

PROGRAMA DE ESTUDIOS

DESARROLLO DE SISTEMAS DE INFORMACIÓN

**ANÁLISIS Y DISEÑO
DE SISTEMAS**

Tema:

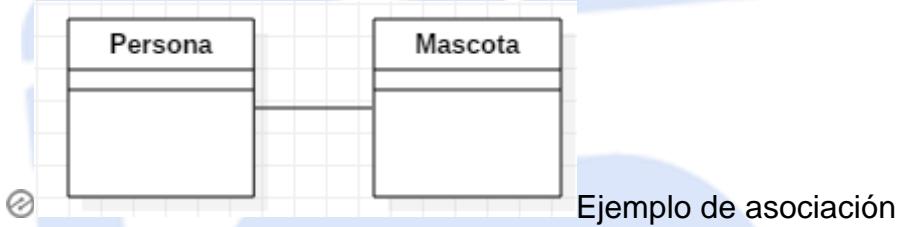
**DIAGRAMA DE CLASES - RELACIONES
ENTRE CLASES: AGREGACIÓN Y
COMPOSICIÓN.**

DIAGRAMA DE CLASES - RELACIONES ENTRE CLASES: AGREGACIÓN.

Asociación

Este tipo de relación es el más común y se utiliza para representar dependencia semántica. Se representa con una simple linea continua que une las clases que están incluidas en la asociación.

Un ejemplo de asociación podría ser: «Una mascota pertenece a una persona».

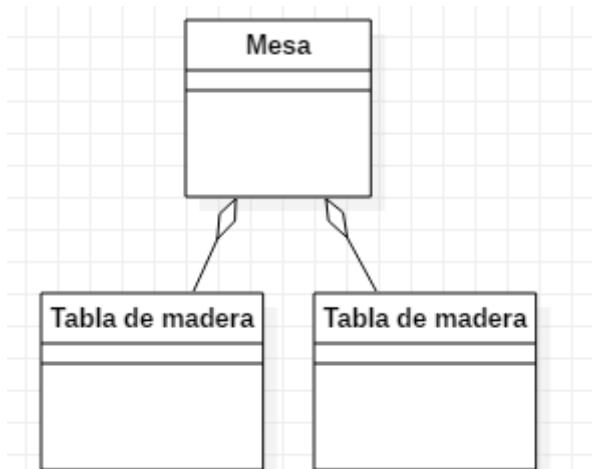


Agregación

Es una representación jerárquica que indica a un objeto y las partes que componen ese objeto. Es decir, representa relaciones en las que **un objeto es parte de otro**, pero aun así debe tener **existencia en sí mismo**.

Se representa con una línea que tiene un rombo en la parte de la clase que es una agregación de la otra clase (es decir, en la clase que contiene las otras).

Un ejemplo de esta relación podría ser: «Las mesas están formadas por tablas de madera y tornillos o, dicho de otra manera, los tornillos y las tablas forman parte de una mesa». Como ves, el tornillo podría formar parte de más objetos, por lo que interesa especialmente su abstracción en otra clase.



Ejemplo de agregación

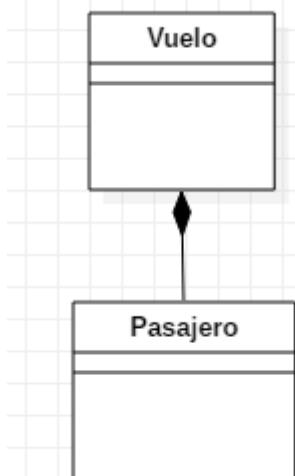
Composición

La composición es similar a la agregación, representa una **relación jerárquica entre un objeto y las partes que lo componen, pero de una forma más fuerte**. En este caso, los elementos que forman parte no tienen sentido de existencia cuando el primero no existe. Es decir, cuando el elemento que contiene los otros desaparece, deben desaparecer todos ya que no tienen sentido por sí mismos sino que dependen del elemento que componen. Además, suelen tener los mismos tiempo de vida. Los componentes no se comparten entre varios elementos, esta es otra de las diferencias con la agregación.



Se representa con una linea continua con un rombo relleno en la clase que es compuesta.

Un ejemplo de esta relación sería: «Un vuelo de una compañía aerea está compuesto por pasajeros, que es lo mismo que decir que un pasajero está asignado a un vuelo»



Ejemplo de composición

Diferencia entre agregación y composición

La diferencia entre agregación y composición es semántica, por lo que **a veces no está del todo definida**. Ninguna de las dos tienen análogos en muchos lenguajes de programación (como por ejemplo Java).

Un «agregado» representa **un todo que comprende varias partes**; de esta manera, un Comité es un agregado de sus Miembros. Una reunión es un agregado de una agenda, una sala y los asistentes. En el momento de la implementación, esta relación no es de contención. (Una reunión no contiene una sala). Del mismo modo, las partes del agregado podrían estar haciendo otras cosas en otras partes del programa, por lo que podrían ser referenciadas por varios objetos que nada tienen que ver. En otras palabras, no existe una diferencia de nivel de implementación entre la agregación y una simple relación de «usos». En ambos casos, un objeto tiene referencias a otros objetos. Aunque no existe una diferencia en la implementación, definitivamente vale la pena capturar la relación en el diagrama UML, tanto porque ayuda a comprender mejor el modelo de dominio, como porque puede haber problemas de implementación que pueden pasar desapercibidos. Podría permitir relaciones de acoplamiento más estrictas en una agregación de lo que haría con un simple «uso», por ejemplo.

La composición, por otro lado, implica un **acoplamiento aún más estricto que la agregación**, y definitivamente implica la contención. El requisito básico es que, si una clase de objetos (llamado «contenedor») se compone de otros objetos (llamados «elementos»), entonces los elementos aparecerán y también serán destruidos como un efecto secundario de crear o destruir el contenedor. Sería raro que un elemento no se declare como privado. Un ejemplo podría ser el nombre y la dirección del Cliente. Un cliente sin nombre o dirección no tiene valor. Por la misma razón, cuando se destruye al cliente, no tiene sentido mantener el nombre y la dirección. (Compare esta situación con la agregación, donde destruir al Comité no debe causar la destrucción de los miembros, ya que pueden ser miembros de otros Comités).

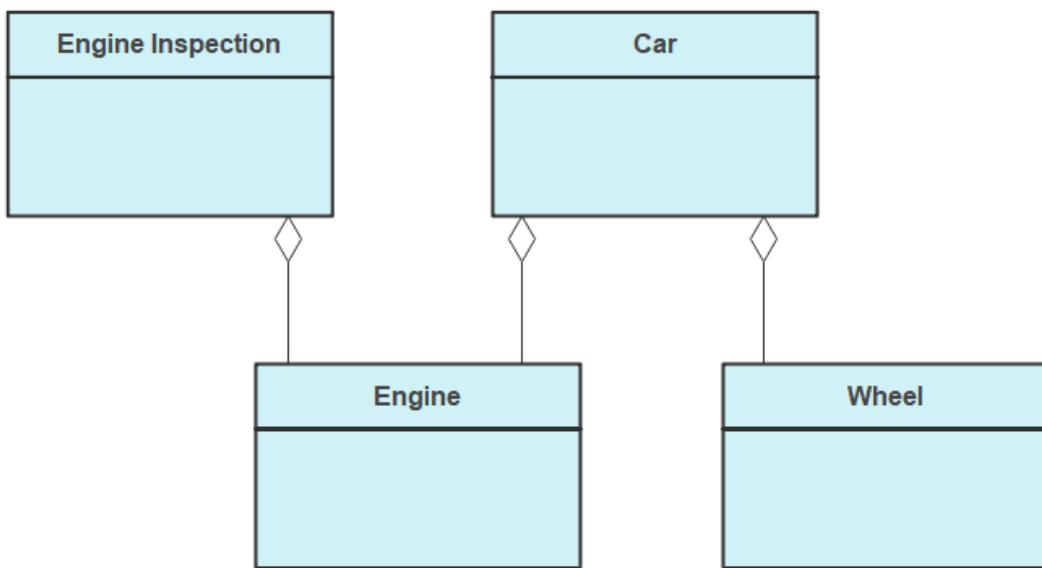
Agregación VS Composición

La Agregación y la Composición son dos subconjuntos de la asociación; sin embargo, la **composición** es un subconjunto más de la relación de agregación. En palabras sencillas, la **agregación** puede definirse como la situación en la que los objetos de una clase pueden acceder a las entidades de otra clase. Mientras que la composición representa una relación bilateral y el objeto forma parte de un solo compuesto a la vez.

Agregaciones Representadas en un Diagrama UML

Agregación es una asociación en la que las entidades se configuran juntas para crear un objeto más complejo. A través de ella se describe un grupo de objetos y su vinculación común. Define un único punto de control, denominado agregado, y el grupo de objetos que representa el conjunto.

En un diagrama UML, los datos fluyen desde el agregado (clase padre) hacia el conjunto (clase hija) con una forma de diamante (sin rellenar) cerca de la primera clase. El clasificador de piezas, o las entidades de ensamblaje, pueden estar vinculadas a más de un agregado y existir de forma independiente.

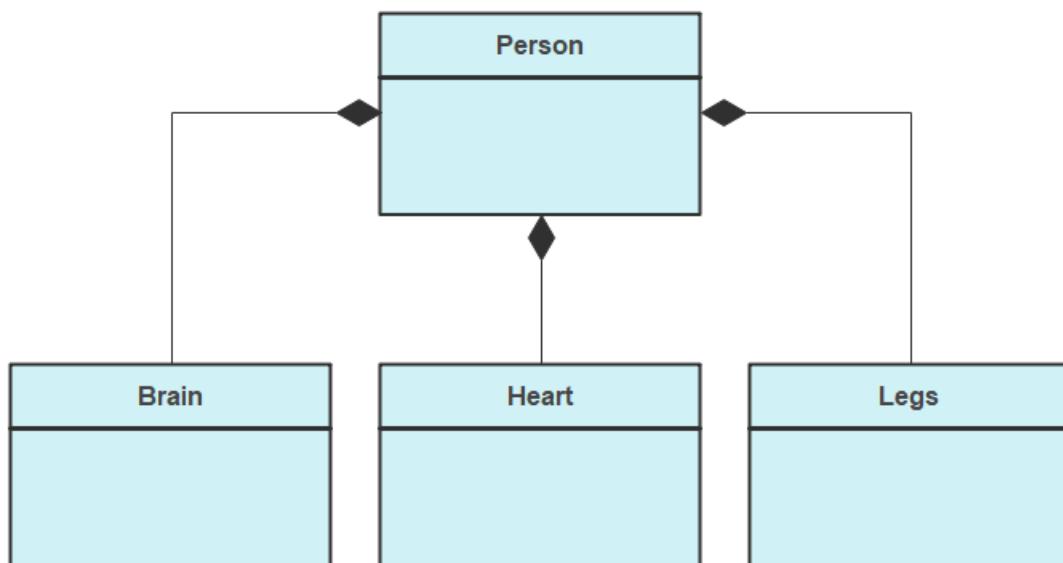


Este ejemplo muestra que el coche siempre necesita una rueda y un motor para funcionar correctamente, mientras que el motor y la rueda no siempre necesitan un coche. Pueden utilizarse con un autobús, una bicicleta, máquinas, un cortacésped y mucho más. Del mismo modo, el motor es necesario para una inspección de motores, pero el motor puede existir como entidad independiente y estar asociado a otros vehículos también.

Composiciones Representadas en un Diagrama UML

Las composiciones son también un tipo de asociación que representa una relación bidireccional y altamente dependiente entre dos clases diferentes. Si la composición se elimina, las otras entidades vinculadas con ella también se eliminan.

Representa la dependencia entre el padre y el hijo. En un diagrama UML, dibuja una línea entre la clase padre y la clase hija con una forma de diamante rellena cerca de la clase padre.



Como se ilustra en este ejemplo, una persona tiene piernas, un corazón y un cerebro. Si el punto central de este diagrama, la persona, se hiere y muere, todas las entidades compuestas también serán descartadas. A diferencia de la agregación, las entidades no pueden existir de forma independiente y son una "parte de" la persona.

Agregación	Composición
Un subconjunto de la asociación	Un subconjunto de la agregación.
Un tipo de asociación débil.	Un tipo de asociación fuerte
Los objetos enlazados son independientes entre sí.	Los objetos enlazados son altamente dependientes entre sí.
Se representa con una línea sólida y una punta de flecha vacía.	Representado por una línea sólida con una punta de flecha rellena.
La agregación se define como una relación "tiene-un".	La composición se define como una relación "parte-de".

FUENTE:

- <https://diagramasuml.com/diagrama-de-clases/>
- <https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/diagrama-de-clases/>
- <https://creately.com/blog/es/diagramas/tutorial-de-diagrama-de-clases/>
- <http://sel.unsl.edu.ar/licenciatura/ingsoft2/UML-DiagramaClaseObjeto.pdf>
- <https://www.edrawsoft.com/es/article/uml-aggregation-vs-composition.html>
- <https://www.seas.es/blog/informatica/agregacion-vs-composicion-en-diagramas-de-clases-uml/>
- BURCH, John; GRUDNISKY, Gary. "Diseño de Sistemas de Información", Grupo Noriega editores.
- SENN, James A. "Análisis y diseño de sistemas de información", 2da. ed., McGraw-Hill.

