

Nombre: Ariel Camilo Sánchez López

Grupo: 302 SIS

### Programación orientada a objetos

Este tipo de programación se incluye dentro de la lista de paradigmas de programación, siendo el más común y de los más eficientes en la mayoría de los casos, ya que cuenta con diferentes características que permite que los programas bajo su nombre puedan contar con una organización y estructura casi orgánica.

Entre las características que hacen especial este tipo de programación podemos encontrar:

**Clases:** Una clase funciona como una plantilla que define como se van a organizar las diferentes estructuras, que atributos va a tener y que funciones puede cumplir. Un ejemplo de eso puede ser una clase llamada **Lápiz**, el concepto de 'lápiz' es conocido por todos y es fácilmente reconocible, se conocen sus características como puede ser su forma, tamaño, color y función, todos estos atributos permiten que la clase sea distinta a las demás.

**Objetos:** ¿Las clases son una plantilla, pero es una plantilla de qué? Acá es donde entran los objetos, estos objetos siguen la plantilla de la clase y son construidos en base a los atributos, funciones y métodos. Un ejemplo puede ser un objeto llamado 'lapiz1', este objeto tiene que regirse por la plantilla de la clase a la que corresponde, por lo que va a tener una forma, un tamaño, un color y una función, como puede ser escribir. Esta forma puede ser cualquiera que corresponda a un lápiz, debe tener un tamaño que corresponda al de un lápiz, un color y la función de escribir. Por otro lado, puede haber otro lápiz con otra forma, otro tamaño, otro color, pero la misma función. Los dos lápices son diferentes, pero tienen la misma plantilla de atributos y funciones, por lo que pertenecen a la misma clase.

**Encapsulación:** Esta característica permite que el objeto exponga o no exponga sus atributos al mundo exterior, en el caso del lápiz su color, forma y tamaño son visibles, pero en el ambiente de los computadores estos atributos son abstractos e invisibles, solo existen en una cadena de bits, por lo que dependiendo de la naturalidad del objeto y de la clase se pueden encapsular algunos atributos y que solo sean visibles algunos otros.

**Abstracción:** Una de las ventajas de la programación orientada a objetos es su fácil mantenimiento, al código estar organizado en diferentes archivos o secciones y a nivel práctico se encuentra en una estructura bien definida, es relativamente simple acceder a cada parte del programa y ver qué sucede en el interior. Aun así, estas estructuras pueden ser tan complejas y extensas que es necesario llevar esa complejidad a simples términos con significado simple y fácilmente de entender, donde se puedan identificar sus atributos y funciones.

**Herencia:** Retomando el ejemplo del lápiz, definimos un lápiz como un 'instrumento de escritura' al igual que otros instrumentos como lo pueden ser un marcador, bolígrafo, etc. Estos tres instrumentos se pueden catalogar en la clase 'InstrumentosDeEscritura' ya que cuentan con atributos y funciones en común. Si el lápiz era una clase y a la vez pertenece a otra clase que cuenta con elementos de sí, podemos decir que la clase lápiz es una subclase de InstrumentosDeEscritura que hereda los atributos de esta última.

**TDA:** Son datos abstractos que se encuentran en la vida cotidiana que permiten abstraer sus atributos y funciones. Un ejemplo puede ser un monitor, es un objeto de la vida cotidiana y

permite que mediante la abstracción se obtengan sus atributos y funciones. Como atributos tiene la resolución, tamaño, colores, etc. Y sus funciones son llevar a un formato visual los datos dados por la computadora.

**Polimorfismo:** En base a la plantilla los objetos son capaces de usar sus interfaces de interacción en otros objetos que compartan comportamientos y sean similares en estructura.