

## Capítulo 1

En Primer lugar, antes de adentrarse en el tema de la programación orientada a objetos, debemos iniciar definiendo que es. La Programación Orientada a Objetos (POO) es un modelo de programación, es decir, un patrón o un estilo de programación que nos da unas pautas sobre cómo trabajar con él. Se basa el sistema de clases y objetos. Este tipo de programación se usa para construir un programa de software en piezas manejables y que puedan utilizarse nuevamente para crear instancias individuales de objetos.

Ciñéndonos a lo que la historia nos dice, podemos ver que la programación orientada a objetos data de mediados de la década de los 50 y 60, con un lenguaje de programación llamado Simula el cual más adelante evolucionaría a uno con mayores avances y mejor rendimiento a la hora del procesamiento, el cual recibía el nombre de Smalltalk.

Luego de esto, hubo un resurgimiento atrayente hacia por las metodologías que utiliza este lenguaje de programación. Más específicamente en lenguajes como C++ y Eiffel. Después de esto este nicho continuó creciendo con la llegada de Java en el 1990 y con el gran lanzamiento de .NET en el año 2002. Hoy en día este tipo de lenguaje solamente sigue avanzando y son unas de las bases fundamentales de donde se sostiene la programación moderna.

**¿Por qué la OOP se ha convertido en uno de los lenguajes más utilizados en la industria hoy en día?**

Básicamente en la década de los 90 los lenguajes más utilizados por empresas para el desarrollo de nuevos sistemas era pascal y C, los cuales te permitían ejecutar un programa siguiendo una serie de pasos continua y sin saltos, lo cual estaba bien si el programa no era muy grande, dado que si este contaba con una amplia cantidad de código era bastante complicado el poder depurarlo, solucionar errores y aun más poder hacerles una mejora después de que ya estaban constituidos. Para intentar combatir el cada vez mayor tamaño de los programas se implementan las funciones, lo cual claramente fue una mejora, pero de todos modos esta no soluciono del todo el problema que presentaban estos lenguajes de programación.

Visto ya todo lo anterior, pasaremos a presentar algunos de los términos o palabras fundamentales de este lenguaje.

### **-Objetos**

Básicamente y dicho con palabras simples, un objeto es una estructura que almacena datos. Dicho de otra manera, un objeto puede ser un individuo del cual se pueden almacenar sus atributos o características como lo son: color de piel, estatura, número de identificación etc. Esto es bastante

útil ya que tiene bastantes aplicaciones en múltiples campos laborales, como: nóminas, cuentas de banco y hasta el poder llevar un inventario.

### **-Abstracción**

Esta función se utiliza para la filtración de los datos innecesarios de un objeto, dado que despendiendo del tipo de programa que se este creando una u otra información puede llegar a estar de más restándole así eficiencia al programa que se está desarrollando por lo cual esta se obvia, un ejemplo de esto pueden llegar a ser el aplicaciones de pedidos las cuales proporcionan constantes actualizaciones en tiempo real mostrándonos el trayecto de nuestro pedido, pero, además de esto la aplicación maneja una red de pedido de las demás personas las cuales a nosotros no nos concierne en lo absoluto por lo cual esto se obvia o se filtra.

### **-Encapsulamiento**

El encapsulamiento es un proceso que se utiliza para resguardar los datos de un objeto y que de esta manera su manejo sea mas seguro y poder ahorrarse todo el depurado del programa. Veamos esto como una oficina de recursos humanos que trabaja con información de todos los empleados de una empresa, esta solamente en caso de tener una solicitud de una persona de la suficiente capacidad permitirá la extracción o modificación de estos datos y de no ser así los encapsulará (Ocultará).

### **-Polimorfismos**

El polimorfismo el proceso que se utiliza para asociar dos o más métodos a un objeto de tal manera que cuando este objeto reciba una solicitud lleguen respuesta a ambos métodos, un ejemplo practico seria con un programa para una tienda que quiere manejar su inventario. Entonces se crea un método para saber si hay productos en stock y se crea otro adjunto a este que haga se diga el precio de este producto.

### **-Herencia**

Tal y como su nombre lo indica, la herencia nos permite hacer que los objetos en este caso hereden atributos o mejor dicho datos de los objetos anteriormente creado, como si estos fuesen platillas para facilitar el trabajo. Un ejemplo de esto puede ser un programa de recursos humanos que ya tenga una plantilla para los empleaos a la cual solamente haya que modificarles cosas básicas como nombre y Id. Ya que lo de más estará dado por los datos que ya poseía la anterior

### **-Agregación**

la agregación de manera simple podíamos decir que es un objeto dentro de otro objeto y así sucesivamente.

## **-References**

Clark, D. (2013). Beginning C# Object-Oriented Programming (2nd Ed.). New York: Apress.