



# *Programación Estructurada*

## *Fundamentos e Historia del Lenguaje C*

*Yeraldi Adolfo Treviño Herrera -  
2079052*

*Lic. Ciencias Computacionales*

## Índice

Introducción.....	2
La historia del lenguaje C.....	3
Fundamentos del lenguaje C.....	4
Variables, constantes y tipos de datos. ....	4
Operadores en el lenguaje C.....	5
Operadores aritméticos.....	5
Operadores relacionales.....	5
Operadores lógicos.....	5
Librerías o bibliotecas.....	6
Librerías más usadas en C.....	6
Estructura de un programa en C.....	7
Conclusión.....	9
Bibliografía.....	10

## Introducción.

El lenguaje C es uno de los lenguajes que actualmente ha tenido mucho tiempo manteniéndose en el mercado y que se reconoce por muchas cosas. A lo largo de la historia de este lenguaje hemos visto que se han podido crear distintas cosas con este mismo, entre ellas; sistemas operativos, lenguajes de programación, compiladores e intérpretes para los lenguajes interpretados como Python.

El lenguaje C es uno de los lenguajes que se ha mantenido desde su creación por la facilidad que este ofrece para la creación de múltiples cosas como las mencionadas anteriormente, ya que este mismo tiene la capacidad de poder manipular de manera directa la memoria, sin la necesidad de ser un lenguaje de nivel alto, ya que el lenguaje combina ambas cosas (la capacidad del lenguaje Ensamblador que nos ayuda a manipular directamente la memoria y la forma de los lenguajes de alto nivel la cual nos facilita la programación en el mismo). Este lenguaje es el antecesor al lenguaje C++, el cual está basado en C, pero con la diferencia que este agrega el paradigma o la programación orientada a objetos, mientras que en C tenemos el paradigma de la programación estructurada y modular, es decir el programar con funciones para organizar y optimizar mejor el código.

Un dato curioso es que el nombre de este lenguaje de programación "C" fue elegido porque es el que seguía de "B" el lenguaje del cual viene C, teniendo así este nombre que hoy en día todos conocemos.

## La historia del lenguaje C.

El lenguaje C, es la evolución de dos lenguajes de programación llamados BCPL y B, los cuales manejan el paradigma imperativo. El lenguaje C fue creado en el año 1972 por Dennis Ritchie junto con Ken Thompson, las cuales son las mismas personas que diseñaron el sistema operativo llamado Unix. Sabemos que la mayor parte de sistemas operativos de hoy en día, están diseñados con el lenguaje de programación C/C++, ya que este mismo facilita que podamos manipular directamente la memoria lo cual da facilidad



para que se puedan programar este tipo de cosas. Antes de 1983, los sistemas operativos se creaban con este lenguaje de programación sin embargo las versiones que se utilizaban podrían variar en muchas cosas, haciendo que todo sistema creado sea totalmente diferente, generando así un problema para los que necesitaran leer el código fuente de estos. No fue hasta el año 1983 que se creó un comité para crear o establecer estándares con este lenguaje, para así poder resolver las discrepancias de las versiones de este lenguaje llamado C, este comité empezó a trabajar en un estándar llamado ASCII C, el cual fue completado en el año 1989, y es el estándar que actualmente se usa para este lenguaje de programación, teniendo como resultado que las versiones no varíen de sí mismas y evitando que haya un desastre con todos los sistemas y que el código sea 100% entendible para todo aquel que lo lea y utilice. Este estándar creado incluye una gran cantidad de bibliotecas con funciones que incluyen cosas como: entradas y salidas, tratamiento de textos, funciones matemáticas, y muchas cosas más.

## Fundamentos del lenguaje C.

El objetivo con el que se creó este lenguaje de programación fue para que sea útil y entendible para los programadores y no programadores, este lenguaje posee las características de ser: moderno, eficiente, portátil, potente y flexible, y Compilado como se puede deducir por su forma de escribir código en él.

### Variables, constantes y tipos de datos.

Sabemos que cualquier programa necesita de los datos para poder trabajar, existen dos tipos de identificadores para los datos “básicos”, una de ellas son las variables; las cuales nos ayudan a guardar un tipo de dato en específico en un espacio de memoria en la computadora y este valor puede ir cambiando conforme la ejecución del programa. El otro tipo de identificador se llama constante; esta también almacena un tipo de dato en la memoria, con la peculiaridad de que este dato no cambiara en toda la ejecución del programa, es decir que este dato tendrá el mismo valor que se le asigne desde que empiece hasta que termine la ejecución de este.

En las variables y constantes podemos guardar distintos tipos de datos entre ellos están los siguientes:

Tipo de dato	Tamaño En Memoria	Descripción
<b>int</b>	2 o 4 bytes	Almacena números sin decimales
<b>float</b>	4 bytes	Almacena números que tiene de 1 a 7 decimales
<b>double</b>	8 bytes	Almacena números que contienen hasta 15 decimales
<b>char</b>	1 byte	Almacena solamente un carácter como: letra, numero o un valor de tipo ASCII

## Operadores en el lenguaje C.

Los operadores nos ayudan a realizar operaciones aritméticas, comparar datos entre sí, y relacionar a los mismos datos. Para lograr esto el lenguaje cuenta con operadores, que no son más que símbolos o combinaciones de ellos los cuales tenemos cualquiera de nosotros en nuestro teclado y se usan para distintas ocasiones. Normalmente utilizamos los operadores aritméticos para hacer operaciones aritméticas, los operadores relacionales los solemos usar en las estructuras repetitivas o de decisión al igual que los lógicos.

### Operadores aritméticos.

Operador	Nombre	Descripción
+	Suma	Sumar
-	Resta	Restar
*	Multiplicación	Multiplicar
/	División	Dividir
%	Modulo	Resto de una división
++	Incremento	Incrementa o suma 1 al valor de una variable
--	Decremento	Decrementa o resta 1 al valor de una variable

### Operadores relacionales.

Operador	Nombre
==	Igual que
!=	Diferente que
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que

### Operadores lógicos.

Operador	Nombre	Descripción
&&	Y	Retorna Verdadero si ambas comparaciones son ciertas
	O	Retorna Verdadero si una de las comparaciones es cierta
!	No	Retorna falso si el resultado es verdadero

## Librerías o bibliotecas.

En C/C++ existen archivos que les llamamos librerías, estas normalmente se guardan con la extensión “.h”, estos archivos contienen funciones ya predeterminadas o predefinidas, que podemos usar para hacer diversas cosas como: entrada y salida de datos, operaciones matemáticas un poco más complejas, acceder al tiempo y día marcado en el sistema operativo, etc. Existen muchas librerías en el lenguaje de programación C y si queremos podemos utilizar todas en un mismo programa, sin embargo, no se suele hacer ya que cada programa tiene su propósito y por lo tanto es posible que no utilice todas las librerías que hay en c. También podemos nosotros mismos crear, mantener y actualizar nuestra propia librería con nuestras propias funciones o nuestros propios métodos, adaptados a nuestras necesidades y comodidades.

En el lenguaje c podemos utilizar la librería poniendo los sig. al inicio de nuestro programa: `#include <nombre_libreria>`

## Librerías más usadas en C.

A pesar de existir una gran variedad de librerías, es muy común que se utilicen unas cuantas, con más frecuencia, algunas de estas librerías son:

`stdio.h`

Esta librería es por así decirlo, obligatorio ponerla en cualquier programa, esto debido a que esta, nos ayuda a mantener la entrada y salida de los datos que ingresemos, si no la ponemos prácticamente no podríamos hacer mucho con nuestros programas. Sus siglas quieren decir: “**standar input output**”.

`<time.h>`

La librería time.h, nos ayuda a gestionar o acceder al tiempo que tenemos en nuestro dispositivo, algunas funciones que contiene esta librería son: generar números aleatorios, medir tiempos de ejecución, tomar la fecha y hora de nuestro dispositivo de manera exacta, entre otras más.

`<math.h>`

**math.h** nos ayuda a hacer operaciones aritméticas un poco más complejas para hacerlas o aplicarlas en un programa, algunas de sus funciones son: cálculo de potencias, el cálculo de raíces cuadradas, las funciones trigonométricas, etc.

`<string.h>`

Esta es otra de las librerías más usadas debido a que en el lenguaje C, no se cuenta con el tipo de dato "string" la cual es una cadena de caracteres, sin embargo esta librería nos ayuda a manipular los arreglos o cadenas de caracteres que nosotros creamos como si fueran este tipo de dato anteriormente mencionado. Algunas funciones que incluye son: comparación de cadena de caracteres, copia de cadena de caracteres, entre otras más.

### Estructura de un programa en C.

Pondremos un pequeño ejemplo o programa para introducirnos un poco a la programación, nuestro objetivo es imprimir un "Hola Mundo" en pantalla o consola. Para esto empezaremos abriendo un editor de código, texto o IDE de nuestra preferencia; crearemos un archivo con la extensión .c, no importa que nombre utilices un ejemplo puede ser **holamundo.c**.

Para que podamos hacer que el programa imprima o nos muestre por consola lo que queremos primeramente tenemos que llamar a la librería de entradas y salidas así que al principio de nuestro archivo pondremos lo siguiente: **#include <stdio.h>**

Después de eso tendremos que crear un método o función principal la cual es necesaria para que se pueda ejecutar de manera correcta el programa, esta función se llamara main, cuando terminemos de hacer eso nuestro archivo debe verse así, sin embargo aún falta agregar una cosa más, en este lenguaje por estándar se retorna el valor 0 en la función main recién creada así que eso es lo que agregaremos, quedando tal que así.

```
#include <stdio.h>
```

```
int main()  
{  
  
}
```

```
#include <stdio.h>
```

```
int main()  
{  
    /* Código aqui */  
    return 0;  
}
```

Finalmente, al haber creado nuestro método main, podemos empezar a escribir código, dentro de las llaves del mismo método y sobre la palabra "return". Para



poder mostrar por consola lo que necesitamos, utilizaremos la función “printf()”, ya que esta es una función de la librería stdio.h mencionada anteriormente, dentro de los paréntesis escribiremos entre comillas dobles lo que deseamos imprimir o mostrar, en este caso “Hola Mundo”, al finalizar, no olvides agregar el punto y coma después del paréntesis, ya que si no se agrega marcará error de sintaxis. Al terminar el código del programa debería quedar como en la imagen que se muestra.

```
#include <stdio.h>

int main()
{
    /* Código aquí */
    printf("Hola Mundo");
    return 0;
}
```

Para poder ver el resultado de nuestro programa debemos de compilar el código, podemos hacerlo de manera automática con un editor de código o manualmente, en este caso lo haremos manualmente, así que abriremos una consola de comandos o conocida como CMD en Windows y navegaremos hasta la carpeta donde está guardado nuestro archivo. Cuando estemos en la carpeta desde la terminal, podremos usar el comando “ls” para ver los archivos y carpetas que tenemos, deberíamos de ver el nombre de nuestro archivo con la extensión .c, una vez estemos seguros de estar en el lugar correcto

```
C:\Users\ytrev\OneDrive\Documentos\Ciencias_Computacionales\Programación
Estructurada\Clase Normal>ls
'Lenguaje C.docx'    holaMundo.c

C:\Users\ytrev\OneDrive\Documentos\Ciencias_Computacionales\Programación
Estructurada\Clase Normal>gcc holaMundo.c -o holaMundo

C:\Users\ytrev\OneDrive\Documentos\Ciencias_Computacionales\Programación
Estructurada\Clase Normal>ls
'Lenguaje C.docx'    holaMundo.c    holaMundo.exe

C:\Users\ytrev\OneDrive\Documentos\Ciencias_Computacionales\Programación
Estructurada\Clase Normal>
```

utilizaremos el sig. Comando: “gcc **nombreArchivo.c** -o **nombreArchivo**”. Cuando hagamos esto, se nos creará un archivo con extensión .exe, el cual será el programa ya terminado, así que en la consola simplemente escribiremos el nombre del archivo con su extensión, una vez ejecutado en la consola nos deberá de aparecer escrita la frase “Hola Mundo”.

```
C:\WINDOWS\system32\cmd. x + v
C:\Users\ytrev\OneDrive\Documentos\Ciencias_Computacionales\Programación
Estructurada\Clase Normal>holaMundo.exe
Hola Mundo
C:\Users\ytrev\OneDrive\Documentos\Ciencias_Computacionales\Programación
Estructurada\Clase Normal>
```

## Conclusión.

Finalmente podemos apreciar que el lenguaje C tiene una historia muy grande por detrás, no se convirtió en un lenguaje de programación bueno de un día para otro, simplemente pasaron varios años para que este pudiera corregir muchos de sus errores y fallas, sin embargo, se puede observar que la potencia del lenguaje tenía y tiene un gran impacto debido a que hubo la necesidad de crear un comité para que se pudiera estandarizar el lenguaje. Todos los cambios que ha tenido lo han hecho el lenguaje de programación preferido para mucho, en el ámbito de sistemas operativos y microprocesadores, debido a su funcionalidad de ensamblador y su forma de lenguaje de programación de nivel alto.

Hablando no solo del lenguaje C, si no de todos los lenguajes de programación en general, podemos concluir que no existe el “mejor lenguaje de programación”, todos los lenguajes de programación son buenos, solamente tienen librerías, bibliotecas u otras cosas que hacen que destaquen en ciertas áreas por la facilidad que este tiene de acceder o crear. En este caso el lenguaje C, es el lenguaje mas usado para la creación de sistemas operativos y microprocesadores, por la facilidad que nos da de acceder a la memoria como se menciono anteriormente, y su historia puede hacer que nazcan mas lenguajes de programación de este estilo para facilitar muchas tareas en un futuro próximo, tal como el caso de rust un nuevo lenguaje de programación que esta siendo muy popular para la creación de algunos entornos para el sistema operativo Linux y que tiene alguna similitud con el actual C.

## Bibliografía.

- Durán, J. (2017, 4 diciembre). ▷ Las librerías más usadas en C. Lenguaje de programación. Recuperado 11 de agosto de 2022, de [https://lenguajedeprogramacion.com/programacion-c/librerias-mas-usadas-c/#La libreria de C para la gestion de tiempos](https://lenguajedeprogramacion.com/programacion-c/librerias-mas-usadas-c/#La%20libreria%20de%20C%20para%20la%20gestion%20de%20tiempos)
- C Operators. (s. f.). Learn C. Recuperado 11 de agosto de 2022, de [https://www.w3schools.com/c/c\\_operators.php](https://www.w3schools.com/c/c_operators.php)
- EXT\_ocw\_informatica\_01: Guion 1: Fundamentos básicos del lenguaje C. (s. f.). Fundamentos básicos del lenguaje C. Recuperado 11 de agosto de 2022, de <https://ocw.uca.es/mod/resource/view.php?id=987#:~:text=El%20Lenguaje%20C%20fue%20creado,y%20poseen%20un%20dise%C3%B1o%20modular.>
- Stack Overflow Developer Survey 2021. (s. f.). Stack Overflow. Recuperado 11 de agosto de 2022, de <https://insights.stackoverflow.com/survey/2021#most-popular-technologies-language-prof>