

# Qué es UML

UML significa **Unified Modeling Language**. Se trata de un **conjunto de diagramas que buscan capturar una "perspectiva" de un sistema informático**: por ejemplo, un diagrama está destinado a documentar los requerimientos del sistema y otro esta orientado a seguir el ciclo de vida de un determinado objeto.

## Diagrama de Clases

Es el diagrama más importante ya que en él se diseñan las clases que serán parte de nuestro sistema. Es el esqueleto de nuestra aplicación.

Una clase se representa con un rectángulo de la siguiente forma:



## Simbología General

La simbología utilizada en el diagrama de clases es la siguiente:

- : privado

+ : público

# : protegido

subrayados: de clase

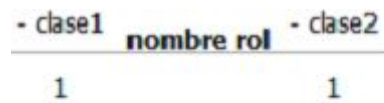
Muchas herramientas de diseño utilizan una simbología que consideran más amigable, basada en colores: por ejemplo **el color rojo denota que un atributo/método es privado y el color verde que es público.**

## Asociación

Es una relación genérica entre dos clases que representa un enlace entre los objetos.

Se caracterizan por tener un **nombre** (nombre de la relación) y una **cardinalidad** (también denominada multiplicidad de la relación).

Las asociaciones se representan con líneas rectas sobre las cuales se puede escribir un texto descriptivo o rol de la relación, así también como el grado de multiplicidad.



## Generalización

Indica que una clase “hereda” atributos y métodos de otra, es decir, que es “hija” de la superclase a la cual se apunta. Se representa con:



## Agregación

Es una relación que indica que un objeto es un componente o parte de otro objeto. Aún así, hay independencia entre ellos. Por ejemplo, Persona y Domicilio.

Se simboliza mediante:



## Composición

Es una relación más fuerte que la agregación. A diferencia del caso anterior, aquí no hay independencia, por lo cual, las partes existen sólo si existe la otra. Se representa con:



## Multiplicidad

Representa la cardinalidad que posee una clase con respecto a otra. Es un concepto muy similar a la multiplicidad que existe entre las distintas entidades de una base de datos.

Los distintos tipos de multiplicidad pueden representarse de la siguiente manera:

Uno a uno: 1 \_\_\_\_\_ 1

Uno a muchos: 1 \_\_\_\_\_ \*

Uno a uno o más: 1 \_\_\_\_\_ 1..\*

Uno a ninguno o a uno: 1 \_\_\_\_\_ 0..1

Combinaciones: 0..1, 3..4, 6..\*

## Ejemplo de Diagrama de clases

