educación IT
CODE YOUR FUTURE

Programación Web .NET Core

Módulo 1

educación IT
CODE YOUR FUTURE

Introducción a la Programación Orientada a Objetos (POO)

Diferencias entre Programación Estructurada y POO

Programación Estructurada

La *programación estructurada* nació como solución a los problemas que presentaba la llamada *programación imperativa*, que se empleó antes, durante mucho tiempo.

Un programa no estructurado es un **programa procedimental**: las instrucciones se ejecutan en el mismo orden en que fueron escritas. Este tipo de programación emplea la instrucción **`goto`**. Una instrucción **`goto`** permite pasar el control a cualquier otra parte del programa, es decir “hacer saltos a otra sección del código”.

Cuando se ejecuta una instrucción **`goto`** la secuencia de ejecución del programa continúa a partir de la instrucción indicada en el salto.

La **programación estructurada está basada en la utilización de subrutinas y estructuras básicas**: secuencia, selección (`if` y `switch`) e iteración (`bucles for` y `while`). De esta manera, se dejó de lado el uso de **`goto`** y actualmente se lo considera una mala práctica de programación.

Programación Orientada a Objetos

La **Programación Orientada a Objetos, o POO** (*OOP - Object Oriented Programming*, según sus siglas en inglés), es un **paradigma de programación que usa objetos y sus interacciones**, con el fin de diseñar aplicaciones y programas informáticos. Está basado en varias técnicas que son sus pilares conceptuales, incluyendo: **herencia, abstracción, polimorfismo y encapsulamiento**.

Su uso se popularizó a principios de la década de los años 1990. En la actualidad, **existen variedad de lenguajes de programación que soportan la orientación a objetos**.

¿Qué son los objetos?

Los objetos son entidades que tienen un determinado estado formado por sus **atributos**, poseen **comportamiento** a través de sus métodos y tienen una **identidad**:

- **El estado** está compuesto de datos, será uno o varios atributos, a los que se habrán asignado valores concretos.
- **El comportamiento** está definido por los métodos o mensajes a los que pueden responder, es decir, qué operaciones pueden realizar.
- **La identidad** es una propiedad de un objeto que lo diferencia del resto; es decir, es su identificador (concepto análogo al de identificador de una variable o una constante).

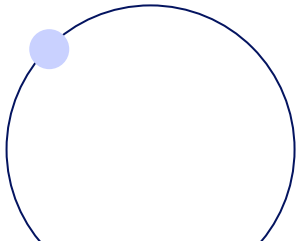
Clase

Un objeto se construye a partir de la *clase* que lo define. Una clase se puede considerar como una **plantilla que permite construir objetos similares en cuanto a su estructura**.

Un objeto contiene toda la información que hace posible definirlo e identificarlo frente a otros objetos pertenecientes a otras clases, e incluso frente a objetos de una misma clase; ya que puede tener valores bien diferentes en sus atributos.

Métodos

A su vez, **los objetos disponen de mecanismos de interacción llamados *métodos***, que favorecen la comunicación entre ellos. Esta comunicación favorece asimismo, el cambio de estado en los propios objetos. Esta característica lleva a tratarlos como unidades indivisibles, en las que no se separa el estado y el comportamiento.



Programación Orientada a Objetos: Ventajas

La POO aporta los medios para que los desarrolladores se enfrenten a nuevos retos, con:

- **Una organización modular muy cercana a la realidad**, porque un sistema se puede pensar como la siguiente definición: “conjunto de objetos que interactúan entre sí”.
- **Procesos de creación, puesta a punto y mantenimiento** de los componentes, los cuales son más sencillos y rápidos.
- **La reutilización y evolución de los componentes existentes**, o de aquellos que provienen de proveedores de software de terceros.
- **Una integración sencilla** para lograr su funcionamiento en entornos gráficos.
- **Una lógica de codificación compatible con las aplicaciones distribuidas**, que reparten sus contenidos en varias máquinas.
- **Un desacoplamiento de la aplicación**, la cual posibilita un trabajo en equipo más eficaz y productivo. Esto significa que **no existe código repetido y solapado (que hace lo mismo), en distintas partes de un sistema**.

Los principios básicos de la POO son:

- **Abstracción:** denota las características esenciales de un objeto del mundo real, sea tangible o intangible, donde se capturan sus comportamientos. El comportamiento correcto en POO consiste en **establecer relaciones entre objetos tan abstractas como sea posible**. Por ejemplo, un origen de datos puede provenir de una entrada por teclado, del contenido de un archivo, de una conexión de red, etc. Si determinada funcionalidad de un objeto recibe como argumento una sucesión de datos, será interesante abstraerse de su origen y considerarla como flujo "genérico de datos" (*stream*), sin pensar si viene del teclado o de un archivo.
- **Encapsulamiento:** significa reunir todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. El encapsulamiento consiste en crear un tipo de **caja negra** que contiene internamente el estado del objeto, y externamente un conjunto de métodos que permitan administrarlo desde el exterior. Esto implica lo siguiente: **el usuario del objeto jamás podrá acceder internamente a sus datos, sólo lo hará desde los métodos**.

- **Ocultamiento:** se refiere a que cada objeto está aislado del exterior, es un módulo natural. Y cada tipo de objeto **expone una interfaz a otros objetos, la cual especifica cómo pueden interactuar unos con otros**.
- **Herencia:** Los objetos **heredan las propiedades y el comportamiento de todas las clases a las que pertenecen**. Por ejemplo, crear una clase de base, llamada *Empleado*, que contiene la información común a todos los empleados de una empresa. Después, crear una clase llamada *Ejecutivo*, que heredará de la clase base *Empleado*, los miembros (propiedades y métodos) de *Empleado* y agregará a esta lista de miembros, los aspectos del tipo ”Ejecutivo”. En este caso, se dice que *Ejecutivo* extiende *Empleado* o que *Ejecutivo* hereda de *Empleado*. Es decir, *Ejecutivo* es la clase derivada y *Empleado* es la clase base.
- **Polimorfismo: permite tener comportamientos similares, asociados a objetos distintos**. La raíz etimológica de la palabra conduce a pensar de manera natural que el objeto puede comportarse de distintas maneras.

educación IT
CODE YOUR FUTURE

¡Sigamos trabajando!