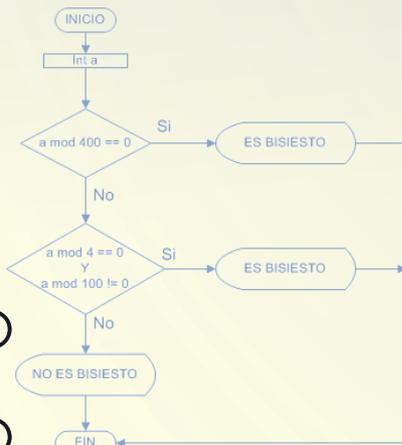
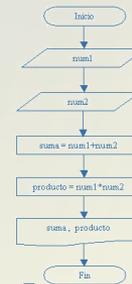


FUNDAMENTOS DE PROGRAMACIÓN

TEORÍA N° 1: INTRODUCCIÓN



FUNDAMENTOS DE PROGRAMACIÓN

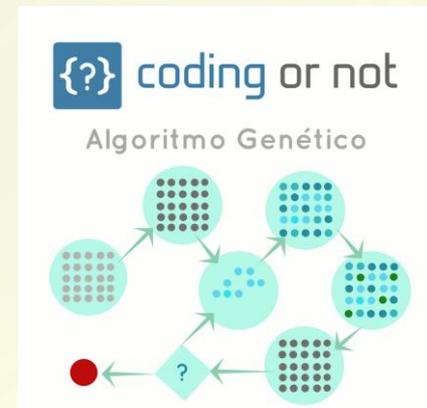
Facultad de Ingeniería
Universidad Nacional de Jujuy

ÍNDICE

- ▶ Algoritmos
- ▶ Solución de problemas mediante computadores
- ▶ Conceptos básicos
- ▶ Clasificación (tipos de datos)
 - ▶ Simples (estándar y definidos por el usuario)
 - ▶ Compuestos o estructurados
- ▶ Elementos básicos: variables y constantes
- ▶ Operadores y precedencia
- ▶ Expresiones
- ▶ Asignación, lectura y escritura

ALGORITMOS

- ▶ Un algoritmo es una secuencia de pasos que de forma ordenada permite resolver un problema específico.
- ▶ Características
 - **Preciso** (indica el orden de pasos)
 - **Definido** (para la misma entrada siempre produce el mismo resultado)
 - **Finito** (tiene un N° finito de pasos)



METODOLOGÍA

- ▶ La *metodología de la programación* provee los métodos, las técnicas y las herramientas que transforman un algoritmo en un programa de computadora.
- ▶ El *objetivo de la programación* es dar solución, usando la computadora, a problemas que resultan engorrosos o difíciles para las personas.
 - Métodos: Indican la secuencia de tareas necesaria para realizar un trabajo.
 - Técnicas: Indican cómo realizar una tarea y las herramientas a utilizar.
 - Herramientas: Son los elementos que se aplican al realizar una tarea.

SOLUCIÓN POR COMPUTADORA

► Fases

1. Análisis del Problema
2. Diseño del Algoritmo
3. Codificación
4. Compilación y Ejecución
5. Verificación y Depuración
6. Documentación y Mantenimiento



ANÁLISIS DEL PROBLEMA

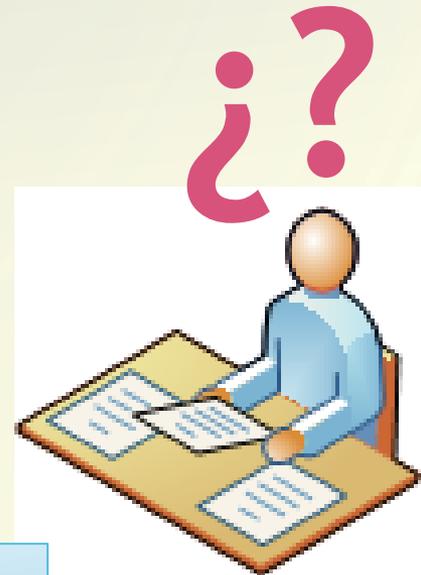
- ▶ Se debe identificar claramente:
 - el problema a resolver,
 - las entradas de datos,
 - los resultados a obtener y
 - el objetivo del algoritmo.

Problema
Sumar 2 valores

Entradas
2 números

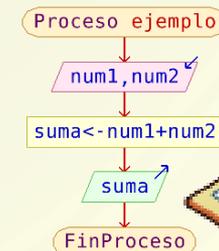
Salida
1 número

Objetivo
Calcular la suma



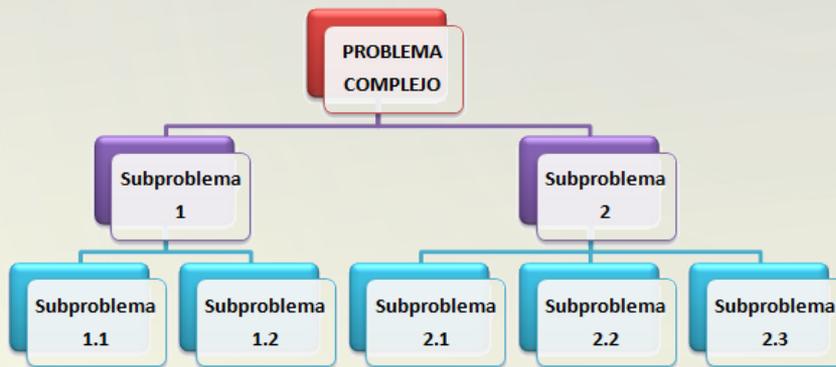
DISEÑO DEL ALGORITMO (1)

- ▶ Se establecen los pasos y el orden en qué deben aplicarse para resolver el problema planteado.
- ▶ Descomposición del problema
 - Diseño Top-Down
 - Refinamiento Sucesivo
 - Herramientas de especificación de algoritmos (diagramas de flujo, diagramas N-S, pseudocódigo, fórmulas, lenguaje natural)



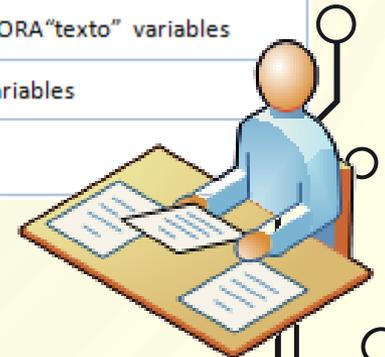
DISEÑO DEL ALGORITMO (2)

Descomposición de Problemas



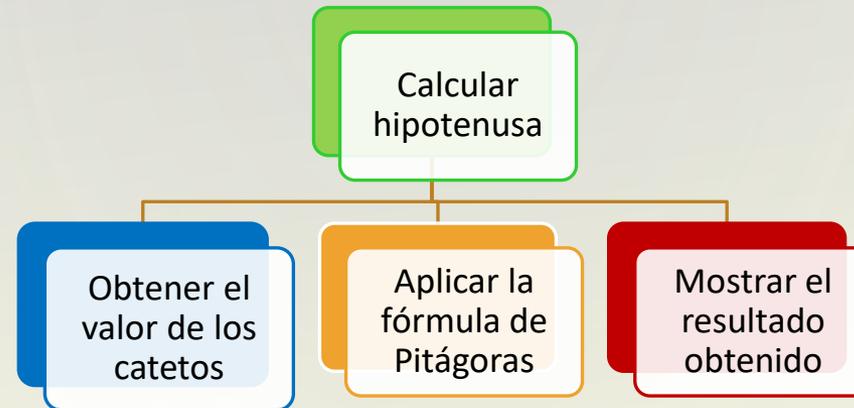
Herramientas de Especificación de Algoritmos

Diagrama de Flujo	Pseudocódigo
	INICIO ... FIN
	ESCRIBIR "texto" variables LEER variables
	variable ← operación
	SI condición ENTONCES FIN_SI
	variable ← función(par1, par2, par3) procedimiento(par1, par2, par3)
	ESCRIBIR PANTALLA "texto" variables
	ESCRIBIR IMPRESORA "texto" variables
	LEER TECLADO variables
	// comentarios

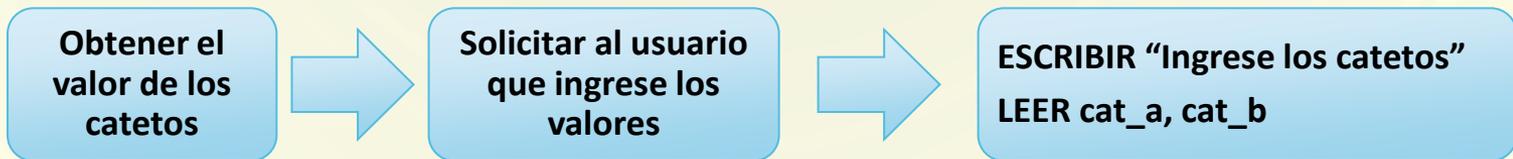


DISEÑO DEL ALGORITMO (3)

Descomposición de Problemas



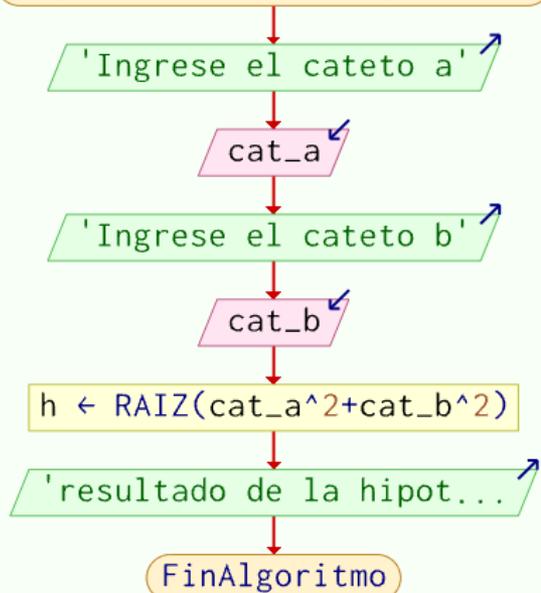
Refinamiento Sucesivo



CODIFICACIÓN (1)

- ▶ Consiste en traducir el diseño del algoritmo a algún lenguaje de programación específico.

Algoritmo Calculo_hipotenusa



FUNDAMENTOS DE PROGRAMACIÓN

Lenguaje de Programación

The screenshot shows the Turbo C++ IDE. The main window displays the source code for a program that prompts for a name and prints it. The code is as follows:

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

main()
{
    char nombre[30];
    char *ch;

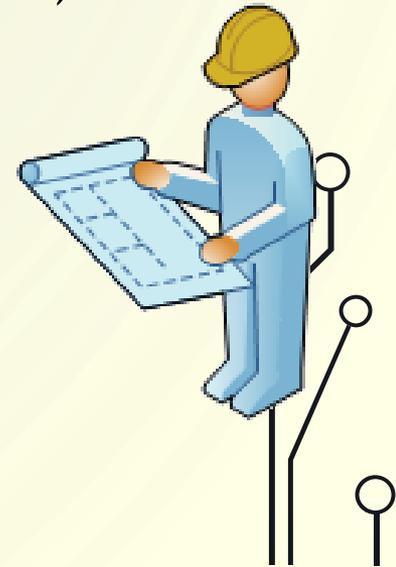
    ch = nombre;
    printf("Ingresar un nombre: ");
    gets(nombre);
    printf("\n El nombre es: ");
    puts(ch);
}
```

The output window shows the program's execution: 'Ingresar un nombre: Rodolfo Panduro' followed by 'El nombre es: Rodolfo Panduro'. The status bar at the bottom indicates 'Program running'.

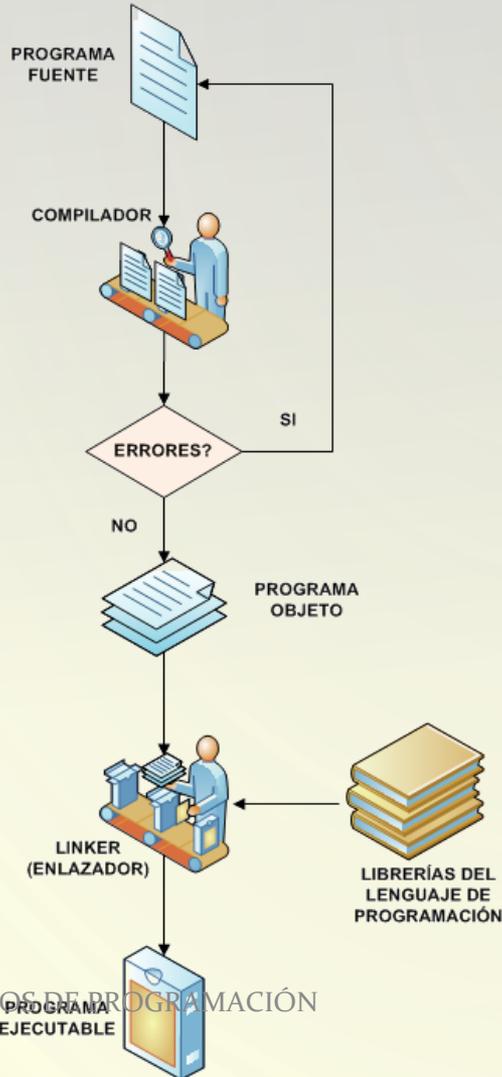
CODIFICACIÓN (2)

► Lenguajes de Programación

- Lenguaje Máquina: secuencias binarias (0's y 1's) que especifican instrucciones y datos directamente comprensibles por la computadora.
- Lenguajes de Bajo Nivel (ensamblador): instrucciones en especificadas códigos especiales llamados nemotécnicos (Por ejemplo, STAA \$0400)
- Lenguajes de Alto Nivel: diseñados para la fácil comprensión del ser humano.



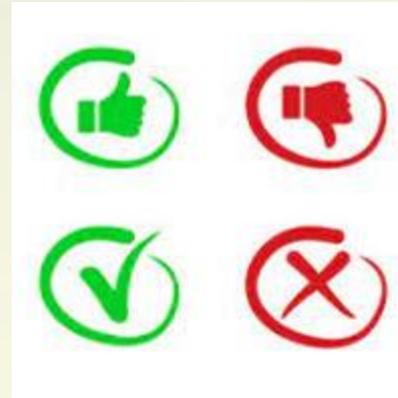
COMPILACIÓN Y EJECUCIÓN



- ▶ Un compilador es programa especial que traduce un programa fuente a lenguaje máquina (programa objeto).
- ▶ Un Linker (enlazador) es un programa que genera un programa ejecutable al agregar librerías (funciones especiales) a un programa objeto.

VERIFICACIÓN Y DEPURACIÓN

- ▶ La verificación comprueba el algoritmo a través de un conjunto de datos de prueba o test (valores normales, extremos y aspectos especiales).
- ▶ La depuración detecta, corrige y/o elimina errores en el programa.
 - Errores de compilación
 - Errores de ejecución
 - Errores de lógica



DOCUMENTACIÓN Y MANTENIMIENTO

▶ Documentación Interna

- Comentarios

▶ Documentación Externa

- Manuales, diseños, análisis del problema

▶ Mantenimiento

- **Correctivo** (localiza y **elimina errores** detectados luego de la entrega del programa)
- **Preventivo** (**mejora las propiedades** de un programa sin alterar su funcionalidad)
- **Adaptativo** (**adapta** un programa a nuevas condiciones del entorno)
- **Perfectivo** (mejora o **añade nuevas funcionalidades** a un programa)

LENGUAJE NATURAL Y DFD

Algoritmo: “Quiero café”

¿Hay café hecho?

Si hay café hecho entonces

Calentar café

Si no hay café hecho

Hacer café

Servir café en taza

Añadir azúcar

Tomar el café.



PSEUDOCÓDIGO

Inicio

Escribir "Calentar agua"

Escribir "Colocar café en el filtro"

Escribir "Verter agua caliente sobre el café"

Escribir "Esperar a que se filtre"

Escribir "Servir el café en una taza"

Escribir "¿Desea agregar azúcar o leche?"

Leer respuesta

Si respuesta = "sí" Entonces

 Escribir "Agregar azúcar o leche"

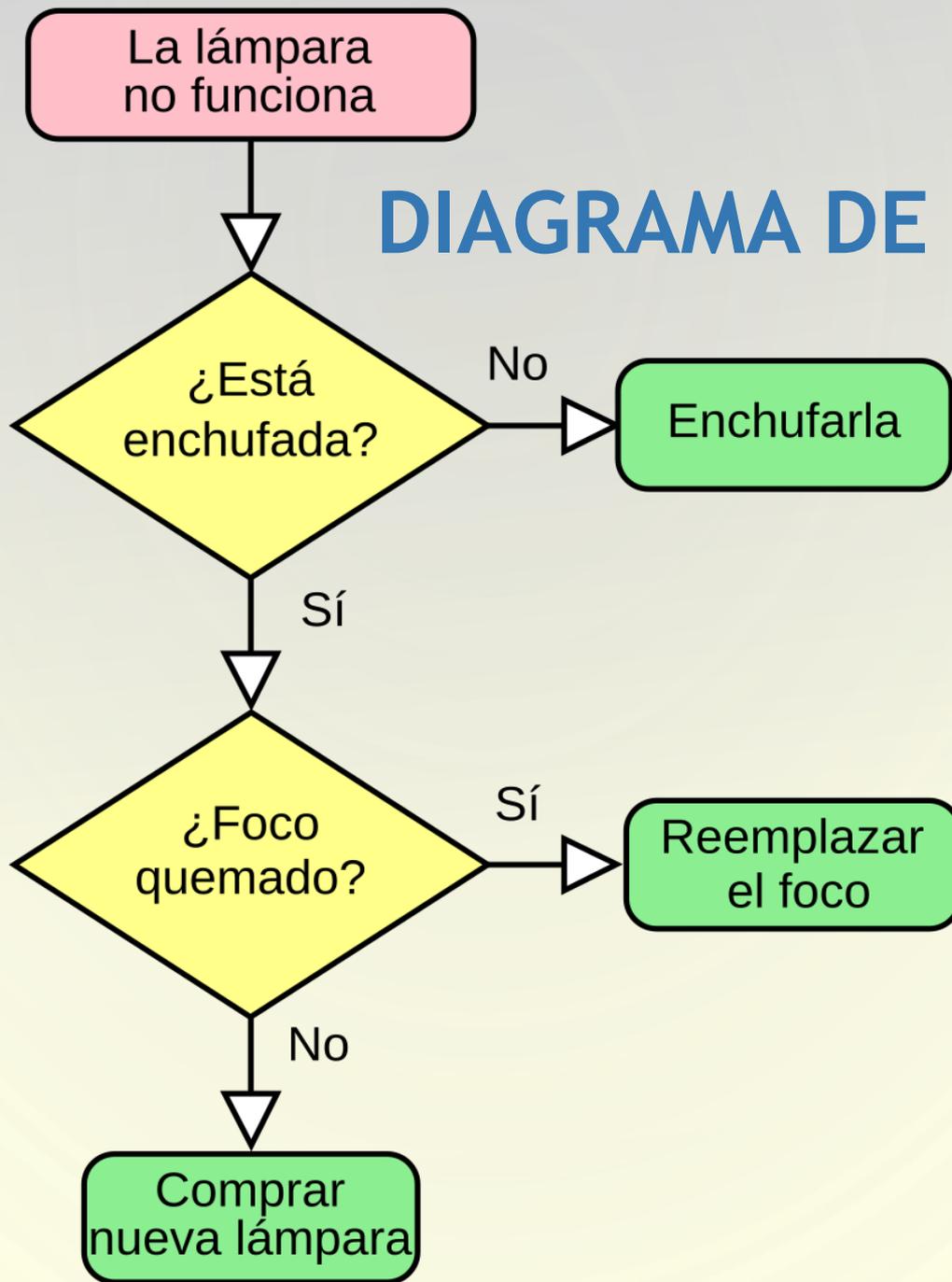
FinSi

Escribir "Disfrutar el café"

Fin

<https://rea.ceibal.edu.uy/elp/programacion-a-traves-de-algoritmos/index.html>

DIAGRAMA DE FLUJO



PSEUDOCÓDIGO

Inicio

Definir nota1, nota2, nota3, promedio como Real

Escribir "Ingrese la primera nota:"

Leer nota1

Escribir "Ingrese la segunda nota:"

Leer nota2

Escribir "Ingrese la tercera nota:"

Leer nota3

$\text{promedio} \leftarrow (\text{nota1} + \text{nota2} + \text{nota3}) / 3$

Escribir "El promedio es:", promedio

Fin

<https://rea.ceibal.edu.uy/elp/programacion-a-traves-de-algoritmos/index.html>

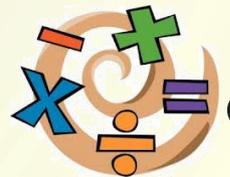
INTERCAMBIO DE IDEAS PARA PROBLEMAS COTIDIANOS



<https://padlet.com/mvillarrubia/intercambio-de-ideas-para-problemas-cotidianos-ydhtbz8is56bveg9>

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.)

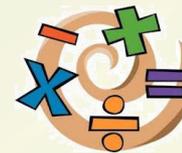


CONCEPTOS BÁSICOS (2)

- 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)

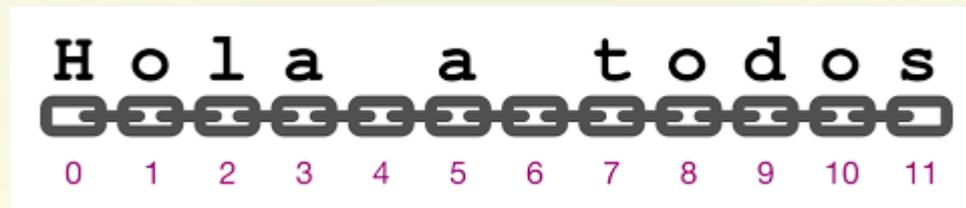
- ¿Qué es un tipo de dato ORDINAL?
 - Se trata de los tipos de datos en los que
 - ✓ Por cada valor se conoce el inmediato anterior (predecesor) y el inmediato siguiente (sucesor)
 - ✓ Existe un primer elemento del conjunto de valores (no tiene predecesor)
 - ✓ Existe un último elemento del conjunto 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		
0		32		64	@	96		128	Ç	160	á	192	ˆ	224	α
1	☺	33	!	65	A	97	a	129	ü	161	í	193	ˆ	225	β
2		34	"	66	B	98	b	130	é	162	ó	194	ˆ	226	Γ
3	♥	35	#	67	C	99	c	131	â	163	ú	195	ˆ	227	π
4	♦	36	\$	68	D	100	d	132	ä	164	ñ	196	ˆ	228	Σ
5	♣	37	%	69	E	101	e	133	à	165	Ñ	197	ˆ	229	σ
6	♠	38	&	70	F	102	f	134	â	166	ª	198	ˆ	230	μ
7	•	39	'	71	G	103	g	135	ç	167	º	199	ˆ	231	γ
8	◻	40	(72	H	104	h	136	ê	168	¿	200	ˆ	232	φ
9	◯	41)	73	I	105	i	137	ë	169	ƒ	201	ˆ	233	θ
10	■	42	*	74	J	106	j	138	è	170	ƒ	202	ˆ	234	Ω
11	♂	43	+	75	K	107	k	139	ï	171	½	203	ˆ	235	δ
12	♀	44	,	76	L	108	l	140	î	172	¼	204	ˆ	236	ò
13	♪	45	-	77	M	109	m	141	ì	173	¾	205	ˆ	237	Ø
14	♫	46	.	78	N	110	n	142	Ä	174	«	206	ˆ	238	ε
15	☀	47	/	79	O	111	o	143	Å	175	»	207	ˆ	239	Ɔ
16	▶	48	0	80	P	112	p	144	É	176		208	ˆ	240	≡
17	◀	49	1	81	Q	113	q	145	æ	177	■	209	ˆ	241	≠
18	↕	50	2	82	R	114	r	146	Æ	178	■	210	ˆ	242	≡
19	!!!	51	3	83	S	115	s	147	ô	179		211	ˆ	243	∩
20	¶	52	4	84	T	116	t	148	ö	180	└	212	ˆ	244	∩
21	§	53	5	85	U	117	u	149	ò	181	├	213	ˆ	245	∩
22	■	54	6	86	V	118	v	150	ù	182	┌	214	ˆ	246	∩
23	↕	55	7	87	W	119	w	151	ù	183	┐	215	ˆ	247	∩
24	↑	56	8	88	X	120	x	152	ÿ	184	┘	216	ˆ	248	∩
25	↓	57	9	89	Y	121	y	153	Ï	185	┘	217	ˆ	249	∩
26	→	58	:	90	Z	122	z	154	Ü	186		218	ˆ	250	∩
27	←	59	;	91	[123	{	155	ç	187	┘	219	ˆ	251	∩
28	┌	60	<	92	\	124		156	£	188	┘	220	ˆ	252	∩
29	↔	61	=	93]	125	}	157	¥	189	┘	221	ˆ	253	∩
30	▲	62	>	94	^	126	~	158	Pt	190	┘	222	ˆ	254	∩
31	▼	63	?	95	_	127	△	159	f	191	┘	223	ˆ	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

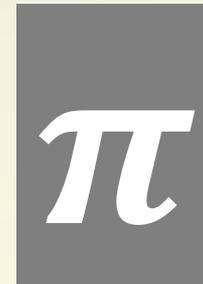
=

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

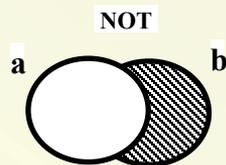
Tipo	Símbolo	Nombre	Función
Paréntesis	()		Anida expresiones
Aritméticos	** ó ^ *, / +, - div, mod	Potencia Producto, división Suma, diferencia División entera, resto	Conectan objetos o campos numéricos
Alfanuméricos	+	Concatenación	Conectan campos alfanuméricos
Relacionales	= < <= > >= <>	Igual a Menor 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) <i>niega un valor</i>	
Negación de a	
A	NO a
Verdadero	Falso
Falso	Verdadero

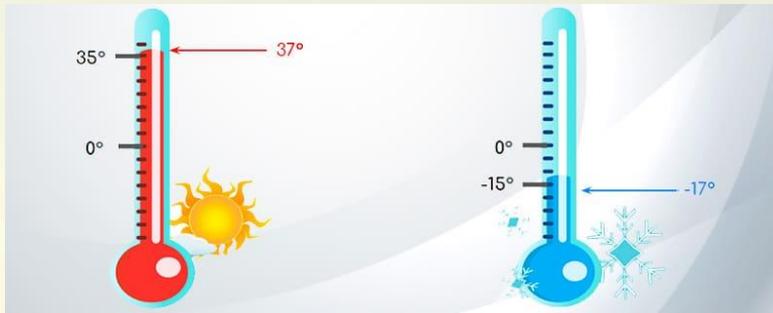
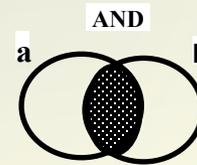


OPERADORES (2)

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.



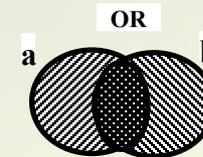
http://prepa8.unam.mx/academia/colegios/matematicas/paginacolmate/applets/tsm/Applets_Geogebra/oplogicos.html

OPERADORES (3)

OPERADORES LÓGICOS

Operador O (OR)		
Disyunción de a y b		
A	b	$a \text{ O } b$
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:

$28 \geq \text{valor}$

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)

$$3 * 6 / \underbrace{(4 + 5)}_9 + 2 * 9 \wedge \underbrace{(1 / 2)}_{0,5}$$

$$\underbrace{3 * 6}_{18} \quad \underbrace{2 * 9 \wedge (1 / 2)}_3$$

$$\underbrace{18 / 9}_2 \quad \underbrace{2 * 9 \wedge 0,5}_6$$

$$\underbrace{2 + 6}_8$$

EXPRESIONES (6)

$(5 \geq 7) \text{ O } (V \Leftrightarrow F) \text{ Y } (5 \text{ div } 2 = 1) \text{ O } \text{NO } F$

$\underbrace{\hspace{10em}}_F$

$\underbrace{\hspace{10em}}_V$

$\underbrace{\hspace{10em}}_2$

$\underbrace{\hspace{10em}}_V$

$\underbrace{\hspace{10em}}_F$

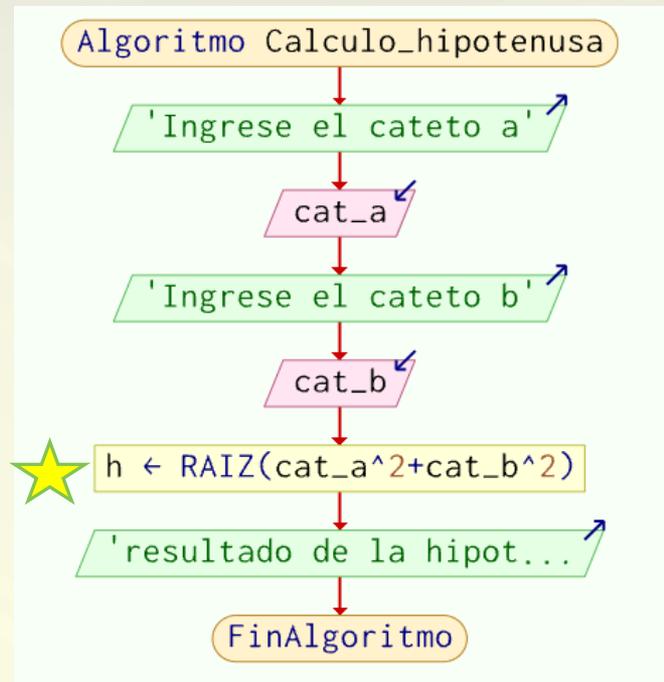
$\underbrace{\hspace{10em}}_F$

$\underbrace{\hspace{10em}}_F$

$\underbrace{\hspace{10em}}_V$

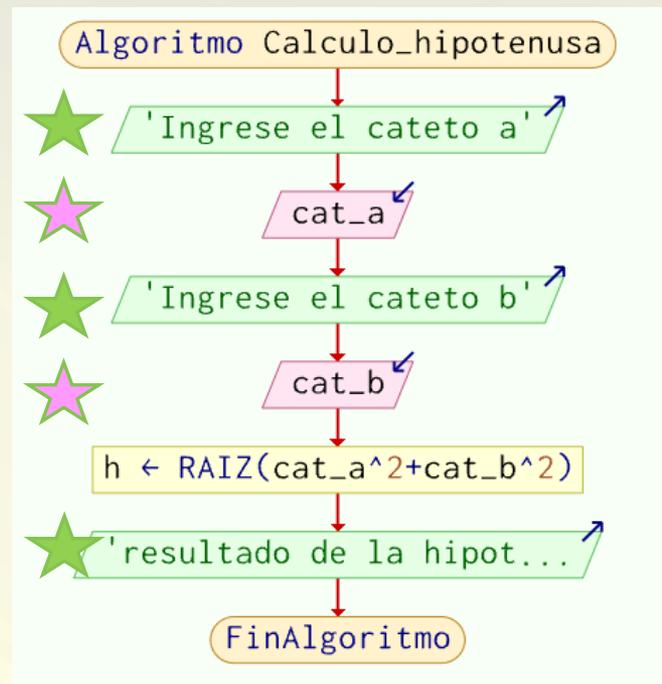
ASIGNACIÓN

- La operación de *asignación* (\leftarrow 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**:





FUNDAMENTOS DE PROGRAMACIÓN