

TP 3 Introducción a la POO
PROGRAMACIÓN II
AZCUY NICOLÁS - DNI 33.368.267
LINK GITHUB:

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

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.

```
u2ej1.java x Estudiante.java [-/A] x Main.java [-/A] x
Source History
1 import java.util.Scanner;
2 public class Main {
3     public static void main(String[] args) {
4         Scanner entrada = new Scanner(System.in);
5         System.out.println("Ingrese nombre: ");
6         String nombre = entrada.nextLine();
7         System.out.println("Ingrese apellido: ");
8         String apellido = entrada.nextLine();
9         System.out.println("Ingrese curso: ");
10        String curso = entrada.nextLine();
11        System.out.println("Ingrese la calificación actual: ");
12        Double notaActual = entrada.nextDouble();
13
14        Estudiante estudiantel = new Estudiante(nombre, apellido, curso, notaActual);
15        estudiantel.mostrarInfo();
16
17        System.out.println("\n¿Qué desea hacer?");
18        System.out.println("1 - Subir calificación");
19        System.out.println("2 - Bajar calificación");
20        System.out.print("Elija una opción (1/2): ");
21        int opcion = entrada.nextInt();
22
23        System.out.print("Ingrese los puntos: ");
24        double puntos = entrada.nextDouble();
25
26        if (opcion == 1) {
27            estudiantel.subirCalificacion(puntos);
28        } else if (opcion == 2) {
29            estudiantel.bajarCalificacion(puntos);
30        } else {
31            System.out.println("Opción no válida.");
32        }
33
34    }
```

```
u2ej1.java x Estudiante.java [-/A] x Main.java [-/A] x
Source History
import java.util.Scanner;
2 public class Estudiante {
3     private String nombre;
4     private String apellido;
5     private String curso;
6     private double calificacion;
7
8     public Estudiante(String nom, String apell, String curs, Double calif) {
9         nombre = nom;
10        apellido = apell;
11        curso = curs;
12        calificacion = calif;
13    }
14    public void mostrarInfo() {
15        System.out.println("Nombre: " + nombre + " " + apellido);
16        System.out.println("Curso: " + curso);
17        System.out.println("Calificación actual: " + calificacion);
18    }
19
20    public void subirCalificacion(double puntos) {
21        calificacion += puntos;
22        System.out.println("La nota actualizada es: " + calificacion);
23    }
24
25    public void bajarCalificacion(double puntos) {
26        calificacion -= puntos;
27        System.out.println("La nota bajó a: " + calificacion);
28    }
29 }
30
```

```
Estudiante > Estudiante >  
Output x  
ej01 (run) x Programación 2 - C:\Users\nicoa\Desktop\Nico\Programación 2 x  
run:  
Ingrese nombre:  
Nico  
Ingrese apellido:  
Azcuy  
Ingrese curso:  
Programacion 2  
Ingrese la calificación actual:  
10  
Nombre: Nico Azcuy  
Curso: Programacion 2  
Calificación actual: 10.0  
  
¿Qué desea hacer?  
1 - Subir calificación  
2 - Bajar calificación  
Elija una opción (1/2): 2  
Ingrese los puntos: 8  
La nota bajó a: 2.0  
BUILD SUCCESSFUL (total time: 34 seconds)
```

2. Registro de Mascotas

a. Crear una clase Mascota con los atributos: nombre, especie, edad.

Métodos requeridos: mostrarInfo(), cumplirAños().

Tarea: Crear una mascota, mostrar su información, simular el paso del tiempo y verificar los cambios.

```
Mascota.java [-/A] x Main.java [-/A] x
Source History
1 import java.util.Scanner;
2 public class Main {
3     public static void main(String[] args) {
4         Scanner entrada = new Scanner(System.in);
5
6         System.out.println("REGISTRO DE MASCOTA");
7         System.out.println("Ingrese el nombre de la mascota: ");
8         String nombre = entrada.nextLine();
9         System.out.println("Ingrese la especie de la mascota: ");
10        String especie = entrada.nextLine();
11        System.out.println("Ingrese la edad de la mascota: ");
12        int edad = entrada.nextInt();
13
14        Mascota mascota = new Mascota(nombre, especie, edad);
15
16        mascota.mostrarInfo();
17
18        System.out.println("¿Cuántos años pasaron desde el último registro?: ");
19        int años = entrada.nextInt();
20
21        for (int i = 0; i < años; i++) {
22            mascota.cumplirAños();
23        }
24        System.out.println("Su mascota" + nombre + " cumplió años. Ahora tiene: " + edad);
25
26        mascota.mostrarInfo();
27    }
28 }
29
```

```
Mascota.java [-/A] x Main.java [-/A] x
Source History
import java.util.Scanner;
2 public class Mascota {
3     private String nombre;
4     private String especie;
5     private int edad;
6
7     public Mascota(String nom, String esp, int ed) {
8         nombre = nom;
9         especie = esp;
10        edad = ed;
11    }
12
13    public void mostrarInfo() {
14        System.out.println("Información de la mascota: ");
15        System.out.println("Nombre: " + nombre);
16        System.out.println("Especie: " + especie);
17        System.out.println("Edad: " + edad + " años");
18    }
19
20    public void cumplirAños() {
21        edad++;
22    }
23 }
24
```

```
Output - ej02 (run) x
run:
REGISTRO DE MASCOTA
Ingrese el nombre de la mascota:
Bobi
Ingrese la especie de la mascota:
Doberman
Ingrese la edad de la mascota:
4
Información de la mascota:
Nombre: Bobi
Especie: Doberman
Edad: 4 años
¿Cuántos años pasaron desde el último registro?:
5
Su mascotaBobi cumplió años. Ahora tiene: 4
Información de la mascota:
Nombre: Bobi
Especie: Doberman
Edad: 9 años
BUILD SUCCESSFUL (total time: 35 seconds)
```

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.

```
Libro.java [-/A] x Main.java [-/A] x
Source History
import java.util.Scanner;
2 public class Libro {
3     private String titulo;
4     private String autor;
5     private int añoPublicacion;
6
7     public Libro(String tit, String aut, int añoPub) {
8         titulo = tit;
9         autor = aut;
10        setAñoPublicacion(añoPub);
11    }
12    public String getTitulo() {
13        return titulo;
14    }
15    public String getAutor() {
16        return autor;
17    }
18    public int getAñoPublicacion() {
19        return añoPublicacion;
20    }
21
22    public void setTitulo(String tit) {
23        titulo = tit;
24    }
25    public void setAutor (String aut) {
26        autor = aut;
27    }
28    public void setAñoPublicacion(int nuevoAño) {
29        int añoActual = 2025;
30        if (nuevoAño < 1440 || nuevoAño > añoActual) {
31            System.out.println("Error: Año de publicación inválido. Debe estar entre 1440 (imprenta de Gutenberg) y " + añoActual + ".");
32        } else {
33            añoPublicacion = nuevoAño;
34        }
35    }
36
37    public void mostrarInfo() {
38        System.out.println("Titulo: " + titulo);
39        System.out.println("Autor: " + autor);
40        System.out.println("Año de publicación: " + añoPublicacion);
41    }
42 }
```

```
Libro.java [-/A] x Main.java [-/A] x
Source History
1 import java.util.Scanner;
2 public class Main {
3     public static void main(String[] args) {
4         Scanner entrada = new Scanner(System.in);
5
6         System.out.println("REGISTRO DE LIBRO");
7         System.out.println("Ingrese el título del libro: ");
8         String titulo = entrada.nextLine();
9         System.out.println("Ingrese el autor: ");
10        String autor = entrada.nextLine();
11        System.out.println("Ingrese año de la publicación: ");
12        int añoPublicacion = entrada.nextInt();
13
14        Libro librol = new Libro(titulo, autor, añoPublicacion);
15
16        librol.mostrarInfo();
17    }
18 }
19 }
```

```
Output x
ej03 (run) x Programación 2 - C:\Users\nicoa\Desktop\Nico\Programación 2 x
run:
REGISTRO DE LIBRO
Ingrese el título del libro:
El Aleph
Ingrese el autor:
Jorge Luis Borges
Ingrese año de la publicación:
3000
Error: Año de publicación inválido. Debe estar entre 1440 (imprenta de Gutenberg) y 2025.
Título: El Aleph
Autor: Jorge Luis Borges
Año de publicación: 0
BUILD SUCCESSFUL (total time: 1 minute 26 seconds)
```

```
Output x
ej03 (run) x Programación 2 - C:\Users\nicoa\Desktop\Nico\Programación 2 x
run:
REGISTRO DE LIBRO
Ingrese el título del libro:
El Aleph
Ingrese el autor:
Jorge Luis Borges
Ingrese año de la publicación:
1949
Título: El Aleph
Autor: Jorge Luis Borges
Año de publicación: 1949
BUILD SUCCESSFUL (total time: 19 seconds)
```

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.

```
Gallina.java [-/A] x Main.java [-/A] x
Source History
import java.util.Scanner;
2 public class Gallina {
3     private int idGallina;
4     private int edad;
5     private int huevosPuestos;
6
7     public Gallina(int id, int eda) {
8         idGallina = id;
9         edad = eda;
10        huevosPuestos = 0;
11    }
12    public void ponerHuevo() {
13        if (edad > 1) {
14            int añosProductivos = edad - 1;
15            huevosPuestos = añosProductivos * 365;
16        } else {
17            huevosPuestos = 0;
18            System.out.println("La gallina " + idGallina + " es demasiado joven para poner huevos.");
19        }
20    }
21    public void envejecer(int años) {
22        edad += años;
23    }
24    public void mostrarEstado() {
25        System.out.println("ESTADO DE LA GALLINA: " + idGallina);
26        System.out.println("Edad: " + edad);
27        System.out.println("Huevos puestos: " + huevosPuestos);
28        System.out.println("-----");
29    }
30    public int getEdad() {
31        return edad;
32    }
33
34    public int getHuevosPuestos() {
35        return huevosPuestos;
36    }
37
38    public int getIdGallina() {
39        return idGallina;
40    }
41 }
```



```
Gallina.java [-/A] x Main.java [-/A] x
Source History
1 import java.util.Scanner;
2 public class Main {
3     public static void main(String[] args) {
4         Scanner entrada = new Scanner(System.in);
5
6         System.out.println("REGISTRO DE LA PRIMER GALLINA");
7         System.out.println("Ingrese el código de la gallina: ");
8         int idGallinal = entrada.nextInt();
9         System.out.println("Ingrese la edad de la gallina: ");
10        int edad1 = entrada.nextInt();
11        Gallina gallinal = new Gallina(idGallinal, edad1);
12
13        System.out.println("REGISTRO DE LA SEGUNDA GALLINA");
14        System.out.println("Ingrese el código de la gallina: ");
15        int idGallina2 = entrada.nextInt();
16        System.out.println("Ingrese la edad de la gallina: ");
17        int edad2 = entrada.nextInt();
18        Gallina gallina2 = new Gallina(idGallina2, edad2);
19
20        System.out.println("");
21        System.out.println("----Estado inicial----");
22        gallinal.mostrarEstado();
23        gallina2.mostrarEstado();
24
25        System.out.println("Las gallinas ponen huevos 1 vez por día, a partir del año de vida.");
26        System.out.println("\n¿Cuántos años vamos a simular de envejecimiento de las gallinas?: ");
27        int añosEnvejecimiento = entrada.nextInt();
28
29        gallinal.envejecer(añosEnvejecimiento);
30        gallina2.envejecer(añosEnvejecimiento);
31
32        gallinal.ponerHuevo();
33        gallina2.ponerHuevo();
34
35        System.out.println("\nEstado final después de " + añosEnvejecimiento + " años:");
36        gallinal.mostrarEstado();
37        gallina2.mostrarEstado();
38    }
39 }
```



Output - ej04 (run) ×

```
run:
REGISTRO DE LA PRIMER GALLINA
Ingrese el código de la gallina:
12
Ingrese la edad de la gallina:
1
REGISTRO DE LA SEGUNDA GALLINA
Ingrese el código de la gallina:
32
Ingrese la edad de la gallina:
0

----Estado inicial----
ESTADO DE LA GALLINA: 12
Edad: 1
Huevos puestos: 0
-----
ESTADO DE LA GALLINA: 32
Edad: 0
Huevos puestos: 0
-----
Las gallinas ponen huevos 1 vez por día, a partir del año de vida.

¿Cuántos años vamos a simular de envejecimiento de las gallinas?:
1
La gallina 32 es demasiado joven para poner huevos.

Estado final después de 1 años:
ESTADO DE LA GALLINA: 12
Edad: 2
Huevos puestos: 365
-----
ESTADO DE LA GALLINA: 32
Edad: 1
Huevos puestos: 0
-----
BUILD SUCCESSFUL (total time: 24 seconds)
```



Output - ej04 (run) x

```
run:
REGISTRO DE LA PRIMER GALLINA
Ingrese el código de la gallina:
12
Ingrese la edad de la gallina:
4
REGISTRO DE LA SEGUNDA GALLINA
Ingrese el código de la gallina:
32
Ingrese la edad de la gallina:
6

----Estado inicial----
ESTADO DE LA GALLINA: 12
Edad: 4
Huevos puestos: 0
-----
ESTADO DE LA GALLINA: 32
Edad: 6
Huevos puestos: 0
-----
Las gallinas ponen huevos 1 vez por día, a partir del año de vida.

¿Cuántos años vamos a simular de envejecimiento de las gallinas?:
2

Estado final después de 2 años:
ESTADO DE LA GALLINA: 12
Edad: 6
Huevos puestos: 1825
-----
ESTADO DE LA GALLINA: 32
Edad: 8
Huevos puestos: 2555
-----
BUILD SUCCESSFUL (total time: 14 seconds)
```

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.

```
NaveEspacial.java [-/A] x Main.java [-/A] x
Source History
1 import java.util.Scanner;
2 public class Main {
3     public static void main(String[] args) {
4         Scanner entrada = new Scanner(System.in);
5
6         System.out.println("REGISTRO DE LA NAVE");
7         System.out.println("Ingrese el nombre de la nave: ");
8         String nombrel = entrada.nextLine();
9         System.out.println("El cohete " + nombrel + " necesita 60 litros de combustible para poder despegar.");
10        System.out.println("Ingrese la cantidad de combustible que indica el tablero: ");
11        int combustibleInicial = entrada.nextInt();
12
13        NaveEspacial nave1 = new NaveEspacial(nombrel, combustibleInicial);
14        System.out.println("Estado inicial de la nave");
15        nave1.mostrarEstado();
16
17        boolean continuar = true;
18        while (continuar) {
19            System.out.println("\nMenú: \n 1 - Despegar\n 2 - Avanzar\n 3 - Recargar combustible\n 4 - Mostrar estado\n 5 - Salir");
20            int opcion = entrada.nextInt();
21
22            switch (opcion) {
23                case 1:
24                    nave1.despegar();
25                    break;
26                case 2:
27                    System.out.println("Ingrese la distancia a avanzar: ");
28                    int distancia = entrada.nextInt();
29                    nave1.avanzar(distancia);
30                    break;
31                case 3:
32                    System.out.println("Ingrese la cantidad de combustible para recargar: ");
33                    int cantidad = entrada.nextInt();
34                    nave1.recargarCombustible(cantidad);
35                    break;
36                case 4:
37                    nave1.mostrarEstado();
38                    break;
39                case 5:
40                    continuar = false;
41                    break;
42                default:
43                    System.out.println("Error, por favor elija una opción válida");
44            }
45        }
46    }
47 }
```

```
NaveEspacial.java [-/A] x Main.java [-/A] x
Source History
import java.util.Scanner;
2 public class NaveEspacial {
3     private String nombre;
4     private int combustible;
5     public NaveEspacial(String nom, int comb) {
6         nombre = nom;
7         combustible = comb;
8     }
9     public void despegar() {
10        if (combustible >= 60) {
11            System.out.println("El cohete " + nombre + " ha despegado correctamente!");
12            combustible -= 60;
13        } else {
14            System.out.println("El cohete " + nombre + " no tiene combustible para despegar!");
15        }
16    }
17    public void avanzar(int distancia){
18        int combustibleNecesario = distancia * 10;
19        if (combustible >= combustibleNecesario) {
20            System.out.println("El cohete " + nombre + " avanzó " + distancia + " km.");
21            combustible -= combustibleNecesario;
22        } else {
23            System.out.println("El cohete " + nombre + " no tiene combustible para avanzar " + distancia + " km.");
24        }
25        System.out.println("Necesita " + combustibleNecesario + " litros, pero solo tiene " + combustible + " litros.");
26    }
27    public void recargarCombustible(int cantidad) {
28        int limiteRecarga = 1000;
29        int espacioDisponible = limiteRecarga - combustible;
30        if (cantidad < espacioDisponible) {
31            combustible += cantidad;
32            System.out.println("Recarga exitosa. Puede continuar viaje.");
33        } else {
34            System.out.println("Usted ya tiene el tanque lleno.");
35        }
36    }
37    public void mostrarEstado() {
38        System.out.println("=== ESTADO DE LA NAVE: " + nombre);
39        System.out.println("Combustible: " + combustible + " litros.");
40        System.out.println("=====");
41    }
42    public String getNombre() {
43        return nombre;
44    }
45    public int getCombustible() {
```

Output ×

Programación 2 - C:\Users\nicoa\Desktop\Nico\Programación 2 ×

ej05 (run) ×

```
run:
REGISTRO DE LA NAVE
Ingrese el nombre de la nave:
Apolo
El cohete Apolo necesita 60 litros de combustible para poder despegar.
Ingrese la cantidad de combustible que indica el tablero:
50
Estado inicial de la nave
=== ESTADO DE LA NAVE: Apolo
Combustible: 50 litros.
=====

Menú:
1 - Despegar
2 - Avanzar
3 - Recargar combustible
4 - Mostrar estado
5 - Salir
1
El coheteApolo no tiene combustible para despegar!

Menú:
1 - Despegar
2 - Avanzar
3 - Recargar combustible
4 - Mostrar estado
5 - Salir
3
Ingrese la cantidad de combustible para recargar:
10
Recarga exitosa. Puede continuar viaje.

Menú:
1 - Despegar
2 - Avanzar
3 - Recargar combustible
4 - Mostrar estado
5 - Salir
1
El coheteApolo ha despegado correctamente!
```

Menú:

- 1 - Despegar
- 2 - Avanzar
- 3 - Recargar combustible
- 4 - Mostrar estado
- 5 - Salir

2

Ingrese la distancia a avanzar:

10

El cohete Apolo no tiene combustible para avanzar 10 km.
Necesita 100 litros, pero solo tiene 0 litros.

Menú:

- 1 - Despegar
- 2 - Avanzar
- 3 - Recargar combustible
- 4 - Mostrar estado
- 5 - Salir

3

Ingrese la cantidad de combustible para recargar:

100

Recarga exitosa. Puede continuar viaje.

Menú:

- 1 - Despegar
- 2 - Avanzar
- 3 - Recargar combustible
- 4 - Mostrar estado
- 5 - Salir

2

Ingrese la distancia a avanzar:

10

El cohete Apolo avanzó 10 km.
Necesita 100 litros, pero solo tiene 0 litros.