

Diagrama de Clases

Los diagramas de clases es uno de los artefactos más utilizados. Muestran claramente la estructura de un sistema específico al modelar sus clases, atributos, operaciones y relaciones entre los objetos. Como las clases son los componentes básicos de los objetos, los diagramas de clases son los componentes básicos del UML.

La figura de clase en sí misma consiste en un rectángulo de tres filas. La fila superior contiene el nombre de la clase, la fila del centro contiene los atributos de la clase y la última expresa los métodos o las operaciones que la clase puede utilizar. Las clases y las subclasses se agrupan para mostrar la relación estática entre cada objeto.

Características de los diagramas de clases:

- Ilustrar modelos de datos para sistemas de información, sin importar qué tan simples o complejos sean.
- Comprender mejor la visión general de los esquemas de una aplicación.
- Expresar visualmente cualesquier necesidades específicas de un sistema y divulgar esa información en toda la empresa.
- Crear diagramas detallados que resalten cualquier código específico que será necesario programar e implementar en la estructura descrita.
- Ofrecer una descripción independiente de la implementación sobre los tipos empleados en un sistema que son posteriormente transferidos entre sus componentes.

Componentes básicos de un diagrama de clases

El diagrama de clases estándar está compuesto por tres partes:

- **Sección superior:** Contiene el nombre de la clase. Esta sección siempre es necesaria, ya sea que estés hablando del clasificador o de un objeto.
- **Sección central:** Contiene los atributos de la clase. Usa esta sección para describir cualidades de la clase. Esto solo es necesario al describir una instancia específica de una clase.
- **Sección inferior:** Incluye operaciones de clases (métodos). Esto está organizado en un formato de lista. Cada operación requiere su propia línea. Las operaciones describen cómo una clase puede interactuar con los datos.

Modificadores de acceso a miembros

Todas las clases poseen diferentes niveles de acceso en función del modificador de acceso (visibilidad). A continuación, te mostramos los niveles de acceso con sus símbolos correspondientes:

- Público (+)
- Privado (-)
- Protegido (#)
- Paquete (~)
- Estático (subrayado)

Alcance de los miembros

Hay dos alcances para los miembros: clasificadores e instancias.

Los clasificadores son miembros estáticos, mientras que las instancias son las instancias específicas de la clase.

Componentes adicionales del diagrama de clases

En función del contexto, las clases de un diagrama de clases pueden representar los objetos principales, las interacciones en la aplicación o las clases que se programarán.

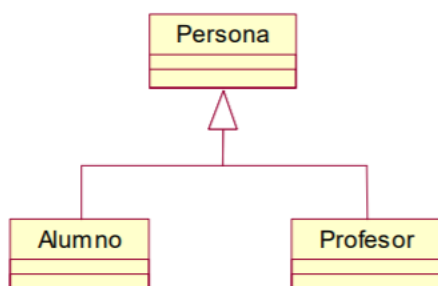
Clases: Una plantilla para crear objetos e implementar un comportamiento en un sistema. En UML, una clase representa un objeto o un conjunto de objetos que comparte una estructura y un comportamiento común. Se representan con un rectángulo que incluye filas del nombre de la clase, sus atributos y sus operaciones. Al dibujar una clase en un diagrama de clases, solo se debe cumplimentar la fila superior. Las otras son opcionales y se usan si deseas agregar más detalles.

- **Nombre:** La primera fila en una figura de clase.
- **Atributos:** La segunda fila en una figura de clase. Cada atributo de una clase está ubicado en una línea separada.
- **Métodos:** La tercera fila en una figura de clase. También conocidos como "operaciones", los métodos se organizan en un formato de lista donde cada operación posee su propia línea.
- **Señales:** Símbolos que representan comunicaciones unidireccionales y asíncronas entre objetos activos.
- **Tipos de datos:** Clasificadores que definen valores de datos. Los tipos de datos pueden modelar tanto enumeraciones como tipos primitivos.
- **Paquetes:** Figuras diseñadas para organizar clasificadores relacionados en un diagrama. Se simbolizan con una figura de un gran rectángulo con pestañas.
- **Interfaces:** Una recopilación de firmas de operaciones o de definiciones de atributo que define un conjunto uniforme de comportamientos. Las interfaces son similares a una clase, excepto por que una clase puede tener una instancia de su tipo, y una interfaz debe poseer, como mínimo, una clase para implementarla.
- **Enumeraciones:** Representaciones de tipos de datos definidos por el usuario. Una enumeración incluye grupos de identificadores que representan valores de la enumeración.
- **Objetos:** Instancias de una clase o clases. Los objetos se pueden agregar a un diagrama de clases para representar instancias prototípicas o concretas.
- **Artefactos:** Elementos modelo que representan las entidades concretas de un sistema de software, como documentos, bases de datos, archivos ejecutables, componentes de software y más.

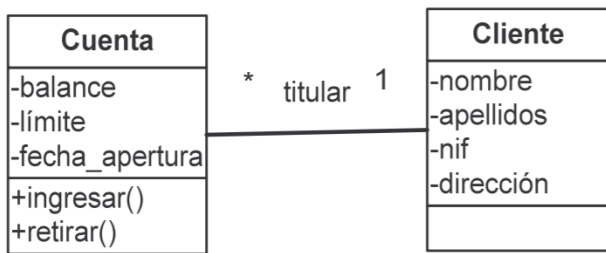
Interacciones

El término "interacciones" se refiere a múltiples relaciones y enlaces que pueden existir en diagramas de objetos y de clases. Algunas de las interacciones más comunes incluyen:

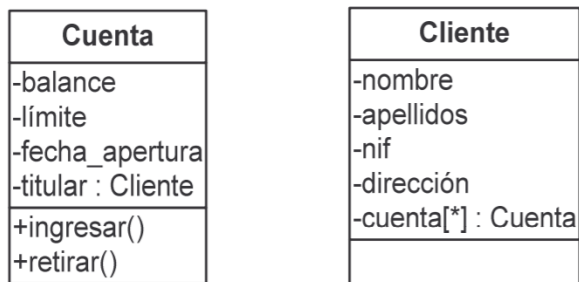
- **Herencia:** El proceso en el que una subclase o clase derivada recibe la funcionalidad de una superclase o clase principal, también se conoce como "generalización". Se simboliza mediante una línea de conexión recta con una punta de flecha cerrada que señala a la superclase.



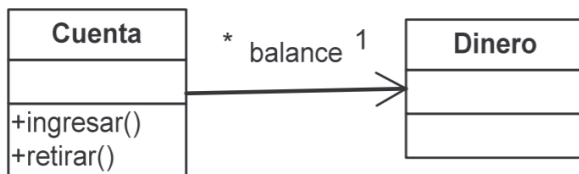
Asociación bidireccional: La relación predeterminada entre dos clases. Ambas clases están conscientes una de la otra y de la relación que tienen entre sí. Esta asociación se representa mediante una línea recta entre dos clases.



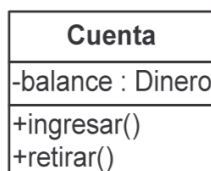
viene a ser lo mismo que



Asociación unidireccional: Una relación un poco menos común entre dos clases. Una clase está consciente de la otra e interactúa con ella. La asociación unidireccional se dibuja con una línea de conexión recta que señala una punta de flecha abierta desde la clase "knowing" a la clase "known".



equivale a



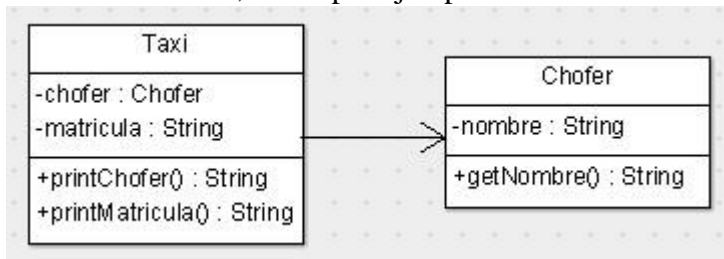
Las relaciones entre las clases reflejan lo que ocurre en el mundo real, establecen las conexiones lógicas. UML posee las siguientes relaciones:

- Dependencia:** Relación más débil que una asociación. Esta relación se da cuando una clase depende y usa los servicios de otra clase. Por ejemplo, una clase A, en un método, recibe un parámetro que es un objeto de un determinado tipo de clase B. La clase A tiene una dependencia sobre la clase B. Gráficamente se simboliza con una línea discontinua con una punta de flecha que apunta del cliente al proveedor.

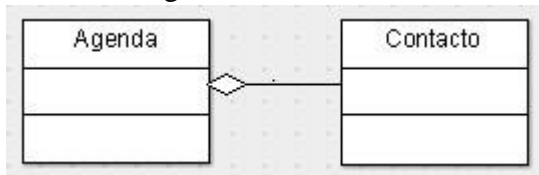


- b) **Asociación:** Relación más fuerte que una dependencia. Es una conexión, cuando entre dos clases se establece un vínculo que las une, entre una clase A y una B, donde los objetos de la clase B estarían relacionados con la clase A en base a criterios que se imponga. Gráficamente se simboliza con una línea continua.

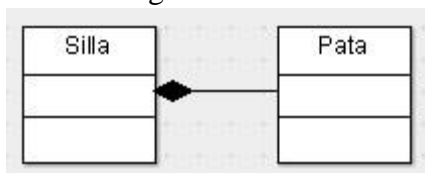
Por ejemplo, la clase Taxi está asociada a la clase Chofer. La clase Taxi podría acceder a un atributo de la clase Chofer, como por ejemplo el nombre.



- a. **Agregación:** es una asociación donde la clase A consta de un conjunto de objetos de la clase B de manera tal que si la clase A desaparece los objetos de la clase B siguen existiendo. Por ejemplo, una Agenda está formada por un conjunto de Contactos, si la asociación se rompe los Contactos siguen existiendo en el sistema, por lo tanto, no desaparecen.



- b. **Composición:** es una asociación fuerte donde la clase A esta formada por un conjunto de objetos de la clase B de manera tal que si la clase A desaparece también desaparecerán los objetos de la clase B. Por ejemplo, una Silla está formada por un conjunto de cuatro Patas, si eliminamos la Silla también deberían desaparecer las Patas. El símbolo es un rombo relleno de color negro en el lado del contenido, en el ejemplo la clase.



- c) **Generalización:** Se refiere a la herencia entre clases: Ejemplo entre una clase Cliente y las clases Clientes Personal y Cliente Sociedad. Otro ejemplo es la clase Sistema de Pago y las clases Tarjeta Crédito y Contrato.
- d) **Realización:** Una clase implementa una interface o contrato que se define.