



---

## TAREA 2

---

### 1. Programación Orientada a Objetos

En programación se tienen varios paradigmas, los cuales son útiles en diferentes circunstancias y para diferentes niveles de conocimiento. El más famoso es el de programación secuencial, uno "contrario" a este es el de programación en paralelo, el cual busca utilizar todo el poder del procesador. Normalmente, cuando se enseña a programar, luego de estudiar la programación estructurada, se introduce un concepto muy útil en muchas ramas del saber, función. La programación funcional es muy útil al estar trabajando con problemas dividibles en subproblemas. Luego, con esta idea en mente, se puede tener paquetes/librerías de funciones para poder reutilizar en varios programas y con mayor facilidad de depuración, a esto se le conoce como programación modular.

Partiendo de la idea de la programación modular y funcional, se introduce el paradigma de programación orientada a objetos (POO), el cual mezcla la idea de programación modular en un solo objeto de programación, llamado "objeto", valga la redundancia. Un objeto está compuesto de atributos (datos) y métodos (funciones) y un conjunto de objetos es una clase. En POO el programa no se divide en tareas sino en modelos. Lenguajes de programación que utilizan este paradigma como base: C++, C#, Java y JavaScript; también es posible utilizar este paradigma en otros lenguajes como Python o R.

#### 1.1. Características

**Abstracción:** Es la base de cualquier programa a realizar, no solo en POO. Es la forma en la que el programador resuelve el problema en cuestión, de modo que el lenguaje sea capaz de entenderlo y de realizar los respectivos procedimientos. Es básicamente traducir la solución general a un algoritmo entendible por la máquina, obviamente siguiendo la POO.

**Encapsulamiento:** Es la organización de todos los elementos del programa, juntando aquellos que pertenezcan a la misma entidad (mismo nivel de abstracción).

**Herencia:** Es la relación que se tiene entre diferentes clases, esto es para facilitar y optimizar el código; también ayuda al polimorfismo, dos métodos que se nombran igual pero con características heredadas de distintas clases, realizarán sus respectivos procesos independientemente de su nombre.

**Polimorfismo:** Son comportamientos diferentes de objetos distintos, con entradas distintas que poseen el mismo nombre. Esto es una característica de muchos lenguajes de programación, el único punto bajo es que esta característica aumenta el tiempo de ejecución del programa.

**Modularidad:** Con lo explicado en los primeros párrafos referido a los diferentes paradigmas de programación, los módulos son subdivisiones de una aplicación, las cuales deben ser lo más independientemente posible de la aplicación en sí, así como de sus partes. La idea es facilitar la depuración y tener un programa más ordenado.