



Clase 05

UML

Programación II



¿Qué es UML?

- UML = *Unified Modeling Language* (Lenguaje Unificado de Modelado).
- Es un **lenguaje gráfico estandarizado** para modelar software.
- No es un lenguaje de programación → es una herramienta de diseño.
- Se usa para planificar antes de programar, comunicar ideas entre programadores y documentar sistemas.

¿Por qué usar UML?

- Permite **visualizar** el sistema antes de implementarlo.
- Es un **lenguaje gráfico estandarizado** para modelar software.
- Sirve como **punto entre la idea y el código**.
- Reduce errores de diseño en proyectos grandes.

Tipos de Diagramas UML

- Estructurales:
 - Diagrama de clases (veremos este).
 - Diagramas de componentes, paquetes
- De comportamiento:
 - Casos de uso
 - Diagramas de secuencia
 - Actividades

Diagrama de Clase

- Representa la **estructura estática** del sistema.
- Incluye: Clases, atributos, métodos y relaciones entre las clases

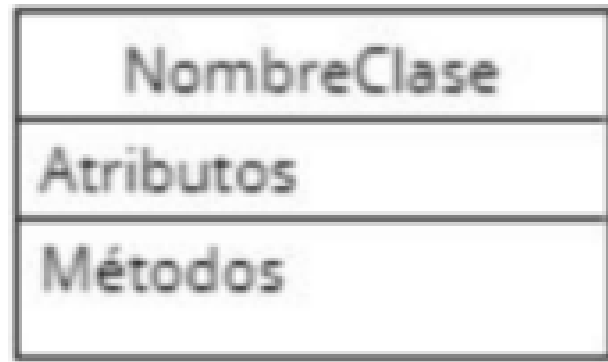
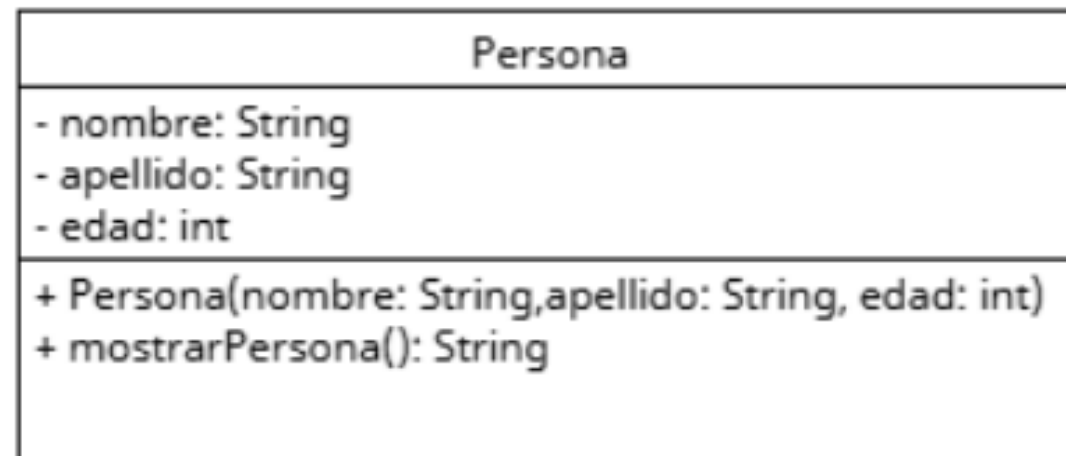


Diagrama de Clase

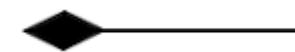
- También se representa el nivel de privacidad de los elementos.
- - para los privados
- + para los públicos
- # para los protegidos





Relaciones en UML (introducción)

- **Asociación:** una clase usa a otra.
- **Agregación:** una clase contiene objetos de otra, pero independientes
- **Composición:** una clase contiene objetos de otra, con dependencia fuerte.
- **Herencia:** una clase hija hereda de una clase padre.



¿Qué es StringBuilder?

- Una clase de Java para trabajar con cadenas modificables.
- A diferencia de String, que es inmutable, StringBuilder permite cambiar el contenido sin crear nuevos objetos.
- Se encuentra en el paquete `java.lang` (no hace falta importarlo).
- Más eficiente cuando hacemos muchas concatenaciones o modificaciones de texto.

Métodos más comunes de StringBuilder

- `append(String texto)` → agrega texto al final.
- `insert(int index, String texto)` → inserta texto en una posición.
- `delete(int start, int end)` → elimina caracteres en un rango.
- `reverse()` → invierte la cadena.
- `toString()` → devuelve el contenido como un String.

```
StringBuilder sb = new StringBuilder("Hola");  
sb.append(" mundo");  
System.out.println(sb); // Hola mundo
```

¿Cuándo usar StringBuilder?

- Cuando se requiere modificar muchas veces una cadena
- Armar un texto dinámico en un bucle.
- Concatenar datos antes de mostrarlos en consola.
- Procesar cadenas grandes donde la eficiencia es importante.

