

Diagrama de clases

Se utilizan para:

- Explorar conceptos del dominio
- Analizar requerimientos
- Mostrar el diseño detallado de software orientado a objetos
- Generalmente contiene: Clases, Interfaces y Relaciones.

CLASES:

- Es la descripción de un conjunto de objetos que comparten los mismos: Atributos, Métodos y Semántica.
- Representan objetos del mundo real y elementos conceptuales.
- Accesibilidad de atributos y operaciones:
Public: +
Private: -
Protected: #
- Clases abstractas
- Tipo de los atributos: se debe poner nombre ":" y el tipo
- Parámetro
- Retornos de los métodos
-

RELACIONES ENTRE CLASES

- Las relaciones son conexiones entre clases
- Modelan la colaboración entre objetos
- Tipos:
 - ◆ Generalización
 - ◆ Realización
 - ◆ Asociación
 - ◆ Agregación
 - ◆ Composición
 - ◆ Clase de asociación
 - ◆ Dependencia

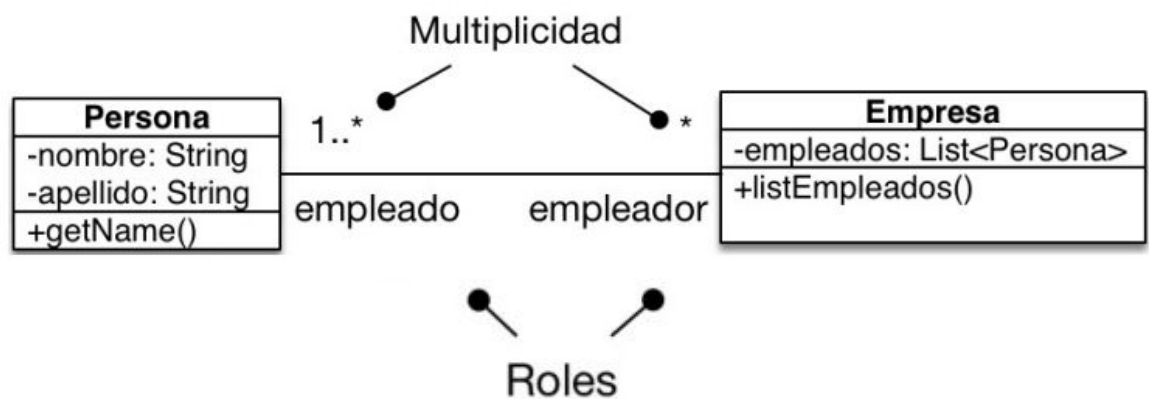
Relación de Generalización

- Indica una relación entre una superclase y una subclase.
- Usamos generalizaciones para modelar conexiones del tipo: "Es un tipo de".
- Por ejemplo supongamos que una clase A es subclase de B, los atributos, operaciones y relaciones comunes se muestran en la superclase B.

Relación de asociación

- Es una relación estructural.

- Es por ejemplo cuando un objeto de la clase A tiene un atributo o una colección del tipo es B.
- Usamos asociaciones para modelar conexiones del tipo: “tiene”, “es de”, “conoce”
- Se dibuja como una línea entre dos clases
- se puede agregar más información a la relación, es decir tener conectores como por ejemplo: trabaja en, tiene, etc.
- La misma clase puede jugar el mismo o diferentes roles en otras asociaciones.
- La multiplicidad indica cuántos elementos de una instancia se relacionan con otra.



Relación de Agregación

- Es un tipo especial de relación de asociación.
- Se utiliza solamente cuando una de las clases representa el “todo” y la/s otra/s la/s “partes” .
- Modelan conexiones del tipo: “Está formado por”.
- Se dibuja como una línea entre dos clases con un rombo en blanco sobre la clase que representa el “todo”.
- Puede incluir información.



Relación de Composición

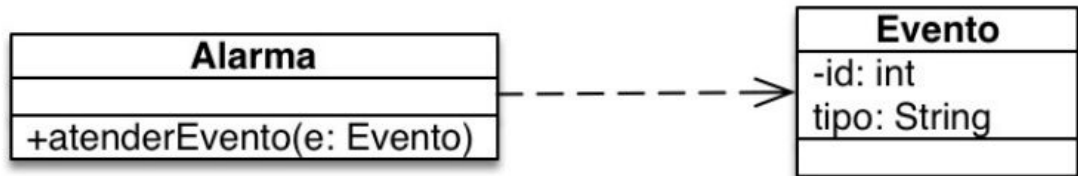
- Es un tipo especial de relación de agregación
- También representa el “todo” y la/s “partes” pero el tiempo de vida de las partes está ligada al del todo, cuando se destruye el todo también se destruyen las partes.
- Se dibuja como una línea entre dos clases con un rombo lleno sobre la clase que representa el “todo”

- Puede incluir información.



Relación de Dependencia

- Es una relación para indicar que una clase usa a otra clase
- Usamos dependencias para modelar conexiones del tipo: “usa”.
- Se dibuja como una línea punteada entre dos clases con una flecha que indica de qué clase se depende.



Relación de Realización

- Indica una relación donde una de las partes especifica un “contrato” y la otra parte garantiza llevarlo a cabo.
- Es una mezcla entre dependencia y generalización
- Usamos realizaciones para modelar conexiones del tipo: “Implementa”
- Se usan principalmente para especificar la relación entre una interface y la clase que provee una operación para ella.