

## Síntesis conceptual

<b>Asignatura:</b> Programación
<b>Unidad:</b> 2. Programación orientada a objetos

### Resumen

La POO, Programación Orientada a Objetos posee claras ventajas frente al resto, pero igualmente posee claras desventajas. Mantiene el código y los datos están entrelazados, mientras que, en la programación estructurada, estos se encuentran separados. La interacción entre distintos objetos permite acercarnos a un modelo que simule el mundo real de la mejor manera. Esto permite crear programas escalables y modulares, hace que abordar programas de gran tamaño sea más sencillo que con la programación estructurada, presenta una interfaz gráfica más depurada y permite reutilizar código sin tener que escribirlo nuevamente.

Dentro de POO destacan estos elementos:

- La clase: Elemento principal de la POO. En la clase queda descrito el posible conjunto de datos y operaciones. De una clase podemos obtener la cantidad de objetos instanciados que se requieran. Algunos de los elementos que contienen son:
  - El campo: elementos que actúan como variables propias de cada instancia de una clase y que definen un atributo de un objeto.
  - Propiedad: determina un atributo de un objeto. La propiedad permite el acceso a un área de almacenamiento mediante un campo definido con anterioridad o utilizando un método creado con ese objetivo.
  - Método: forma en que se implementa un algoritmo que se aplica a todas las instancias de una clase, puede poseer dos formas: función y procedimiento.
  - Constructor: tipo especial de método que pertenece a una clase concreta y utilizado para instanciar objetos que pertenecen a esta.
  - Destructor: se encarga de destruir un objeto y liberar aquellos recursos que este tuviera asignados.
- El objeto: La instancia de una determinada clase se llama objeto. En otras palabras, podemos decir que un objeto resulta de la creación en tiempo de ejecución de un conjunto de datos, el cual es almacenado en memoria y que cuenta con una estructura como la descrita en su clase. Entre los objetos encontramos los simples y los compuestos. Los simples están formados por datos primitivos (un dígito, un carácter, un booleano, etc.), mientras que los compuestos están formados por datos primitivos, y además por objetos pertenecientes a otras clases.
- El encapsulamiento: La encapsulación es la agrupación en un mismo elemento de los datos y las operaciones relacionadas con estos.
- Abstracción: La abstracción es un mecanismo que permite reducir la complejidad de los objetos minimizando la información a representar, dejando solo los aspectos más relevantes.
- Reutilización: Este modelo de programación permite reutilizar el código de forma sencilla.

- **Visibilidad:** La visibilidad determina la manera en que los miembros de una clase son accedidos desde otros puntos del programa. Distinguimos tres tipos de visibilidad: privada, pública y protegida.
- **Relaciones entre clases:** Muchas clases se relacionan entre sí de diferentes tipos:
  - **Herencia:** Clase derivada de otra clase denominada clase base, clase padre, clase ancestral o superclase, y la clase que hereda se llama clase extendida, clase hija, clase derivada o subclase.
  - **Asociación, composición y agregación:**
    - **Asociación:** relación en la que los objetos pueden pertenecer a la misma o a varias clases distintas, las cuales interactúan entre sí.
    - **Agregación:** asociación opcional, los componentes son capaces de existir por sí mismos y de participar de otras relaciones de composición con otros objetos.
    - **Composición:** asociación obligatoria, del compuesto dependen los componentes y sin él no tiene sentido que existan.
- **Polimorfismo:** El polimorfismo es la propiedad que permite a los objetos adoptar distintas formas en tiempos de ejecución, además de permitir que se utilice como objeto de clase base un objeto de clase derivada. Puede aparecer de dos formas:
  - **Asignación polimorfa:** sucede cuando se contiene en un elemento declarado de clase base un objeto de clase derivada. Para esto, el lenguaje debe permitir la relajación del sistema de tipos.
  - **Ejecución polimorfa:** sucede al invocar métodos con el mismo nombre sobre instancias que son de diferente clase sin tener constancia de ello.

## Conceptos fundamentales

- **Abstracción:** mecanismo por el que se presentan solamente los aspectos fundamentales de un objeto, omitiendo la información poco relevante.
- **Algoritmo:** instrucciones lógicas y ordenadas que siempre proporcionan un resultado, se emplea como base para resolver otros problemas.
- **Encapsulación:** agrupación en un mismo elemento de los datos y las operaciones relacionadas con estos.
- **Implementar:** llevar a la práctica una idea o proyecto, generalmente para resolver un problema o necesidad.
- **Instanciar:** crear instancias de una clase, es decir, individualizar un elemento concreto de la idea de un tipo de objeto abstracto.