



Generalización - Herencia: Su propósito es el reúso de comportamiento para ahorrar código. Se define el comportamiento una sola vez en una clase padre para ser heredado por clases hijas. Es una relación que se da entre clases y no se traslada a los objetos porque la identidad de los objetos no se puede duplicar. Para lograr la herencia la clase padre debe romper su propiedad de encapsulamiento, de esta manera su implementación es visible para las clases hijas. Vale la pena usar herencia cuando se puede reutilizar todo el comportamiento

Agregación: Establece que un todo está formado por partes. Una clase “todo” esta formada por clases “partes”

Composición: Es una relación mas estricta que la agregación ya que el “todo” no puede existir sin sus “partes”

Asociación: Establece que no hay una clase más importante que otra. Una clase utiliza a otra en un determinado momento