ACTIVIDAD A REALIZAR

1. **Analizar las indicaciones y el modelado UML.**
2. **Pasarlo a JAVA**

Ejercicio 1: Gestión de Cursos

Problema: Una universidad necesita modelar la información básica de sus cursos.
Cada curso debe contener los siguientes datos:

- **Nombre** del curso (por ejemplo, "Programación I"),
- **Código** único que identifique el curso (por ejemplo, "INF101"),
- **Cantidad de créditos** asignados al curso (por ejemplo, 5 créditos).

Además, el sistema debe proporcionar las siguientes funcionalidades a través de métodos:

- **Mostrar la información completa del curso**: El método debe desplegar en pantalla el nombre, el código y la cantidad de créditos actuales del curso.
- **Actualizar la cantidad de créditos**: El método debe permitir asignar un nuevo valor al atributo creditos, reemplazando el valor anterior.

- **Clase:** Curso
- **Atributos:**

  nombreCurso:

  String code:

  String

  creditos: int
- **Métodos:**

  infoCurso(): void

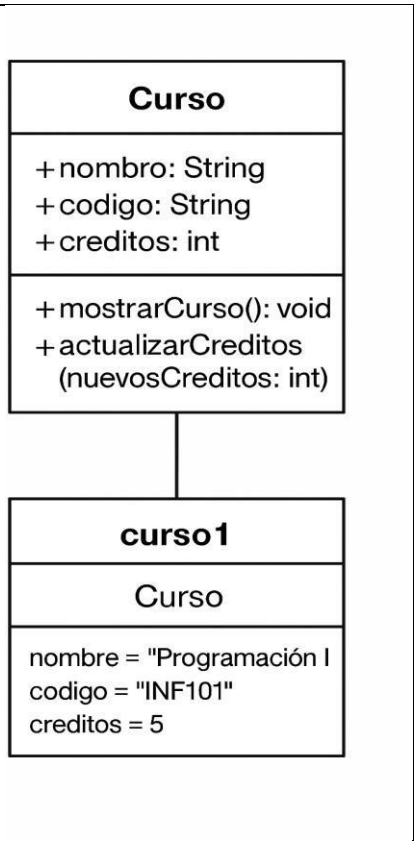
  actualizarCreditos(nuevosCreditos: int): void
- **Objeto:**

  curso1: Curso

  - nombre = "Programación "
  - codigo = "PROG101"
  - creditos = 3

  curso2: Curso

  - nombre = "Matematicas"
  - codigo = "MAT101"
  - creditos = 4



COMPLETAR LA TABLA, INDICACIONES Y A LA DERECHA EL UML, SEGUIR EL EJEMPLO ANTERIOR (Gestión de Cursos) Y CREAR EN JAVA LA PRACTICA

```
C:\Users\XCORP\.jdks\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Users\XCORP\A
Nombre del curso: Programación
Código del curso: PROG101
Créditos del curso: 3
Nombre del curso: Matemáticas
Código del curso: MAT101
Créditos del curso: 4
Nuevos créditos del curso: 5

Proceso terminado con código de salida 0
```

Problema 2: Gestión de Registro de Libros

Una biblioteca necesita desarrollar un sistema para registrar y administrar los libros disponibles en su colección.

Cada libro debe contener la siguiente información:

- **Título** del libro (por ejemplo, "Don Quijote"),
- **Autor** del libro (por ejemplo, "Miguel de Cervantes"),
- **Año de publicación** (por ejemplo, 1605).

Además, el sistema debe permitir:

- **Mostrar toda la información** del libro registrado,
- **Actualizar el año de publicación** en caso de que se corrija o actualice la edición.

Se solicita:

1. **Crear una clase** llamada Libro que contenga:

   Los atributos: titulo, autor y anioPublicacion.

   Los métodos: mostrarInformacion() para desplegar la información y establecerAnio(nuevoAnio: int) para actualizar el año de publicación.

2. **Crear dos objeto** llamado libro1, 2 a partir de la clase Libro

- **Clase:** Libros
- **Atributos:**

String titulo;

String autor;

int anioPublicacion;

- **Métodos:**
  public void infoLibro()

  public void actualizarAnioPublicacion()
- **Objeto:**

  Libros libro1 = new Libros();

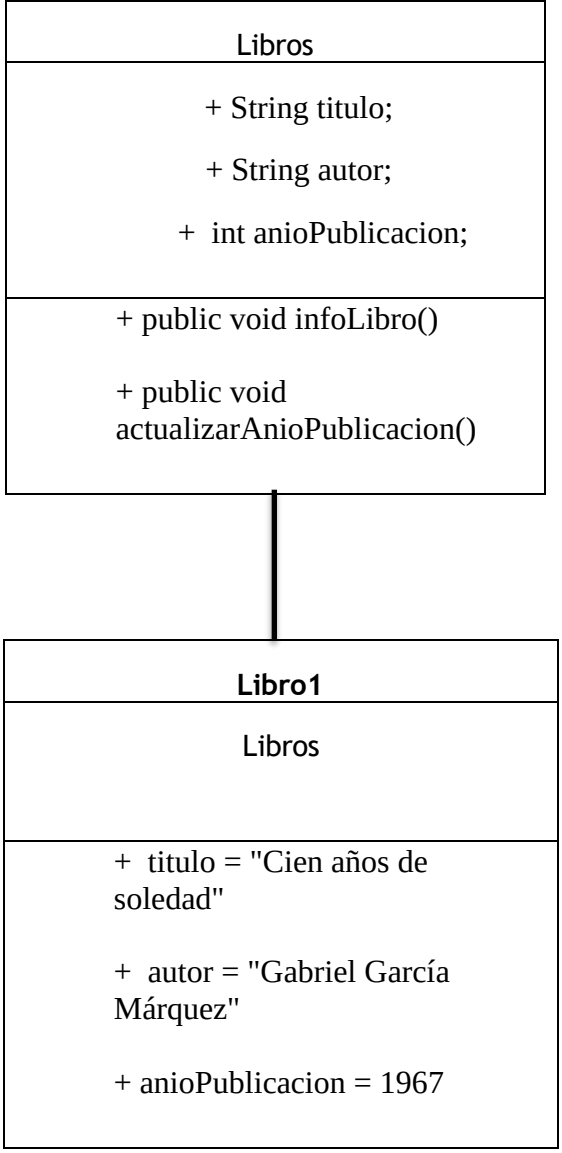
libro1.titulo = "El Principito";

libro1.autor = "Antoine de Saint-Exupéry";

libro1.anioPublicacion = 1943;

Libros libro2 = new Libros();

libro2.titulo = "Cien años de soledad";
libro2.autor = "Gabriel García Márquez";
libro2.anioPublicacion = 1967;



```
C:\Users\XCORP\.jdks\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Users\XCOF
Información del libro 1:
Titulo: El Principito
Autor: Antoine de Saint-Exupéry
Año de Publicacion: 1943

Información del libro 2:
Titulo: Cien años de soledad
Autor: Gabriel García Márquez
Año de Publicacion: 1967
Ingrese el nuevo año de publicacion del libro 1:
1941
Año de publicacion actualizado a: 1941

Información actualizada del libro 1:
Titulo: El Principito
Autor: Antoine de Saint-Exupéry
Año de Publicacion: 1941

Proceso terminado con código de salida 0
```

COMPLETAR LA TABLA, INDICACIONES Y A LA DERECHA EL UML, SEGUIR EL EJEMPLO ANTERIOR (Gestión de Cursos) Y CREAR EN JAVA LA PRACTICA

Gestión de Jugadores 3

Un club deportivo necesita modelar la información básica de sus jugadores.

Cada jugador debe contener los siguientes datos:

- **Nombre** del jugador (por ejemplo, "Lionel Messi"),
- **Número** que identifica al jugador en el equipo (por ejemplo, 10),
- **Posición** en la que juega (por ejemplo, "Delantero").

Además, el sistema debe proporcionar las siguientes funcionalidades a través de métodos:

- **Mostrar la información completa del jugador**: El método debe desplegar en pantalla el nombre, el número y la posición del jugador.
- **Actualizar la posición del jugador**: El método debe permitir asignar un nuevo valor al atributo posicion, reemplazando el valor anterior.

- **Clase:** Jugador
- **Atributos:**

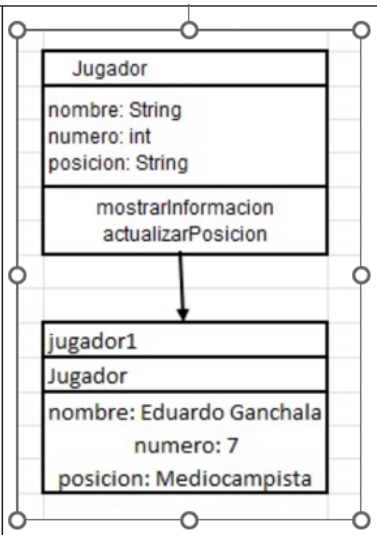
  nombre: String

  numero: int

  posicion: String

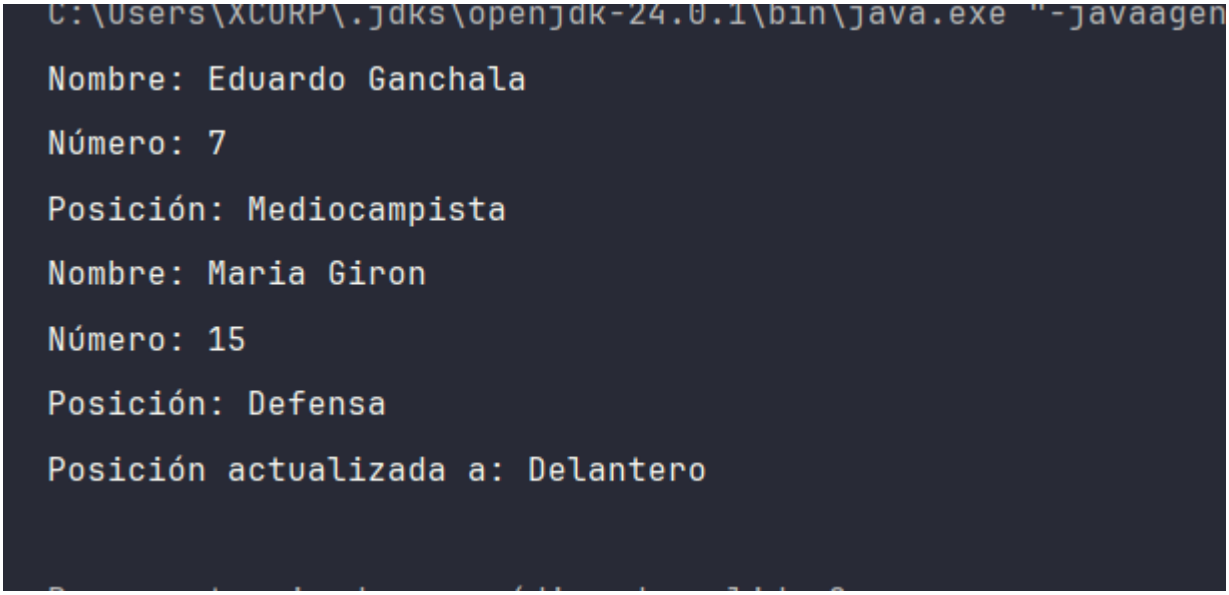
- **Métodos:**

```
public void mostrarInformacion()
public void actualizarPosicion(String
nuevaPosicion)
```



- **Objeto:**

```
jugador1: Jugador
  • nombre="Eduardo Ganchala"
  • numero= 7
  • posición="Mediocampista"
jugador2: Jugador
  • nombre="Maria Giron"
  • numero= 15
  • posición= "Defensa"
```



COMPLETAR LA TABLA, INDICACIONES Y A LA DERECHA EL UML, SEGUIR EL EJEMPLO ANTERIOR (Gestión de Cursos) Y CREAR EN JAVA LA PRACTICA

Gestión de Ciclistas 4

Problema:

Una federación de ciclismo necesita modelar la información de sus ciclistas para gestionar su desempeño y progreso.

Cada ciclista debe registrar:

- **Nombre** del ciclista (por ejemplo, "Egan Bernal"),
- **Edad** del ciclista (por ejemplo, 26),
- **Kilómetros recorridos** durante la temporada (por ejemplo, 4200 km).

El sistema debe proporcionar las siguientes funcionalidades a través de métodos:

- **Mostrar toda la información del ciclista** (nombre, edad y kilómetros recorridos).
- **Actualizar los kilómetros recorridos** añadiendo los kilómetros de una nueva competencia.
- **Calcular el promedio de kilómetros recorridos por mes**, suponiendo que la temporada dura 12 meses.
- **Crear 3 objetos**

- **Clase:** Ciclista
- **Atributos:**

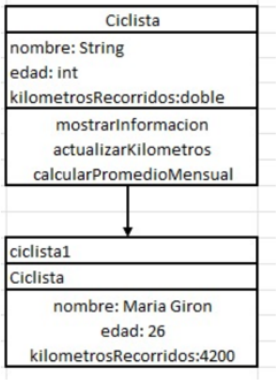
  nombre: String

  edad: int

  kilometrosRecorridos:double

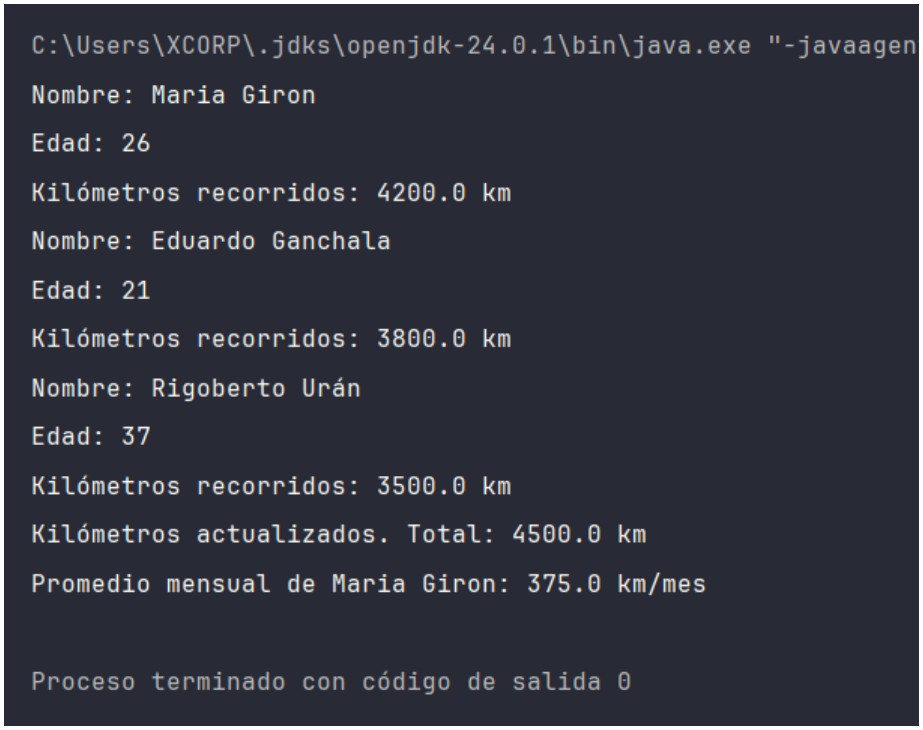
- **Métodos:**

```
public void mostrarInformacion()
public void actualizarKilometros(double
nuevosKilometros)
public double calcularPromedioMensual()
```



- **Objeto:**

```
ciclista1:Ciclista
nombre:"Maria Giron"
edad: 26
kilometrosRecorridos: 4200
ciclista2:Ciclista
nombre:"Eduardo Ganchala"
edad: 21
kilometrosRecorridos: 3800
ciclista1:Ciclista
nombre:"Rigoberto Urán"
edad: 37
kilometrosRecorridos: 3500
```



TRABAJAR EN GRUPO DISCUTIR LA ACTIVIDAD, APRENDER POR IGUAL, EXPONER LA SIGUIENTE CLASE, NO OLVIDE DE SUBIR AL GITHUB EL CODIGO Y EL PDF