

programación

Trabajo practico: Introducción a POO

Alumno

Rios Benito Sebastián

Comisión: M2025-4

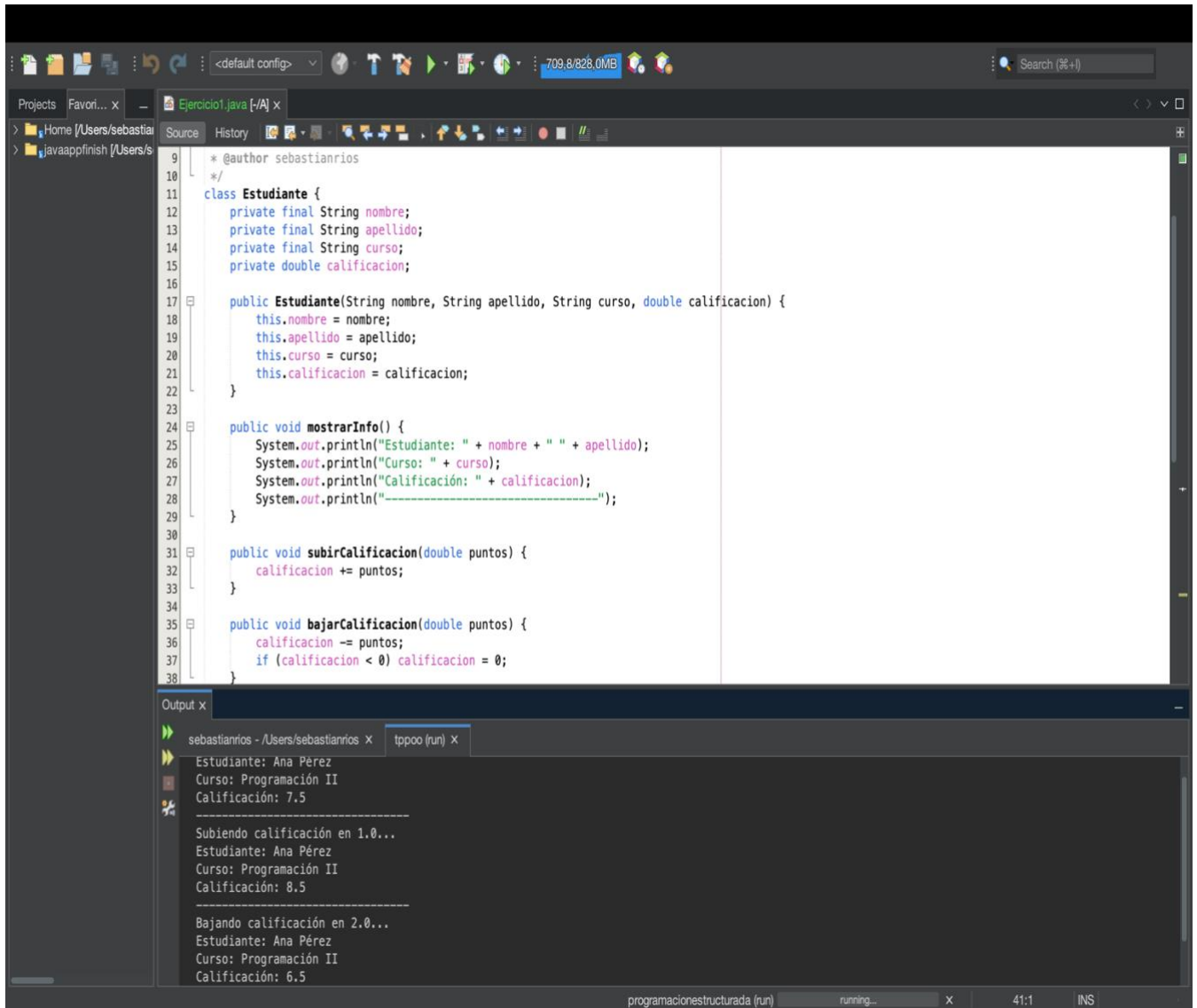
Profesor

Ariel enferrel **Tutor**

David López

28 de agosto 2025

Ejercicio 1



The screenshot shows an IDE window with a Java file named 'Ejercicio1.java'. The code defines a class 'Estudiante' with private final attributes for name, surname, course, and grade. It includes methods for displaying information, increasing the grade, and decreasing the grade. The output window shows the execution of these methods, demonstrating the state changes to the grade.

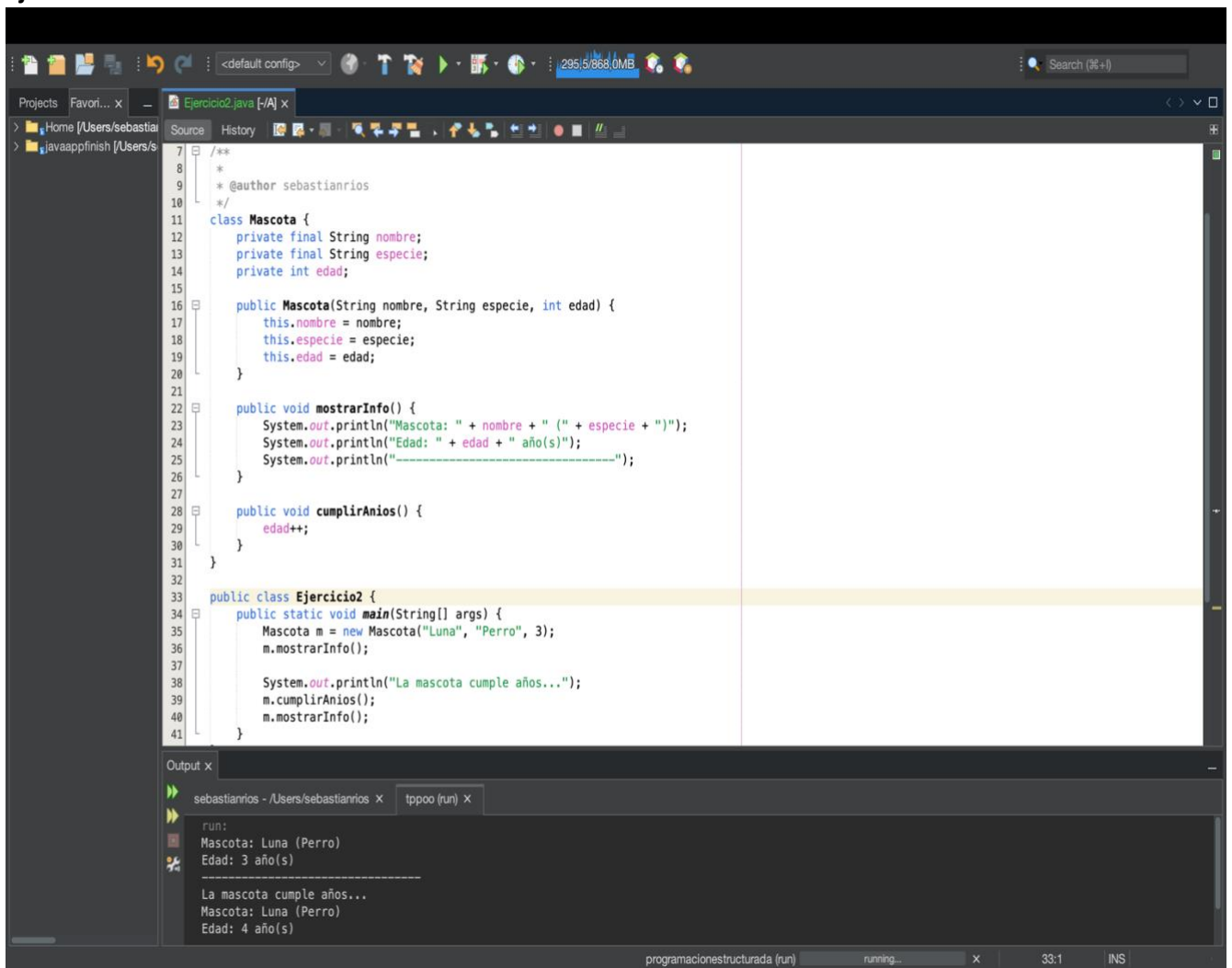
```
9  * @author sebastianrios
10 */
11 class Estudiante {
12     private final String nombre;
13     private final String apellido;
14     private final String curso;
15     private double calificacion;
16
17     public Estudiante(String nombre, String apellido, String curso, double calificacion) {
18         this.nombre = nombre;
19         this.apellido = apellido;
20         this.curso = curso;
21         this.calificacion = calificacion;
22     }
23
24     public void mostrarInfo() {
25         System.out.println("Estudiante: " + nombre + " " + apellido);
26         System.out.println("Curso: " + curso);
27         System.out.println("Calificación: " + calificacion);
28         System.out.println("-----");
29     }
30
31     public void subirCalificacion(double puntos) {
32         calificacion += puntos;
33     }
34
35     public void bajarCalificacion(double puntos) {
36         calificacion -= puntos;
37         if (calificacion < 0) calificacion = 0;
38     }
39 }
```

Output x

```
sebastianrios - /Users/sebastianrios x tppoo (run) x
Estudiante: Ana Pérez
Curso: Programación II
Calificación: 7.5
-----
Subiendo calificación en 1.0...
Estudiante: Ana Pérez
Curso: Programación II
Calificación: 8.5
-----
Bajando calificación en 2.0...
Estudiante: Ana Pérez
Curso: Programación II
Calificación: 6.5
```

programacionestructurada (run) running... x 41:1 INS

Ejercicio 2



The screenshot shows an IDE window with a Java file named `Ejercicio2.java`. The code defines a `Mascota` class with attributes `nombre`, `especie`, and `edad`, and methods `mostrarInfo()` and `cumplirAños()`. A `Ejercicio2` class contains a `main` method that creates a `Mascota` object and calls these methods twice.

```
7  /**
8   *
9   * @author sebastianrios
10  */
11  class Mascota {
12      private final String nombre;
13      private final String especie;
14      private 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 + ")");
24          System.out.println("Edad: " + edad + " año(s)");
25          System.out.println("-----");
26      }
27
28      public void cumplirAños() {
29          edad++;
30      }
31  }
32
33  public class Ejercicio2 {
34      public static void main(String[] args) {
35          Mascota m = new Mascota("Luna", "Perro", 3);
36          m.mostrarInfo();
37
38          System.out.println("La mascota cumple años...");
39          m.cumplirAños();
40          m.mostrarInfo();
41      }
42  }
```

The Output window shows the execution results:

```
run:
Mascota: Luna (Perro)
Edad: 3 año(s)
-----
La mascota cumple años...
Mascota: Luna (Perro)
Edad: 4 año(s)
```

Ejercicio 3

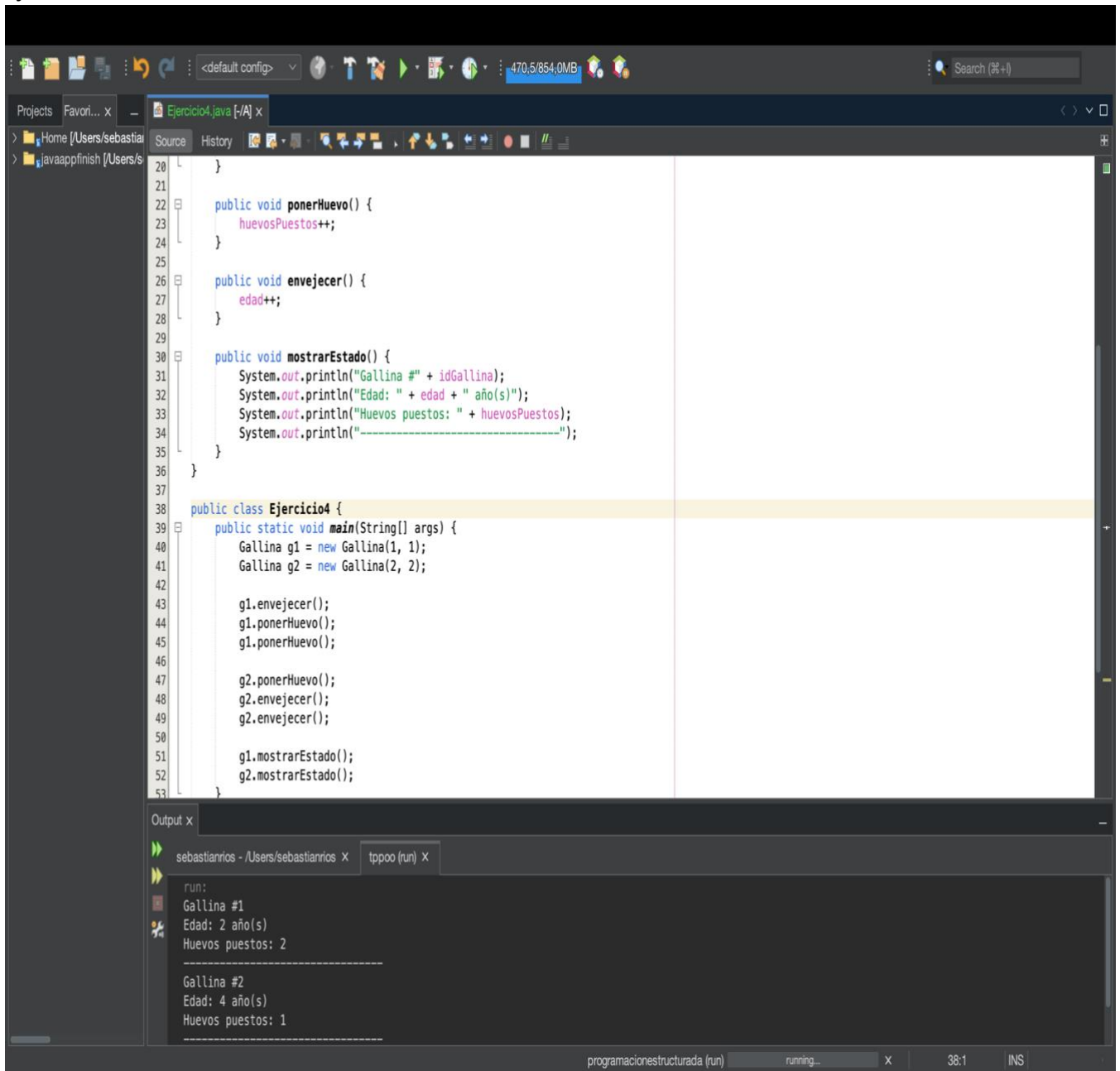
The screenshot shows an IDE with a Java file named `Ejercicio3.java`. The code defines a `Libro` class with attributes `titulo`, `autor`, and `anioPublicacion`. It includes a constructor, getters, and a setter for the year. The `setAnioPublicacion` method includes validation logic to ensure the year is between 1450 and the current year. A `mostrarInfo` method prints the book details. The `Ejercicio3` class contains a `main` method that demonstrates the class usage.

```
9  * @author sebastianrios
10  */
11  final class Libro {
12      private final String titulo;
13      private final String autor;
14      private 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          int anioMin = 1450;
28          int anioMax = java.time.Year.now().getValue();
29          if (anio < anioMin || anio > anioMax) {
30              System.out.println("Año inválido (" + anio + "). Debe estar entre " + anioMin + " y " + anioMax + ".");
31              return;
32          }
33          this.anioPublicacion = anio;
34      }
35
36      public void mostrarInfo() {
37          System.out.println("Libro: \"\" + titulo + "\" de \"\" + autor + \"\" (" + anioPublicacion + ")");
38          System.out.println("-----");
39      }
40  }
41
42  public class Ejercicio3 {
43      public static void main(String[] args) {
```

The Output window shows the execution results:

```
sebastianrios - /Users/sebastianrios x tppoo (run) x
Intento de modificar año con valor inválido...
Año inválido (1200). Debe estar entre 1450 y 2025.
Libro: "Cien años de soledad" de Gabriel García Márquez (1967)
-----
Intento de modificar año con valor válido...
Libro: "Cien años de soledad" de Gabriel García Márquez (1982)
-----
OUTPUT SUCCESSFUL (Análisis de código)
```

Ejercicio 4



The screenshot shows an IDE with a Java project named 'Ejercicio4'. The code defines a 'Gallina' class with methods for laying eggs, aging, and displaying state, and a 'Ejercicio4' class with a 'main' method that creates two 'Gallina' objects and calls their methods. The output window shows the results of the program execution.

```
20 }
21
22 public void ponerHuevo() {
23     huevosPuestos++;
24 }
25
26 public void envejecer() {
27     edad++;
28 }
29
30 public void mostrarEstado() {
31     System.out.println("Gallina #" + idGallina);
32     System.out.println("Edad: " + edad + " año(s)");
33     System.out.println("Huevos puestos: " + huevosPuestos);
34     System.out.println("-----");
35 }
36 }
37
38 public class Ejercicio4 {
39     public static void main(String[] args) {
40         Gallina g1 = new Gallina(1, 1);
41         Gallina g2 = new Gallina(2, 2);
42
43         g1.envejecer();
44         g1.ponerHuevo();
45         g1.ponerHuevo();
46
47         g2.ponerHuevo();
48         g2.envejecer();
49         g2.envejecer();
50
51         g1.mostrarEstado();
52         g2.mostrarEstado();
53     }
54 }
```

Output:

```
run:
Gallina #1
Edad: 2 año(s)
Huevos puestos: 2
-----
Gallina #2
Edad: 4 año(s)
Huevos puestos: 1
-----
```

Ejercicio 5

The screenshot shows an IDE with a Java class named `NaveEspacial` and its execution output. The class has three private final attributes: `nombre` (String), `combustible` (int), and `capacidadMax` (int). It has three public methods: `NaveEspacial` (constructor), `despegar`, and `avanzar`.

```
11 class NaveEspacial {
12     private final String nombre;
13     private int combustible;
14     private final int capacidadMax;
15
16     public NaveEspacial(String nombre, int combustibleInicial, int capacidadMax) {
17         this.nombre = nombre;
18         this.capacidadMax = Math.max(capacidadMax, 1);
19         if (combustibleInicial < 0) combustibleInicial = 0;
20         this.combustible = Math.min(combustibleInicial, this.capacidadMax);
21     }
22
23     public void despegar() {
24         if (combustible <= 0) {
25             System.out.println("No hay combustible para despegar.");
26             return;
27         }
28         System.out.println(nombre + " ha despegado.");
29     }
30
31     public void avanzar(int distancia) {
32         if (distancia <= 0) {
33             System.out.println("La distancia debe ser positiva.");
34             return;
35         }
36         int consumo = distancia;
37         if (consumo > combustible) {
38             System.out.println("Combustible insuficiente. Necesitas " + consumo + " y tienes " + combustible + ".");
39             return;
40         }
41         combustible -= consumo;
42         System.out.println(nombre + " avanzó " + distancia + " unidades. Combustible restante: " + combustible);
43     }
44 }
```

The output window shows the following execution log:

```
Intento de avanzar sin recargar...
Combustible insuficiente. Necesitas 60 y tienes 50.
Recargando combustible...
Recargado: +40. Combustible actual: 90
Avanzando correctamente...
Amanecer avanzó 70 unidades. Combustible restante: 20
Nave: Amanecer | Combustible: 20/100
```

At the bottom of the IDE, the status bar indicates "programacionestructurada (run)" is running, with a memory usage of 304.3/854.0MB, and the system clock shows 10:4 on INS Unix (LF).