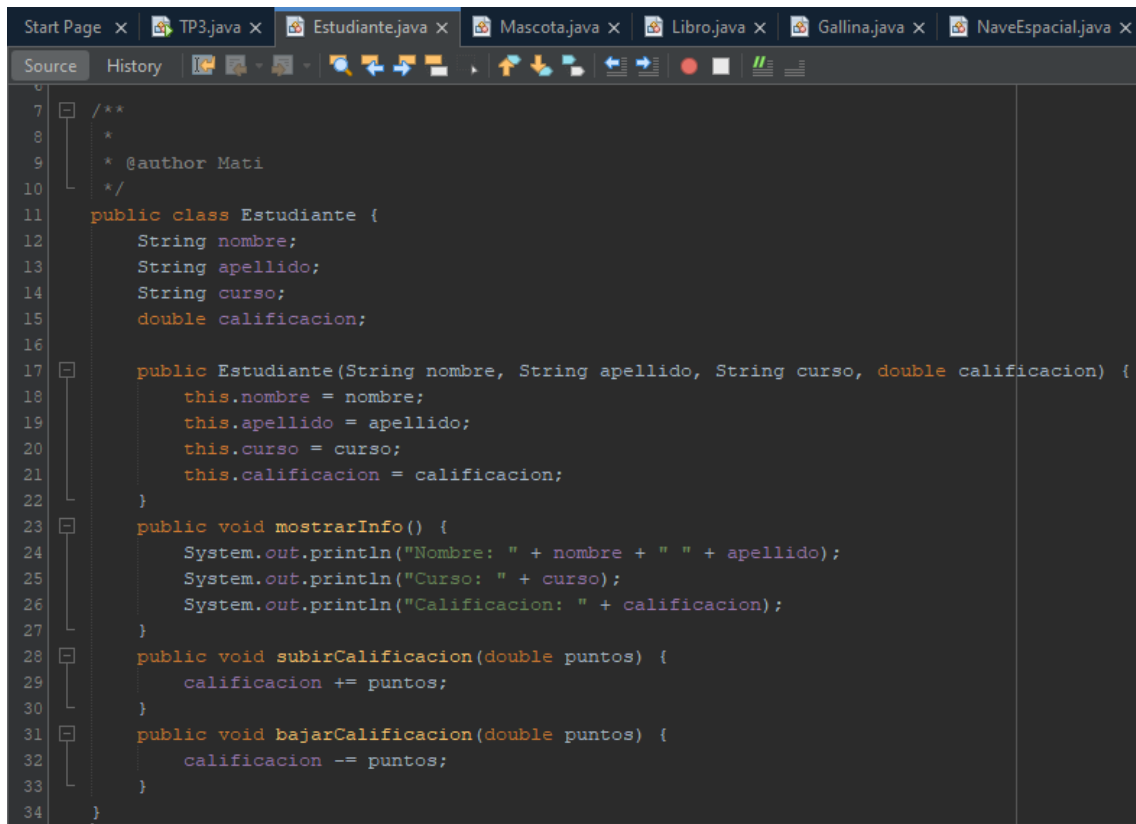
PROGRAMACIÓN II

Trabajo Práctico 3: Introducción a la Programación Orientada a Objetos

Alumno: Matias Luna

GitHub: <https://github.com/matiaslunaa?tab=repositories>

1. Registro de Estudiantes



```
7  /**
8   *
9   * @author Mati
10  */
11  public class Estudiante {
12      String nombre;
13      String apellido;
14      String curso;
15      double calificacion;
16
17      public Estudiante(String nombre, String apellido, String curso, double calificacion) {
18          this.nombre = nombre;
19          this.apellido = apellido;
20          this.curso = curso;
21          this.calificacion = calificacion;
22      }
23      public void mostrarInfo() {
24          System.out.println("Nombre: " + nombre + " " + apellido);
25          System.out.println("Curso: " + curso);
26          System.out.println("Calificacion: " + calificacion);
27      }
28      public void subirCalificacion(double puntos) {
29          calificacion += puntos;
30      }
31      public void bajarCalificacion(double puntos) {
32          calificacion -= puntos;
33      }
34  }
```

2. Registro de Mascotas

```
Start Page x TP3.java x Estudiante.java x Mascota.java x Libro.java x Gallina.java x NaveEspacial.java x
Source History
1
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package tp3;
6
7  /**
8   *
9   * @author Mati
10  */
11  public class Mascota {
12      String nombre;
13      String especie;
14      int edad;
15
16      public Mascota(String nombre, String especie, int edad) {
17          this.nombre = nombre;
18          this.especie = especie;
19          this.edad = edad;
20      }
21
22      public void mostrarInfo() {
23          System.out.println("Mascota: " + nombre + " | Especie: " + especie + " | Edad: " + edad);
24      }
25
26      public void cumplirAnios() {
27          edad++;
28      }
29  }
```

3. Encapsulamiento con la Clase Libro

```
Start Page x TP3.java x Estudiante.java x Mascota.java x Libro.java x Gallina.java x NaveEspacial.java x
Source History
5  package tp3;
6
7  /**
8   *
9   * @author Mati
10  */
11  public class Libro {
12      String titulo;
13      String autor;
14      int anioPublicacion;
15
16      public Libro(String titulo, String autor, int anioPublicacion) {
17          this.titulo = titulo;
18          this.autor = autor;
19          setAnioPublicacion(anioPublicacion);
20      }
21
22      public String getTitulo() { return titulo; }
23      public String getAutor() { return autor; }
24      public int getAnioPublicacion() { return anioPublicacion; }
25
26      public void setAnioPublicacion(int anio) {
27          if (anio > 0 && anio <= 2025) {
28              this.anioPublicacion = anio;
29          } else {
30              System.out.println("Año invalido. No se actualizo.");
31          }
32      }
33
34      public void mostrarInfo() {
35          System.out.println("Libro: " + titulo + " | Autor: " + autor + " | Año: " + anioPublicacion);
36      }
37  }
```

4. Gestión de Gallinas en Granja Digital

```
Start Page x TP3.java x Estudiante.java x Mascota.java x Libro.java x Gallina.java x NaveEspacial.java x
Source History
8
9  * @author Mati
10 */
11 public class Gallina {
12     int idGallina;
13     int edad;
14     int huevosPuestos;
15     public Gallina(int idGallina, int edad) {
16         this.idGallina = idGallina;
17         this.edad = edad;
18         this.huevosPuestos = 0;
19     }
20     public void ponerHuevo() {
21         huevosPuestos++;
22     }
23     public void envejecer() {
24         edad++;
25     }
26     public void mostrarEstado() {
27         System.out.println("Gallina " + idGallina + " | Edad: " + edad + " | Huevos puestos: " + huevosPuestos);
28     }
29 }
```

5. Simulación de Nave Espacial

```
Start Page x TP3.java x Estudiante.java x Mascota.java x Libro.java x Gallina.java x NaveEspacial.java x
Source History
11 public class NaveEspacial {
12     String nombre;
13     int combustible;
14     final int MAX_COMBUSTIBLE = 100;
15
16     public NaveEspacial(String nombre, int combustible) {
17         this.nombre = nombre;
18         this.combustible = Math.min(combustible, MAX_COMBUSTIBLE);
19     }
20     public void despegar() {
21         int costoDespegue = 10;
22         if (combustible >= costoDespegue) {
23             combustible -= costoDespegue;
24             System.out.println(nombre + " ha despegado. Combustible restante: " + combustible);
25         } else {
26             System.out.println("No hay suficiente combustible para despegar.");
27         }
28     }
29     public void avanzar(int distancia) {
30         int consumo = distancia * 2;
31         if (combustible >= consumo) {
32             combustible -= consumo;
33             System.out.println(nombre + " avanza " + distancia + " km. Combustible restante: " + combustible);
34         } else {
35             System.out.println("No hay suficiente combustible para avanzar " + distancia + " km. Necesita " + consumo + " unidades.");
36         }
37     }
38     public void recargarCombustible(int cantidad) {
39         if (cantidad <= 0) {
40             System.out.println("Cantidad invalida para recargar.");
41             return;
42         }
43         combustible = Math.min(combustible + cantidad, MAX_COMBUSTIBLE);
44         System.out.println("Se recargaron " + cantidad + " unidades. Combustible actual: " + combustible);
45     }
46     public void mostrarEstado() {
47         System.out.println("Nave: " + nombre + " | Combustible: " + combustible);
48     }
49 }
```

RESULTADO FINAL POR CONSOLA:

