

UML y su aplicación en la Programación Orientada a Objetos en Java

1. ¿Qué es UML?

UML (Unified Modeling Language) es un **lenguaje de modelado estandarizado** que nos permite representar sistemas de software de forma **visual**.

No es un lenguaje de programación, sino una **herramienta de comunicación** que ayuda a:

- Representar cómo estará organizado un sistema.
- Comprender mejor el problema antes de escribir código.
- Explicar a otros cómo funciona el diseño (compañeros, clientes, profesores).

Podés pensar UML como los **planos de una casa**: antes de construir, necesitamos un diseño claro.

2. ¿Por qué usar UML?

- Facilita la **planificación** de proyectos.
- Evita errores comunes al escribir código directamente.
- Permite **documentar** sistemas, lo cual es clave en trabajos en equipo.
- Es un puente entre la **teoría de POO** y la implementación en **Java**.

3. Diagramas más importantes de UML

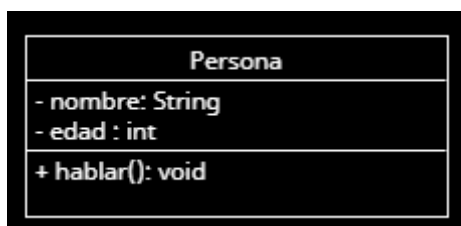
UML tiene muchos diagramas, pero en Programación 2 vamos a enfocarnos en los más usados en **POO con Java**:

3.1. Diagrama de Clases

El más relevante para nosotros. Representa:

- **Clases** y sus atributos.
- **Métodos** (operaciones).
- **Relaciones** entre clases (asociaciones, herencia, dependencias).

Ejemplo simple:



Traducción a Java:

```
public class Persona {  
  
    private String nombre;  
    private int edad;  
  
    public void hablar() {  
        System.out.println("Hola, soy una persona");  
    }  
}
```

3.2. Diagrama de Objetos

Muestra **instancias** de clases y sus valores concretos.

Ejemplo:

persona1: Persona

nombre = "Ana"

edad = 25

3.3. Diagramas de Casos de Uso

Muestran cómo los **usuarios (actores)** interactúan con el sistema.

Ejemplo: en una aplicación de música:

- Actor: Usuario
- Caso de uso: Reproducir canción, Crear lista de reproducción.

3.4. Diagramas de Secuencia

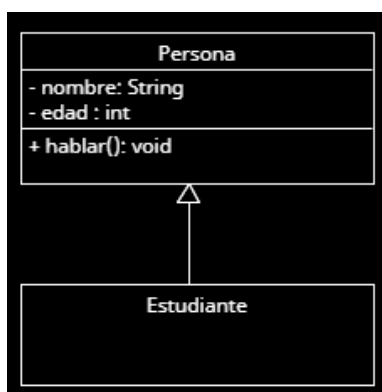
Representan cómo los objetos se envían **mensajes en el tiempo**.

Útiles para entender la dinámica del sistema.

4. UML aplicado a la Programación en Java

El paso natural después del modelado es implementar el diseño en Java.

Ejemplo de relación **herencia** en UML:



Código en Java:

```
public class Persona {  
  
    private String nombre;  
    private int edad;  
  
    public void hablar() {  
        System.out.println("Hola, soy una persona");  
    }  
}  
  
public class Estudiante extends Persona {  
  
    private String carrera;  
}
```

5. Herramientas para modelar UML

- **UMLetino** (recomendado en esta materia).
- Draw.io
- StarUML
- Lucidchart

Todas permiten dibujar diagramas fácilmente y exportarlos para tus proyectos.