

Diagrama de Clases:

Es un tipo de diagrama **estático** que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro. En un diagrama de clases se pueden distinguir principalmente dos elementos: clases y sus relaciones, (Flores Cueto & Bertolotti Zuñiga, 2013).

Se debe tener en cuenta los conceptos principales de: Clase, asociación, generalización, dependencia, realización interfaz.

Objetivo principal: describir las clases.

Vista estática:

Modela los conceptos del dominio de la aplicación, así como los conceptos internos inventados como parte de la implementación de la aplicación.

Es estática porque no describe el comportamiento del sistema dependiente del tiempo, que se describe en otras vistas.

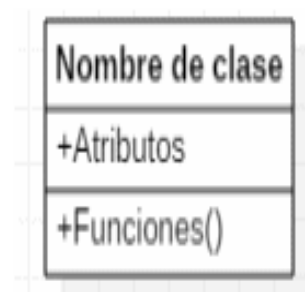
Los componentes principales son: las clases y sus relaciones: asociación, generalización y varias clases de dependencia, tales como: realización y uso.

Clase:

Elemento principal del diagrama y representa, como su nombre indica, una clase dentro del paradigma de la orientación a objetos, que describe un concepto del dominio de la aplicación o de la solución de la aplicación.

Está compuesta por tres elementos: nombre de la clase, atributos, funciones. Estos elementos se incluyen en la representación (o no, dependiendo del nivel de análisis).

Para representar la clase con estos elementos se utiliza una caja que es dividida en tres zonas utilizando para ello líneas horizontales:



Notación de una clase

La primera de las zonas se utiliza para el nombre de la clase. En caso de que la clase sea abstracta se utilizará su nombre en cursiva.

La segunda de las zonas se utiliza para escribir los atributos de la clase, uno por línea y utilizando el siguiente formato:

visibilidad nombre_atributo : tipo = valor-inicial { propiedades }

De la misma manera que con los atributos, se suele simplificar indicando únicamente el nombre de la función y, en ocasiones, el tipo devuelto.

Tanto los atributos como las funciones incluyen al principio de su descripción la visibilidad que tendrá. Esta visibilidad se identifica escribiendo un símbolo y podrá ser:

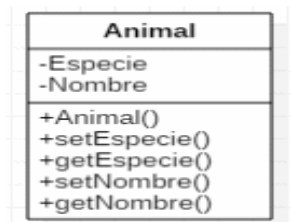
(+) Pública. Representa que se puede acceder al atributo o función desde cualquier lugar de la aplicación.

(-) Privada. Representa que se puede acceder al atributo o función únicamente desde la misma clase.

(#) Protegida. Representa que el atributo o función puede ser accedida únicamente desde la misma clase o desde las clases que hereden de ella (clases derivadas).

Estos tres tipos de visibilidad son los más comunes. No obstante, pueden incluirse otros en base al lenguaje de programación que se esté usando (no es muy común). Por ejemplo: (/) Derivado o (~) Paquete.

Un ejemplo de clase podría ser el siguiente:

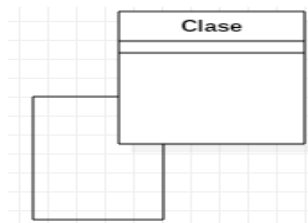


Ejemplo de una clase

En caso de que un atributo o función sea estático, se representa en el diagrama subrayando su nombre.

Una característica estática se define como aquella que es compartida por cada clase y no instanciada para cada uno de los objetos de esa clase. Es un concepto muy común.

Relaciones:



Relación reflexiva

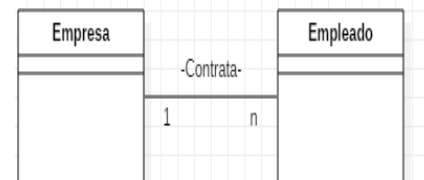
Una relación identifica una dependencia. Esta dependencia puede ser entre dos o más clases (más común) o una clase hacia sí misma (menos común, pero existen), este último tipo de dependencia se denomina dependencia reflexiva.

Las relaciones se representan con una línea que une las clases, esta línea variará dependiendo del tipo de relación

Las relaciones en el diagrama de clases tienen varias propiedades, que dependiendo la profundidad que se quiera dar al diagrama se representarán o no. Estas propiedades son las siguientes:

Multiplicidad. Es decir, el número de elementos de una clase que participan en una relación. Se puede indicar un número, un rango... Se utiliza n o $*$ para identificar un número cualquiera.

Nombre de la asociación. En ocasiones se escribe una indicación de la asociación que ayuda a entender la relación que tienen dos clases. Suelen utilizarse verbos como por ejemplo: «Una empresa contrata a n empleados»

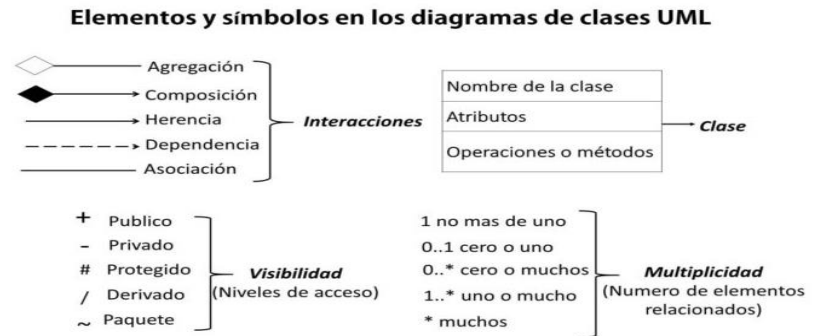


Ejemplo de relación Empresa-Empleado

Tipos de relaciones

Un diagrama de clases incluye los siguientes tipos de relaciones:

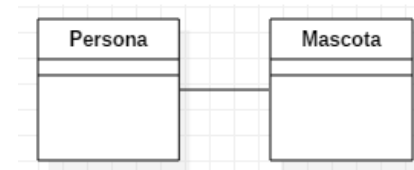
- Asociación.
- Agregación.
- Composición.
- Dependencia.
- Herencia.



Asociación

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

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



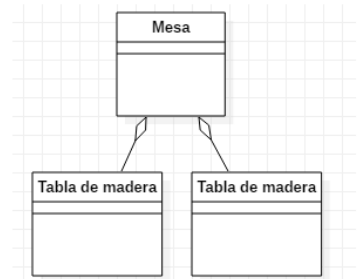
Ejemplo de asociación

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.



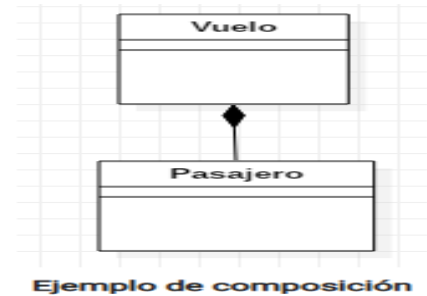
Ejemplo de agregación

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 tiempos de vida. Los componentes no se comparten entre varios elementos, esta es otra de las diferencias con la agregación.

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

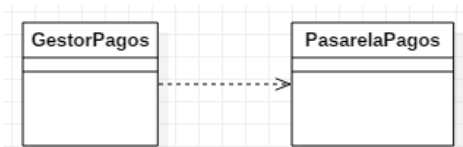
Un ejemplo de esta relación sería: «Un vuelo de una compañía aérea está compuesto por pasajeros, que es lo mismo que decir que un pasajero está asignado a un vuelo»



Ejemplo de composición

Dependencia

Se utiliza este tipo de relación para representar que una clase requiere de otra para ofrecer sus funcionalidades. Es muy sencilla y se representa con una flecha discontinua que va desde la clase que necesita la utilidad de la otra flecha hasta esta misma.



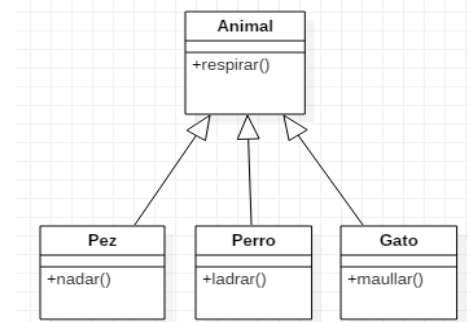
Ejemplo de dependencia

Herencia

Otra relación muy común en el diagrama de clases es la herencia. Este tipo de relaciones permiten que una clase (clase hija o subclase) reciba los atributos y métodos de otra clase (clase padre o superclase). Estos atributos y métodos recibidos se suman a los que la clase tiene por sí misma. Se utiliza en relaciones «es un».

Un ejemplo de esta relación podría ser: Un pez, un perro y un gato son animales.

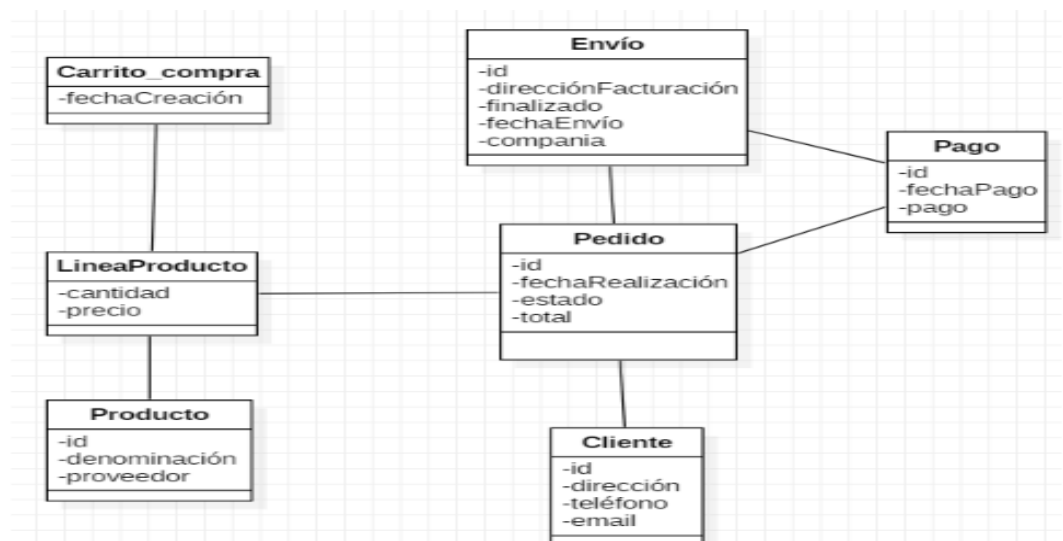
En este ejemplo, las tres clases (Pez, Perro, Gato) podrán utilizar la función respirar, ya que lo heredan de la clase animal, pero solamente la clase Pez podrá nadar, la clase Perro ladrar y la clase Gato maullar. La clase Animal podría plantearse ser definida abstracta, aunque no es necesario.



Ejemplo de herencia

Ejemplo:

Diagrama de clases de una tienda



Ejemplo de **clases** para un diagrama de clases de una tienda web

Ejemplo de clases con sus atributos que se podrían incluir en un diagrama de clases de una tienda online:

Usuario:	Producto:	Carrito de Compras:	Orden de compra:	Categoría:	Comentarios:
idUsuario: Identificador único del usuario. nombre: Nombre completo del usuario. correoElectronico: Dirección de correo electrónico del usuario. contraseña: Contraseña del usuario. dirección: Dirección de envío del usuario. métodoDePago: Método de pago preferido por el usuario.	idProducto: Identificador único del producto. nombre: Nombre del producto. descripción: Descripción detallada del producto. precio: Precio del producto. stock: Cantidad de unidades disponibles en el inventario.	idCarrito: Identificador único del carrito de compras. productos: Lista de productos que el usuario ha añadido al carrito. subtotal: Monto total del carrito antes de aplicar impuestos y descuentos. impuestos: Monto total de impuestos aplicados al carrito.	idOrden: Identificador único de la orden de compra. productos: Lista de productos comprados en la orden. subtotal: Monto total de la orden antes de aplicar impuestos y descuentos. impuestos: Monto total de impuestos aplicados a la orden. envío: Monto del costo de envío de la orden. total: Monto total de la orden incluyendo impuestos, descuentos y costo de envío.	idCategoría: Identificador único de la categoría. nombre: Nombre de la categoría.	idComentario: Identificador único del comentario. producto: Identificador del producto al que se refiere el comentario. usuario: Identificador del usuario que escribió el comentario. comentario: Contenido del comentario. fecha: Fecha de creación del comentario.

Ejemplo:

Diagrama de clases de la aplicación de la Taquilla de teatro que ha automatizado sus operaciones.

Este diagrama muestra varias clases importantes, tales como: **Cliente**, **Reserva**, **Entrada** y **Representación**.

Los **clientes** pueden tener muchas **reservas**, pero cada reserva es hecha por un **cliente**.

Las Reservas son de dos clases: **suscripción y reservas individuales**.

Ambos reservan entradas:

1. En un caso, solamente una entrada;
2. En el otro caso, varias entradas.

Cada entrada es parte de una suscripción o de una reserva individual, pero no de ambas.

Cada representación tiene muchas entradas disponibles, cada una con un número de asiento único.

Una representación se puede identificar por una obra, una fecha y una hora

