# ACTIVIDAD: Métodos Estáticos en Programación Orientada a Objetos

Los métodos estáticos son funciones que pertenecen a una clase en lugar de a una instancia específica de esa clase. Esto significa que se pueden invocar sin necesidad de crear un objeto de la clase. Los métodos estáticos son particularmente útiles para operaciones que no requieren el estado de un objeto, lo que los hace ideales para realizar tareas que son comunes a todos los objetos de la clase. Por ejemplo, se pueden utilizar para definir funciones utilitarias, realizar cálculos matemáticos, o gestionar recursos que no dependen del estado de una instancia.

Como hemos visto hasta ahora las clases nos sirven como modelos con un fin: *crear objetos*. Las clases tienen métodos y estos pueden invocarse desde el objeto que ha sido creado. Sin embargo, las clases no se limitan solamente a eso, servir de modelo para la creación de un objeto, sino que dentro de las clases nosotros podemos definir métodos independientes llamados **métodos estáticos** utilizando la palabra reservada **static** antes del nombre del método:

class NombreDeLaClase {

public static void nombreDelMetodo(){

}

}

En el contexto de la programación orientada a objetos (POO), los métodos estáticos ofrecen un enfoque diferente a la encapsulación de la lógica. A menudo, se asocian con funciones de utilidad que se pueden llamar sin crear instancias. Por ejemplo, en una clase MathUtils que contiene un método estático CalculateSquare, se podría calcular el cuadrado de un número sin necesidad de instanciar la clase: MathUtils.CalculateSquare(5). Esto no va en contra de los principios de la POO, sino que complementa su uso. La POO se centra en la creación de objetos que representan entidades del mundo real, y los métodos estáticos pueden ayudar a manejar operaciones que no se asocian directamente a un objeto, facilitando así la organización del código y la reutilización de funciones comunes.

En el contexto de la programación orientada a objetos (POO), los métodos estáticos ofrecen un enfoque diferente a la encapsulación de la lógica.

La invocación de un método estático se realiza a través del nombre de la clase, seguido del operador de acceso a miembros (punto). Por ejemplo, en C#, si se tiene una clase llamada MathOperations con un método estático llamado Add, se puede invocar de la siguiente manera: MathOperations.Add(5, 10). Esta sintaxis es clara y refleja que el método estático no está vinculado a una instancia de la clase, lo que facilita la lectura y el entendimiento del código.

Una diferencia clave entre los métodos estáticos y los métodos de instancia (o "normales") radica en su relación con el estado del objeto. Los métodos de instancia pueden acceder a los atributos de la clase y manipular el estado del objeto en función de su contexto. En cambio, los métodos estáticos no pueden acceder a los atributos de instancia a menos que se les pase explícitamente una referencia al objeto. Esto significa que los métodos estáticos son más limitados en su interacción con los datos del objeto, pero pueden ser más eficientes en términos de rendimiento, ya que no requieren la creación de instancias.

Entre las características de los métodos estáticos se incluyen: no pueden ser sobreescritos en una subclase, ya que están ligados a la clase base, y no pueden acceder a miembros de instancia directamente. Además, son una herramienta poderosa para la organización del código, permitiendo agrupar funciones relacionadas en una clase específica sin necesidad de crear instancias. Por estas razones, los métodos estáticos son ampliamente utilizados en la programación, especialmente en el desarrollo de bibliotecas y frameworks, donde la reutilización de código es crucial.

**Preguntas para Reflexionar**

¿Qué son los métodos estáticos y en qué se diferencian de los métodos de instancia en programación orientada a objetos?

¿Por qué los métodos estáticos no van en contra de los principios de la programación orientada a objetos?

¿Cómo se invocan los métodos estáticos y cuál es su sintaxis en un lenguaje de programación como C#?

¿En qué situaciones es más apropiado utilizar un método estático en lugar de un método de instancia?

¿Cuáles son algunas de las limitaciones de los métodos estáticos en comparación con los métodos de instancia?