

PROGRAMA DE ESTUDIOS

DESARROLLO DE SISTEMAS DE INFORMACIÓN

**HERRAMIENTAS DE
PROGRAMACION –
C#**

Tema

INTRODUCCIÓN AL CURSO

PARADIGMAS DE LOS LENGUAJES DE PROGRAMACIÓN

Un paradigma de programación es una forma de abordar la resolución de problemas mediante la programación. Los paradigmas de programación definen cómo se estructura y se organiza el código, y cómo se interactúa con la computadora. Hay varios paradigmas de programación, cada uno con sus propias características y ventajas.

TIPOS DE PARADIGMAS

Paradigma Imperativo

El paradigma imperativo es el más común y se basa en la ejecución de instrucciones secuenciales para lograr un objetivo. En este paradigma, el programador especifica exactamente qué acciones debe realizar la computadora y en qué orden.

Características clave:

- **Secuencialidad:** El código se ejecuta en orden secuencial.
- **Mutabilidad:** Los valores de las variables pueden cambiar durante la ejecución del programa.
- **Control de flujo:** El programador controla el flujo de ejecución del programa mediante instrucciones de control de flujo (if, while, for, etc.).

Ejemplos de lenguajes imperativos: C, C++, Java, Python

Paradigma Funcional

El paradigma funcional se centra en la evaluación de funciones puras, que no tienen efectos secundarios y siempre devuelven el mismo resultado para los mismos argumentos. En este paradigma, el programador define funciones que se componen para lograr un objetivo.

Características clave:

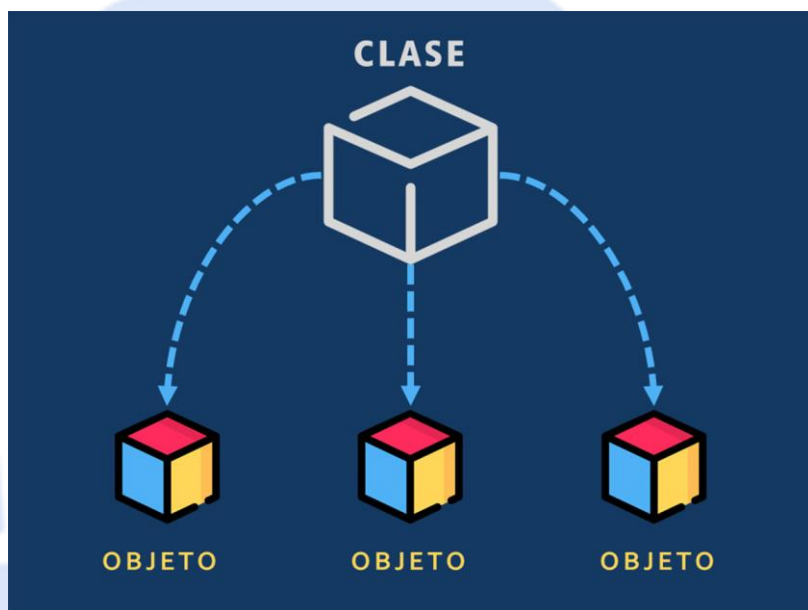
- **Inmutabilidad:** Los valores de las variables no cambian durante la ejecución del programa.

- Funciones puras: Las funciones no tienen efectos secundarios y siempre devuelven el mismo resultado para los mismos argumentos.
- Composición: Las funciones se componen para lograr un objetivo.

Ejemplos de lenguajes funcionales: Haskell, Lisp, Scala, Clojure

Paradigma Orientado a Objetos

El paradigma orientado a objetos se centra en la creación de objetos que interactúan entre sí para lograr un objetivo. En este paradigma, el programador define clases que representan objetos y sus comportamientos.



Características clave:

- Abstracción: Los objetos se definen en términos de sus características y comportamientos.
- Encapsulación: Los objetos ocultan sus detalles internos y solo exponen sus interfaces.

- Herencia: Las clases pueden heredar comportamientos de otras clases.

Ejemplos de lenguajes orientados a objetos: Java, C#, C++, Python

Paradigma Declarativo

El paradigma declarativo se centra en la especificación de lo que se quiere lograr, sin preocuparse por cómo se logra. En este paradigma, el programador define reglas y restricciones que se deben cumplir.

Características clave:

- Declaratividad: El programador especifica lo que se quiere lograr, sin preocuparse por cómo se logra.
- Reglas y restricciones: Se definen reglas y restricciones que se deben cumplir.

Ejemplos de lenguajes declarativos: Prolog, SQL

Paradigma Lógico

El paradigma lógico se centra en la deducción de conclusiones a partir de hechos y reglas. En este paradigma, el programador define reglas y hechos que se utilizan para deducir conclusiones.

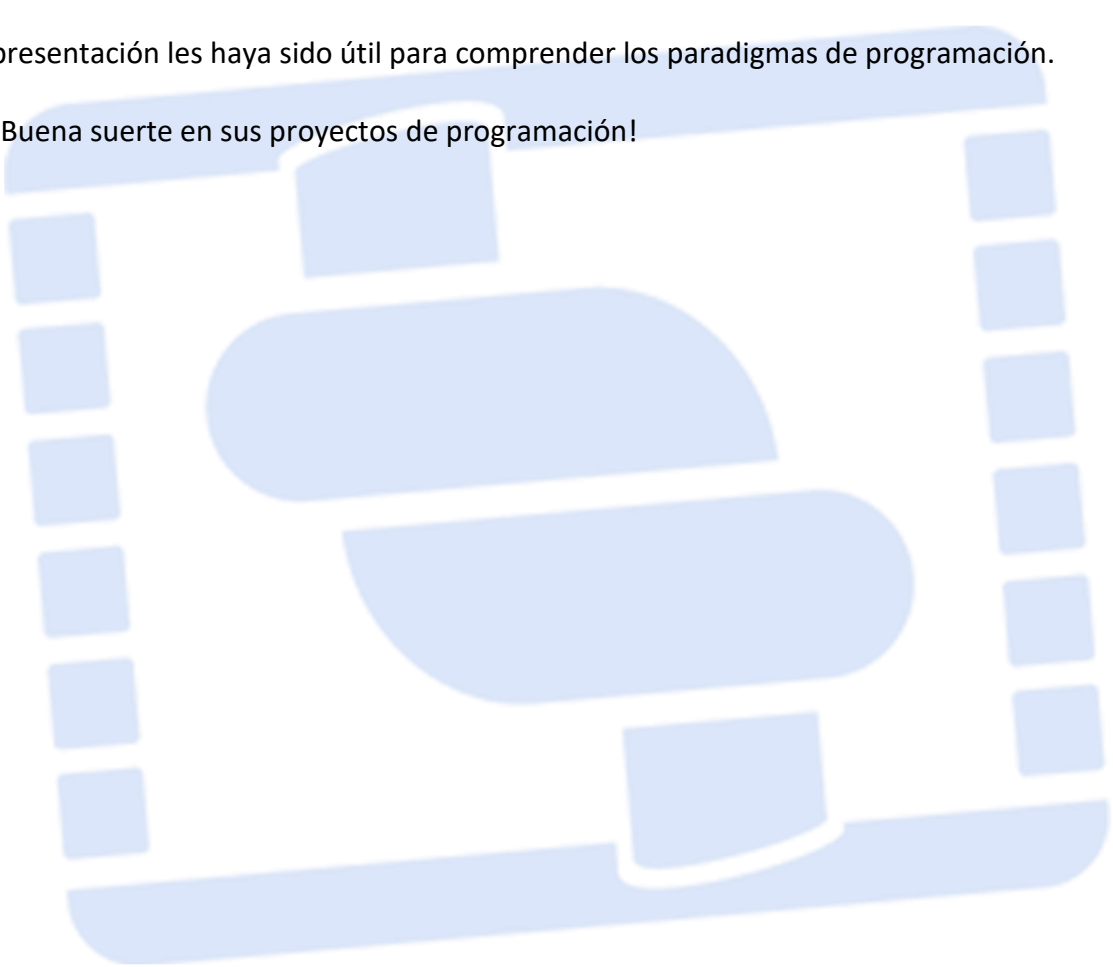
Características clave:

- Deducción: El programa deduce conclusiones a partir de hechos y reglas.
- Reglas y hechos: Se definen reglas y hechos que se utilizan para deducir conclusiones.

Ejemplos de lenguajes lógicos: Prolog, Mercury

Conclusión

En resumen, cada paradigma de programación tiene sus propias características y ventajas. Los programadores deben entender los conceptos clave de cada paradigma para elegir el más adecuado para resolver un problema específico. Espero que esta presentación les haya sido útil para comprender los paradigmas de programación.
¡Buena suerte en sus proyectos de programación!





INSTITUTO
KHIPU