

CUADERNO VIRTUAL

DAVID ALEJANDRO BALLESTEROS PADILLA

UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS

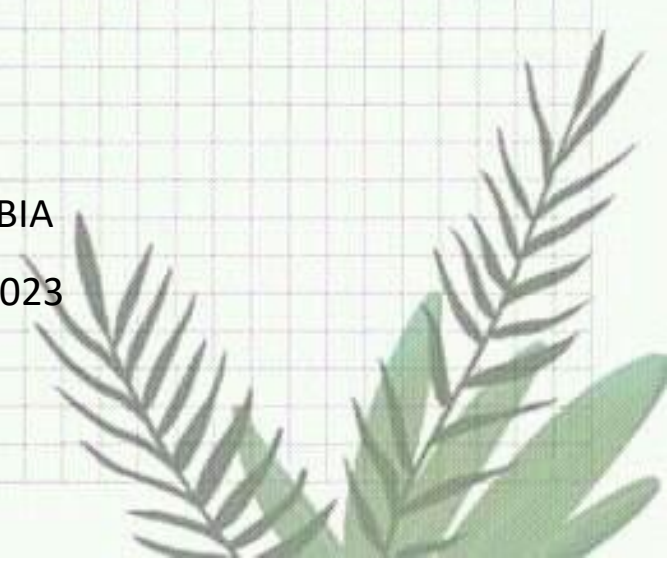
INGENIERIA ELECTRONICA

INGENIERA: ALEXANDRA ABUCHAR PORRAS

GRUPO 7

BOGOTA D.C, COLOMBIA

9 DE NOVIEMBRE DE 2023



# Programación Básica

## Horarios:

Lunes de 2 a 4pm Salón 301

Miércoles de 4 a 6pm Salón 503

Jueves de 2 a 4pm Salón 412

## Valores de los cortes

1 corte: 30%

2 corte: 40%

Final: 30%

1 firma vale=0.5 decimas

Por cada 3 puntos es una firma

Trabajo 10 agosto

Realizar un Mapa mental del syllabus de programación básica.





Trabajo 10 de agosto

Averiguar bibliografía de un libro de programación básica.

<http://www.fdi.ucm.es/profesor/luis/Fp/FP.pdf>

Trabajo 15 de agosto

Realizar una exposición grupal sobre las competencias digitales.

Trabajo 17 agosto

Consultar números decimales del 1 al 60 y pasarlos a binario, octal y hexadecimal.

Dec	Hex	Oct	Bin	Dec	Hex	Oct	Bin	Dec	Hex	Oct	Bin	Dec	Hex	Oct	Bin
0	0	000	000000000	16	10	020	00010000	32	20	040	00100000	48	30	060	00110000
1	1	001	000000001	17	11	021	00010001	33	21	041	00100001	49	31	061	00110001
2	2	002	000000010	18	12	022	00010010	34	22	042	00100010	50	32	062	00110010
3	3	003	000000011	19	13	023	00010011	35	23	043	00100011	51	33	063	00110011
4	4	004	000000100	20	14	024	00010100	36	24	044	00100100	52	34	064	00110100
5	5	005	000000101	21	15	025	00010101	37	25	045	00100101	53	35	065	00110101
6	6	006	000000110	22	16	026	00010110	38	26	046	00100110	54	36	066	00110110
7	7	007	000000111	23	17	027	00010111	39	27	047	00100111	55	37	067	00110111
8	8	010	00001000	24	18	030	00011000	40	28	050	00101000	56	38	070	00111000
9	9	011	00001001	25	19	031	00011001	41	29	051	00101001	57	39	071	00111001
10	A	012	00001010	26	1A	032	00011010	42	2A	052	00101010	58	3A	072	00111010
11	B	013	00001011	27	1B	033	00011011	43	2B	053	00101011	59	3B	073	00111011
12	C	014	00001100	28	1C	034	00011100	44	2C	054	00101100	60	3C	074	00111100
13	D	015	00001101	29	1D	035	00011101	45	2D	055	00101101	61	3D	075	00111101
14	E	016	00001110	30	1E	036	00011110	46	2E	056	00101110	62	3E	076	00111110
15	F	017	00001111	31	1F	037	00011111	47	2F	057	00101111	63	3F	077	00111111

Trabajo 17 agosto

Realizar infografía sobre evolución de los computadores en software y hardware

-Antecedentes

-Personajes

-Lenguajes de programación

-Características



# HISTORIA DEL HARDWARE

## ¿CUÁLES SON SUS ORÍGENES?



EL HARDWARE HA EXISTIDO DESDE LA INVENCIÓN DE LA COMPUTADORA. LOS PRIMEROS COMPONENTES DE HARDWARE SE FABRICABAN A PARTIR DE INTERRUPTORES Y COMPONENTES MECÁNICOS BÁSICOS.

A MEDIADOS DEL SIGLO XX, EL HARDWARE SE BASABA PRINCIPALMENTE EN LA TECNOLOGÍA DE TUBOS DE VACÍO, QUE FINALMENTE FUE REEMPLAZADA POR EL TRANSISTOR. DÉCADA DE 1960.

HOY EN DÍA, EL HARDWARE SE UTILIZA AMPLIAMENTE EN TODO TIPO DE SISTEMAS Y DISPOSITIVOS INFORMÁTICOS, DESDE COMPUTADORAS DE ESCRITORIO Y PORTÁILES HASTA TELÉFONOS INTELIGENTES Y RELOJES INTELIGENTES.



### CHARLES BABBAGE

CIENTÍFICO A QUIEN SE LE OCURRIÓ POR PRIMERA VEZ LA IDEA DE UNA COMPUTADORA

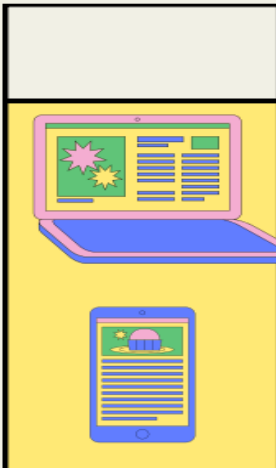
### JACK S. KILBY Y ROBERT NOYCE

CREADORES DEL PRIMER CIRCUITO INTEGRADO (CHIP)

### JOHN MCCARTHY

CREADOR DEL TÉRMINO "INTELIGENCIA ARTIFICIAL" GRAN PIONERO EN ESTE CAMPO, TRAJÓ MUCHOS LOGROS

## GENERACIONES DE HARDWARE



DESDE LOS PRIMEROS SISTEMAS INFORMÁTICOS, EL HARDWARE HA SUFRIDO IMPORTANTES CAMBIOS, PARA PERMITIR ORDENADORES MÁS RÁPIDOS Y CAPACES DE REALIZAR UNA MAYOR VARIEDAD DE TAREAS. TENIENDO EN CUENTA LOS AVANCES MÁS SIGNIFICATIVOS, EXISTEN AL MENOS CUATRO GENERACIONES DE HARDWARE:

ALGUNOS PERIÓDICOS HAN CAMBIADO SU TRADICIONAL FORMATO PARA PODER SEGUIR LLEGANDO A LA MAYOR CANTIDAD DE PERSONAS. AHORA, TODOS PUEDEN LEER EL PERIÓDICO EN SUS TABLETAS O TELÉFONOS CELULARES.

- PRIMERA GENERACIÓN (1945-1956). ERAN COMPUTADORAS QUE FUNCIONABAN CON TUBOS DE VACÍO.
- SEGUNDA GENERACIÓN (1957-1963). SE INVENTARON LOS TRANSISTORES, GRACIAS A LOS CUALES LAS DIMENSIONES DE LAS COMPUTADORAS SE REDUJERON SIGNIFICATIVAMENTE.
- TERCERA GENERACIÓN (1964-1971). SE DESARROLLARON LOS PRIMEROS CIRCUITOS INTEGRADOS IMPRESOS EN CHIPS DE SILICIO, QUE PERMITIERON AUMENTAR LA VELOCIDAD Y EFICIENCIA DEL PROCESAMIENTO DE LA INFORMACIÓN.
- CUARTA GENERACIÓN (1971-1981). SURGIERON LOS MICROPROCESADORES QUE PERMITIERON CREAR UNA COMPUTADORA PERSONAL (PC).

FUENTE: [HTTPS://CONCEPTO.DE/HARDWARE/#IXZZ8BEQMRPX6](https://concepto.de/hardware/#ixzz8BEQMRPX6)  
[HTTPS://WWW.ARIMETRICS.COM/GLOSARIO-DIGITAL/HARDWARE#:~:TEXT=HARDWARE%20Y%20SOFTWARE-,ORIGEN%20DEL%20HARDWARE,DE%20CHARLES%20BABBAGE%20EN%201822.](https://www.arimetrics.com/glosario-digital/hardware#:~:text=hardware%20y%20software-,origen%20del%20hardware,de%20charles%20babbage%20en%201822.)

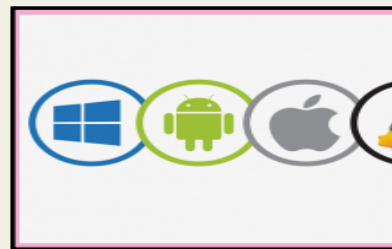
# HISTORIA DEL SOFTWARE

## ¿CUÁLES SON SUS ORÍGENES?



EL SIGNIFICADO DE SOFTWARE SE REMONTA A LOS PRIMEROS MOMENTOS DE LA INFORMÁTICA, CUANDO LOS OPERADORES HUMANOS PROGRAMABAN MÁQUINAS DIRECTAMENTE MEDIANTE EL USO DE TARJETAS PERFORADAS O INGRESANDO COMANDOS EN TERMINALES.

SE DESARROLLARON PROGRAMAS INFORMÁTICOS QUE PODÍAN AUTOMATIZAR TAREAS RUTINARIAS, AUMENTANDO TANTO LA EFICIENCIA COMO LA PRECISIÓN DE ESTOS SISTEMAS. HOY EN DÍA, DESEMPEÑA UN PAPEL VITAL EN TODO, DESDE NUESTRA VIDA DIARIA HASTA LOS SISTEMAS DE INFRAESTRUCTURA CRÍTICA, Y EVOLUCIONA CONSTANTEMENTE PARA SATISFACER LAS NECESIDADES DE LOS USUARIOS DE TODO EL MUNDO.



### ADA LOVELACE

FUE LA PRIMERA PROGRAMADORA EN EL MUNDO.

### DENNIS RITCHIE

EL CREADOR DE UNIX Y DE UNOS DE LOS LENGUAJES MAS FAMOSOS EN EL MUNDO "C".

### BILL GATES

CREADOR DE LA LLAMADA INDUSTRIA DEL SOFTWARE Y DE LA EMPRESA MICROSOFT.

## GENERACIONES DE SOFTWARE

### PRIMERA GENERACIÓN: LENGUAJE DE MÁQUINA

CADA COMPUTADORA TIENE UN SOLO LENGUAJE DE PROGRAMACIÓN QUE SU PROCESADOR PUEDE EJECUTAR; BUENO, ESE ES TU IDIOMA NATIVO O LENGUAJE DE MÁQUINA.

### SEGUNDA GENERACIÓN: LENGUAJE ENSAMBLADOR

EVITO QUE LOS PROGRAMADORES TUVIERAN QUE PROGRAMAR DIRECTAMENTE EN CÓDIGO BINARIO, SE DESARROLLARON PROGRAMAS PARA TRADUCIR INSTRUCCIONES A CÓDIGO DE MÁQUINA. ESTOS PROGRAMAS SE DENOMINARON ENSAMBLADORES O DE BAJO NIVEL.

### TERCERA GENERACIÓN: LENGUAJES DE ALTO NIVEL

ESTOS IDIOMAS SON SIMILARES AL INGLÉS Y FACILITAN EL TRABAJO DE LOS DESARROLLADORES DE SOFTWARE PARA QUE NO SE TENGA QUE CENTRAR EN EL FUNCIONAMIENTO INTERNO DEL PROCESADOR COMO EN LOS LENGUAJES DE GENERACIONES ANTERIORES Y PUEDEN MANEJAR MEJOR LA APLICACIÓN QUE ESTÁN PROGRAMANDO.

### CUARTA GENERACIÓN: LENGUAJES ORIENTADOS AL USUARIO

EL SOFTWARE DE ESTOS LENGUAJES GENERA LA MAYOR PARTE DEL FLUJO DE UN PROGRAMA DE FORMA AUTOMÁTICA. UN PROGRAMADOR QUE TRABAJA EN UN LENGUAJE DE TERCER NIVEL ESCRIBE INSTRUCCIONES SOBRE QUÉ HACER Y CÓMO HACERLO TENIENDO INTERFAZ GRÁFICA PARA USAR DE FORMA FÁCIL Y SENCILLA.

### QUINTA GENERACIÓN: LENGUAJES NATURALES

LOS LENGUAJES NATURALES ESTÁN MÁS CERCA DEL LENGUAJE HUMANO QUE SUS PREDECESORES, LOS LENGUAJES 4GL CADA VEZ HAY MÁS PROGRAMAS DE BASES DE DATOS QUE SE PUEDEN CONSULTAR UTILIZANDO HERRAMIENTAS DE CONSULTA EN LENGUAJE NATURAL.

#### FUENTE:

[HTTP://CV.UOC.EDU/MODULS/XW02\\_79049\\_00373/WEB/MAIN/M4/V2\\_2.HTML](http://cv.uoc.edu/moduls/xw02_79049_00373/web/main/m4/v2_2.html)  
[HTTP://WWW.TIPOSDESFTWARE.COM/HISTORIA-DEL-SOFTWARE.HTM](http://www.tiposdesoftware.com/historia-del-software.htm)



## **Sistemas numéricos**

Los sistemas numéricos son métodos o notaciones que representan números utilizando símbolos o combinaciones de dígitos. Los sistemas más comunes son el decimal (base 10), binario (base 2), octal (base 8) y hexadecimal (base 16). Cada sistema tiene su base específica y utiliza diferentes dígitos para representar valores, lo que permite realizar operaciones matemáticas y expresar cantidades de manera diversa.

Los sistemas binarios, octal y hexadecimal son sistemas numéricos alternativos utilizados para representar números.

### **Sistema Binario (Base 2):**

El sistema binario utiliza dos símbolos: 0 y 1. Cada posición en un número binario representa una potencia de 2. Es ampliamente utilizado en sistemas informáticos y digitales debido a su naturaleza de representar estados de encendido/apagado (0 y 1).

### **Sistema Octal (Base 8):**

El sistema octal utiliza ocho símbolos: 0, 1, 2, 3, 4, 5, 6 y 7. Cada posición en un número octal representa una potencia de 8. Aunque fue utilizado más comúnmente en el pasado, hoy en día su uso es menos frecuente, especialmente en informática.

### **Sistema Hexadecimal (Base 16):**

El sistema hexadecimal utiliza dieciséis símbolos: 0-9 y A-F (donde A representa 10, B representa 11, y así sucesivamente hasta F que representa 15). Cada posición en un número hexadecimal representa una potencia de 16. Se utiliza comúnmente en informática y programación, especialmente para representar valores binarios de manera más compacta y legible.



## Números octales

Suma:

$$\begin{array}{r} + 7560(8) \\ + 752(8) \\ \hline 10532(8) \end{array}$$

$$\begin{array}{r} + 7654(8) \\ + 762(8) \\ \hline 0656(8) \end{array}$$

Resta:

$$\begin{array}{r} 5752(8) \\ - 327(8) \\ \hline 5423(8) \end{array}$$

Multipliación:

$$\begin{array}{r} 756(8) \\ \times 23(8) \\ \hline 1252(8) \end{array}$$

$$\begin{array}{r} 6321(8) \\ \times 632(8) \\ \hline 5173272(8) \end{array}$$

División:

$$\begin{array}{r} 7635(8) \\ / 721(8) \\ \hline 122(8) \end{array}$$



## Números binarios

Suma:

$$\begin{array}{r} + 11101(2) \\ + 1101(2) \\ \hline 101010(2) \end{array}$$

$$\begin{array}{r} + 10101(2) \\ + 1010(2) \\ \hline 1011(2) \end{array}$$

Resta:

$$\begin{array}{r} - 1110(2) \\ - 100(2) \\ \hline 1010(2) \end{array}$$

$$\begin{array}{r} - 10011(2) \\ - 1101(2) \\ \hline 110(2) \end{array}$$

Multiplicación:

$$\begin{array}{r} x 111101(2) \\ x 1010(2) \\ \hline 1001100010(2) \end{array}$$

$$\begin{array}{r} x 11100(2) \\ x 101(2) \\ \hline 10001100(2) \end{array}$$



División:

$$\begin{array}{r} 11010(2) \\ / 111(2) \\ \hline 0101(2) \end{array}$$

**Números hexadecimales:**

Suma

$$\begin{array}{r} ABF8(16) \\ + FBA7(16) \\ \hline 1A79F(16) \end{array}$$

Resta

$$\begin{array}{r} BBCA(16) \\ - 7A8(16) \\ \hline B422(16) \end{array}$$

Multiplicación

$$\begin{array}{r} 45A5(16) \\ \times 7BF(16) \\ \hline 21BBED2(16) \end{array}$$

División

$$\begin{array}{r} A1E81E(16) \\ / F6(16) \\ \hline A87D(16) \end{array}$$

Trabajo

Realizar dos ejercicios por cada operación básica en binario, octal y hexadecimal.



## Binarios:

Suma

$$\begin{array}{r} + 10101(2) \\ + 0010(2) \\ \hline 10101(2) \end{array}$$

$$\begin{array}{r} + 11100(2) \\ + 1100(2) \\ \hline 101000(2) \end{array}$$

$$\begin{array}{r} + 10110(2) \\ + 11110(2) \\ \hline 110100(2) \end{array}$$

Resta

$$\begin{array}{r} - 101101(2) \\ - 10101(2) \\ \hline 11000(2) \end{array}$$

$$\begin{array}{r} - 110011(2) \\ - 01110(2) \\ \hline 1101(2) \end{array}$$

$$\begin{array}{r} - 0110011(2) \\ - 001110(2) \\ \hline 001101(2) \end{array}$$

Multiplicación

$$\begin{array}{r} \times 101110(2) \\ 101(2) \\ \hline 11100110(2) \end{array}$$

$$\begin{array}{r} \times 101011(2) \\ 110(2) \\ \hline 100000010(2) \end{array}$$

$$\begin{array}{r} \times 110011(2) \\ 101(2) \\ \hline 11111111(2) \end{array}$$

División

$$\begin{array}{r} / 11110(2) \\ 101(2) \\ \hline 110(2) \end{array}$$

$$\begin{array}{r} / 1101110(2) \\ 100(2) \\ \hline 11011(2) \end{array}$$

$$\begin{array}{r} / 1011101(2) \\ 110(2) \\ \hline 0111(2) \end{array}$$

Conversión

$$1001100010(2)=49(10)$$

$$1001100010(2)=255(10)$$

$$1001100010(2)=108(10)$$



## Octales

Suma

$$\begin{array}{r} + 56371(8) \\ + 13457(8) \\ \hline 72050(8) \end{array}$$

$$\begin{array}{r} + 674532(8) \\ + 5670(8) \\ \hline 702422(8) \end{array}$$

$$\begin{array}{r} + 763265(8) \\ + 12765(8) \\ \hline 776242(8) \end{array}$$

Resta

$$\begin{array}{r} - 54345(8) \\ - 5721(8) \\ \hline 46425(8) \end{array}$$

$$\begin{array}{r} - 67432(8) \\ - 32167(8) \\ \hline 35243(8) \end{array}$$

$$\begin{array}{r} - 16175(8) \\ - 7523(8) \\ \hline 1452(8) \end{array}$$

Multiplicación

$$\begin{array}{r} x 125672(8) \\ x 521(8) \\ \hline 36125732(8) \end{array}$$

$$\begin{array}{r} x 65312(8) \\ x 125(8) \\ \hline 10672422(8) \end{array}$$

$$\begin{array}{r} x 23154(8) \\ x 442(8) \\ \hline 10783110(8) \end{array}$$

División

$$\begin{array}{r} / 7635(8) \\ / 721(8) \\ \hline 10.5(8) \end{array}$$

$$\begin{array}{r} / 3504(8) \\ / 42(8) \\ \hline 66(8) \end{array}$$

$$\begin{array}{r} / 45382(8) \\ / 25(8) \\ \hline 1622(8) \end{array}$$

Conversión

$$540(8)=49(10)$$

$$540(8)=255(2)$$

$$540(8)=108(16)$$



$$6700(8)=49(10)$$

$$6700(8)=255(2)$$

$$6700(8)=108(16)$$

### Hexadecimal

Suma

$$\begin{array}{r} + AAB91(16) \\ + B1245(16) \\ \hline 1BBFD6(16) \end{array}$$

$$\begin{array}{r} + B2345(16) \\ + 2F2CA(16) \\ \hline E261F(16) \end{array}$$

$$\begin{array}{r} + CD235(16) \\ + 32C2(16) \\ \hline D14F7(16) \end{array}$$

Resta

$$\begin{array}{r} - B3547(16) \\ - A2355(16) \\ \hline 111F2(16) \end{array}$$

$$\begin{array}{r} - F94152(16) \\ - C12591(16) \\ \hline 382BC1(16) \end{array}$$

$$\begin{array}{r} - 32CDF(16) \\ - 5327(16) \\ \hline 2D9A8(16) \end{array}$$

Multiplicación

$$\begin{array}{r} \times 5529(16) \\ \times 23(16) \\ \hline BA49B(16) \end{array}$$

$$\begin{array}{r} \times EC01(16) \\ \times 54(16) \\ \hline 4D7054(16) \end{array}$$

$$\begin{array}{r} \times 8011(16) \\ \times BA(16) \\ \hline 4D7054(16) \end{array}$$

División

$$\begin{array}{r} / 4DB2F0(16) \\ 50(16) \\ \hline F8A3(16) \end{array}$$

$$\begin{array}{r} / 10DE50(16) \\ A2(16) \\ \hline 1AA8(16) \end{array}$$

$$\begin{array}{r} / 156C03(16) \\ 2B(16) \\ \hline 7F89(16) \end{array}$$



## Habilidades de pensamiento

Son los procesos mentales que permiten a las personas procesar cualquier tipo de información; adquirir conocimientos y resolver problemas.

Estas cosas son importantes porque permiten a los humanos pensar y actuar de determinadas maneras, adquiriendo conocimientos; Desarrollar las habilidades, actitudes y competencias necesarias para el desarrollo del pensamiento lógico.

Ejercicios para entrenar la habilidad de pensamiento

Encontrar las 8 diferencias en cada imagen:



A partir de la imagen e intenta decir el color y no la palabra

AMARILLO	AZUL	NARANJA
NEGRO	ROJO	VERDE
MORADO	AMARILLO	ROJO
NARANJA	VERDE	NEGRO
AZUL	ROJO	MORADO
VERDE	AZUL	NARANJA



# Trabajo

Realizar mapa conceptual de las habilidades de pensamiento.



## Competencias steam

- Science technology engineer art mathematics
- Surgió en 1990
- Método científico
- Uso de tecnologías digitales
- Metodologías científicas

## Desarrollo de habilidades de pensamiento

- Permite adquirir conocimientos
- Resolver problemas
- ¿Por qué es importante?
- Nos prepara para la vida
- Potencia pensamientos lógicos
- Estructurar habilidades y actitudes



## **Algoritmos**

Es un conjunto de procedimientos lógicos o matemáticos que se pueden seguir para resolver un problema en un número limitado de pasos.

Sus características son:

Un algoritmo debe ser preciso e indicar el orden en el que se completa cada paso.

Se debe definir el resultado del algoritmo. Si sigue el algoritmo dos veces con los mismos datos de entrada, obtendrá los mismos resultados cada vez.

Un algoritmo debe ser finito. Si se sigue un algoritmo, entonces el algoritmo debe completarse en un punto determinado, es decir, el algoritmo debe tener un número limitado de pasos.

## **Seudocódigo**

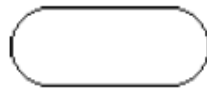
En algoritmos expresados más formalmente, se utiliza como representación intermedia, antes de ser traducido o codificado con un lenguaje de programación.

## **Diagrama de flujo**

Es una representación gráfica del algoritmo. Donde se expresan los pasos del algoritmo, mediante un esquema con símbolos predefinidos.

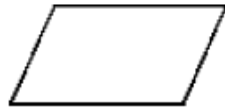


## Tipos de símbolos en el diagrama de flujo



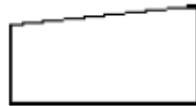
### Inicio/Final

Se utiliza para indicar el inicio y el final de un diagrama; del Inicio sólo puede salir una línea de flujo y al Final sólo debe llegar una línea.



### Entrada General

Entrada/Salida de datos en General (en esta guía, solo la usaremos para la Entrada).



### Entrada por teclado

Instrucción de entrada de datos por teclado. Indica que el computador debe esperar a que el usuario teclee un dato que se guardará en una variable o constante.



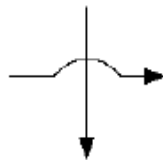
### Llamada a subrutina

Indica la llamada a una subrutina o procedimiento determinado.



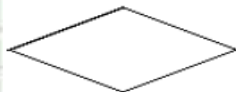
### Acción/Proceso General

Indica una acción o instrucción general que debe realizar el computador (cambios de valores de variables, asignaciones, operaciones aritméticas, etc).



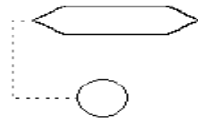
### Flujo

Indica el seguimiento lógico del diagrama. También indica el sentido de ejecución de las operaciones.



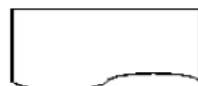
### Decisión

Indica la comparación de dos datos y dependiendo del resultado lógico (falso o verdadero) se toma la decisión de seguir un camino del diagrama u otro.



### Iteración

Indica que una instrucción o grupo de instrucciones deben ejecutarse varias veces.



### Salida Impresa

Indica la presentación de uno o varios resultados en forma impresa.



### Salida en Pantalla

Instrucción de presentación de mensajes o resultados en pantalla.



### Conector

Indica el enlace de dos partes de un diagrama dentro de la misma página.

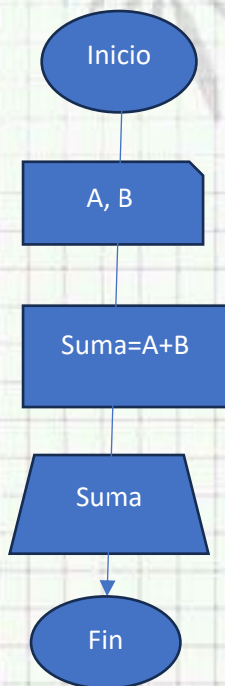


### Conector

Indica el enlace de dos partes de un diagrama en páginas diferentes.



## Diagrama de flujo



## Pseudocódigo

Inicio

Lea

A, B

Haga

Suma=A+B

Muestre

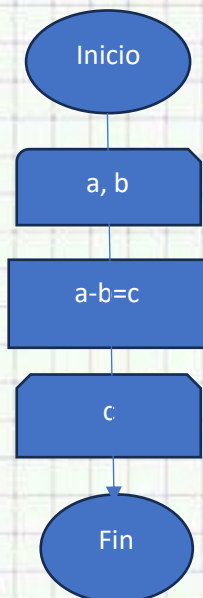
Suma

Fin

Realizar el algoritmo y diagrama de:

a) Restar dos números

## Diagrama de flujo



## Pseudocódigo

Inicio

Lea

a, b

Haga

a-b=c

Muestre

c

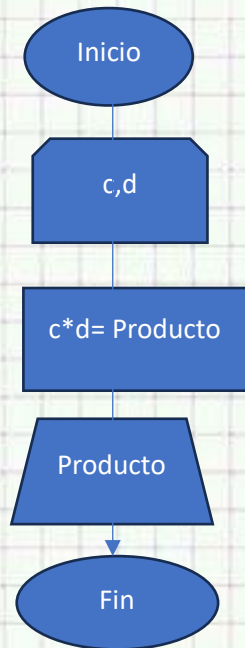
Fin



Realizar el algoritmo y diagrama de:

b) Multiplicar dos números

### Diagrama de flujo



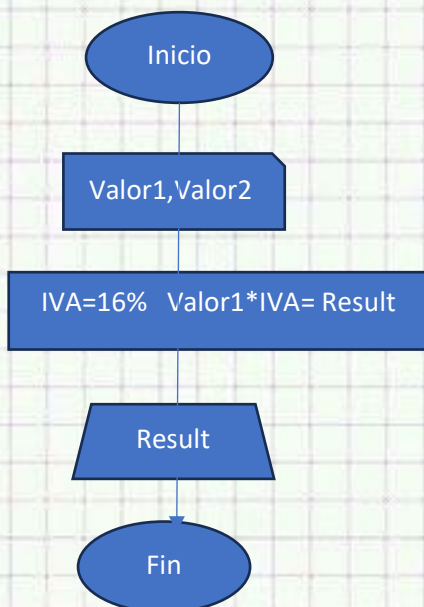
### Pseudocódigo

Inicio  
Lea  
c, d  
Haga  
c\*d=Producto  
Muestre  
Producto  
Fin

Realizar el algoritmo y diagrama de:

c) Hallar el valor del IVA =16%

### Diagrama de flujo



### Pseudocódigo

Inicio  
Lea  
Valor1, Valor2  
Haga  
IVA=16% Valor1\*IVA=Result  
Muestre  
Result  
Fin



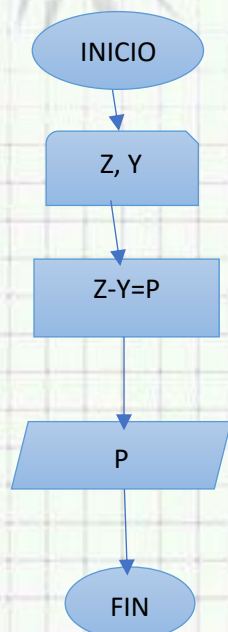
## Cuadro sobre los tipos de datos

Tipos de datos	Características
Numéricos	Entero: Son números sin parte fraccionaria y con estos se pueden realizar la suma, resta, multiplicación y división.
Lógico	Booleano: Tipo de dato de verdad, True o false, un dato desconocido es un valor nulo.  Cadena: Es el conjunto de características incluyendo letras, datos alfanuméricos, números, espacios entre otros.
Carácter	Carácter: Representado por comillas y toma un valor de los caracteres representados en la máquina.

Operadores	Representación y significado
Relacionales	Comparan datos numéricos, de serie, lógicos, de caracteres. Verdadero (1) o Falso (0). Sentencia If
Aritméticos	Usa datos numéricos e indican las operaciones matemáticas. (+) (-) (*) (/) (pow) (%) (sqrt).
Lógicos	Combinan dos valores booleanos y devolver un resultado verdadero, falso o nulo. (&) es y (  ) es o (!) es No.



## Diagrama de flujo



## Pseudocódigo

Inicio

Lea

Z, Y (Variables)

Haga

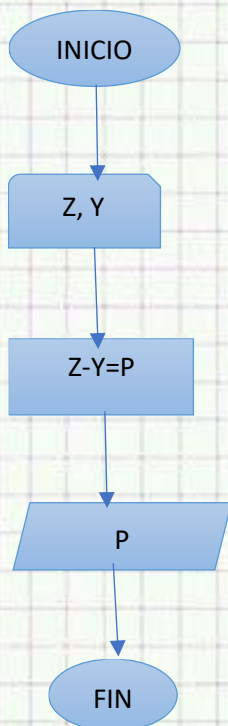
$Z - Y = P$  (Proceso)

Muestre

P

Fin

## Diagrama de flujo



## Pseudocódigo

Inicio

Lea

A, B, C, D

Haga

$A+B+C+D=Z$

Muestre

Z

Fin



## Operaciones Primitivas:

Se realizan directamente en el lenguaje de programación.

Tipos de datos

(+) Suma

(-) Resta

(\*) Multiplicación

(/) División

Carácter numérico

(<) Menor

(>) Mayor

(<=) Menor o igual

(>=) Mayor o igual

(i=) Distinto que

Realizar pseudocódigo que pida el nombre a la persona y le desee feliz día

Definir

$\pi = 3.1416$

Iva= 16%

## Seudocódigo

Inicio

Lea

Nomb= David

Haga

“¿Cómo te llamas?” + nomb



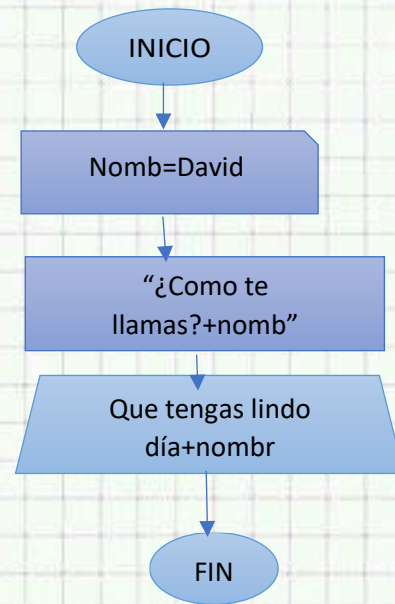
Haga

“Que tengas lindo día” +  
nomb

Muestre

Que tengas lindo día  
David

Fin



Inicio

Lea

X // cantidad de dólares

\$4000 = a // Valor cop

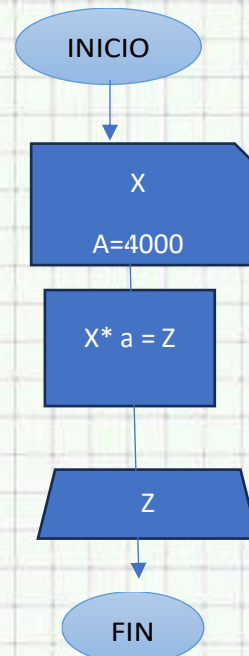
Haga

$X * a = Z$

Muestre

Z

Fin





## Variables y Constantes

**Variable:** Son contenedores que se utilizan para almacenar información que puede cambiar durante la ejecución del programa.

**Constantes:** A diferencia de las variables, las constantes son valores que no pueden ser modificados una vez que se les asigna un valor inicial. Se utilizan para representar valores fijos que no cambiarán a lo largo de la ejecución del programa.

## Reglas de prioridad

Expresiones matemáticas que contienen dos o más operadores necesitan un conjunto de reglas que determinen el orden en la ejecución de sus operaciones.

## Sintaxis

Son las reglas y estructuras gramaticales que deben seguirse al escribir código en un lenguaje de programación específico.

## Estructuras de control

Las estructuras de control en programación son bloques de código que permiten controlar el flujo de ejecución del programa. Estas estructuras son fundamentales para tomar decisiones, repetir acciones y controlar el comportamiento del código en función de diferentes condiciones.

Hay tres tipos principales de estructuras de control:



Hay tres tipos principales de estructuras de control:

### **Estructuras de control condicionales:**

if: Permite ejecutar un bloque de código si una condición es verdadera.

else: Se utiliza junto con "if" para ejecutar un bloque de código cuando la condición del "if" es falsa.

else if (en algunos lenguajes): Permite comprobar múltiples condiciones después de un "if" inicial.

### **Estructuras de control de bucles:**

for: Se utiliza para ejecutar un bloque de código un número específico de veces.

while: Ejecuta un bloque de código mientras una condición sea verdadera.

do-while (en algunos lenguajes): Similar al "while", pero garantiza que el bloque de código se ejecute al menos una vez antes de verificar la condición.

### **Estructuras de control de salto:**

break: Se utiliza para salir de un bucle.

continue: Permite pasar a la siguiente iteración de un bucle, omitiendo parte del código restante en esa iteración.

goto (en algunos lenguajes): Permite saltar a una etiqueta específica en el código.



## Ejercicios en DFD

1. Desarrolle un programa que al escribir la hora, si la hora es menor que las 12 del día indique buenos días; de lo contrario diga buenas tardes.

INICIO

|

Preguntar la hora

|

Si la hora es menor que 12

|

└─ Mostrar "Buenos días"

|

Si no

|

└─ Mostrar "Buenas tardes"

|

FIN

2. Desarrolle un programa que pregunte la edad el año de nacimiento y calcule la edad ;si la edad es mayor de 18 años, le permita el ingreso al concierto musical ,de lo contrario diga aun no eres mayor de edad

INICIO

|

Preguntar el año de nacimiento

|

Calcular la edad (año actual - año de nacimiento)

|

Si la edad es mayor o igual a 18

|



└─ Mostrar "¡Bienvenido al concierto musical!"

|

Si no

|

└─ Mostrar "Aún no eres mayor de edad para ingresar al concierto."

|

FIN

3. Desarrolle un programa que pregunte la edad el año de nacimiento y calcule la edad ;si la edad es menor o igual a 25 años, le permita sacar cita media en la EPS, de lo contrario diga indique que ya no puede ser beneficiario.

INICIO

|

Pedir año de nacimiento

|

Calcular la edad (año actual - año de nacimiento)

|

Si edad  $\leq$  25

|

└─ Permitir sacar cita en la EPS

|

Si no

|

└─ Mostrar "Ya no puede ser beneficiario"

|

FIN



4. Desarrollo un programa que solicite 3 números en 3 variables A, B y C. Y determine cual es el mayor y cual es el menor.

INICIO

|

Solicitar tres números: A, B y C

|

Encontrar el mayor y el menor entre A, B y C

|

Mostrar el número mayor y el número menor

|

FIN

### **Estructura condicional simple**

Es evaluar una condición y realizar una acción si esa condición es verdadera. Es una manera de controlar el flujo del programa basado en una única evaluación.

### **Estructura condicional doble**

Es evaluar una condición y realizar una acción si esa condición es verdadera, y otra acción si la condición es falsa. Es una forma de controlar el flujo del programa cuando se tienen dos posibles resultados basados en una única evaluación.

### **Bucles**

Los bucles son estructuras fundamentales en programación que permiten ejecutar repetidamente un bloque de código mientras se cumple una condición específica o por un número determinado de veces. Estos son algunos de los bucles más comunes:

#### **Bucle for:**

El bucle for se utiliza para iterar sobre una secuencia de valores conocida, como una colección de elementos, durante un número específico de veces.



Bucle while:

El bucle while ejecuta un bloque de código mientras una condición sea verdadera.

### Ejercicios DFD

1. Un programa que imprima 10 veces Universidad distrital Francisco Jose de caldas

INICIO

|

Inicializar contador en 1

|

Mientras contador sea menor o igual a 10

|

Imprimir "Universidad Distrital Francisco José de Caldas"

Incrementar contador en 1

|

FIN

2. Un programa que imprima los números de 20 a 1.

INICIO

|

Inicializar variable número en 20

|

Mientras número sea mayor o igual a 1

|

|---> Imprimir número

|---> Decrementar número en 1

|

FIN



3. Un programa que imprima N veces los números pares.

INICIO

|

Inicializar variable número en 20

|

Mientras número sea mayor o igual a 1

|

Imprimir número

Decrementar número en 1

|

FIN

4. Desarrolle un algoritmo que le permita a Juan hacer la sumatoria de los números enteros comprendidos entre el 1 y el 10.

INICIO

|

Inicializar variables: total = 0, contador = 1

|

Mientras contador sea menor o igual a 10

|

total = total + contador

Incrementar contador en 1

|

Mostrar total

|

FIN



1. Hallar el promedio de las edades de 5 personas, ese valor elevarlo al cuadrado y multiplicarlo por dos

Inicio

Lea

V1, V2, V3, V4, V5

V6=5 V7=2

Haga

$V1+V2+V3+V4+V5 = \text{Prom}$

$\text{Prom}/V6 = \text{Result}$

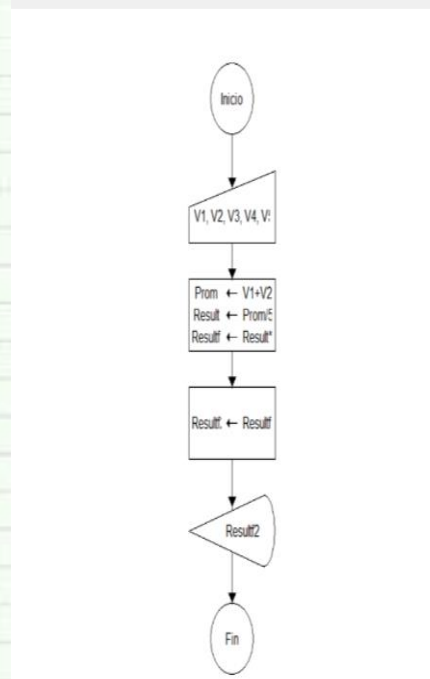
$\text{Result} * \text{Result} = \text{El result}$

$\text{El result} * V7 = \text{Result Final}$

Muestre

ResultFinal

Fin



2. Se administra una finca por cada vaca se necesitan 2 personas, cuantas personas se necesitan para 20 vacas.

Inicio

Lea

V

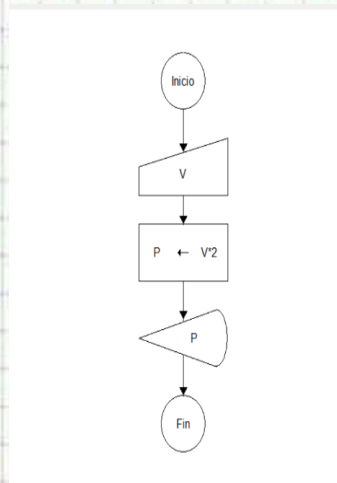
Haga

$P = V * 2$

Muestre

P

Fin





### 3. Halle la hipotenusa de un triángulo rectángulo

Inicio

Lea

h, Co, Ca

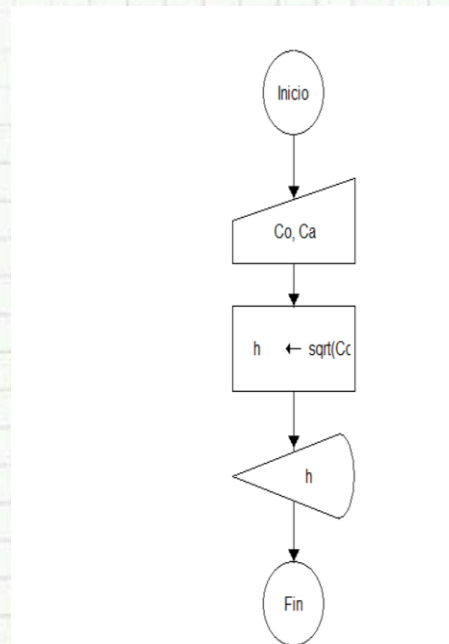
Haga

$H = \sqrt{Co^2 + Ca^2}$

Muestre

H

Fin



### 4. Halle el área de un triángulo

Inicio

Lea

Base, Altura

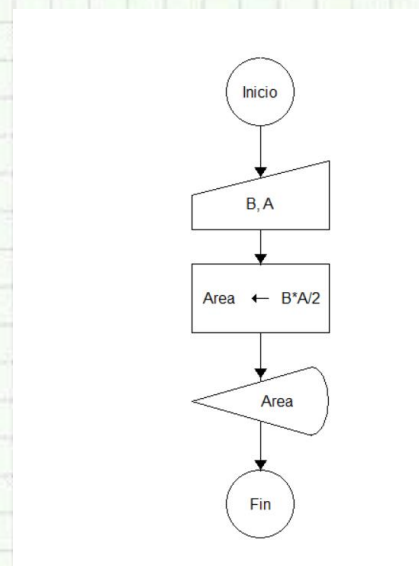
Haga

$\text{Base} \times \text{Altura} / 2 = R$

Muestre

R

Fin





## 5. Área de un triángulo rectángulo.

Inicio

Lea

$L1 * L2$

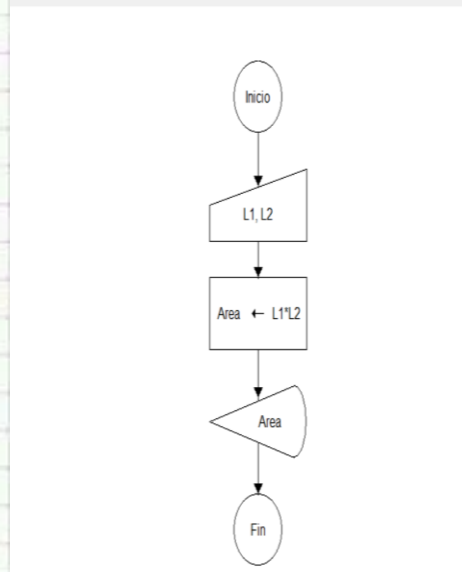
Haga

$L1 * L2 = B$

Muestre

B

Fin



## 6. Encontrar el valor de un producto con un descuento cualquiera.

Inicio

Lea

$V1 \text{ Desc}=60\%$

Haga

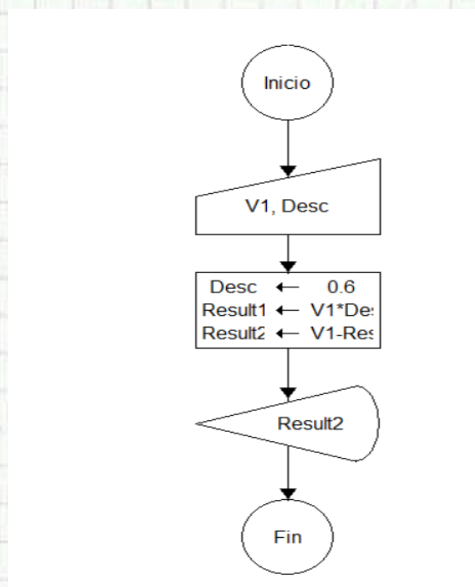
$V1 * \text{Desc} = \text{Result1}$

$V1 - \text{Result1} = \text{Result2}$

Muestre

Result2

Fin





7. Hallar la cantidad de frutas que se reparten si hay 6 frutas y 12 niños para que todos los niños tengan frutas

Inicio

Lea

F=6

N=12

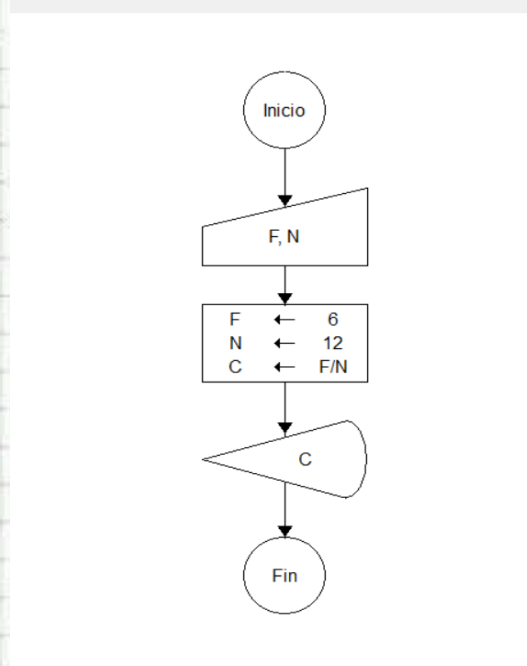
Haga

$F/N=C$

Muestre

C

Fin



8. Hay x televisores, televisores tienen alto a y ancho x, hallar las pulgadas

Inicio

Lea

h,x

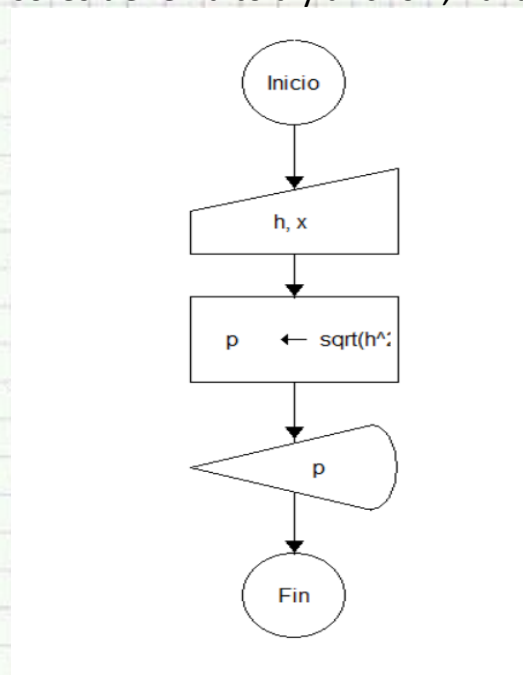
Haga

$P=H^2+X^2$

Muestre

P

Fin





9. Halle el área de un círculo con un radio cualquiera y halle el área del cuadrado.

Inicio

Lea

R

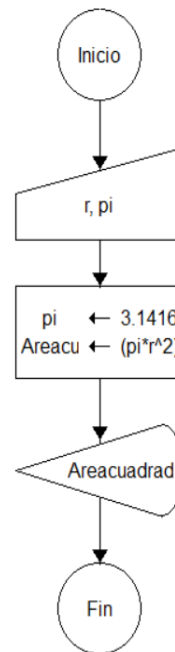
Haga

$A = (\pi * r^2) * (\pi * r^2)$

Muestre

A

Fin





### Cuadro sobre los tipos de datos

<b>Tipos de datos</b>	<b>Características</b>
Numéricos	Entero: Son números sin parte fraccionaria y con estos se pueden realizar la suma, resta, multiplicación y división.
Lógico	Booleano: Tipo de dato de verdad, True o false, un dato desconocido es un valor nulo.  Cadena: Es el conjunto de características incluyendo letras, datos alfanuméricos, números, espacios entre otros.
Carácter	Carácter: Representado por comillas y toma un valor de los caracteres representados en la máquina.

<b>Operadores</b>	<b>Representación y significado</b>
Relacionales	Comparan datos numéricos, de serie, lógicos, de caracteres. Verdadero (1) o Falso (0). Sentencia If
Aritméticos	Usa datos numéricos e indican las operaciones matemáticas. (+) (-) (*) (/)(pow)(%)(sqrt).
Lógicos	Combinan dos valores booleanos y devolver un resultado verdadero, falso o nulo. (&&) es y (  ) es o (!) es No.



## Problemas computacionales

### Decisiones:

En C++, tanto la estructura if como la estructura switch son herramientas fundamentales para el control de flujo en un programa, permitiendo tomar decisiones basadas en condiciones específicas.

### IF

La estructura if es una construcción que evalúa una condición y ejecuta un bloque de código si esa condición es verdadera.

### SWITCH

La estructura switch evalúa el valor de una variable y ejecuta un bloque de código específico dependiendo del valor que tenga esa variable.

### Entrada y salida de datos:

### Bibliotecas

Es un conjunto de funcionalidades predefinidas que se pueden utilizar en un programa para realizar tareas específicas sin necesidad de volver a escribir todo el código desde cero. Estas bibliotecas contienen conjuntos de funciones, clases y/o definiciones que están diseñadas para ser reutilizadas en diferentes programas.

Ejemplos:

```
#include <iostream> // Incluir la biblioteca estándar iostream
```

```
#include <vector> // Incluir la biblioteca estándar vector
```

### PRINTF

Es una función de salida comúnmente utilizada en C++. Se encuentra en la biblioteca estándar de C, <cstdio>, y se usa para imprimir texto y datos formateados en la consola o en otro flujo de salida. Su sintaxis básica es:

```
printf("formato", argumentos);
```



# Ejercicios en c++:

## 1. Realizar una resta, Multiplicación y División.

The image shows two screenshots of a C++ IDE. The top screenshot displays a program for subtraction. The code includes the `<iostream>` header, uses the `std` namespace, and defines a `main` function that prompts the user for two numbers, calculates their difference, and prints the result. The console output shows the user entering '900 500' and the program outputting 'La resta es: 400'. The bottom screenshot displays a program for multiplication. The code follows a similar structure, prompting for two numbers, calculating their product, and printing the result. The console output shows the user entering '700 40' and the program outputting 'La multiplicacion es: 28000'.

```
1 1. /*Realizame una resta en c++*/
2
3
4
5
6 #include <iostream>
7
8 using namespace std;
9
10 int main() {
11     float num1, num2, resultado_resta;
12
13     cout << "Ingrese dos numeros para la resta: ";
14     cin >> num1 >> num2;
15
16     resultado_resta = num1 - num2;
17
18     cout << "La resta es: " << resultado_resta << endl;
19
20     return 0;
21 }
22
```

input

Ingrese dos numeros para la resta: 900 500  
La resta es: 400

...Program finished with exit code 0  
Press ENTER to exit console.

```
1 1. /*Realizame una multiplicacion en c++*/
2
3
4
5
6 #include <iostream>
7
8 using namespace std;
9
10 int main() {
11     float num1, num2, resultado_multiplicacion;
12
13     cout << "Ingrese dos numeros para la multiplicacion: ";
14     cin >> num1 >> num2;
15
16     resultado_multiplicacion = num1 * num2;
17
18     cout << "La multiplicacion es: " << resultado_multiplicacion << endl;
19
20     return 0;
21 }
```

input

Ingrese dos numeros para la multiplicacion: 700 40  
La multiplicacion es: 28000

...Program finished with exit code 0  
Press ENTER to exit console.



```
main.cpp
1- /*****
2 Realizame una division en c++
3
4 *****/
5
6 #include <iostream>
7
8 using namespace std;
9
10 int main() {
11     float num1, num2, resultado_division;
12
13     cout << "Ingrese dos numeros para la division: ";
14     cin >> num1 >> num2;
15
16     if (num2 != 0) {
17         resultado_division = num1 / num2;
18         cout << "La division es: " << resultado_division << endl;
19     } else {
20         cout << "Error: No se puede dividir por cero." << endl;
21     }
22
23     return 0;
24 }
```

input

Ingrese dos numeros para la division: 100 2  
La division es: 50

...Program finished with exit code 0  
Press ENTER to exit console.

2. Don Carlos tiene una tienda y necesita que realices un programa que al escribir el costo de producto y realice el valor del IVA y posteriormente totalice el pago

```
main.cpp
1- /*****
2
3 Don Carlos tiene una tienda y necesita que realices un programa que al escribir el costo de producto
4 y realice el valor del IVA y posteriormente totalice el pago
5
6 *****/
7
8 #include <iostream>
9 using namespace std;
10
11 int main() {
12     float costoProducto, iva, totalPago;
13
14     // Pedir al usuario el costo del producto
15     cout << "Ingrese el costo del producto: $";
16     cin >> costoProducto;
17
18     // Calcular el valor del IVA (suponiendo una tasa del 16%)
19     iva = costoProducto * 0.16;
20
21     // Calcular el total a pagar
22     totalPago = costoProducto + iva;
23
24     // Mostrar los resultados
25     cout << "El valor del IVA es: $" << iva << endl;
26     cout << "El total a pagar es: $" << totalPago << endl;
27 }
```

input

Ingrese el costo del producto: \$50000  
El valor del IVA es: \$8000  
El total a pagar es: \$58000



3. María tiene un descuento del 10% de la compra total de los productos, es necesario realizar el total de las compras y posteriormente aplique el descuento y le indique cuanto ha ahorrado.

```
main.cpp
1- /*****
2-
3- María tiene un descuento del 10% de la compra total de Los productos,
4- es necesario realizar el total de Las compras y posteriormente aplique el descuento y Le indique cuanto ha ahorrado.
5-
6- *****/
7- #include <iostream>
8- using namespace std;
9-
10- int main() {
11-     float totalCompras, descuento, totalConDescuento, ahorro;
12-
13-     cout << "Ingrese el total de las compras: $";
14-     cin >> totalCompras;
15-
16-     descuento = totalCompras * 0.1;
17-     totalConDescuento = totalCompras - descuento;
18-     ahorro = descuento;
19-
20-     cout << "El total de las compras es: $" << totalCompras << endl;
21-     cout << "Descuento (10%): $" << descuento << endl;
22-     cout << "Total con descuento: $" << totalConDescuento << endl;
23-     cout << "Ha ahorrado: $" << ahorro << endl;
24-
25-     return 0;
26- }
27-
input
El total de las compras es: $300000
Descuento (10%): $30000
Total con descuento: $270000
Ha ahorrado: $30000

...Program finished with exit code 0
Press ENTER to exit console.
```

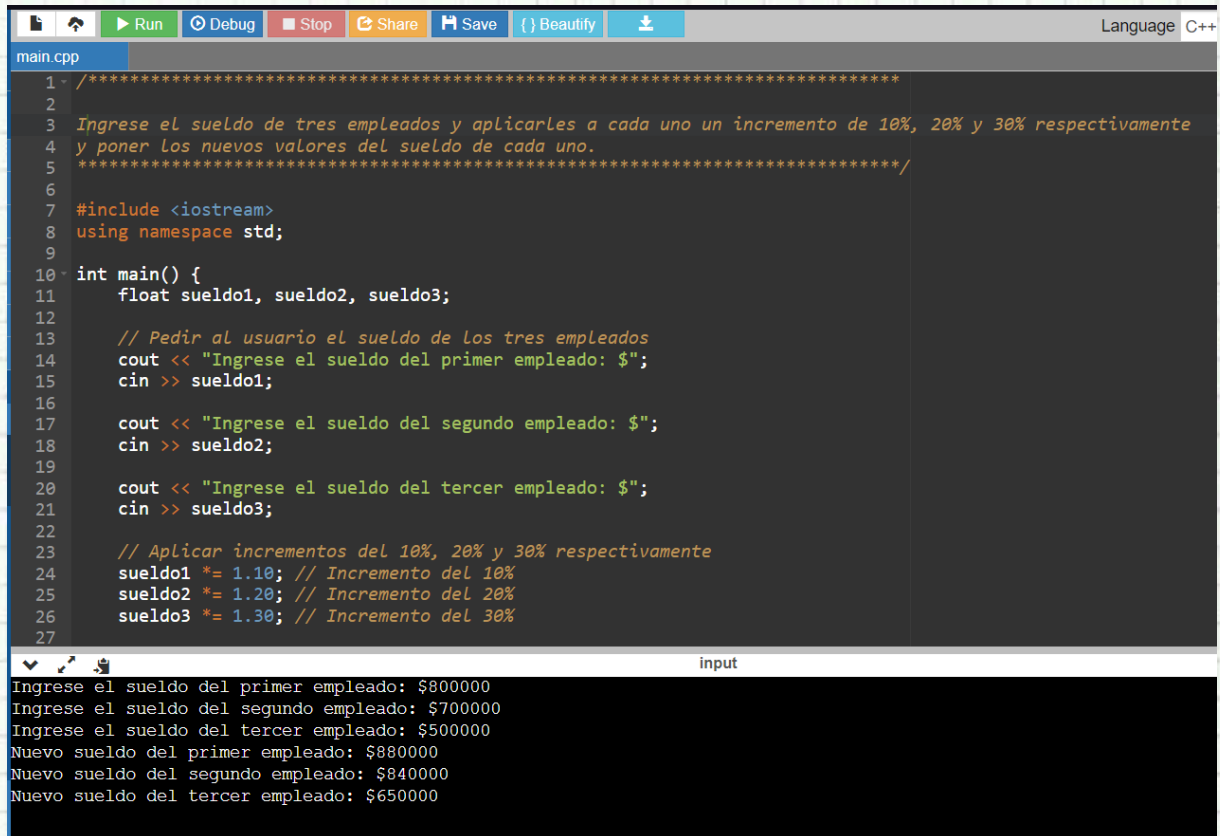
4. Calcular el área de un triángulo, conociendo la fórmula  $\text{Área} = \text{Base} * \text{Altura} / 2$

```
main.cpp
1- /*****
2-
3- Calcular el área de un triángulo, conociendo la fórmula  $\text{Área} = \text{Base} * \text{Altura} / 2$ .
4-
5- *****/
6- #include <iostream>
7- using namespace std;
8-
9- int main() {
10-     float base, altura, area;
11-
12-     cout << "Ingrese la base del triángulo: ";
13-     cin >> base;
14-
15-     cout << "Ingrese la altura del triángulo: ";
16-     cin >> altura;
17-
18-     area = (base * altura) / 2;
19-
20-     cout << "El área del triángulo es: " << area << endl;
21-
22-     return 0;
23- }
24-
25-
input
Ingrese la base del triángulo: 20
Ingrese la altura del triángulo: 30
El área del triángulo es: 300

...Program finished with exit code 0
Press ENTER to exit console.
```



5. Ingrese el sueldo de tres empleados y aplicarles a cada uno un incremento de 10%, 20% y 30% respectivamente y poner los nuevos valores del sueldo de cada uno.



```
1  /*****
2
3  Ingrese el sueldo de tres empleados y aplicarles a cada uno un incremento de 10%, 20% y 30% respectivamente
4  y poner los nuevos valores del sueldo de cada uno.
5  *****/
6
7  #include <iostream>
8  using namespace std;
9
10 int main() {
11     float sueldo1, sueldo2, sueldo3;
12
13     // Pedir al usuario el sueldo de los tres empleados
14     cout << "Ingrese el sueldo del primer empleado: $";
15     cin >> sueldo1;
16
17     cout << "Ingrese el sueldo del segundo empleado: $";
18     cin >> sueldo2;
19
20     cout << "Ingrese el sueldo del tercer empleado: $";
21     cin >> sueldo3;
22
23     // Aplicar incrementos del 10%, 20% y 30% respectivamente
24     sueldo1 *= 1.10; // Incremento del 10%
25     sueldo2 *= 1.20; // Incremento del 20%
26     sueldo3 *= 1.30; // Incremento del 30%
27 }
```

input

```
Ingrese el sueldo del primer empleado: $800000
Ingrese el sueldo del segundo empleado: $700000
Ingrese el sueldo del tercer empleado: $500000
Nuevo sueldo del primer empleado: $880000
Nuevo sueldo del segundo empleado: $840000
Nuevo sueldo del tercer empleado: $650000
```

## Trabajo

Realizar una calculadora en c++ y crear un programa que calcule los diferentes volúmenes de figura.



## Calculadora

```
main.cpp
1- /*****
2
3  Calculadora en c++
4  *****/
5  #include <iostream>
6  using namespace std;
7
8  int main() {
9      char operador;
10     float num1, num2, resultado;
11
12     cout << "Ingrese el primer número: ";
13     cin >> num1;
14
15     cout << "Ingrese el operador (+, -, *, /): ";
16     cin >> operador;
17
18     cout << "Ingrese el segundo número: ";
19     cin >> num2;
20
21     switch (operador) {
22     case '+':
23         resultado = num1 + num2;
24         break;
25     case '-':
26         resultado = num1 - num2;
27         break;
```

input

```
Ingrese el primer número: 80
Ingrese el operador (+, -, *, /): *
Ingrese el segundo número: 60
El resultado es: 4800

...Program finished with exit code 0
Press ENTER to exit console.
```

## Volumenes

```
main.cpp
1- /*****
2
3  Calcular los volúmenes de diferentes figuras geometricas
4
5  *****/
6  #include <iostream>
7  #include <cmath>
8  using namespace std;
9
10 const float PI = 3.14159;
11
12 // Función para calcular el volumen de un cubo
13 float volumenCubo(float lado) {
14     return lado * lado * lado;
15 }
16
17 // Función para calcular el volumen de una esfera
18 float volumenEsfera(float radio) {
19     return (4.0 / 3.0) * PI * pow(radio, 3);
20 }
21
22 // Función para calcular el volumen de un cilindro
23 float volumenCilindro(float radio, float altura) {
24     return PI * pow(radio, 2) * altura;
25 }
26
27 int main() {
```

input

```
Calculadora de volúmenes:
1. Volumen de un cubo
2. Volumen de una esfera
3. Volumen de un cilindro
Seleccione la figura (1-3): 1
Ingrese el lado del cubo: 5
El volumen del cubo es: 125
```