



INTRODUCCIÓN AL LENGUAJE DE PROGRAMACIÓN

1. ¿QUÉ ES UN LENGUAJE DE PROGRAMACIÓN?

Un lenguaje de programación es un lenguaje diseñado para escribir el conjunto de instrucciones que una computadora debe ejecutar.

Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a una computadora.

2. LENGUAJES DE PROGRAMACIÓN MAS UTILIZADOS

Existen más de 8,000 lenguajes de programación, aunque solo una pequeña fracción de ellos se utiliza de manera generalizada en la industria.

En cuanto a los lenguajes de programación más populares en la actualidad, aquí tienes algunos de los más destacados:

PYTHON: Destacando en análisis de datos, aprendizaje automático y desarrollo web. Su sintaxis fácil de aprender y sus vastas bibliotecas lo hacen ideal tanto para principiantes como para programadores experimentados.

JAVASCRIPT: El rey indiscutible del desarrollo web. Desde los elementos interactivos que ves en las páginas hasta la lógica detrás de escena, JavaScript hace que la web funcione.

JAVA: Tiene fuerza dominante en grandes empresas, gracias a su estabilidad y capacidad para manejar proyectos masivos. También es fundamental para el desarrollo de aplicaciones Android.

C#: La elección preferida para el desarrollo de videojuegos inmersivos y software empresarial. Ofrece potencia y flexibilidad para ambos campos.

C++: Un lenguaje de alto rendimiento utilizado principalmente para programación de sistemas, sistemas embebidos, desarrollo de videojuegos, software de audio e imagen, y más. Es la base de muchos otros lenguajes.

C: Considerado por muchos como el lenguaje de programación más influyente de la historia, ya que sirvió de base para C++, C#, Java, Python y JavaScript, entre otros.

PHP: Aunque ha perdido algo de popularidad en los últimos años, sigue siendo importante para el desarrollo web, especialmente para proyectos existentes y la gestión de contenido.

Esta lista representa los lenguajes más utilizados e importantes actualmente. La elección del lenguaje a aprender o utilizar a menudo depende del tipo de proyecto que tengas en mente y de tus intereses.

3. LENGUAJE DE ALTO Y BAJO NIVEL:

3.1. Lenguajes de bajo nivel:

- Son lenguajes que están muy cerca del hardware de la computadora.
- Ejemplos: lenguaje de máquina (binario) y lenguaje ensamblador.
- Ventajas: alto rendimiento y control preciso del hardware.
- Desventajas: difíciles de escribir y entender, específicos de cada arquitectura de computadora.

3.2. Lenguajes de alto nivel:

- Son lenguajes más cercanos al lenguaje humano.
- Ejemplos: Python, Java, C++, JavaScript.



- Ventajas: más fáciles de escribir y entender, portables (pueden ejecutarse en diferentes plataformas).
- Desventajas: menor rendimiento en comparación con los lenguajes de bajo nivel.

4. ESTRUCTURA GENERAL DE UN PROGRAMA.

Un programa de computadora típicamente tiene la siguiente estructura:

Entrada de datos: El programa recibe datos del usuario o de otras fuentes.

Procesamiento de datos: El programa realiza cálculos y operaciones con los datos.

Salida de datos: El programa muestra los resultados al usuario o los envía a otro destino.

5. CODIFICACIÓN Y EJECUCIÓN DE UN PROGRAMA.

5.1. ¿Qué es C++?

C++ es un lenguaje de programación creado a mediados de 1980. La intención de su creador fue extender el exitoso lenguaje "C" con mecanismos que permitan la manipulación de objetos.

5.2. ¿Qué es un compilador?

Permite traducir el código fuente de un programa en lenguaje máquina. De esta manera un programador puede diseñar un programa en un lenguaje mucho más cercano a cómo piensa un ser humano, para luego compilarlo a un lenguaje más manejable por una computadora.



Estructura Básica de un programa en C++:

```

#include<iostream>    ← Declaración de librerías
using namespace std; ← Definición del espacio de nombres std
int main()           ← Declaración de la función principal (main)
{                    ← Inicio de bloque
    // cuerpo del programa
    return 0;
}                    ← fin de bloque
    
```

Diagramas de flujo en C++. - Para aprender a programar es recomendable primeramente aprender a realizar algoritmos mediante Pseudocódigo y Diagramas de Flujo, como nosotros ya sabemos realizar diagramas de flujo el siguiente paso será codificar al lenguaje C++ los algoritmos realizados mediante diagramas de flujo, para lo cual, primero veremos la representación en C++ de los símbolos de los diagramas de flujo:

Nota.- En C++ es necesario declarar a un principio las variables que serán utilizadas en el programa.



Variables: Una variable se define como espacios de memoria que almacena tipos de datos, una variable es un identificador único que se utiliza para almacenar todos los datos generados durante la ejecución de un programa, por ejemplo, num1, num2, con, con1, con2, con3, sum, suma, A, B, C, etc.

Existen ciertas **reglas** en cuanto a variables:

- Las variables deben ser claras y con referencia directa al problema.
- No debe tener espacios blancos, ni símbolos extraños en ellas (por ejemplo, con 1, num 1, &e, %r, etc.)
- No deben ser “palabras reservadas” del lenguaje (por ejemplo: cin, if, while, void, switch, else, int, short, etc.).

Tipos de datos en C++ (char, int, float, double, long y string).

Tipo de dato	Palabra reservada c++	¿Qué representa?	Ejemplo de asignación de variables
Entero	int	Un entero hasta 10 dígitos	int num=0
Entero	short int	Un entero corto hasta 5 dígitos	short int num1=12345
Entero	long int	Un entero con mayor rango de validez hasta 19 dígitos.	long int num2=1234567889
Real	float	Un real, contiene de 6 – 7 dígitos decimales	float numero=1.5
Real	double	Un real de doble precisión, contiene de 15 – 16 dígitos decimales	double decimal=6.4545454
Real	long double	Un real de doble precisión y mayor, contiene de 15 – 16 dígitos decimales o mas	long double deci =23,1212121212
Carácter	char	Un carácter	char letra='a'
Cadena	char	Una cadena de caracteres	char palabra[10]="HOLA"
Booleano o lógico	bool	Verdad (true) y falso (false)	bool estado=true bool estado=1

Participación: Para la asignación de una variable utilizamos el símbolo de la igualdad “=”

Tipo de dato	Palabra reservada c++	Ejemplo de asignación de variables	Ejemplo de asignación de variables
Entero	int		
Entero	short int		
Entero	long int		
Real	float		
Real	double		
Real	long double		
Carácter	char		
Cadena	char		
Booleano	bool		



Asignación de variables

En C++, la asignación de valores a variables se realiza utilizando el operador de asignación "=". Por ejemplo, `int edad = 25;` declara una variable entera llamada edad y le asigna el valor 25.

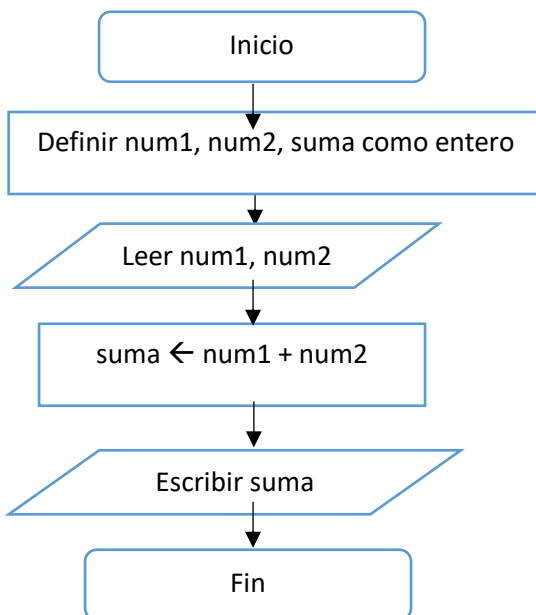
Operadores aritméticos en C++. - Un operador es un símbolo especial que indica al compilador que debe efectuar una operación matemática o lógica.

C++ reconoce los siguientes operadores aritméticos:

Operación	Operador	Ejemplo	Acción
Suma	+	$X + Y$	Suma X más Y
Resta	-	$X - Y$	Resta X menos Y
Multiplicación	*	$X * Y$	Multiplica X por Y
División	/	X / Y	Divide X entre Y, regresa el cociente
Potencia	pow	pow(X,Y)	Eleva X a la Y
Residuo	%	$X \% Y$	Regresa el residuo de X entre Y
Raíz cuadrada	sqrt	sqrt(Y)	Raíz cuadrada de Y

SUMA	+
RESTA	-
MULTIPLICACIÓN	*
DIVISIÓN	/
MÓDULO O RESIDUO	%

Ejemplo: Algoritmo que suma dos números introducidos por teclado.



PSEUDOCODIGO	C++
Principal Inicio Definir num1, num2, suma como entero Leer num1 Leer num2 suma ← num1 + num2 Mostrar suma Fin	<pre> #include <iostream> using namespace std; main() { int num1, num2, suma; cout << "Introduzca el primer numero:"; cin >> num1; cout << "Introduzca el primer numero:"; cin >> num2; suma = num1 + num2; cout << "La suma es: "; cout << suma; return 0; } </pre>

Analizamos lo que realiza el programa: A continuación, se presenta algunas variables propias o reservadas del lenguaje c++.

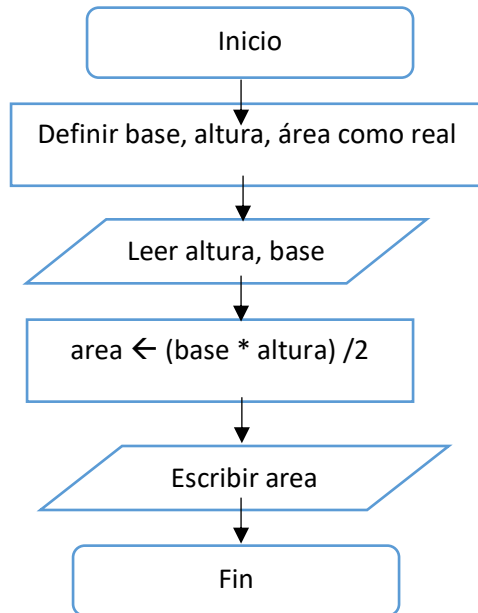
- **Int** Es el tipo de dato que nos indica que el dato introducido será de tipo entero.
- **Cin >>** Significa entrada de datos
- **Cout <<** Significa salida o mostrar datos.



Nota: Para compilar el código, se debe crear un documento en blanco y guardarlo con la extensión “.cpp”, por ejemplo “prueba.cpp”

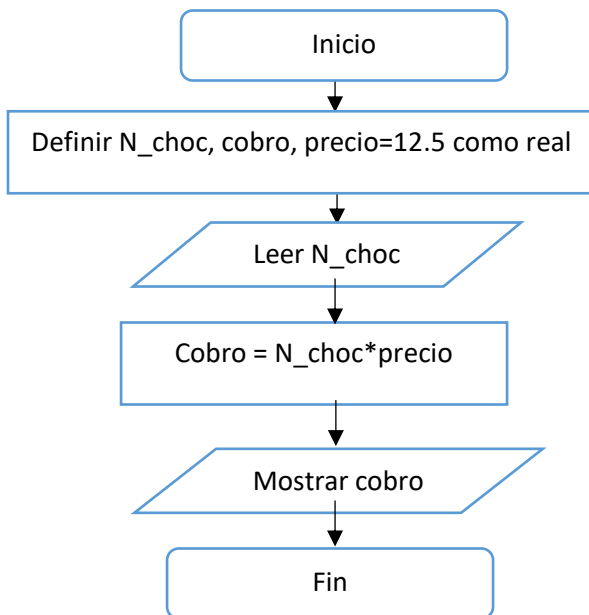
Ejercicios en clase

- 1) Realice el diagrama de flujo, pseudocódigo y código fuente que permite hallar el área de un triángulo.



PSEUDOCODIGO	C++
Principal Inicio Definir base, altura, area como entero Leer base Leer altura area ← (base * altura) / 2 Escribir area Fin	<pre> #include <iostream> using namespace std; main() { int base, altura; int area; cout << "Introduzca la base:"; cin >> base; cout << "Introduzca la altura:"; cin >> altura; area = (base*altura)/2; cout << "El área es: "; cout << area; return 0; } </pre>

- 2) Realice el diagrama de flujo, pseudocódigo y código fuente. Una tienda vende chocolates a 12,5 bolivianos, cuanto recaudará si vende N chocolates.



PSEUDOCODIGO	C++
Principal Inicio Define N_choc, cobro como real Define precio=12,5 como real Leer N_choc cobro=N_choc*precio Mostrar cobro Fin	<pre> #include <iostream> using namespace std; main() { float N_choc, cobro float precio=12,5 cout << "Introduzca la cantidad de chocolates: "; cin >> N_choc; cobro=N_choc*precio; cout << "El total a cobrar es: "; cout << cobro; return 0; } </pre>

**PRACTICA EN CLASE**

- 1) Realice el diagrama de flujo y código fuente que permite hallar la suma y el promedio de tres números.

Entrada			Salida	
2	3	4	9	3
4	6	5	15	5

- 2) Realice el diagrama de flujo y código fuente que permite hallar el volumen de un cubo.

Entrada		Salida
3		27
2		8

- 3) Realice el diagrama de flujo y código fuente que permite hallar el área y el volumen de un círculo.

Entrada	Salida
3	28,2744 63 ,6174

- 4) Realice el diagrama de flujo y código fuente. En una tienda venden helados a un valor de 3,5 bolivianos por unidad, calcular cuánto se debe pagar por N helados.

Ejemplos: Entrada	Salida
5	17.5
10	35

ACTIVIDAD

- ¿Qué es un lenguaje de programación?
- ¿Qué es un compilador?
- ¿Para qué se utiliza la variable cin>>?
- ¿Para qué se utiliza la variable cout<<?
- Realice el diagrama de flujo y código fuente. Un cierto tipo de tela se vende a 16 bolivianos por metro, calcular cuánto se debe pagar por N metros

Ejemplos: Entrada	Salida
5	45
10	90

6. Realice el diagrama de flujo y código fuente. Hallar el área y el perímetro de un Rectángulo.

Ejemplos: Entrada	Salida
4 5	20 18
5 10	50 30