TECNICATURA UNIVERSITARIA
EN PROGRAMACIÓN
A DISTANCIA

TP 3 Introducción a la POO
PROGRAMACIÓN II

AZCUY NICOLÁS - DNI 33.368.267

LINK GITHUB:
https://github.com/nazcuy/Programaci-n-II/tree/master/Trabajos%20Pr%C3%A1cticos%20Programaci%C3%B3n%202/TP3

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 × Estudiante.java [-/A] × Main.java [-/A] ×

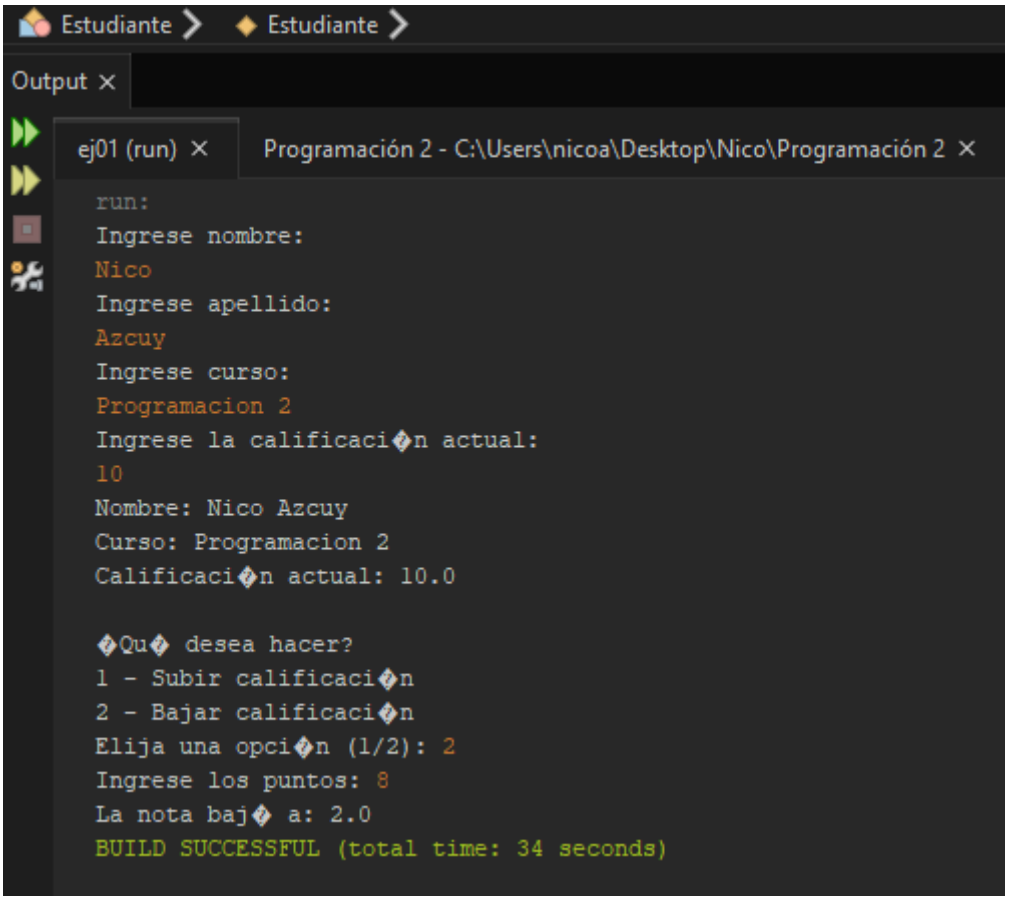
Source History

```java
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner entrada = new Scanner(System.in);
        System.out.println("Ingrese nombre: ");
        String nombre = entrada.nextLine();
        System.out.println("Ingrese apellido: ");
        String apellido = entrada.nextLine();
        System.out.println("Ingrese curso: ");
        String curso = entrada.nextLine();
        System.out.println("Ingrese la calificación actual: ");
        Double notaActual = entrada.nextDouble();

        Estudiante estudiante1 = new Estudiante(nombre, apellido, curso, notaActual);
        estudiante1.mostrarInfo();

        System.out.println("\n¿Qué desea hacer?");
        System.out.println("1 - Subir calificación");
        System.out.println("2 - Bajar calificación");
        System.out.print("Elija una opción (1/2): ");
        int opcion = entrada.nextInt();

        System.out.print("Ingrese los puntos: ");
        double puntos = entrada.nextDouble();

        if (opcion == 1) {
            estudiante1.subirCalificacion(puntos);
        } else if (opcion == 2) {
            estudiante1.bajarCalificacion(puntos);
        } else {
            System.out.println("Opción no válida.");
        }

    }
```

u2ej1.java × Estudiante.java [-/A] × Main.java [-/A] ×

Source History

```java
import java.util.Scanner;
public class Estudiante {
    private String nombre;
    private String apellido;
    private String curso;
    private double calificacion;

    public Estudiante(String nom, String apell, String curs, Double calif) {
        nombre = nom;
        apellido = apell;
        curso = curs;
        calificacion = calif;
}
    public void mostrarInfo() {
        System.out.println("Nombre: " + nombre + " " + apellido);
        System.out.println("Curso: " + curso);
        System.out.println("Calificación actual: " + calificacion);
    }

    public void subirCalificacion(double puntos) {
        calificacion += puntos;
        System.out.println("La nota actualizada es: " + calificacion);
    }

        public void bajarCalificacion(double puntos) {
        calificacion -= puntos;
        System.out.println("La nota bajó a: " + calificacion);
    }
}

```


Estudiante
Estudiante
Output
ej01 (run)
Programación 2 - C:\Users\nicoa\Desktop\Nico\Programación 2
run:
Ingrese nombre:
Nico
Ingrese apellido:
Azcuy
Ingrese curso:
Programacion 2
Ingrese la calificaci□n actual:
10
Nombre: Nico Azcuy
Curso: Programacion 2
Calificaci□n actual: 10.0
□Qu□ desea hacer?
1 - Subir calificaci□n
2 - Bajar calificaci□n
Elija una opci□n (1/2): 2
Ingrese los puntos: 8
La nota baj□ 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(), cumplirAnios().

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

Mascota.java [-/A] × Main.java [-/A] ×

Source History

```java
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner entrada = new Scanner(System.in);

        System.out.println("REGISTRO DE MASCOTA");
        System.out.println("Ingrese el nombre de la mascota: ");
        String nombre = entrada.nextLine();
        System.out.println("Ingrese la especie de la mascota: ");
        String especie = entrada.nextLine();
        System.out.println("Ingrese la edad de la mascota: ");
        int edad = entrada.nextInt();

        Mascota mascota = new Mascota(nombre, especie, edad);

        mascota.mostrarInfo();

        System.out.println("¿Cuántos años pasaron desde el último registro?: ");
        int anios = entrada.nextInt();

        for (int i = 0; i < anios; i++) {
            mascota.cumplirAnios();
        }
        System.out.println("Su mascota" + nombre + " cumplió años. Ahora tiene: " + edad);

        mascota.mostrarInfo();
    }
}
```

Mascota.java [-/A] × Main.java [-/A] ×

Source History

```java
import java.util.Scanner;
public class Mascota {
    private String nombre;
    private String especie;
    private int edad;

    public Mascota(String nom, String esp, int ed) {
        nombre = nom;
        especie = esp;
        edad = ed;
    }

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

    public void cumplirAnios() {
        edad++;
    }
}
```

Output - ej02 (run) ×

```
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
Informaci□n de la mascota:
Nombre: Bobi
Especie: Doberman
Edad: 4 a□os
□Cu□ntos a□os pasaron desde el □ltimo registro?:
5
Su mascotaBobi cumpli□ a□os. Ahora tiene: 4
Informaci□n de la mascota:
Nombre: Bobi
Especie: Doberman
Edad: 9 a□os
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] × Main.java [-/A] ×

Source History

```java
import java.util.Scanner;
public class Libro {
    private String titulo;
    private String autor;
    private int añoPublicacion;

    public Libro(String tit, String aut, int añoPub) {
        titulo = tit;
        autor = aut;
        setAñoPublicacion(añoPub);
    }
    public String getTitulo() {
        return titulo;
    }
    public String getAutor() {
        return autor;
    }
    public int getAñoPublicacion() {
        return añoPublicacion;
    }

    public void setTitulo(String tit) {
        titulo = tit;
    }
    public void setAutor (String aut) {
        autor = aut;
    }
    public void setAñoPublicacion(int nuevoAño) {
        int añoActual = 2025;
        if (nuevoAño < 1440 || nuevoAño > añoActual) {
            System.out.println("Error: Año de publicación inválido. Debe estar entre 1440 (imprenta de Gutenberg) y " + añoActual + ".");
        } else {
            añoPublicacion = nuevoAño;
        }
    }

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

Libro.java [-/A] × Main.java [-/A] ×

Source History

```java
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner entrada = new Scanner(System.in);

        System.out.println("REGISTRO DE LIBRO");
        System.out.println("Ingrese el título del libro: ");
        String titulo = entrada.nextLine();
        System.out.println("Ingrese el autor: ");
        String autor = entrada.nextLine();
        System.out.println("Ingrese año de la publicación: ");
        int añoPublicacion = entrada.nextInt();

        Libro libro1 = new Libro(titulo, autor, añoPublicacion);

        libro1.mostrarInfo();

    }
}
```

Output ×

ej03 (run) × Programación 2 - C:\Users\nicoa\Desktop\Nico\Programación 2 ×

```
run:
REGISTRO DE LIBRO
Ingrese el t□tulo del libro:
El Aleph
Ingrese el autor:
Jorge Luis Borges
Ingrese a□o de la publicaci□n:
3000
Error: A□o de publicaci□n inv□lido. Debe estar entre 1440 (imprenta de Gutenberg) y 2025.
T□tulo: El Aleph
Autor: Jorge Luis Borges
A□o de publicaci□n: 0
BUILD SUCCESSFUL (total time: 1 minute 26 seconds)
```

Output ×

ej03 (run) × Programación 2 - C:\Users\nicoa\Desktop\Nico\Programación 2 ×

```
run:
REGISTRO DE LIBRO
Ingrese el t□tulo del libro:
El Aleph
Ingrese el autor:
Jorge Luis Borges
Ingrese a□o de la publicaci□n:
1949
T□tulo: El Aleph
Autor: Jorge Luis Borges
A□o de publicaci□n: 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] × Main.java [-/A] ×

Source History

```java
import java.util.Scanner;
public class Gallina {
    private int idGallina;
    private int edad;
    private int huevosPuestos;

    public Gallina(int id, int eda) {
        idGallina = id;
        edad = eda;
        huevosPuestos = 0;
    }
    public void ponerHuevo() {
        if (edad > 1) {
            int añosProductivos = edad -1;
            huevosPuestos = añosProductivos * 365;
        } else {
            huevosPuestos = 0;
            System.out.println("La gallina " + idGallina + " es demasiado joven para poner huevos.");
        }
    }
    public void envejecer(int años) {
        edad+= años;
    }
    public void mostrarEstado() {
        System.out.println("ESTADO DE LA GALLINA: " + idGallina);
        System.out.println("Edad: " + edad);
        System.out.println("Huevos puestos: " + huevosPuestos);
        System.out.println("----------------------");
    }
    public int getEdad() {
        return edad;
    }

    public int getHuevosPuestos() {
        return huevosPuestos;
    }

    public int getIdGallina() {
        return idGallina;
    }
}
```

Gallina.java [-/A] × Main.java [-/A] ×

Source History

```java
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner entrada = new Scanner(System.in);

            System.out.println("REGISTRO DE LA PRIMER GALLINA");
            System.out.println("Ingrese el código de la gallina: ");
            int idGallina1 = entrada.nextInt();
            System.out.println("Ingrese la edad de la gallina: ");
            int edad1 = entrada.nextInt();
            Gallina gallina1 = new Gallina(idGallina1, edad1);

            System.out.println("REGISTRO DE LA SEGUNDA GALLINA");
            System.out.println("Ingrese el código de la gallina: ");
            int idGallina2 = entrada.nextInt();
            System.out.println("Ingrese la edad de la gallina: ");
            int edad2 = entrada.nextInt();
            Gallina gallina2 = new Gallina(idGallina2, edad2);

            System.out.println("");
            System.out.println("----Estado inicial----");
            gallina1.mostrarEstado();
            gallina2.mostrarEstado();

            System.out.println("Las gallinas ponen huevos 1 vez por día, a partir del año de vida.");
            System.out.println("\n¿Cuántos años vamos a simular de envejecimiento de las gallinas?: ");
            int añosEnvejecimiento = entrada.nextInt();

            gallina1.envejecer(añosEnvejecimiento);
            gallina2.envejecer(añosEnvejecimiento);

            gallina1.ponerHuevo();
            gallina2.ponerHuevo();

            System.out.println("\nEstado final después de " + añosEnvejecimiento + " años:");
            gallina1.mostrarEstado();
            gallina2.mostrarEstado();
    }
}
```

Output - ej04 (run) ×

```
run:
REGISTRO DE LA PRIMER GALLINA
Ingrese el c◆digo de la gallina:
12
Ingrese la edad de la gallina:
1
REGISTRO DE LA SEGUNDA GALLINA
Ingrese el c◆digo 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 d◆a, a partir del a◆o de vida.

◆Cu◆ntos a◆os vamos a simular de envejecimiento de las gallinas?:
1
La gallina 32 es demasiado joven para poner huevos.

Estado final despu◆s de 1 a◆os:
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) ×

```
run:
REGISTRO DE LA PRIMER GALLINA
Ingrese el c□digo de la gallina:
12
Ingrese la edad de la gallina:
4
REGISTRO DE LA SEGUNDA GALLINA
Ingrese el c□digo 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 d□a, a partir del a□o de vida.

□Cu□ntos a□os vamos a simular de envejecimiento de las gallinas?:
2

Estado final despu□s de 2 a□os:
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] × Main.java [-/A] ×

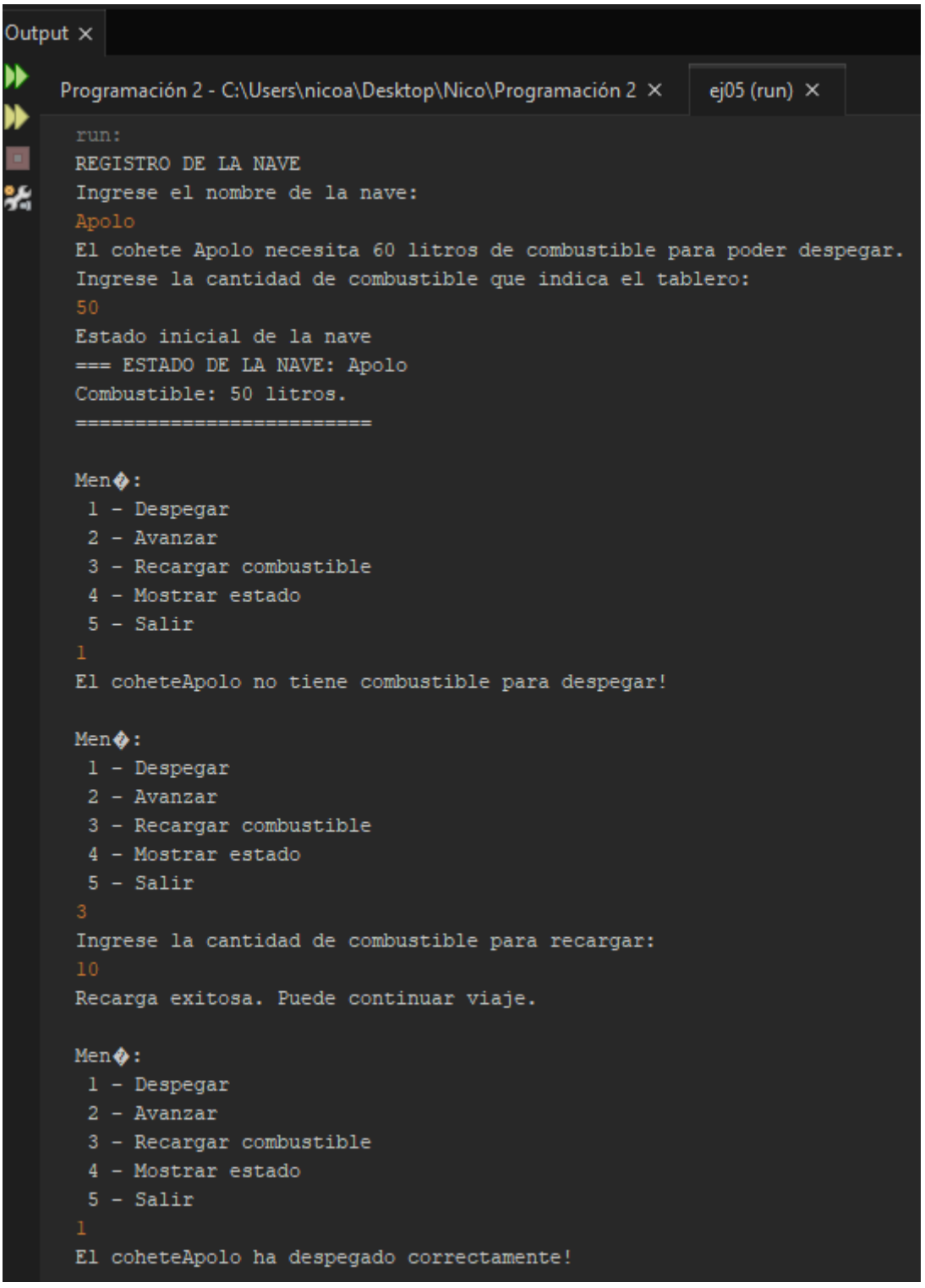
Source History

```java
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner entrada = new Scanner(System.in);

        System.out.println("REGISTRO DE LA NAVE");
        System.out.println("Ingrese el nombre de la nave: ");
        String nombre1 = entrada.nextLine();
        System.out.println("El cohete " + nombre1 + " necesita 60 litros de combustible para poder despegar.");
        System.out.println("Ingrese la cantidad de combustible que indica el tablero: ");
        int combustibleInicial = entrada.nextInt();

        NaveEspacial nave1 = new NaveEspacial(nombre1, combustibleInicial);
        System.out.println("Estado inicial de la nave");
        nave1.mostrarEstado();

        boolean continuar = true;
        while (continuar) {
            System.out.println("\nMenú: \n 1 - Despegar\n 2 - Avanzar\n 3 - Recargar combustible\n 4 - Mostrar estado\n 5 - Salir");
            int opcion = entrada.nextInt();

            switch (opcion) {
                case 1:
                    nave1.despegar();
                    break;
                case 2:
                    System.out.println("Ingrese la distancia a avanzar: ");
                    int distancia = entrada.nextInt();
                    nave1.avanzar(distancia);
                    break;
                case 3:
                    System.out.println("Ingrese la cantidad de combustible para recargar: ");
                    int cantidad = entrada.nextInt();
                    nave1.recargarCombustible(cantidad);
                    break;
                case 4:
                    nave1.mostrarEstado();
                    break;
                case 5:
                    continuar = false;
                    break;
                default:
                    System.out.println("Error, por favor elija una opción válida");
            }
```

NaveEspacial.java [-/A] × Main.java [-/A] ×

Source History

```java
import java.util.Scanner;
public class NaveEspacial {
    private String nombre;
    private int combustible;
    public NaveEspacial(String nom, int comb) {
        nombre = nom;
        combustible = comb;
    }
    public void despegar() {
        if (combustible >= 60) {
            System.out.println("El cohete " + nombre + " ha despegado correctamente!");
            combustible -= 60;
        } else {
            System.out.println("El cohete " + nombre + " no tiene combustible para despegar!");
        }
    }
    public void avanzar(int distancia){
        int combustibleNecesario = distancia * 10;
        if (combustible >= combustibleNecesario) {
            System.out.println("El cohete " + nombre + " avanzó " + distancia + " km.");
            combustible -= combustibleNecesario;
        } else {
            System.out.println("El cohete " + nombre + " no tiene combustible para avanzar " + distancia + " km.");
        }
        System.out.println("Necesita " + combustibleNecesario + " litros, pero solo tiene " + combustible + " litros.");
    }
    public void recargarCombustible(int cantidad) {
        int limiteRecarga = 1000;
        int espacioDisponible = limiteRecarga - combustible;
        if (cantidad < espacioDisponible) {
            combustible += cantidad;
            System.out.println("Recarga exitosa. Puede continuar viaje.");
        } else {
            System.out.println("Usted ya tiene el tanque lleno.");
        }
    }
    public void mostrarEstado() {
        System.out.println("=== ESTADO DE LA NAVE: " + nombre);
        System.out.println("Combustible: " + combustible + " litros.");
        System.out.println("========================");
    }
    public String getNombre() {
        return nombre;
    }
    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.
==========
Men:
1 - Despegar
2 - Avanzar
3 - Recargar combustible
4 - Mostrar estado
5 - Salir
1
El coheteApolo no tiene combustible para despegar!
Men:
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.
Men:
1 - Despegar
2 - Avanzar
3 - Recargar combustible
4 - Mostrar estado
5 - Salir
1
El coheteApolo ha despegado correctamente!

```
Men�:
 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.

Men�:
 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.

Men�:
 1 - Despegar
 2 - Avanzar
 3 - Recargar combustible
 4 - Mostrar estado
 5 - Salir
2
Ingrese la distancia a avanzar:
10
El coheteApolo avanz� 10 km.
Necesita 100 litros, pero solo tiene 0 litros.
```