

# Programación orientada a objetos

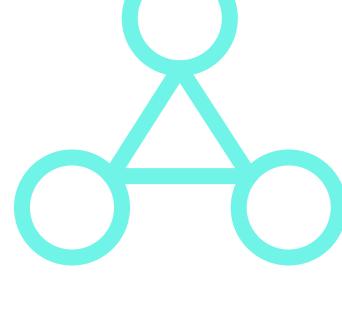
Básicamente una clase es una plantilla de código que sirve para instanciar objetos, esto es crear objetos en memoria, la cual debe seguir los principios básicos de la programación orientada a objetos, los cuales son los siguientes:

## Encapsulamiento



Hay que mantener el estado de un objeto privado, esto quiere decir que sólo el objeto pueda modificar características internas; en la POO un objeto no puede comunicarse o alcanzar métodos o propiedades privados de otro objeto.

## Abstracción



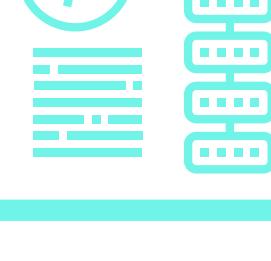
Al aplicar el encapsulamiento, sólo unos cuántos métodos quedan accesibles externamente, lo que genera una interfaz de comunicación, esto quiere decir que un objeto no tiene porqué saber cómo ésta construido otro objeto o cómo funciona por dentro, sólo tiene que saber cómo comunicarse con él; tal cual como se interactúa con los objetos reales, muchas veces no se sabe cómo están construidos o cómo funcionan, el objeto con el que se interactúa expone una interfaz con la que se pueden realizar acciones en él.

## Herencia



Muchas veces los objetos son bastante similares, pueden compartir una lógica común, pero no siempre exactamente lo mismo; la herencia permite el re-uso del código al crear una clase hija con base en una clase padre, heredando las propiedades y métodos de la clase padre, haciendo posible que se escriban en la hija nuevas propiedades y métodos específicos de la clase hija.

## Poliformismo



Ese pequeño cambio en cómo un objeto realiza una acción determinada es el polimorfismo y normalmente se logra sobrescribiendo o sobrecargando (overriding) los métodos de la clase padre.

