Fundamentos de Programación

TEORÍA Nº 2

UNIDAD 2: INTRODUCCIÓN A LA PROGRAMACIÓN

Facultad de Ingeniería
Universidad Nacional de Jujuy

ÍNDICE

- Conceptos básicos
- Clasificación
  - Simples (estándar y definidos por el usuario)
  - Compuestos o estructurados
- Elementos básicos: variables y constantes
- Operadores y precedencia
- Expresiones
- Asignación, lectura y escritura

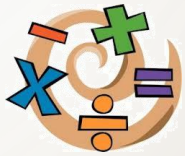
CONCEPTOS BÁSICOS (1)

- Un programa se compone de dos elementos esenciales:
  - *instrucciones* que realizan operaciones específicas (sumar, restar, calcular promedios, listar valores, etc.)
  - *datos* que representan objetos o eventos del mundo real (precios, estado civil, velocidades, fechas, superficies, medidas, nombre de una persona, etc.)

CONCEPTOS BÁSICOS (2)

- Un *tipo de dato* hace referencia a un conjunto de valores.

- Un *tipo de dato abstracto* (TDA) comprende tanto el conjunto de valores como las operaciones que pueden aplicárseles



- Una *estructura de datos* se refiere a la implementación física de un tipo de dato abstracto.

CLASIFICACIÓN

TIPOS ESTÁNDAR (1)

- Numéricos
  - Enteros: subconjunto de los números enteros, se trata de números sin parte decimal, que pueden ser positivos o negativos. Por ejemplo: -123, 0, 48, etc.
  - Reales: subconjunto de los números reales, se trata de números con parte entera y parte decimal, que pueden ser positivos o negativos. Por ejemplo: -234.33, 0.0, 78.21, etc.

Tipo de dato	Puede representar
Entero	Días, personas, productos, etc.
Real	Promedios, medidas, dinero, etc.

TIPOS ESTÁNDAR (2)

- Caracteres
  - El tipo carácter representa una letra (**‘a’**, **‘A’**), un dígito (**‘0’**, **’9’**) o símbolo especial (**‘@’**, **‘&’**, **‘#’**).
- Lógicos
  - El tipo lógico o booleano puede tomar sólo 2 valores: **Verdadero (V)** o **Falso (F)**. Se utiliza para representar la ocurrencia o no de un suceso o condición. Se considera que Falso es menor Verdadero.

Tipo de dato	Puede representar
Carácter	Inicial de un nombre, sexo de una persona en un formulario, etc.
Lógico	Resultado de un análisis clínico, determinación de la veracidad de un dicho, etc.

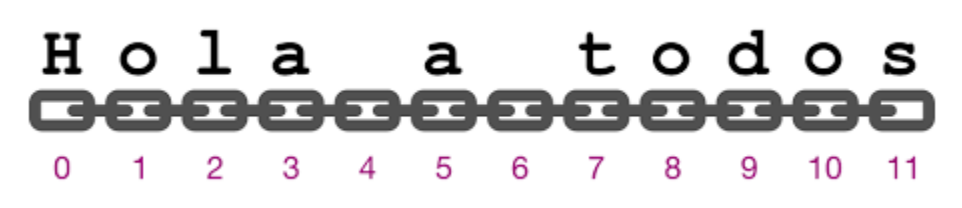
TIPOS ESTÁNDAR (3)

- ¿Qué es un tipo de dato ORDINAL?
  - Se trata de los tipos de datos en los que
    - ✓ Por cada valor se conoce el inmediato anterior (predecesor) y el inmediato siguiente (sucesor)
    - ✓ Existe un primer elemento del conjunto de valores (no tiene predecesor)
    - ✓ Existe un último elemento del conjunto de valores (no tiene sucesor)
- Ejemplo
  - Caracteres: E — Anterior: D Siguiente: F
  - Enteros: 86 — Anterior: 85 Siguiente: 87
  - Reales: 10.85 — Anterior: 10,849 o 10,8495 o 10,8499
  - Lógicos: VERDADERO — Anterior: FALSO Siguiente: No existe




TIPOS COMPUESTOS (1)

- Cadena de caracteres: es un **conjunto de caracteres** (incluido el espacio en blanco) reconocidos por la computadora , los que se almacenan en posiciones de memorias contiguas.
- Ejemplos de cadenas: “esta es una cadena”, “soy cadena”, “hola mundo”.



CODIFICACIÓN DE LA INFORMACIÓN (2)
TABLA ASCII

D	P	D	P	D	P	D	P	D	P	D	P	D	P	D	P
0		32		64	@	96		128	Ç	160	á	192	└	224	α
1	☺	33	!	65	A	97	a	129	ü	161	í	193	┴	225	β
2		34	"	66	B	98	b	130	é	162	ó	194	┬	226	Γ
3	♥	35	#	67	C	99	c	131	â	163	ú	195	├	227	π
4	♦	36	$	68	D	100	d	132	ä	164	ñ	196	-	228	Σ
5	♣	37	%	69	E	101	e	133	à	165	Ñ	197	┼	229	σ
6	♠	38	&	70	F	102	f	134	å	166	ª	198	╞	230	µ
7	•	39	´	71	G	103	g	135	ç	167	º	199		231	γ
8	◘	40	(	72	H	104	h	136	ê	168	¿	200	╚	232	φ
9	○	41	)	73	I	105	i	137	ë	169	⌐	201	╔	233	θ
10	■	42	*	74	J	106	j	138	è	170	¬	202	╩	234	Ω
11	♂	43	+	75	K	107	k	139	ï	171	½	203	╦	235	δ
12	♀	44	,	76	L	108	l	140	î	172	¼	204	╠	236	∂
13	♪	45	-	77	M	109	m	141	ì	173	¡	205	═	237	∅
14	♫	46	.	78	N	110	n	142	Ä	174	«	206	╬	238	∈
15	☼	47	/	79	O	111	o	143	Å	175	»	207	╧	239	∩
16	►	48	0	80	P	112	p	144	É	176		208	╨	240	≡
17	◄	49	1	81	Q	113	q	145	æ	177	▮	209	╤	241	±
18	↕	50	2	82	R	114	r	146	Æ	178	▮	210	╥	242	≥
19	!!	51	3	83	S	115	s	147	ô	179	▮	211	╙	243	≤
20	¶	52	4	84	T	116	t	148	ö	180	┤	212	╘	244	⌠
21	§	53	5	85	U	117	u	149	ò	181	╡	213	╒	245	⌡
22	▬	54	6	86	V	118	v	150	û	182	╢	214	╓	246	÷
23	↨	55	7	87	W	119	w	151	ù	183	╖	215	╫	247	≈
24	↑	56	8	88	X	120	x	152	ÿ	184	╕	216	╪	248	°
25	↓	57	9	89	Y	121	y	153	Ö	185	╣	217	┘	249	∙
26	→	58	:	90	Z	122	z	154	Ü	186	║	218	┌	250	·
27	←	59	;	91	[	123	{	155	¢	187	╗	219	▮	251	√
28	∟	60	<	92	\	124	|	156	£	188	╝	220	▪	252	ⁿ
29	↔	61	=	93	]	125	}	157	¥	189	╜	221	▌	253	²
30	▲	62	>	94	^	126	~	158	Pt	190	╛	222	▐	254	▮
31	▼	63	?	95	_	127	⌂	159	*f*	191	┐	223	▪	255	

D: Código decimal.
P: Escritura del carácter correspondiente al código en la pantalla.

CODIFICACIÓN DE LA INFORMACIÓN (3)

Letra	Decimal ASCII	Binario
p	112	01110000
e	101	01100101
r	114	01110010
r	114	01110010
o	111	01101111

01110000 01100101 01110010 01110010 01101111

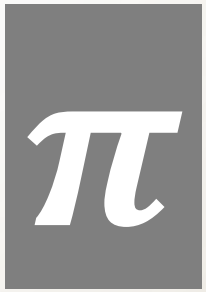
=

perro

- https://es.convertbinary.com/de-binario-a-texto/
- https://es.convertbinary.com/texto-a-binario/

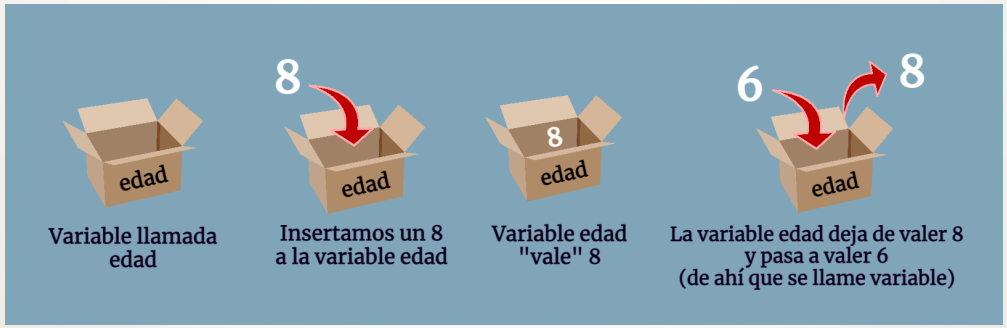
CONSTANTES Y VARIABLES

- Constantes: son aquellos objetos de datos cuyo valor se mantiene invariable (no cambia) durante la ejecución de un programa. Una constante recibe su valor al momento de la compilación del programa y este valor no será modificado durante la ejecución.



CONSTANTES Y VARIABLES

- Variables: son aquellos objetos de datos **cuyo valor se modifica** durante la ejecución de un programa a través de las operaciones que éste realiza.
- Tanto las *variables* como las *constantes* tienen un tipo de dato asociado.



nombre, sexo
curso, estado

OPERADORES (1)

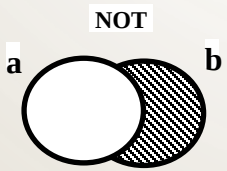
- Los operadores ejecutan acciones sobre los datos. La siguiente tabla presenta los operadores más utilizados en programación

Tipo	Símbolo	Nombre	Función
Paréntesis	( )		Anida expresiones
Aritméticos	** ó ^ *, / +, - div, mod	Potencia Producto, división Suma, diferencia División entera, resto	Conectan objetos o campos numéricos
Alfanuméricos	+	Concatenación	Conectan campos alfanuméricos
Relacionales	= < <= > >= <>	Igual a Menor que Menor o igual que Mayor que Mayor o igual que Distinto a	Conectan objetos, campos o expresiones de cualquier tipo. Su evaluación da como resultado "Verdadero" o "Falso".
Lógicos	NOT AND OR	Negación Conjunción Disyunción	Conectan expresiones de tipo lógico. Su evaluación da como resultado "Verdadero" o "Falso".

OPERADORES (2)
OPERADORES LÓGICOS

Los operadores lógicos "or", "not" y "and" se utilizan comúnmente en programación y en la lógica proposicional para construir expresiones booleanas que ayudan a determinar la veracidad o falsedad de una proposición.

Operador NO (NOT) *niega un valor*	
Negación de a	
A	NO a
Verdadero	Falso
Falso	Verdadero

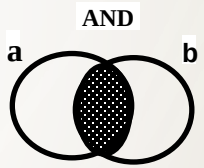


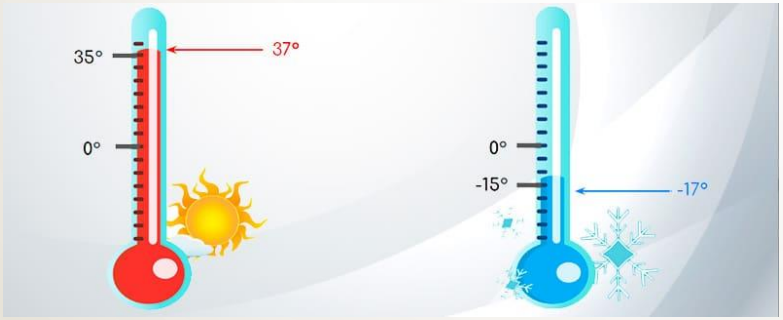
http://prepa8.unam.mx/academia/colegios/matematicas/paginacolmate/applets/tsm/Applets_Geogebra/oplogicos.html

OPERADORES (2)
OPERADORES LÓGICOS

	Operador Y (AND)		
	Conjunción de a y b		
	A	b	a Y b
	Verdadero	Verdadero	Verdadero
	Verdadero	Falso	Falso
	Falso	Verdadero	Falso
	Falso	Falso	Falso

AND (Y): Combina dos condiciones simples y produce un resultado verdadero solo si los dos componentes son verdaderos.



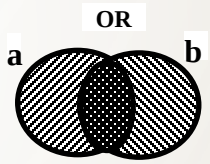


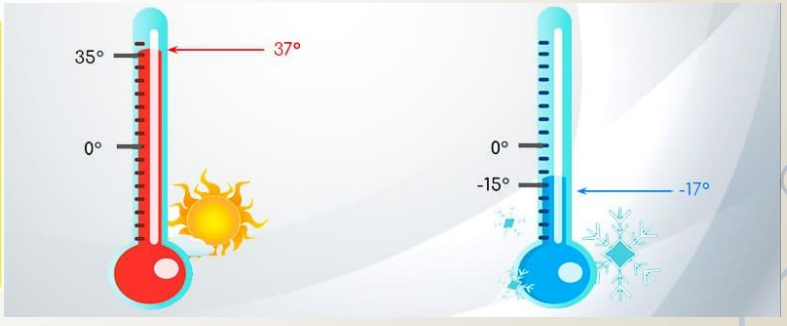
http://prepa8.unam.mx/academia/colegios/matematicas/paginacolmate/applets/tsm/Applets_Geogebra/oplogicos.html

OPERADORES (3)
OPERADORES LÓGICOS

	Operador O (OR)		
	Disyunción de a y b		
	A	***b***	***a O b***
	Verdadero	Verdadero	Verdadero
	Verdadero	Falso	Verdadero
	Falso	Verdadero	Verdadero
	Falso	Falso	Falso

OR (O): es verdadera si uno de los componentes es verdadero.





http://prepa8.unam.mx/academia/colegios/matematicas/paginacolmate/applets/tsm/Applets_Geogebra/oplogicos.html

OPERADORES (4)
OPERADORES DE RELACIÓN

Los operadores de relación se utilizan para expresar condiciones y describen una relación entre dos valores. El resultado de una expresión lógica es un valor de tipo lógico VERDADERO o FALSO.

Operador	Significado	Ejemplo
<	Menor que	a<b
<=	Menor o igual que	a<=b
=	Igual a	a=a
<>	Distinto a	a<>b
>=	Mayor o igual que	b>=a
>	Mayor que	b>a
Observación expresiones verdaderas en columna ejemplo considerando a=10 y b=20		

EXPRESIONES (1)

- Son combinaciones de constantes, variables, símbolos de operación y nombres de funciones especiales.
- De acuerdo a los datos y operadores que contengan las expresiones, éstas pueden ser *aritméticas*, *alfanuméricas* y *lógicas*.

Por ejemplo:

28 >= valor	Expresión lógica
3*6-12	Expresión aritmética
“Hola”+”mundo”	Expresión alfanumérica

EXPRESIONES (2)
PRECEDENCIA DE OPERADORES

- ¿En qué orden se resuelven las operaciones?

Operador	Prioridad
NO, ^	Más alta (se evalúa primero)
***, /, div, mod, Y**	
+, -, O	
<, <=, =, <>, >=, >	Más baja (se evalúa al final)
Si se utilizan paréntesis, las expresiones encerradas se evalúan primero.	

REGLA ASOCIADA IZQUIERDA: *Los operadores en una misma expresión o subexpresión con igual nivel de prioridad (tal como * y /) se evalúan de izquierda a derecha.*

EXPRESIONES (3)

- Reglas para escribir/resolver expresiones
  - Las operaciones entre paréntesis se resuelven primero, iniciando con los paréntesis más internos.
  - Las operaciones se resuelven de acuerdo a la tabla de prioridades.
  - Expresiones con operadores de igual prioridad, al mismo nivel, se resuelven de izquierda a derecha.

Expresión Original

$$\frac{-b+\sqrt{b^2-4\times a\times c}}{2\times a}$$

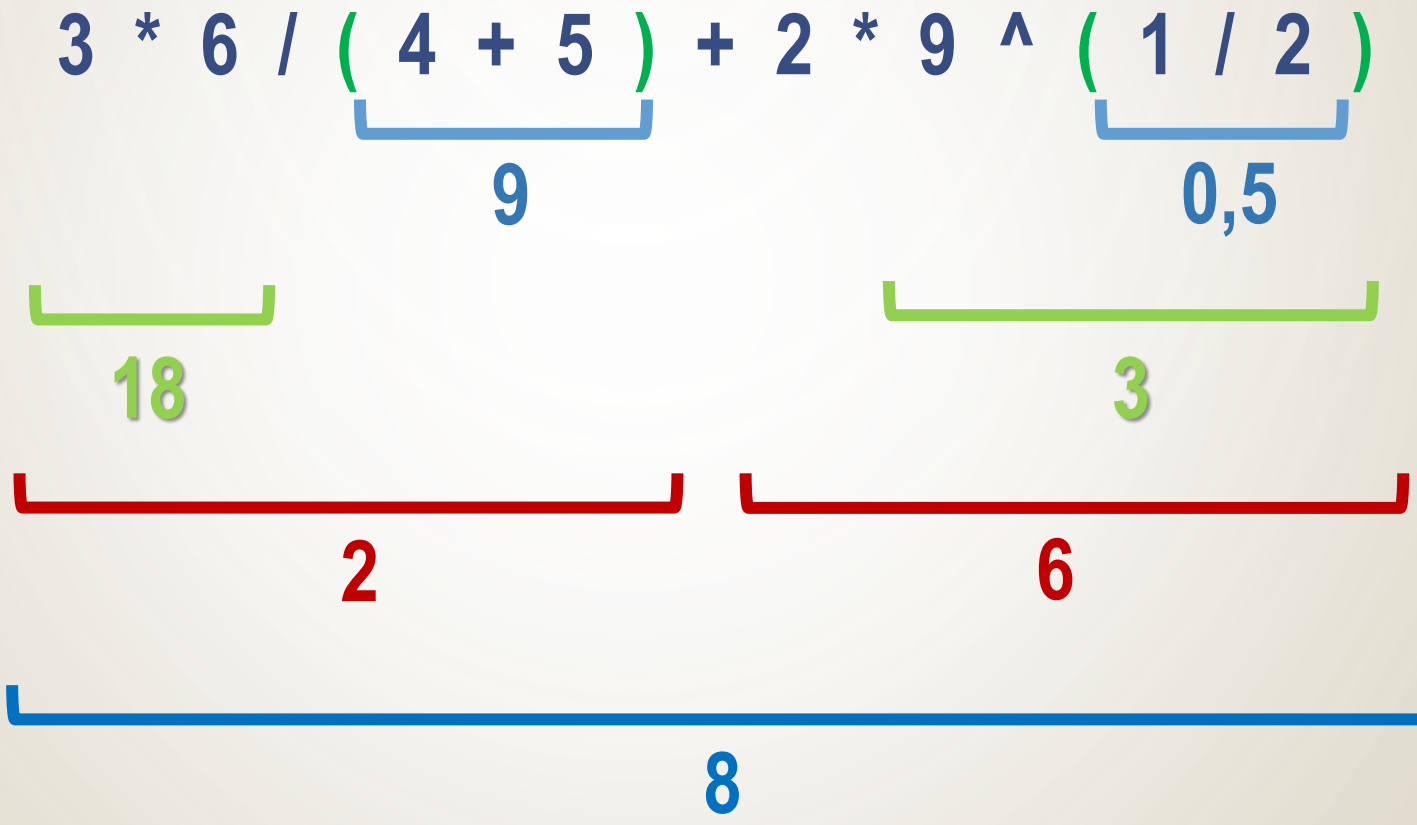
Expresión Algorítmica

(-b + (b ^ 2 – 4 * a * c) ^ (1 / 2))/(2 * a)

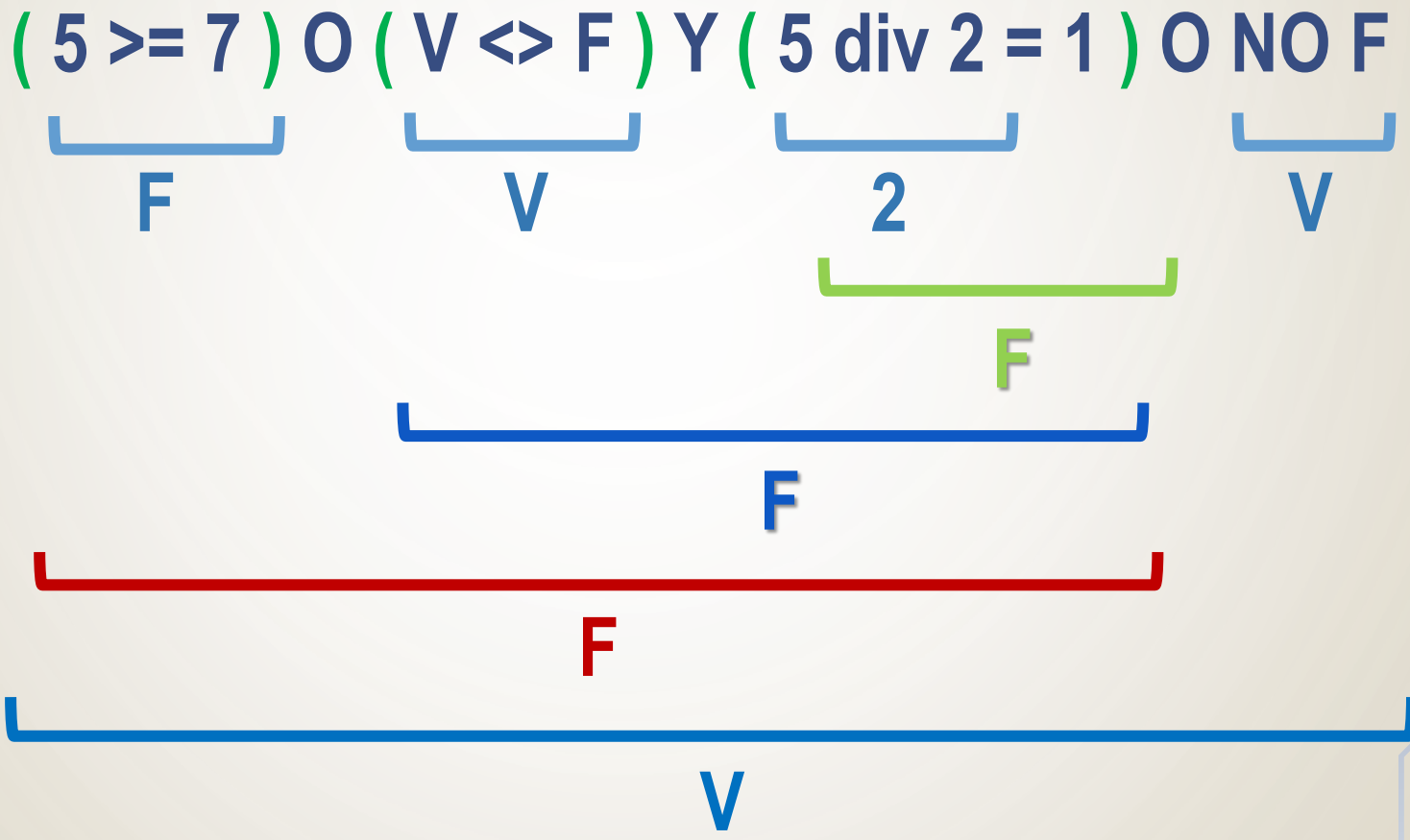
EXPRESIONES (4)

Expresión Original	Expresión Algorítmica
$2 \times m^2 + 4 \times m + 5$	2 * m ^ 2 + 4 * m + 5
$\frac{2 \times a}{b + c} + 5 \times \sqrt{b}$	2 * a / (b + c) + 5 * b ^ (1 / 2)
$\sqrt{b^2 + c^2}$	(b ^ 2 + c ^ 2) ^ (1 / 2)
$\frac{\frac{7 + a}{2 \times b}}{\sqrt[4]{3 \times c + 5}}$	(7 + a) / (2 * b) / (3 * c + 5) ^ (1 / 4)

EXPRESIONES (5)

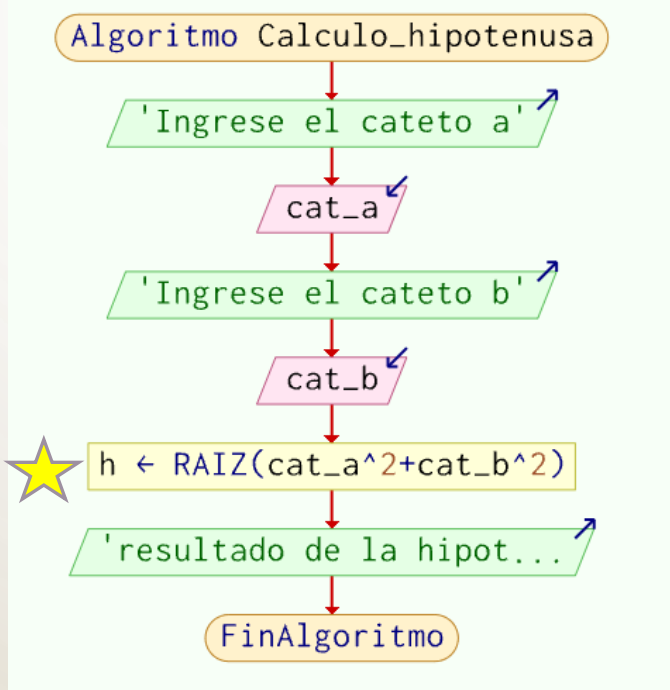


EXPRESIONES (6)



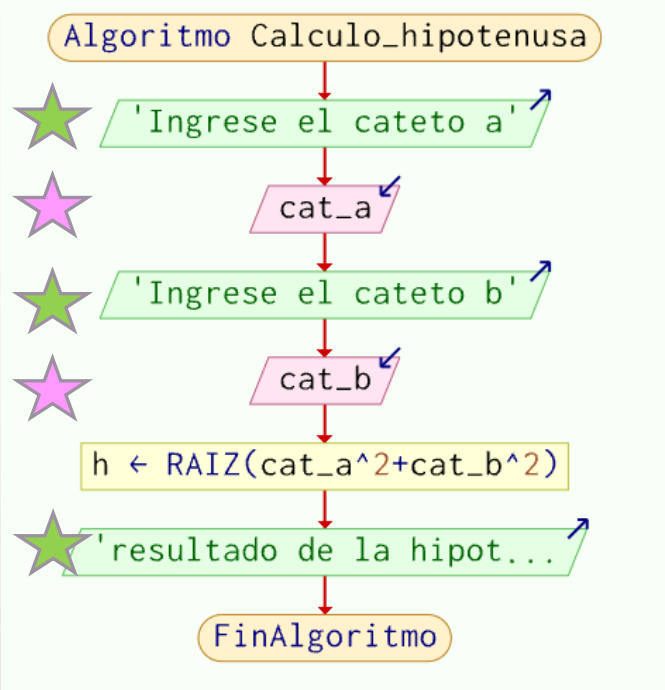
ASIGNACIÓN

- La operación de *asignación* (← en pseudocódigo) es el modo de darle valores a una variable.
- Es una operación destructiva.



ENTRADA/SALIDA

- La entrada y salida de información del proceso realizado por una computadora puede llevarse a cabo mediante las operaciones de Lectura y Escritura:



C++ (1) - INSTALACIÓN

- **Dev-C++** es un entorno de desarrollo integrado para programar en lenguaje C/C++. Usa MinGW, una versión de GCC, como compilador. Dev-C++ puede además ser usado en combinación con Cygwin y cualquier otro compilador basado en GCC. El Entorno está desarrollado en el lenguaje Delphi de Borland. Fuentes y sitio de descarga https://www.bloodshed.net/index.html.
- **CodeBlocks** es un entorno de desarrollo integrado libre y multiplataforma para el desarrollo de programas en lenguaje C++. Está basado en la plataforma de interfaces gráficas WxWidgets, por lo que puede usarse libremente en diversos sistemas operativos, y está bajo GNU. Es una herramienta para desarrollar programas en C++ que ofrece una interfaz sencilla a los usuarios. Fuentes y sitio de descarga http://www.codeblocks.org/downloads/binaries.
- Opciones online: http://cpp.sh/, https://www.jdoodle.com/online-compiler-c++, entre otros.
- App: http://play.google.com/store/apps/details?id=com.duy.c.cpp.compiler

C++ (2) - ESTRUCTURA GENERAL DE PROGRAMA

- **Declaraciones**
  - Librerías (*include* indica al compilador qué librerías incluir en el programa objeto para generar el ejecutable)
  - Módulos (tipo y argumentos de los módulos)
  - Variables Globales (tipos e identificadores)
  - Constantes
- **Programa Principal**
  - La función *main* contiene declaraciones de variables e instrucciones necesarias para controlar las operaciones que ejecuta el programa.
- **Módulos**
  - Se especifica el código correspondiente a cada módulo o componente de programa.

ESTRUCTURA GENERAL DE PROGRAMA (2)

```
/* Archivos de cabecera */
#include <archivo_cabecera.h>
#include <archivo_cabecera.h>
```

Librerías del Lenguaje

```
/* Prototipos de funciones del programador */
tipo_dato función1 (argumentos);
tipo_dato función2 (argumentos);
```

Módulos definidos por el programador

```
/* Variables y constantes globales */
tipo_dato variable_global;
const tipo_dato constante_global=valor;
#define PI 3.14
```

ESTRUCTURA GENERAL DE PROGRAMA (3)

```
/* Algoritmo principal */
tipo_dato main(argumentos)
{ /*Variables locales del algoritmo principal */
   tipo_dato nombre_variable1, nombre_variable2;
   tipo_dato nombre_variable3, nombre_variable4;
   ...
   /* Instrucciones del algoritmo principal */
   función1(argumentos);
   ...
   función2(argumentos);
   ...
   return valor;
}
```

ESTRUCTURA GENERAL DE PROGRAMA (4)

```
/* Código completo de las funciones del programador*/

tipo_dato función1 (argumentos)
{
   /* Variables locales e instrucciones del módulo */
}

tipo_dato función2 (argumentos)
{
   /* Variables locales e instrucciones del módulo */
}
```

TIPOS DE DATOS EN C/C++

Nombre	Descripción	Tamaño	Rango
char	Caracter (código ASCII)	8 bits	Con signo: -128 ... 127 Sin signo: 0 ... 255
short int (short)	Número Entero corto	16 bits	Con signo: -32768 ... 32767 Sin signo: 0 ... 65535
int	Número Entero	32 bits	Con signo: -2147483648 ... 2147483647 Sin signo: 0 ... 4294967295
long int (long)	Número Entero largo	64 bits	Con signo: -9223372036854775808, 9223372036854775807 Sin signo: 0 ... 18446744073709551615
float	Número real	32 bits	3,4*10-38 ... 3,4*10+38 (6 decimales)
double	Número real en doble precisión	64 bits	1,7*10-308 ... 1,7*10+308 (15 decimales)
long double	Número real largo de doble precisión	80 bits	3,4*10-4932 ... 1,1*10+4932
bool	Valor booleano	1 bit	true (VERDADERO) o false (FALSO)

OPERADORES EN C/C++ (1)

ARITMÉTICOS

Operador	Acción	Ejemplo	Resultado
-	Resta	X = 5 - 3	X vale 2
+	Suma	X = 5 + 3	X vale 8
*	Multiplicación	X = 2 * 3	X vale 6
/	División	X = 6 / 3	X vale 2
%	Módulo	X = 5 % 2	X vale 1
--	Decremento	X = 1; X--	X vale 0
++	Incremento	X = 1; X++	X vale 2

OPERADORES EN C/C++ (2)

LÓGICOS

Operador	Acción	Ejemplo	Resultado
&&	AND Lógico	A && B	Si ambos son verdaderos se obtiene verdadero(true)
||	OR Lógico	A || B	Verdadero si alguno es verdader
!	Negación Lógica	!A	Negación de a

OPERADORES EN C/C++ (3)

RELACIONALES

Operador	Relación	Ejemplo	Resultado
<	Menor	X = 5; Y = 3; if(x < y) x+1;	X vale 5 Y vale 3
>	Mayor	X = 5; Y = 3; if(x > y) x+1;	X vale 6 Y vale 3
<=	Menor o igual	X = 2; Y = 3; if(x <= y) x+1;	X vale 3 Y vale 3
>=	Mayor o igual	X = 5; Y = 3; if(x >= y) x+1;	X vale 6 Y vale 3
==	Igual	X = 5; Y = 5; if(x == y) x+1;	X vale 6 Y vale 5
!=	Diferente	X = 5; Y = 3; if(x ¡= y) y+1;	X vale 5 Y vale 4

OPERADORES EN C/C++ (4)

DE ASIGNACIÓN

Operador	Acción	Ejemplo	Resultado
=	Asignación Básica	X = 6	X vale 6
*=	Asigna Producto	X *= 5	X vale 30
/=	Asigna División	X /= 2	X vale 3
+=	Asigna Suma	X += 4	X vale 10
-=	Asigna Resta	X -= 1	X vale 5
%=	Asigna Modulo	X %= 5	X vale 1

EJERCICIOS (1)

- Identifique el tipo de los datos de las siguientes variables, de acuerdo al tipo de Dato que contiene:

Variable	Dato contenido	Tipo de Dato
M	'D'	
Letra	"verdadero"	
Es	100	
L2	4.25	
Num	FALSO	

- Escriba las siguientes expresiones algebraicas como expresiones algorítmicas:

$$\left(\left(\frac{x+5y}{x}\right)^2 - \frac{7x+1}{5y}\right)^3$$

- Analice las siguientes operaciones de asignación y determine los valores finales de las variables A, B, y C.

Asignación	A	B	C
A←22*2 div 5			
B←24 mod 2 + 5			
C←A			

EJERCICIOS (2)

- Considerando las tablas de verdad de los operadores lógicos, resuelva las siguientes expresiones lógicas:

Expresión lógica	Resultado	Observación
(num <= 33) Y (num> 13)		num es una variable numérica que tiene valor 20
(4 <> cuenta) O (5 >= 4)		cuenta es una variable numérica que tiene valor 4
Resta = nro1 – nro2		nro1 y nro2 son variables numéricas que valen 10 y 18, respectivamente. Resta = -8

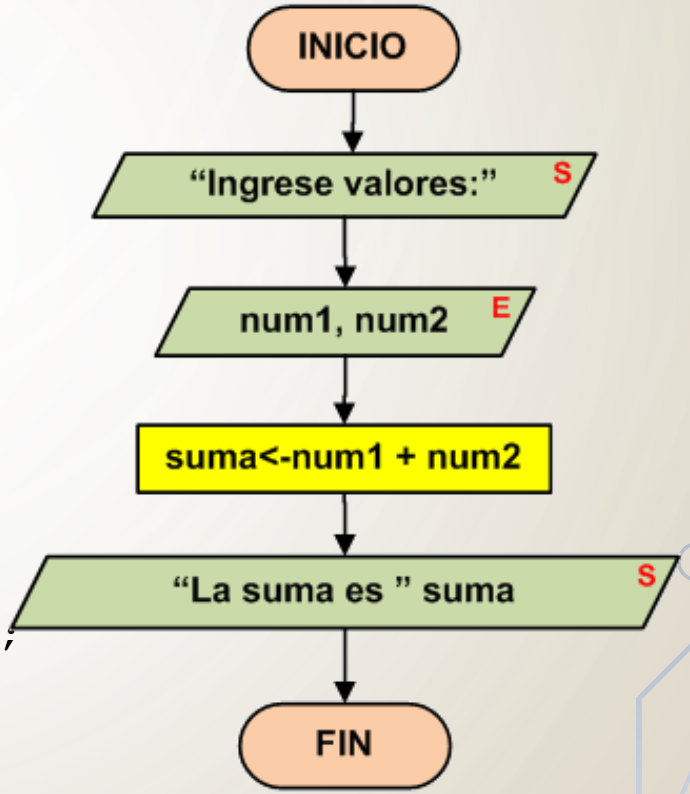
- Resuelva las siguientes expresiones teniendo en cuenta la prioridad de los operadores.

Expresión	Resultado	Observación
5 * x ^ 2 + 5 * y + 12		x=2, y=4
(12 + 8* 3) div 2		
64 / x * r - s * r mod r * 2		x=8, r=4, s=44
40 / 4 * 5 + 3		

EJERCICIOS (3)

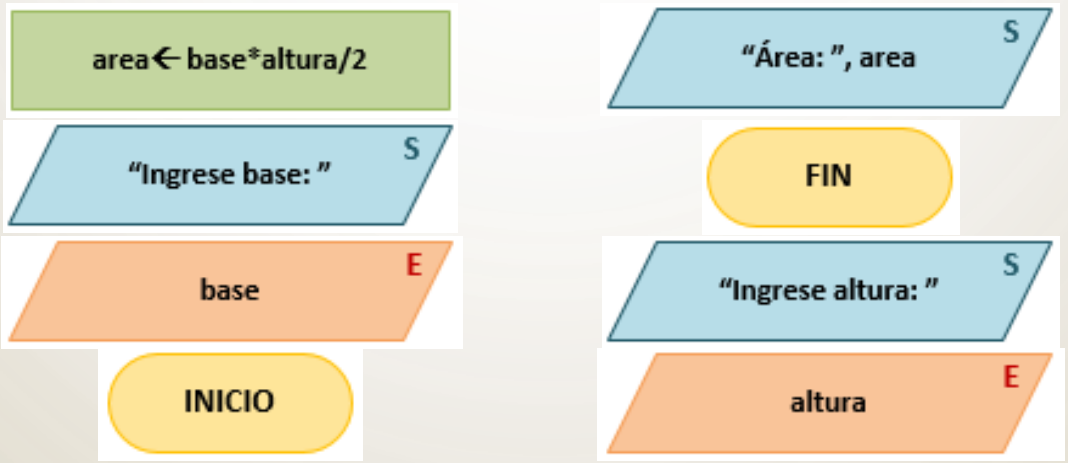
- Diseñe un algoritmo que sume 2 valores ingresados por el usuario. Utilice diagrama de flujo y C++.

```
#include <iostream>

using namespace std;

main()
{ int num1,num2,suma;
  cout << "Ingrese valor: ";
  cin >> num1;
  cout << "Ingrese valor: ";
  cin >> num2;
  suma=num1+num2;
  cout << "Resultado" << suma << endl;
  system("pause");
}
```



EJERCICIOS (3)

- Diseñe un algoritmo (diagrama de flujo y pseudocódigo) que muestre el mensaje “Mi primer PROGRAMA EN C++”.
- Un programador diseñó un algoritmo para calcular el área de un triángulo, sin embargo, olvidó conectar los símbolos del diagrama de flujo que definen el orden de los pasos del algoritmo. ¿Podrías ordenar y conectar los símbolos de la solución propuesta por este programador? Realice la solución en C++.



BIBLIOGRAFÍA

- Sznajdleder, Pablo Augusto. Algoritmos a fondo. Alfaomega. 2012.
- Joyanes Aguilar, Luis. Fundamentos de Programación. Mc Graw Hill. 1996.
- Hernández, Roberto *et al.* Estructuras de datos y algoritmos. Prentice Hall. 2001.
- Fundamentos Básicos de Programación en C++. Martínez del Rio. Jaén. 2015
- Curso de C++. https://www.programarya.com/Cursos/C++