

- *Resumen*

El lenguaje de programación C es uno de los más clásicos y fundamentales en el mundo de la informática. Fue creado en los años 70 por Dennis Ritchie y Brian Kernighan con la intención de tener un lenguaje eficiente, flexible y que permitiera escribir programas potentes sin ser tan complejo como el lenguaje ensamblador. Aunque fue diseñado inicialmente para programar sistemas operativos (como UNIX), hoy en día se puede usar para desarrollar casi cualquier tipo de software.

C se considera un lenguaje estructurado, lo que significa que te permite organizar el código en partes más pequeñas y manejables llamadas funciones. Sin embargo, a diferencia de otros lenguajes estructurados como Pascal, en C no puedes poner funciones dentro de otras funciones. Otro punto fuerte del lenguaje es que está pensado para que cada instrucción en C se pueda traducir fácilmente a instrucciones de bajo nivel, lo cual hace que los programas escritos en C sean muy rápidos.

Con el tiempo, muchas compañías hicieron sus propias versiones de C, lo que causó diferencias entre ellas. Para solucionar esto, en 1983 se creó un estándar oficial conocido como ANSI C, que es el que la mayoría de los programadores usa hoy en día y que funciona en cualquier tipo de computadora.

Una de las grandes ventajas de C es que es portátil, es decir, puedes escribir un programa en una computadora y adaptarlo fácilmente para que funcione en otra. Además, C combina lo mejor de dos mundos: tiene características de lenguajes de alto nivel (como funciones y estructuras de control) y también te da acceso a bajo nivel, como trabajar directamente con memoria a través de punteros.

Un programa en C siempre tiene una estructura parecida: se empieza con comentarios (para explicar qué hace el código), luego se incluyen archivos o bibliotecas con funciones útiles, y después viene la función principal, llamada main, donde se declara qué variables se van a usar y se escriben las instrucciones que forman el programa. Si se necesitan más funciones, el programador las puede crear y usar desde main.

Entre las principales características de C se encuentran su simplicidad, su flexibilidad para programar con distintos estilos, su sistema de tipos que evita errores graves, el uso de un preprocesador para manejar macros o incluir archivos, y su capacidad para acceder directamente a la memoria. También permite crear estructuras (struct) para agrupar datos relacionados,

y aunque no tiene soporte nativo para programación orientada a objetos, se pueden simular muchas de esas funciones usando punteros y estructuras.

Eso sí, C carece de algunas cosas modernas, como la recolección automática de basura (tú mismo tienes que liberar la memoria), el soporte nativo para objetos o para multihilo. Aun así, estas limitaciones hacen que C sea más directo y dé al programador el control total sobre lo que pasa, lo que lo convierte en una opción muy eficiente. De hecho, solo el código escrito cuidadosamente en ensamblador puede superar en rendimiento al de C, aunque esa diferencia cada vez es menor gracias a los avances en los compiladores.

Por último, declarar variables en C es muy sencillo: primero se pone el tipo (como int, float, etc.) y luego el nombre. Por ejemplo: `int contador;`, `float precio = 10.5;`, o `int a, b, c = 0;`.

En resumen, C es un lenguaje poderoso, rápido y confiable, ideal para aprender a programar de verdad y entender cómo funcionan los computadores por dentro. No tiene tantas comodidades como otros lenguajes modernos, pero lo que le falta en facilidades, lo compensa con control, eficiencia y versatilidad.

- *¿Por qué Lenguaje de la Programación C?*

Escogí este texto porque es algo que vamos a ver mas adelante y me puede funcionar para entender un poco del tema, ya que me encuentro perdida en el tema. Porque ofrece una introducción clara y completa al lenguaje de programación C, uno de los pilares fundamentales en el desarrollo de software. Me pareció importante porque no solo explica qué es C y cómo se estructura un programa, sino que también presenta sus características, ventajas y limitaciones de forma sencilla. Además, permite entender mejor el valor histórico y técnico de C, y cómo ha influido en otros lenguajes modernos como C++, Java o Python. Es un buen punto de partida para cualquier persona que quiera aprender programación.