

TEXT CLASS REVIEW

TEMAS A TRATAR EN EL CUE

- ¿Qué es un diagrama de clases?
- ¿Para qué sirven los diagramas de clases?
- Notación de un diagrama de clases.
- Atributos y métodos en un diagrama de clases.
- Tipos de Relaciones entre clases.
- Leyendo un diagrama de clases.
- Escribiendo un diagrama de clases.
- Ejemplo de un diagrama de clases.

¿QUÉ ES UN DIAGRAMA DE CLASES?

Un diagrama de clases en Lenguaje Unificado de Modelado (UML), es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones, y las relaciones entre los objetos.

Son un tipo de diagrama de estructura porque describen lo que debe estar presente en el sistema que se está modelando.

El UML se estableció como un modelo estandarizado para describir un enfoque de programación orientada a objetos (POO). Como las clases son los componentes básicos de los objetos, los diagramas de clases son los componentes básicos del UML. Los diversos componentes en un diagrama de clases pueden representar las clases que se programarán en realidad, los objetos principales o la interacción entre clases y objeto

¿PARA QUÉ SIRVEN 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 las 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.

NOTACIÓN 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 se esté hablando del clasificador o de un objeto.
- **Sección central:** contiene los atributos de la clase. Esta sección describe 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.

Seccion Superior
- Seccion : Central
+ Seccion Inferior()

ATRIBUTOS Y MÉTODOS EN UN DIAGRAMA DE CLASE

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 (+):** indica que el atributo será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.
- **Privado (-):** indica que el atributo sólo será accesible desde dentro de la Clase (sólo sus métodos lo pueden acceder).



- **Protegido (#):** indica que el atributo no será accesible desde fuera de la clase, pero si podrá ser accesado por métodos de la clase, además de las subclases que se deriven.

Los métodos u operaciones de una clase son la forma en cómo ésta interactúa con su entorno. Pueden tener las siguientes características:

- **público (+):** indica que el método será visible tanto dentro como fuera de la Clase, es decir, es accesible desde todos lados.
- **privado (-):** indica que el método sólo será accesible desde dentro de la clase (sólo otros métodos de la clase lo pueden acceder).
- **Protegido (#):** indica que el método no será accesible desde fuera de la clase, pero si podrá ser accesado por métodos de la clase, además de métodos de las subclases que se deriven.

REPRESENTACIÓN DE UNA CLASE

clasePersona
+ nombre : string
+ apellidos : string
- nro_Pasaporte : string
+ llamar()

Podemos ver la representación de una clase. Lo que podemos deducir de esta:

- Nombre de la clase:
clasePersona
- Atributos
 - **nombre:** público del tipo string.
 - **apellidos:** público del tipo string.
 - **nro_pasaporte:** privado del tipo string.
- Métodos
 - **llamar():** público, sin parámetros.

TIPOS DE RELACIONES ENTRE CLASES

Una relación identifica una dependencia. Esta dependencia puede ser entre dos o más clases (más común), o una clase hacia ella 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, y 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. Son las siguientes:

- **Multiplicidad:** es 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. Por ejemplo:
 - 1..*, que se lee de uno a muchos.
 - 1..1, que se lee de 1 a 1.
 - *..*, que se lee de muchos a muchos.
- **Nombre de la asociación:** es describir una indicación de la asociación que ayuda a entender la relación que tienen dos clases. Suelen utilizarse verbos, por ejemplo: "Una empresa contrata a muchos (*) empleados"



Además, existen notaciones específicas para los otros tipos de relación, como son:

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.

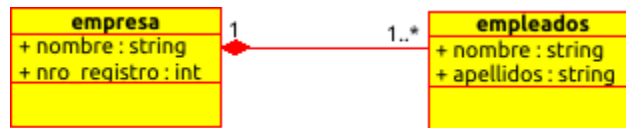
Un ejemplo de asociación podría ser: "un perro pertenece a un animal".



COMPOSICIÓN

Hace referencia a una relación entre clases que indica pertenencia. Y una es dependiente a la otra. Se representa con un rombo lleno en la clase, cuya instancia contiene las instancias de la otra clase. Para saber si una relación es correcta, se realizan las siguientes preguntas:

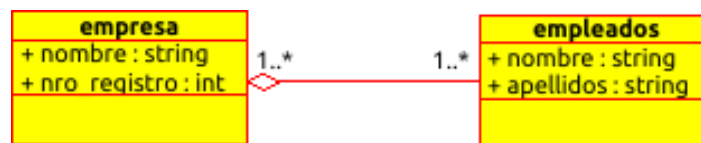
- ¿Una clase1 está compuesta por otra clase2?
- ¿Si clase1 deja de existir la clase2 también?



En este ejemplo observamos que una empresa se compone de empleados, y además si la empresa deja de existir, el empleado también. Es decir, es una relación fuerte entre las dos.

AGREGACIÓN

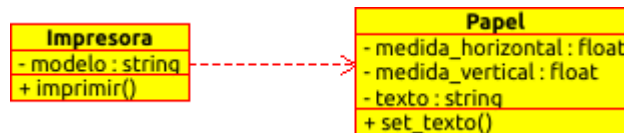
Es una representación jerárquica que indica a un objeto, y las partes que lo componen. Es decir, representa relaciones en las que un objeto es parte de otro, pero aun así debe tener existencia en sí mismo. Se representa con un rombo hueco en la clase, cuya instancia es una agregación de las instancias de la otra.



En este caso, si la clase empresa es eliminada, la clase empleados se mantiene. Tanto Agregación como Composición son para tipos de relación (Todo/parte) donde una clase hace parte de otra, no exactamente que tiene una clase.

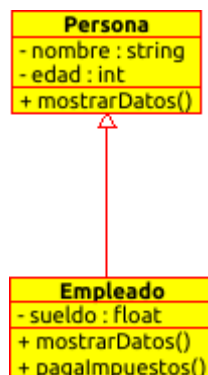
DEPENDENCIA

Una línea discontinua con una flecha apuntando a otra clase de dependencia. La relación puede tener un estereotipo que se coloca junto a la línea punteada. Es una relación más débil que una asociación, por lo general es una relación de uso.



HERENCIA

Una de las características principales de la orientación a objetos es poder asociar a los objetos (clases en UML) con características y operaciones que puedan heredarse entre ellos, siempre y cuando estos estén ligados a un entorno en común. Esta relación se representa como una línea continua con una flecha hueca en el extremo que apunta a la superclase.



LECTURA DE UN DIAGRAMA DE CLASES

1. Lectura de la estructura del objeto de la clase que 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.
 - **Sección inferior:** incluye operaciones de clases (métodos).



2. Lectura de la multiplicidad de la clase.
3. Lectura de las relaciones que tengan estas clases.

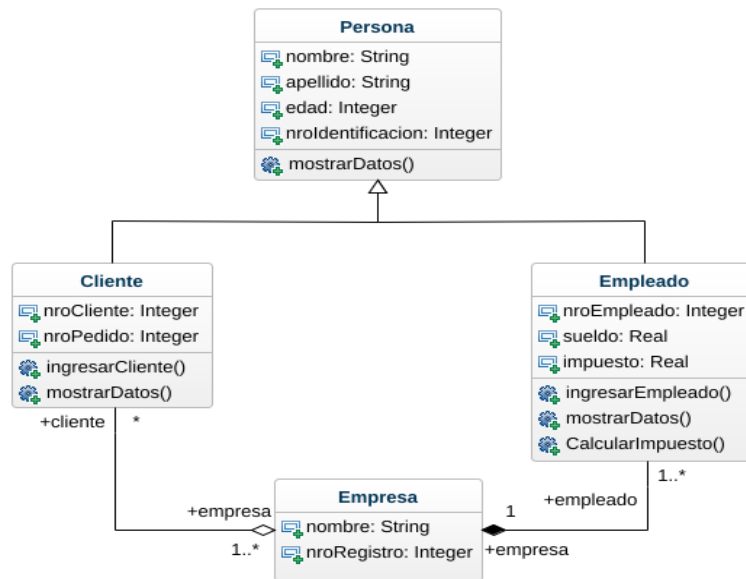
ESCRITURA DE UN DIAGRAMA DE CLASES

Los diagramas de clase van de la mano con el diseño orientado a objetos. Por lo tanto, conocer lo básico de este tipo de diseño es una parte clave para poder dibujar diagramas de clase eficientemente.

Estos tipos de diagramas son solicitados cuando se está describiendo la vista estática del sistema, o sus funcionalidades. Unos pequeños pasos que puedes utilizar de guía para construirlos son los siguientes:

1. Identifica los nombres de las clases a partir de los objetos. El primer paso es identificar los objetos primarios del sistema. Las clases suelen corresponder a sustantivos dentro del dominio del problema.
2. Distinguir las relaciones entre clases. El siguiente paso es determinar cómo cada una de las clases u objetos están relacionadas entre sí. Encontrando los puntos en común y las abstracciones entre ellos; esto ayuda a agruparlos al dibujar el diagrama de clase.
3. Crea la estructura. Primero, agrega los nombres de clase y vincularlos con los conectores apropiados, prestando especial atención a la cardinalidad o las herencias. Deja los atributos y funciones para más tarde, una vez que esté la estructura del diagrama resuelta.
4. Identificar los atributos con su tipo de dato.
5. Identificar las operaciones.

EJEMPLO DE UN DIAGRAMA DE CLASES



En este ejemplo se observa un diagrama de clases que pertenece a una empresa compuesta por clientes y empleados. Para su escritura se observa que tanto clientes como empleados tienen atributos comunes que son: nombre, apellidos, edad, y número de identificación que se generaliza como atributos en la clase Persona, esta clase posee un método de mostrar datos de la persona.

La clase Cliente posee dos atributos particulares: el número de cliente y el número de pedido; adicionalmente, esta clase posee el método ingresar cliente y mostrar datos del cliente.

La clase Empleado posee tres atributos que son: el número de empleado, sueldo, e impuesto. Adicionalmente, posee los métodos de ingresar empleado, mostrar datos del cliente y cálculo de impuestos.

La clase Empresa posee dos atributos que son: el nombre y el número de registro de la empresa.

Con relación a la lectura de la clase, observamos que posee una relación de herencia de la clase Cliente y Empleado con respecto a Persona. Observamos también que el cliente posee una relación de agregación con la clase Empresa, leyendo la multiplicidad tenemos que cero o varios clientes pertenecen a una o muchas empresas, y que una o varias empresas posee cero clientes.