

Clase 2: Diagrama de clases y de Flujo

Francisco Riquelme



UML

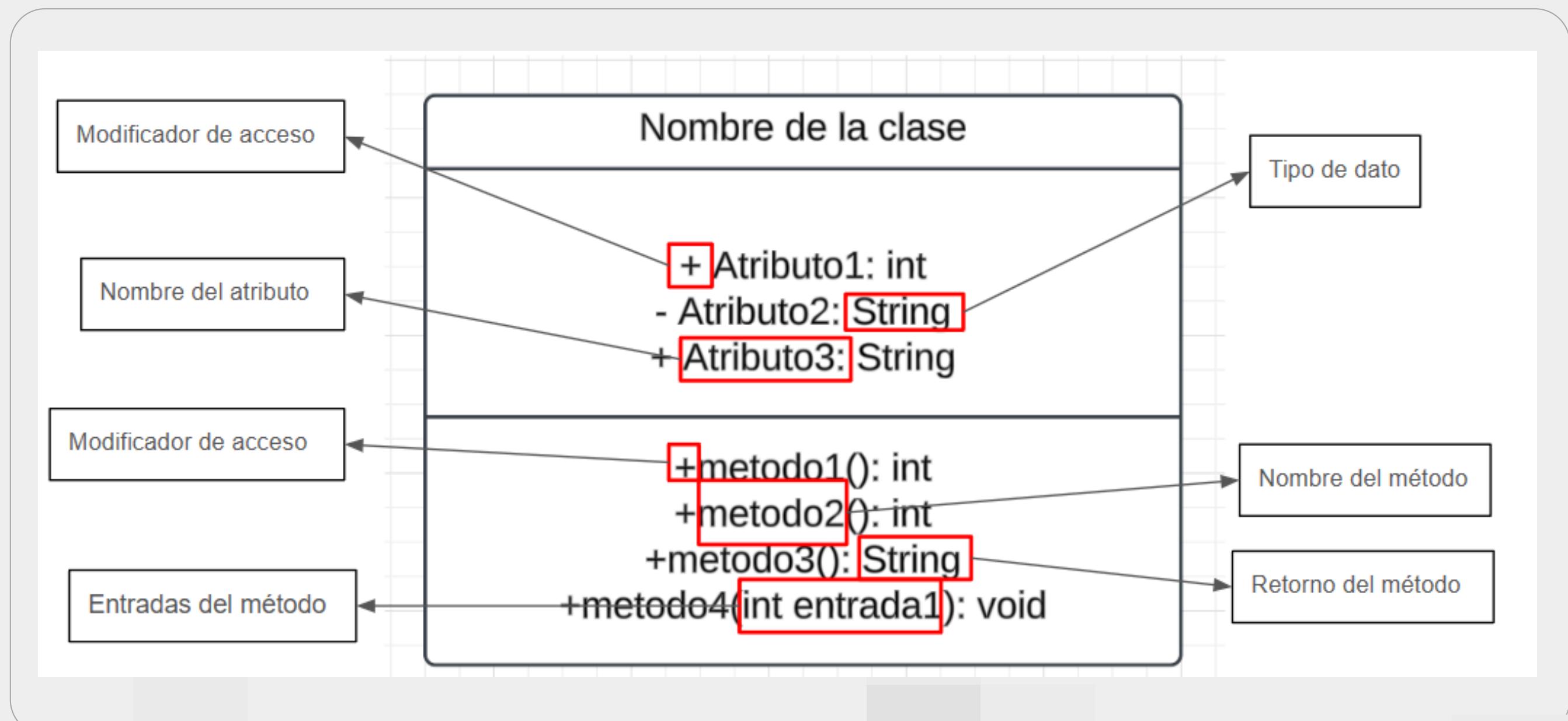
(Unified Modeling Language)

UML (Unified Modeling Language)

- Es un lenguaje estándar para modelar software orientado a objetos.
- Su objetivo fue unificar y crear un lenguaje común para el desarrollo de software
- Existen múltiples tipos de diagramas en este lenguaje
 - Diagrama de Clases
 - Modelo Entidad-Relación, Modelo de Datos
 - Diagrama de Arquitectura
 - Diagrama de Componentes

Diagrama de clases

Estructura



Cardinalidad/Multiplicidad

La cardinalidad define cuántas instancias de una clase pueden estar relacionadas con otra.

Ejemplos:

- En un videojuego un jugador puede tener muchas armas (*), pero cada personaje solo puede tener una mochila (1).
- En una red social, un usuario puede tener muchos amigos (*), pero cada perfil sólo puede estar asociado a una cuenta de correo (1).

En un sistema bancario, un cliente puede tener múltiples cuentas (*), pero cada cuenta pertenece solo a un cliente (1).

Relaciones

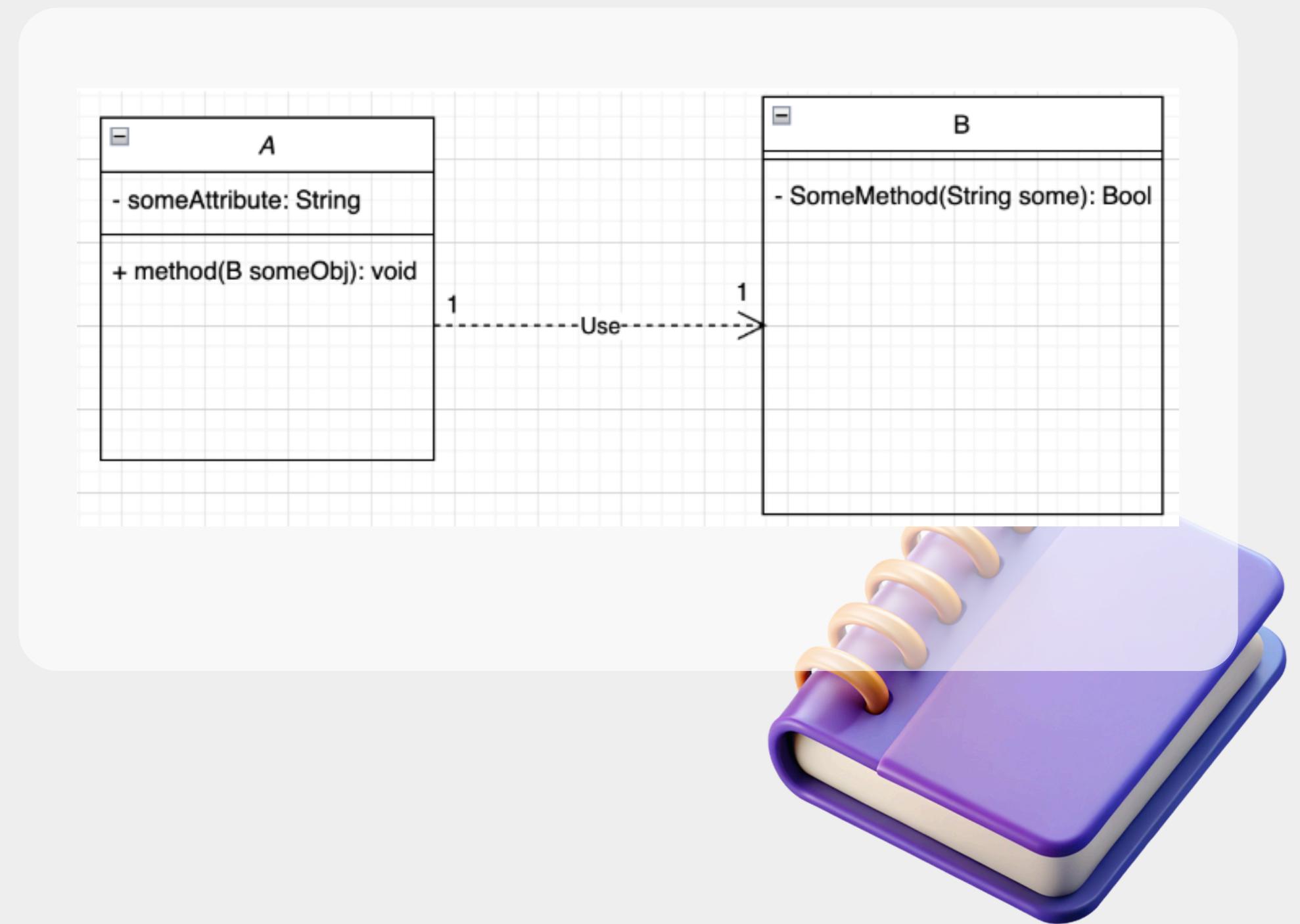
Dependencia.

Una dependencia es cuando:

- Una clase A utiliza de forma temporal otra clase B.
B no es parte de A y sólo es usado de forma breve por A.

Es un tipo de relación débil puesto que no hay relación permanente. La dependencia genera un grado menor de acople.

- Una dependencia se expresa a través del uso de una línea con puntos.

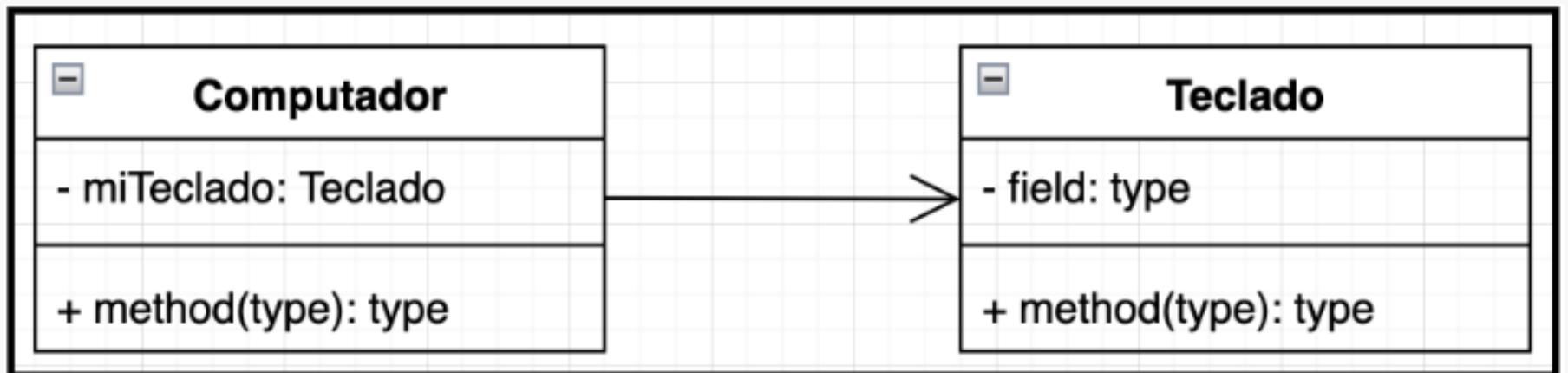




Asociación.

Una asociación es cuando una clase A tiene declarado como atributo una clase B. Luego, A puede usar B en alguno de sus métodos.

- Corresponde a una relación fuerte y a nivel de UML se representa mediante una línea sólida

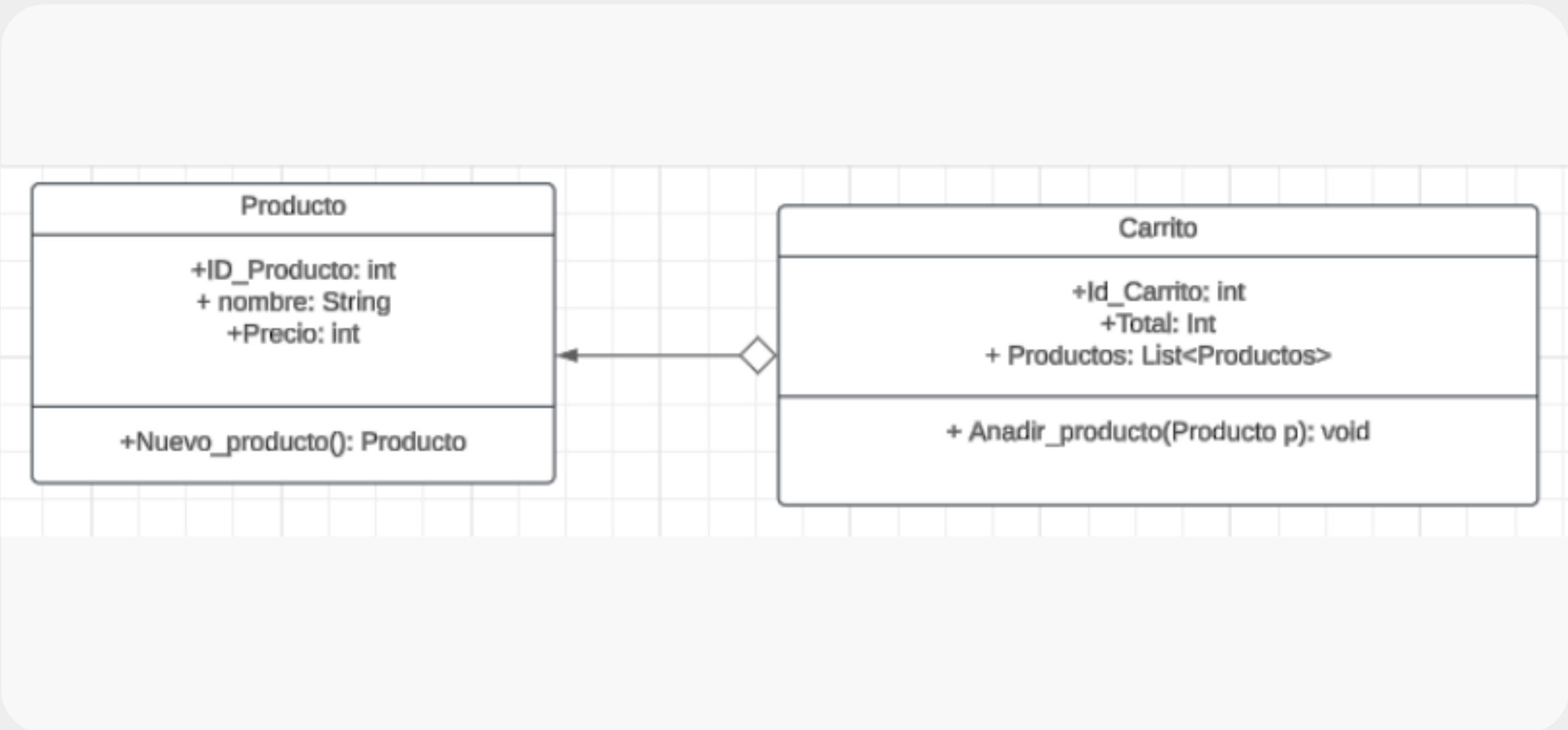




Agregación

En una agregación existe un “dueño/contenedor” que contiene otra clase, pero si este dueño deja de existir, la clase contenida puede seguir existiendo.

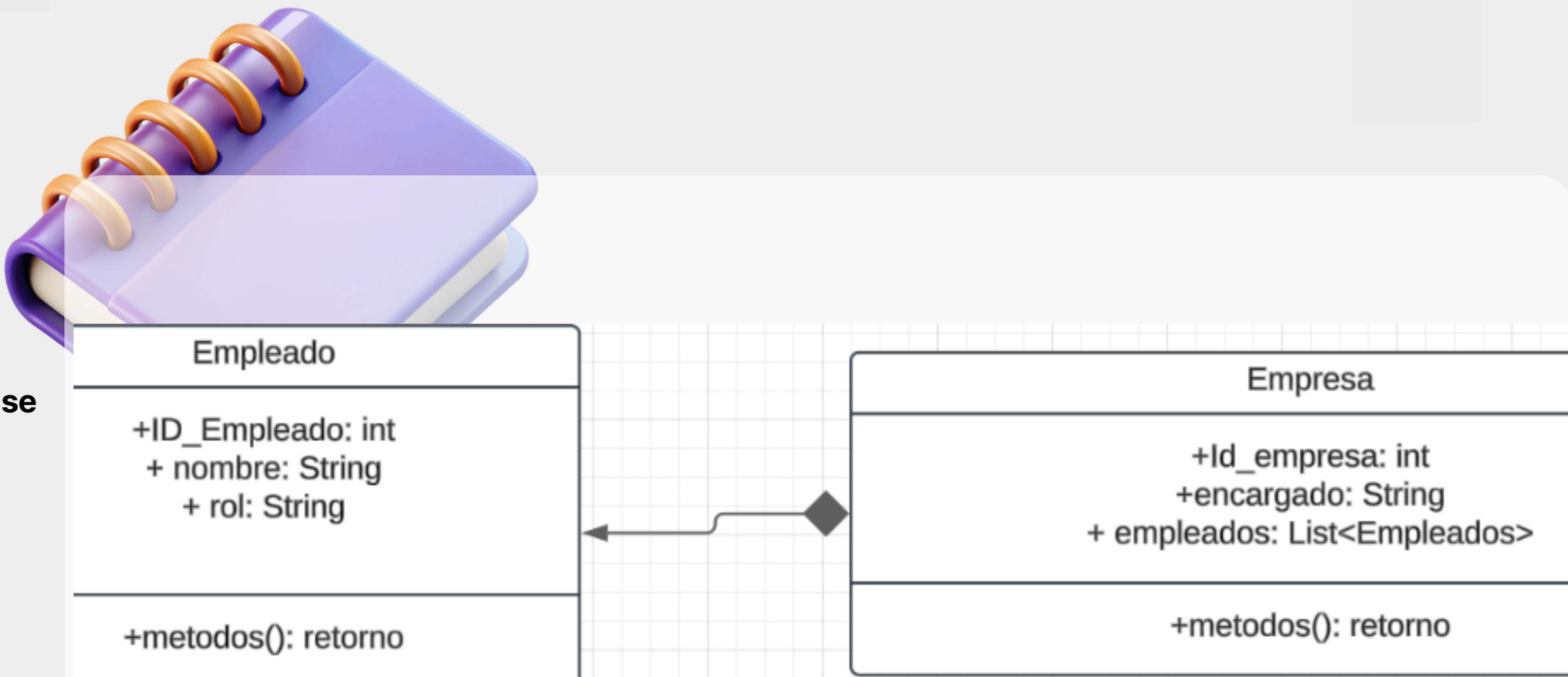
- Es una relación débil que se representa con un rombo vacío.
Relación de existencia débil (rombo no pintado).



Composición

En una composición existe un “dueño/contenedor” que contiene otra clase, pero si este dueño deja de existir, la clase contenida también deja de existir.

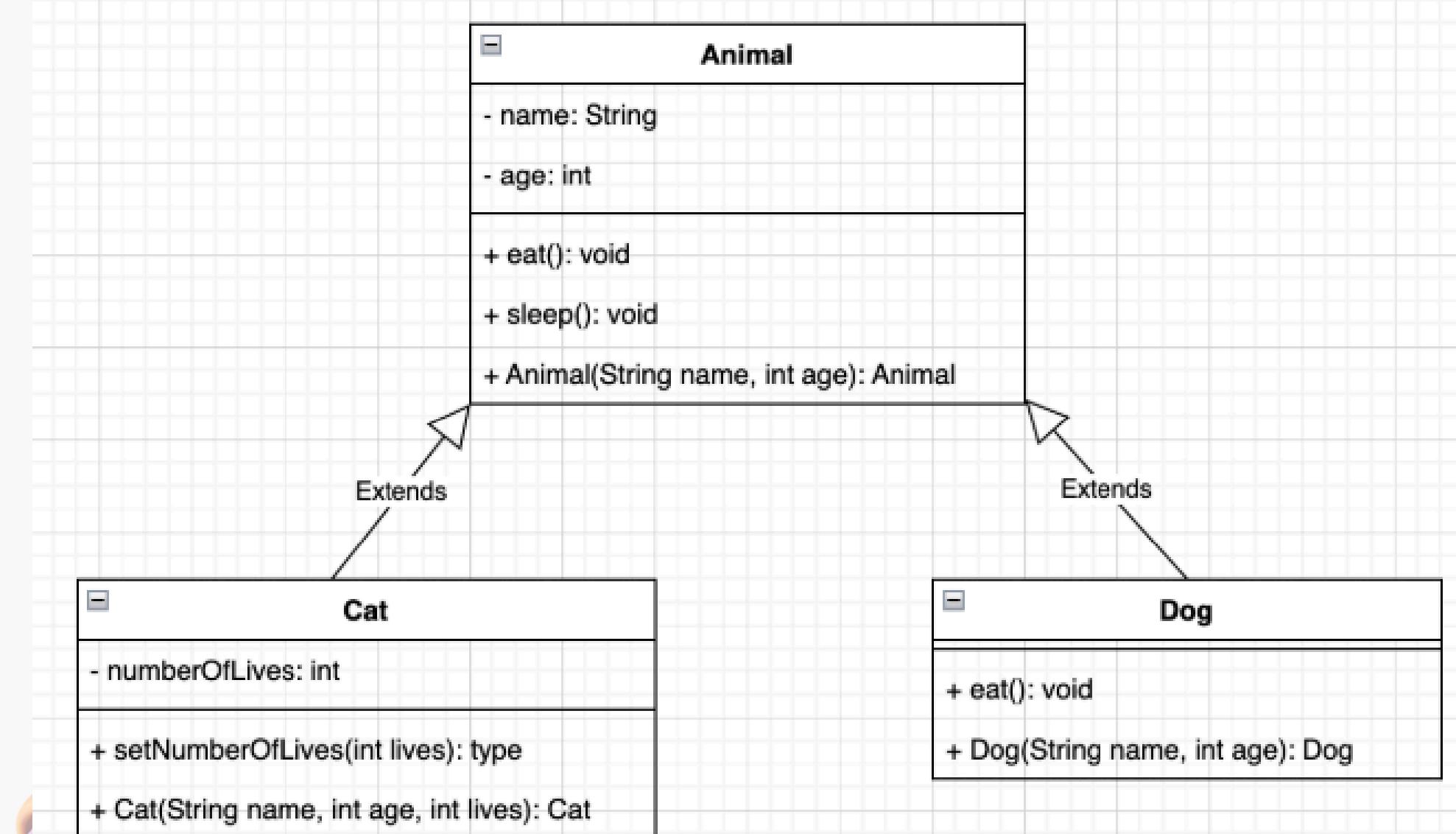
- Relación fuerte (rombo pintado).



Herencia (extends)

Implica dos tipos de clases, la clase padre es de la que se deriva la otra clase y la clase derivada (clase hija) hereda a la clase padre y puede usar los atributos y métodos definidos en la clase padre.

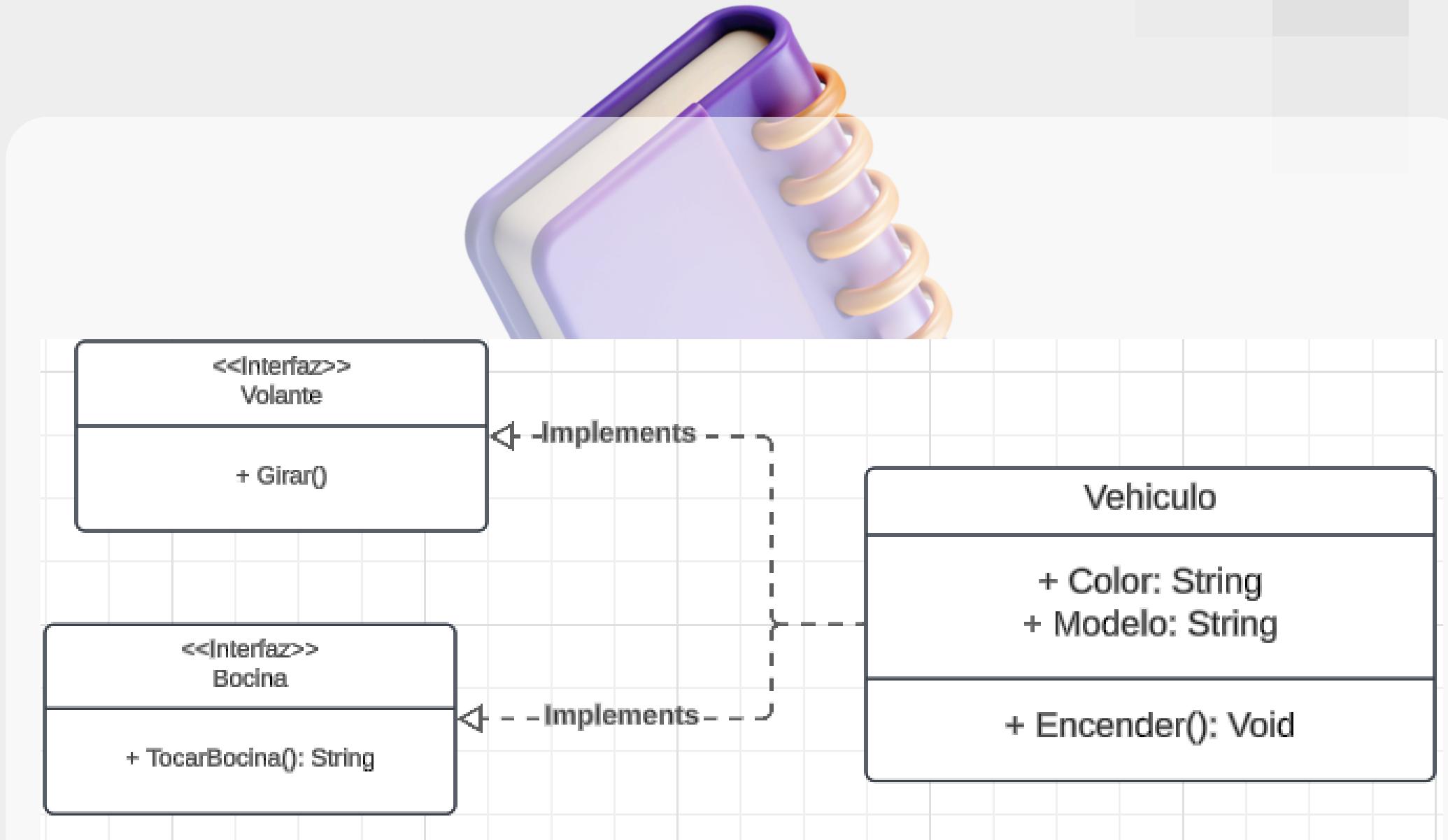
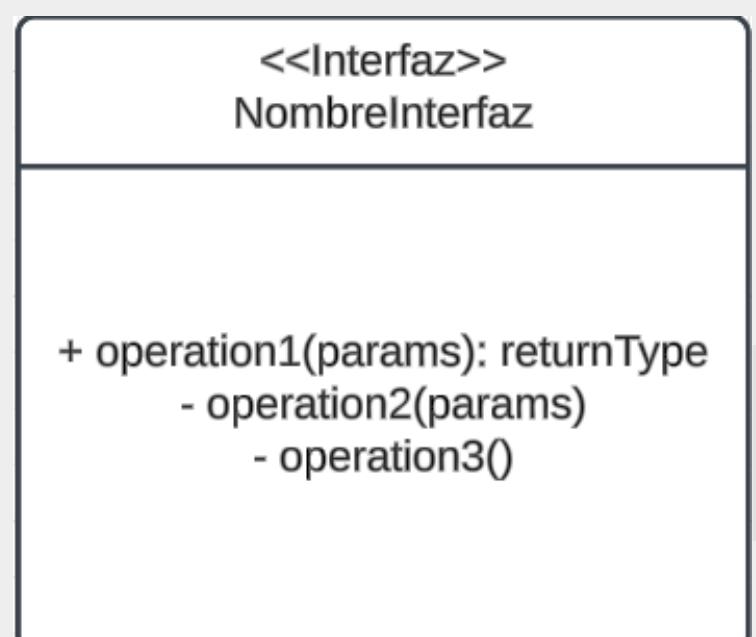
- Relación fuerte (Línea sólida y triangulo cerrado).
- No utiliza cardinalidad.



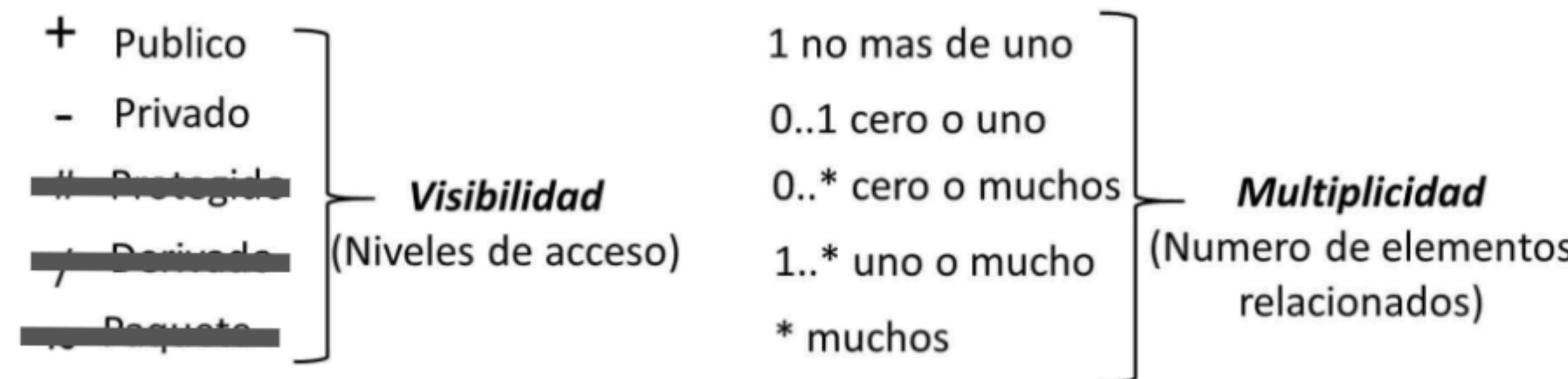
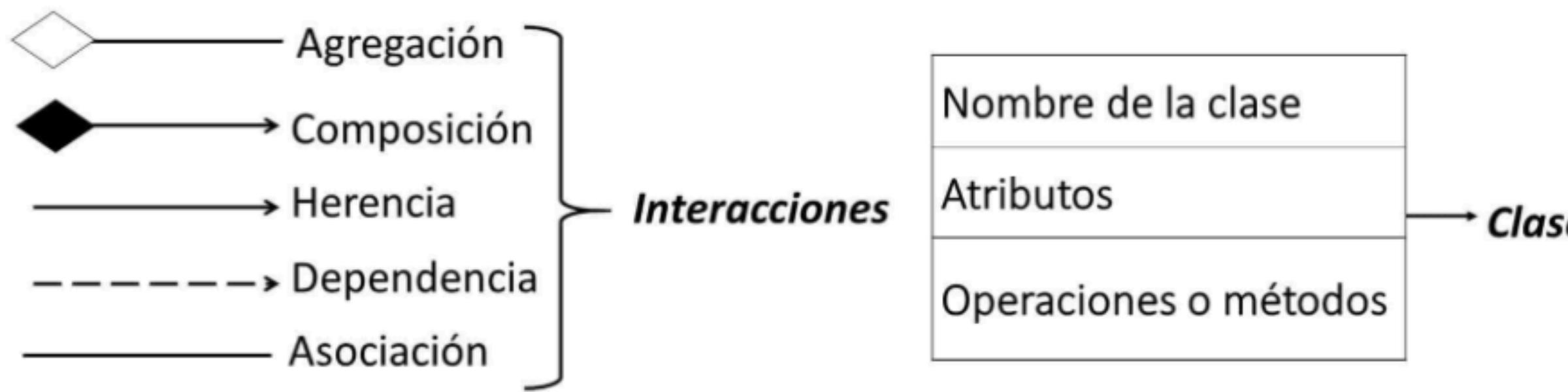
Interface (Implements)

Se utiliza para especificar un conjunto de métodos que una clase debe implementar.

- No utiliza cardinalidad.
- La clase debe implementar todos los métodos de su interface asociada



Elementos y símbolos en los diagramas de clases UML



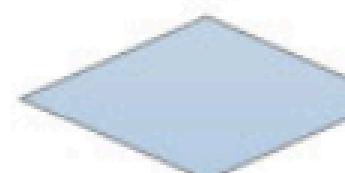
Diagramas de flujo

Los diagramas de flujo son una representación general de procesos.

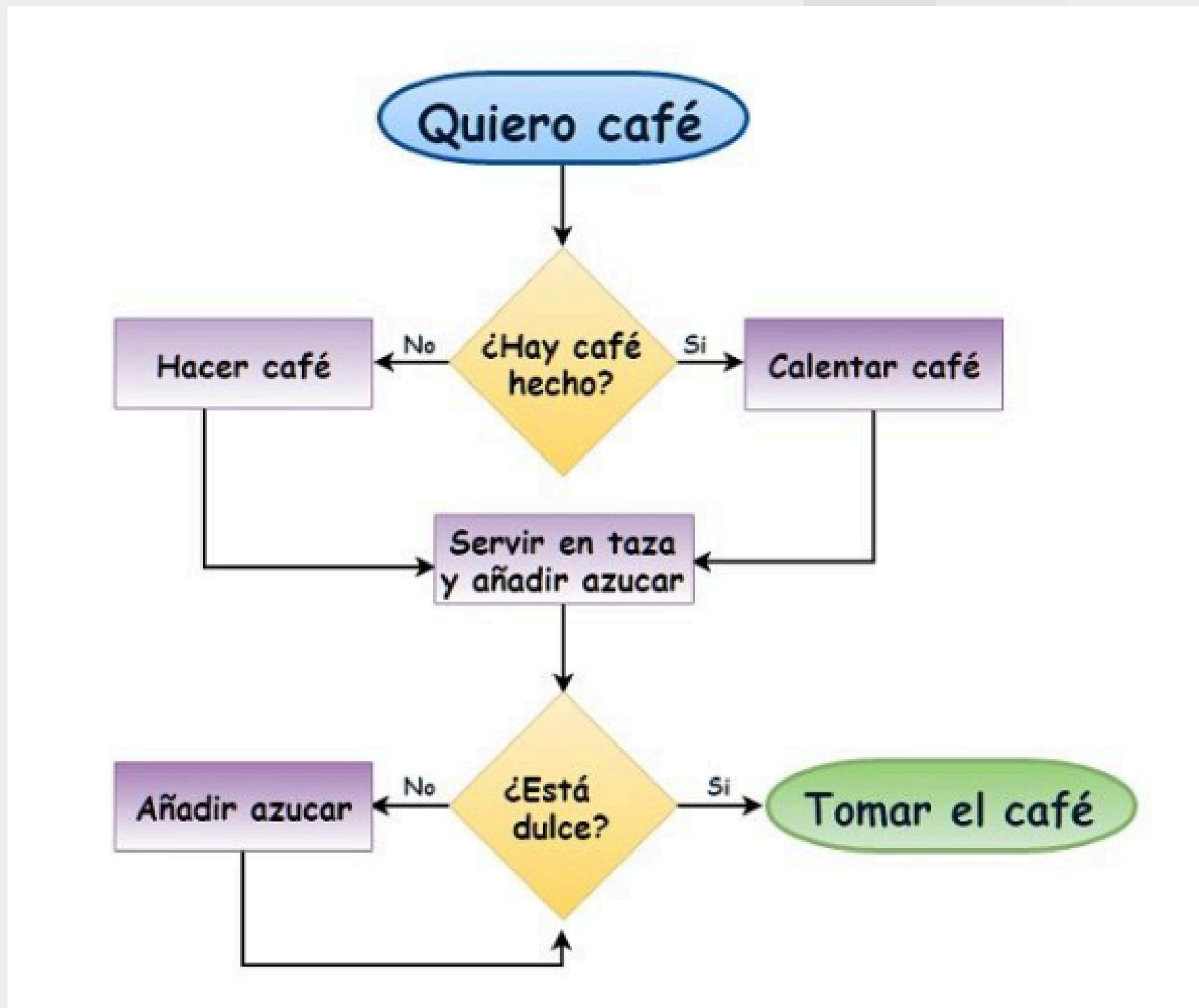
Dentro de la programación sirve para diseñar algoritmos antes de escribir código.



Componentes

Símbolo	Nombre	Función
	Inicio / Final	Representa el inicio y el final de un proceso
	Línea de Flujo	Indica el orden de la ejecución de las operaciones. La flecha indica la siguiente instrucción.
	Entrada / Salida	Representa la lectura de datos en la entrada y la impresión de datos en la salida
	Proceso	Representa cualquier tipo de operación
	Decisión	Nos permite analizar una situación, con base en los valores verdadero y falso

Ejemplo:



Actividad 1

Se desea diseñar un diagrama de clases sobre la información de las reservas de una empresa dedicada al alquiler de automóviles, teniendo en cuenta que:

Un determinado cliente puede tener en un momento dado hechas varias reservas.

De cada cliente se desean almacenar su DNI, nombre, dirección y teléfono. Además dos clientes se diferencian por un código único.

Una reserva la realiza un único cliente pero puede involucrar varios coches.

Es importante registrar la fecha de inicio y final de la reserva, el precio del alquiler de cada uno de los coches, los litros de gasolina en el depósito en el momento de realizar la reserva, el precio total de la reserva y un indicador de si el coche o los coches han sido entregados.

Todo coche tiene siempre asignado un determinado garaje que no puede cambiar. De cada coche se requiere la matrícula, el modelo el color y la marca.

Cada reserva se realiza en una determinada agencia.



Actividad 2

CJ ha caído de un edificio y recibió un golpe en su cabeza. Debido al golpe se le ha olvidado cómo sumar dos números. Nos ha pedido ayuda para desarrollar un diagrama de flujo para recordar cómo hacerlo. Con este diagrama de flujo, CJ tendrá todo bajo control y podrá sumar como un verdadero experto.



Actividad 3

Después de que CJ recuperó su habilidad para sumar, decidió que es hora de dar el siguiente paso: registrarse en una página web. Sin embargo, debido a su golpe en la cabeza, necesita ayuda para recordar cómo completar el proceso de registro. CJ quiere hacerlo bien para no cometer errores.

Nos solicitó crear un diagrama de flujo para recordar cómo registrarse en una página web que tiene los siguientes requisitos:

- 1- Solicitar datos del usuario: nombre, correo electrónico y contraseña.
- 2- Verificar que el correo sea válido (contiene @ y un dominio).
- 3- Comprobar si la contraseña es segura (mínimo 8 caracteres).
- 4- Revisar si el correo ya está registrado en la base de datos.

Si está registrado, mostrar un mensaje de error.

Si no está registrado, continuar con el proceso.

- 7- Guardar los datos del usuario en la base de datos.

- 6- Mostrar un mensaje de éxito y permitir al usuario iniciar sesión.

