

# Elaboración de diagramas de clases

## Contenido

- POO: Conceptos básicos .....2
- Diagramas de clases .....2
  - Clases.....2
    - Visibilidad .....2
    - Tipo de dato .....3
  - Relaciones .....3
    - Asociación: .....3
    - Clase asociación .....3
    - Herencia .....3
    - Composición y Agregación .....3
    - Dependencia .....4

# POO: Conceptos básicos

Un objeto es una instancia de una clase. Tiene una serie de atributos y métodos, además de identidad propia. Los comportamientos de un objeto tendrán una clase en común. Los principios del modelo Orientado a Objetos son:

- **Abstracción:** Las características de un objeto de tal forma que lo distinga de los demás. Indican su trabajo y cómo se comunica con los demás. Permite seleccionar las características más importantes e identificar comportamientos comunes
- **Encapsulación:** Son unos elementos que pertenecen a la misma entidad y a la misma abstracción. Permite más cohesión de los componentes y separa la parte accesible (pública) de la no accesible (privada)
- **Modularidad:** Se puede dividir en módulos independientes entre ellos, es decir, que los módulos han de acoplarse entre ellos lo menos posible
- **Herencia:** Algunas clases pueden tener características de una clase superior. Se establece una jerarquía, donde una **superclase o clase padre** “otorga” sus métodos a su **subclase o clase hijo**. La herencia permite el polimorfismo y el encapsulamiento
- **Polimorfismo:** Cuando dos instancias y objetos responden a la llamada de métodos con mismo nombre, pero distinto comportamiento.
- **Ocultamiento:** Protege las propiedades de un objeto contra su modificación (privado)
- **Tipificación:** Definición de un objeto para que no se pueda intercambiar con objetos de distintos tipos
- **Concurrencia:** Un objeto activo frente a un inactivo
- **Persistencia:** Propiedad de un objeto en función de su trascendencia
- **Recolección de basura:** Se encarga de destruir los objetos que queden sin referencia para desvincular la memoria asociada

## Diagramas de clases

Son la representación gráfica y estática de la estructura de un sistema, valorando sus clases e interacciones representadas en bloques. Se forma por clases y relaciones

### Clases

Son la unidad básica y contienen información de un objeto. Es una descripción de objetos que comparten los mismos atributos y métodos. Se dividen en:

- Parte superior: Nombre de clase
- Parte central: Atributos ({+/-} nombre: tipo)
- Parte inferior: Métodos ({+/-} nombre (valores entrada): valor salida)

### Visibilidad

Cuando se crean atributos se indica también su visibilidad, que puede ser **pública** (+, se puede acceder desde cualquier sitio), **privada** (-, solo se accede desde la propia clase), **protegida** (#, solo se puede acceder desde la misma clase o sus hijas), y **package** (~, se verán únicamente en su mismo paquete)

## Tipo de dato

Se pueden emplear los siguientes:

- **Integer/int:** Valor entero
- **String:** Cadena de caracteres
- **Boolean:** booleana (true/false)
- **Float/Double:** Decimales
- **Char:** Caracteres
- **Date:** Fecha
- **Time:** Hora
- **DateTime:** Fecha y Hora

## Relaciones

Son las vinculaciones entre las clases y objetos. Poseen un nombre y una cardinalidad llamada multiplicidad. Representan el numero de instancias que puede haber entre esas clases, como el modelo EER de SQL. En cada extremo se fija la multiplicidad mínima y máxima:

- **0..1:** Cero o una vez
- **1:** Una y solo una vez
- **0..\*:** Cero a muchas veces
- **1..\*:** Una a muchas veces
- **M..N:** De M a N veces

## Asociación:

Puede ser unidireccional o bidireccional, dependiendo de si uno conoce la existencia del otro. Una unidireccional solo se puede mover de origen a destino, sin que el destino sepa que la clase origen exista. La clase origen tendrá un objeto de tipo destino. La bidireccional se puede mover de un lado a otro sin problema.

Además, las asociaciones pueden ser de distintos tipos:

- **Reflexiva:** A uno mismo, grado 1
- **Binaria:** Entre 2 clases
- **Ternaria, cuaternaria, etc.:** Entre 3, 4...

## Clase asociación

Nace de dos clases que necesitan una información necesaria para su relación. Recibe el estatus de clase y las instancias serán como elementos de clases, y se pueden incluso vincular con otras clases.

## Herencia

Se pueden organizar las clases de forma jerárquica por medio de la herencia. Se hace con una flecha vacía apuntando a la superclase.

## Composición y Agregación

Ambas asociaciones consisten en que un objeto puede estar compuesto de varios objetos llamados componentes.

La composición es una asociación fuerte, la supresión del compuesto supone que los componentes desaparezcan, será de 1 a varios, con un rombo lleno en el compuesto. Los componentes solo pueden aparecer en un compuesto.

La agregación es la relación débil, en la cual los componentes deben ser compartidos por varios compuestos y la destrucción de cualquiera de ellos no implica la desaparición de ningún objeto. Se representa con un rombo vacío, y es de 1 a varios

## Dependencia

Una relación unidireccional donde una clase utiliza otra.