



ANÁLISIS Y DISEÑO DE SOFTWARE

ELEMENTOS BÁSICOS DE UN DIAGRAMA DE CLASES

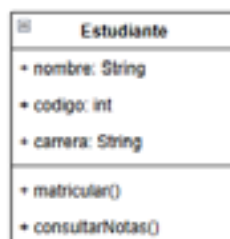
ELEMENTOS BÁSICOS DE UN DIAGRAMA DE CLASES

1. Clases

La clase es el componente principal del diagrama. Representa un conjunto de objetos con características y comportamientos comunes. En la notación UML, una clase se representa como un rectángulo dividido en tres compartimentos: el nombre de la clase, sus atributos y sus métodos (Teniente López et al., 2015).

Ejemplo:

Una clase llamada Estudiante puede incluir atributos como nombre, código y carrera, y métodos como matricular() o consultarNotas().



2. Atributos

Los atributos describen las propiedades o datos que posee una clase. Se anotan en el segundo compartimento del rectángulo y su formato incluye el nivel de visibilidad (+ público, - privado, # protegido), el nombre del atributo y su tipo.

Ejemplo:

- edad: int, indica que el atributo edad es privado y de tipo entero.

3. Métodos (Operaciones)

Los métodos representan el comportamiento o las acciones que una clase puede ejecutar. Se ubican en el tercer compartimento del rectángulo. Al igual que los atributos, indican visibilidad, nombre del método y, opcionalmente, parámetros y tipo de retorno (Teniente López et al., 2015).

Ejemplo:

+ calcularPromedio(): float señala un método público que retorna un valor tipo decimal.

4. Relaciones

Las relaciones permiten establecer cómo interactúan o se vinculan las clases entre sí. Los tipos más comunes son:

- **Asociación.** Representa una relación general entre dos clases. Puede tener multiplicidad (como 1, 0..*, etc.).

Ejemplo. Un Profesor se asocia con múltiples cursos.

- **Herencia (Generalización).** Define que una clase hija hereda atributos y métodos de una clase padre.

Ejemplo. Empleado hereda de Persona.

- **Agregación.** Indica una relación de “todo/parte” débil, donde las partes pueden existir independientemente del todo.

Ejemplo. Una Universidad agrega Facultades.

- **Composición.** Es una forma fuerte de agregación, donde las partes no pueden existir sin el todo.

Ejemplo. Un libro compone capítulos.

5. Interfaces y clases abstractas

Las interfaces definen un contrato que otras clases pueden implementar. Se representan con una clase estereotipada con <<interface>>.

Las clases abstractas no pueden ser instanciadas directamente y contienen métodos abstractos. Se escriben en cursiva o usando el estereotipo <<abstract>>.

Ejemplo. Una interfaz Imprimible con el método imprimir() puede ser implementada por múltiples clases como Factura o Reporte.



Este conjunto de elementos permite construir representaciones fieles del diseño de un sistema orientado a objetos, facilitando el análisis, la documentación y la comunicación entre los distintos miembros del equipo de desarrollo. Al emplear diagramas de clases con precisión, se mejora la comprensión del sistema y se sientan las bases para una implementación coherente y bien estructurada.