Relaciones de clases en UML

DigitalHouse >
Coding School

Certified Tech Developer
The Ultimate Degree

Índice

1 Relación de asociación

Relación de asociación

La relación de asociación conocida como una relación del tipo **“tiene un”** se establece cuando un objeto de una clase colabora con uno o más objetos de otra clase.

tiene un

Relación de Asociación

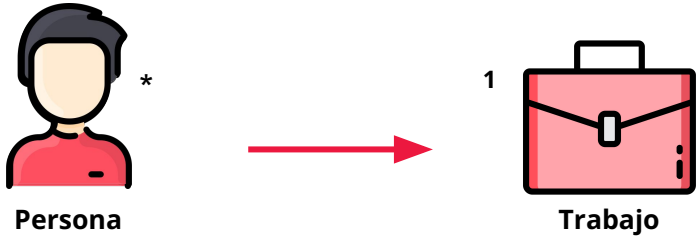
En este diagrama diremos que la clase Persona conoce a una instancia de la clase Mascota, de ahí que la dirección de la flecha es de la clase Persona hacia clase Mascota.

Esta relación representa que Persona tiene un atributo del tipo Mascota.

2 | Navegación

Navegación

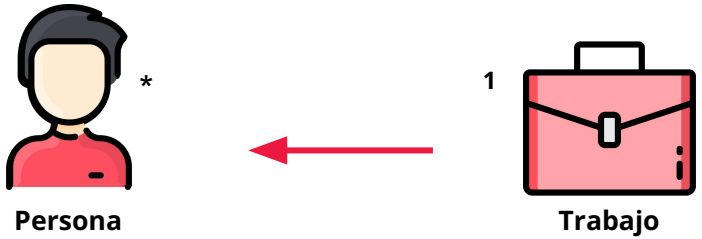
Cuando una asociación lleva una flecha indica una dirección de recorrido (de navegación). Implica que es posible para un objeto en un extremo acceder al objeto del otro extremo porque el primero contiene referencias específicas a este último (al que apunta la flecha), no siendo cierto en el sentido contrario.



En este ejemplo diremos que es la Persona la que tiene un atributo (objeto) de la clase Trabajo.

Navegación

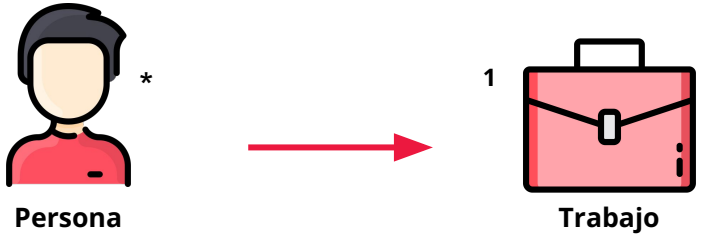
Recordá que siempre depende del contexto.
Si lo que necesitamos es por cada trabajo saber todas las personas que trabajan allí, entonces, el sentido sería al revés.



3 Multiplicidad o cardinalidad

Multiplicidad o cardinalidad

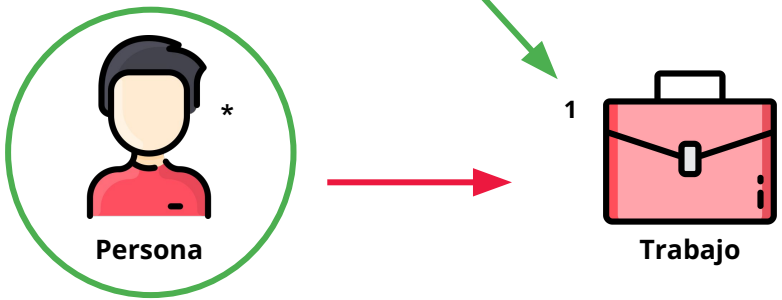
La **multiplicidad** también llamada **cardinalidad** especifica el número de instancias de una clase que puede estar relacionadas con una única instancia de una clase asociada. La multiplicidad limita el número de objetos relacionados.



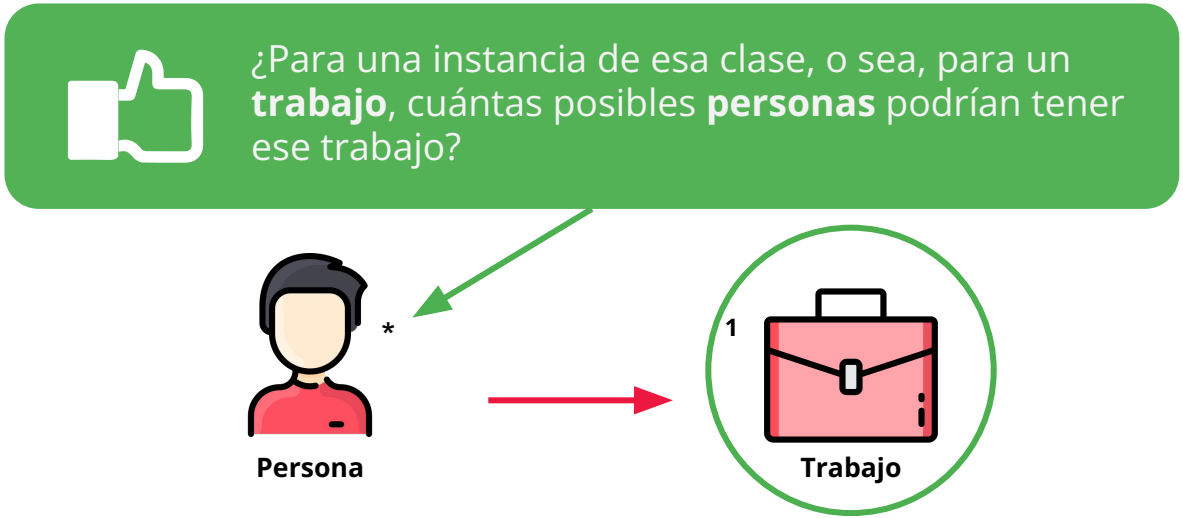
Persona tiene un solo trabajo y ese trabajo es realizado por muchas personas. Como verás el “muchos” lo representamos con un “*” o con una N.

Para establecer las multiplicidades, primero, nos paramos en una de las clases, por ejemplo, la **clase Persona** y paso siguiente debemos hacernos la siguiente pregunta:

¿Para una instancia de esa clase, en este caso de la clase **Persona**, cuántas posibles instancias podría tener de la clase a la que está asociada, en este caso **Trabajo**?



Luego nos paramos en la otra clase, en nuestro caso la clase **Trabajo** y nos debemos hacer la misma pregunta

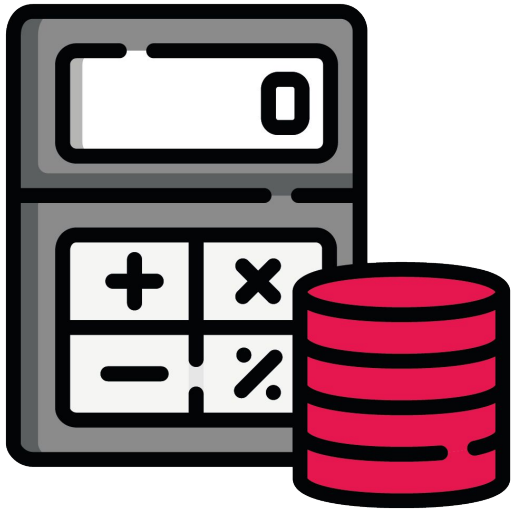


La multiplicidad depende de suposiciones y —como venimos nombrando— muchas veces, del **contexto del problema**. Tener poca información del contexto suele hacer incierta la multiplicidad.

Por ejemplo, la asociación que acabamos de ver entre Persona y Trabajo ¿es (1 a muchos) o (muchos a muchos)? **Esto depende del contexto.**

Una aplicación para el cálculo de impuestos podría permitir que una **persona trabajase en múltiples trabajos**.

Por otra parte, un sindicato de trabajadores del gremio del automóvil que mantuviera registros de sus **afiliados** trabajadores podría considerar **irrelevantes los segundos trabajos**.



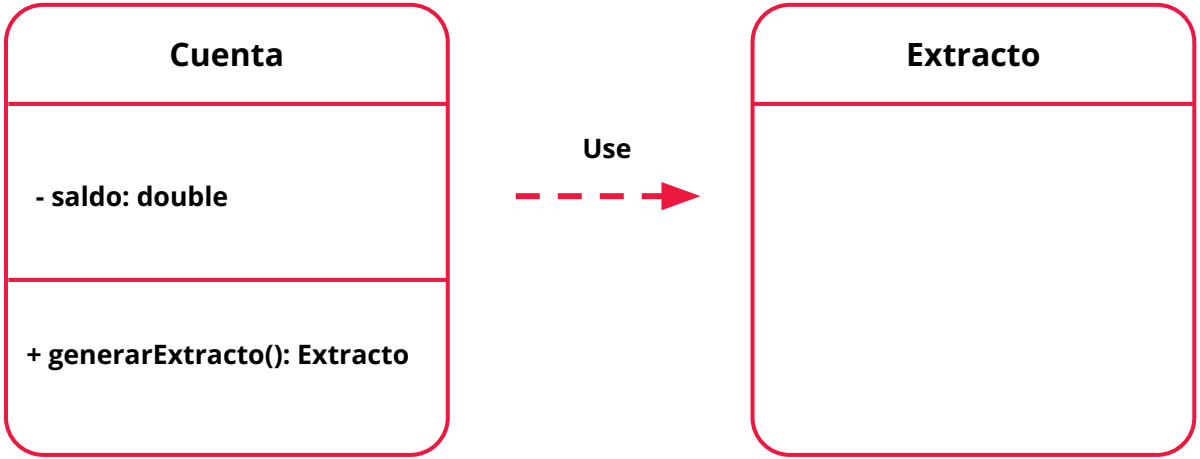
4 | Relación de uso

Relación de uso

Una **relación de uso** es un tipo de asociación que como lo indica su nombre es una relación del tipo **“usa un”**. La particularidad frente al otro tipo de asociación **“tiene un”** es que no hay una referencia de una clase a la otra, sino que en este caso, la relación se da porque hay algún método que devuelve o recibe como parámetro una variable que es del tipo de la otra clase.

usa un

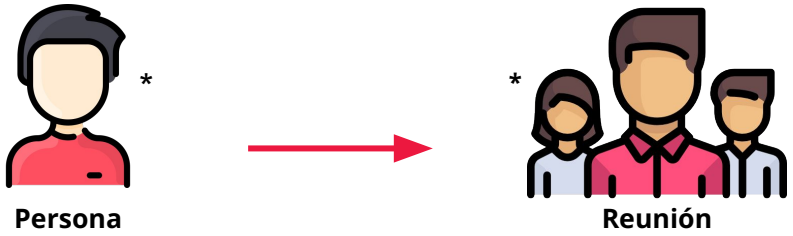
En el ejemplo a continuación la clase **Cuenta** tiene un **método** que **devuelve un Extracto**, pero no necesita tener una instancia Extracto dentro de la Cuenta.



5 | Ejemplos

Mucho a muchos

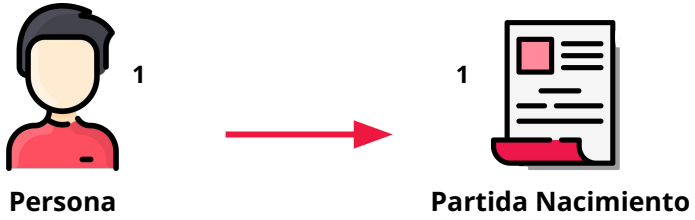
Una **persona** tiene **muchas reuniones** y en una misma **reunión** participan **muchas personas**. A diferencia del modelo relacional donde las relaciones muchos a muchos se transforman en una nueva entidad en el modelo orientado a objetos es posible tener relaciones mucho a muchos.



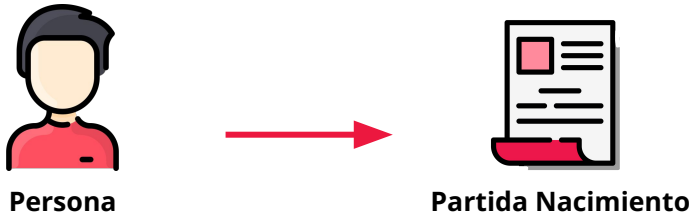
Es posible tener relaciones “mucho a muchos”.

Uno a uno

Una **persona** tiene **una** sola **partida de nacimiento** y una **partida de nacimiento** es de **una única persona**.

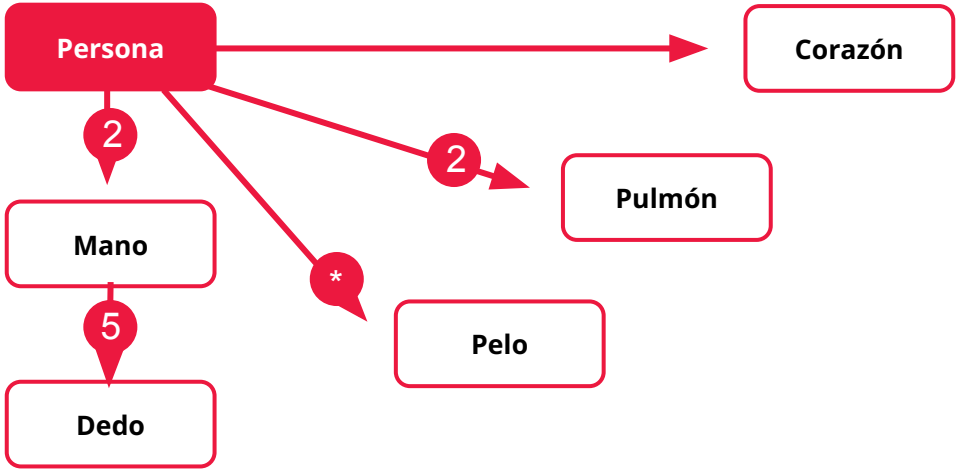


Vale aclarar que cuando la multiplicidad es 1 no se suele escribir en el diagrama.



Uno a muchos o muchos a uno

También podemos tener multiplicidades con cantidad de instancias limitadas en el ejemplo que vemos a continuación una **persona** puede tener **2 pulmones** y **cada pulmón es de una única persona** y, como vimos en el ejemplo anterior, no hay que olvidar que si no escribimos nada en la multiplicidad se entiende que es 1.




DigitalHouse>
Coding School