



UNIVERSIDAD NACIONAL
DE MAR DEL PLATA



PROGRAMACIÓN I

Capítulo 1

Creación de un programa simple (estructura secuencial)

Algoritmo:

Secuencia ordenada y finita de pasos,
exenta de ambigüedades
que lleve a la solución de un problema dado.
Cada paso describe una acción.



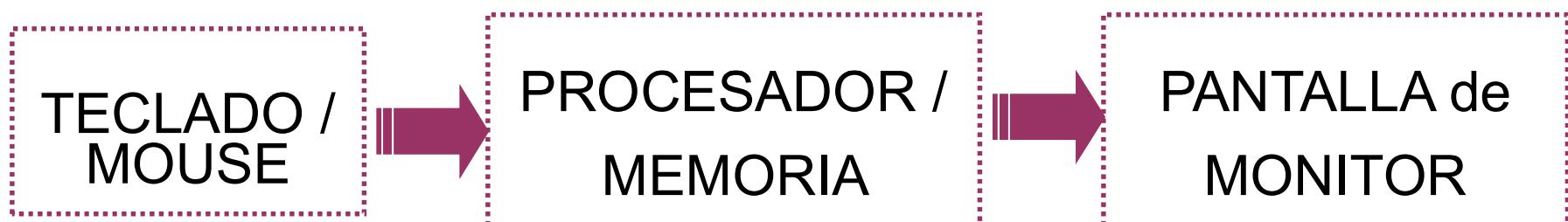
Procesador, entidad capaz de comprender un enunciado y ejecutarlo.

Para desarrollar algoritmos (programas) que ejecutara **una computadora** se debe conocer que tipo de **tareas** puede comprender y realizar, como así también **la sintaxis** para expresarlas.

Como punto de partida, se considera un modelo computacional muy sencillo



La computadora que corresponde a este modelo es



Los datos son
tipeados en el teclado
e ingresan por lectura
en orden,

van sufriendo
transformaciones para
llegar a los resultados
requeridos y

su visualización en el
monitor se realiza
mediante escritura

Estas especificaciones requieren del concepto de variable, nombre simbólico que se asocia a un valor durante la ejecución del programa, pero que no puede determinarse (está vacía) en el momento que se construye o formula el programa .

Las instrucciones de un programa se ejecutan en el orden en el que han sido escritas, con una sintaxis específica (Pascal) que comprende la computadora, deben ser eficaces para que produzcan la salida correcta para todas las posibles variaciones de la entrada (incluso para valores críticos).

En general, los programas leen datos, los operan y escriben los resultados.

SINTAXIS DEL LENGUAJE PASCAL

Utilizaremos el lenguaje Pascal para desarrollar, codificar y ejecutar las soluciones. A continuación se describe algunos elementos y ciertas características generales.

Programa Pascal

Program Compras;

DECLARATIVA

Var

N, CantCerveza, CantPapas : integer;
CantMani : integer;

Begin

Readln(N);
CantCerveza := Round (N * 0.5 /0.75);
CantPapas := Round (N * 30 /60);
CantMani := Round (N * 20 /35);
Writeln(CantCerveza, CantPapas, CantMani);

End.

E
J
E
C
U
T
A
B
L
E

Principales características del lenguaje Pascal

Entidades que intervienen en el programa:

- Palabras reservadas - Identificadores (Variables, Constantes)
- Tipos de datos y operadores. Expresiones.
- Acciones básicas: Lectura, Escritura, Asignación.
- Ejemplos.

Palabras Reservadas

Existen palabras ***propias*** del lenguaje con un uso específico como: Program, Begin, End, Function, etc. que no pueden utilizarse como identificadores de otras entidades

Identificadores

Llamamos **Identificador** al nombre que asignamos a las distintas entidades que intervienen en el programa (variables, constantes, etc), debe comenzar con una letra y a continuación letras, dígitos o el carácter '_'.

El lenguaje Pascal, no distingue entre mayúsculas y minúsculas.

Es conveniente que el identificador sea representativo de su rol en el programa. Por ejemplo si se trata de una variable, del valor que almacena (Precio, Nombre, etc.)

No hay restricción respecto a la longitud, pero debe considerarse la legibilidad así como también la facilidad para escribir y comprender el código.

Variables - Constantes

$$ax^2 + bx + c$$

$$(a + b)^n$$

$$\pi r^2$$

$$F(x) = \cos(x) + \lg(x)$$

VARIABLE, objeto de un programa cuyo valor puede “variар”, son celdas de memoria donde se almacenan datos.

Tienen dos atributos o características asociadas

- Identificador o nombre
- Tipo (del valor que almacena)

CONSTANTE, objeto de un programa cuyo valor puede no “variар”.

$$\pi r^2$$

$$3x^2 + 0.5x - 3.4$$

Sección declarativa

VAR {obligatorio}

<id1> : *<tipo1>*;

<id2> , *<id3>* : *<tipo2>*;

{cuando tienen el mismo tipo}

.....

Ejemplos:

Var

Inscriptos : word;

Numero: integer;

Altura, Superficie : real;

Dia, Mes: byte;

CONST {opcional}

<identif> = *<valor>*;

.....

Ejemplos:

Const

Anio=2020;

AlturaMax= 2.20;

Cartel='Cuidado con el perro';

IVA= 21;

Principales características del lenguaje Pascal

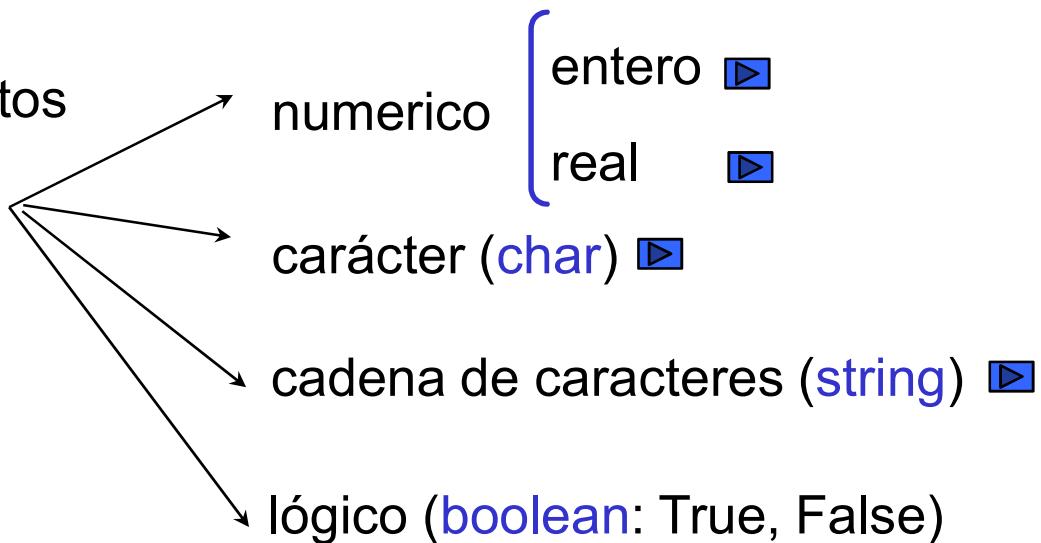
Entidades que intervienen en el programa:

- Palabras reservadas - Identificadores (Variables, Constantes)
- Tipos de datos y operadores. Expresiones.
- Acciones básicas: Lectura, Escritura, Asignación.
- Ejemplos.

Las variables

almacenan distintos

tipos de valores



- Los valores booleanos no pueden ser ingresados (lectura),
sí visualizados (escritura)

Tipo entero

Tipos	Rango numérico	Bytes (Bits)
Byte	0..255	1 (8)
Shortint	-128..127	1 (8)
Integer	-32768..32767	2 (16)
Longint	-2147483648..2147483647	4 (32)
Int64	-9223372036854775808..9223372036854775807	8 (64)
Word	0..65535	2 (16)
Longword	0..4294967295	4(32)
QWord	0 .. 18446744073709551615	8(64)



Tipo real

Tipos	Rango	Decimales significativos	Bytes (Bits)
Single	$1.5^{-45} .. 3.4^{38}$	7-8	4 (32)
Real	$2.9^{-39} .. 1.7^{38}$	15-16	8 (64)
Double	$5.0^{-324} .. 1.7^{308}$	15-16	8 (64)
Extended	$3.6^{-4951} .. 1.1^{4932}$	17-18	10 (80)



Tipo char

A continuación se presenta una parte (la mas utilizada) de la tabla ASCII DOS, cada carácter tiene internamente una representación numérica.

Se vera que el carácter ‘A’ se codifica internamente con el número 65 y el carácter blanco como el 32 .

Para visualizar un carácter debe mantener presionada la tecla **alt** mientras digita su respectivo código numérico.

alt 65 se visualiza la letra ‘A’

ASCII	0	1	2	3	4	5	6	7	8	9
30	-		blanco	i	"	#	\$	%	&	'
40	()	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	¿	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[\]	^	-	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{		}	~		ç	ü

Notar que es diferente la representación del valor entero 5 (internamente un 5 en binario) del carácter '5' (internamente un 53 en binario)

Como se ve en la tabla, existe un orden entre los caracteres y es posible compararlos en forma aislada '**A**' < '**h**'

y tambien una secuencia de caracteres (cadena) '**Papa**' > '**PAPA**'

Tipo String

String: es una secuencia de 255 caracteres como máximo, delimitada entre apóstrofos.

Es posible limitar el número de caracteres de una string especificando entre corchetes dicha cantidad, por ejemplo string [n], con n entero menor a 255.

La longitud exacta de la cadena almacenada se obtiene con la función length(...), dicho valor coincide con la posición del último carácter.

Var

Parrafo: string;

Nombre: string[10];



Sección declarativa

VAR {obligatorio}

<id1> : *<tipo1>*;

<id2> , *<id3>* : *<tipo2>*;

{cuando tienen el mismo tipo}

.....

Ejemplos:

Var

Inscriptos : word;

Numero: integer;

Altura, Superficie : real;

Dia, Mes: byte;

CONST {opcional}

<identif> = *<valor>*;

.....

Ejemplos:

Const

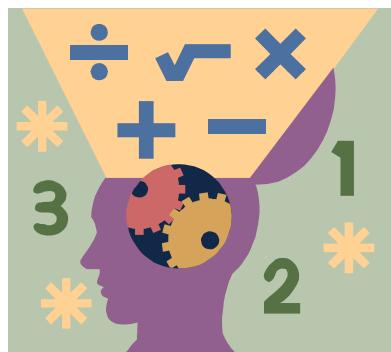
Anio=2020;

AlturaMax= 2.20;

Cartel='Cuidado con el perro';

IVA= 21;

Cada tipo de dato, tiene operadores asociados:



*Sólo para números enteros
MOD resto entero
DIV cociente entero*

Orden de precedencia : aritméticos y alfanumérico, lógicos, relacionales

Dicho orden se altera utilizando paréntesis

Si en una expresión el orden de precedencia de los operadores es el mismo, se resuelve de izquierda a derecha.

Los operandos son enteros o reales.

Considerando las siguientes declaraciones

C, D, X, Z, Y : **integer**;

A, B, E : **real**;



Son correctas

$$A * B / (6 - C)$$

$$D \bmod 5$$

$$Z + 50 \bmod 2$$

$$Y * Y - E$$

$$(Z + 50) \bmod 2$$



Son incorrectas

$$A X^2 + B X + C$$

$$B \bmod 3$$

$$C / 3 \bmod 5$$



Los operandos son carácter o cadena.
Considerando las siguientes declaraciones

Car : char;

Cad : string;

Son correctas

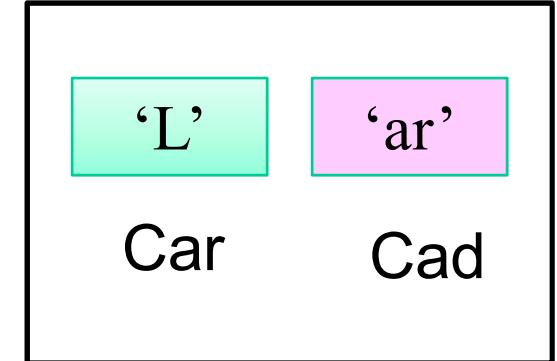
Car + 'A'

'79' + '68'

Cad + 'ena'

Car + 68

79 + '68'



Son incorrectas



Los operandos son de cualquier tipo.

La comparación es entre valores del mismo tipo.

El resultado es lógico (boolean : TRUE , FALSE)

Considerando las siguientes declaraciones

Car : **char**;

Cad : **string**;

Z, Y : **integer**;

E : **real**;

‘L’	‘Par’	3	2	3.5
Car	Cad	Z	Y	E

Son correctas

‘Papa’ > Cad

2 <= E

Z + Y * 5 <> 0

Car = 5

Z <> ‘Z’

Son incorrectas





Los operandos son de tipo lógico (True False).

Considerando las siguientes declaraciones

Logico : boolean;

Cad : string;

Z, Y : integer;

False	‘Par’	3	2
Logico	Cad	Z	Y

Son correctas

$(2 \leq Y) \text{ AND } (\text{Cad} > \text{'Pala'})$

Logico **OR** (Z \neq Y)

Son incorrectas

Z **AND** Logico

NOT Cad



Principales características del lenguaje Pascal

Entidades que intervienen en el programa:

- Identificadores (Variables, Constantes), Palabras reservadas.
- Tipos de datos y operadores. Expresiones.
- **Acciones básicas: Lectura, Escritura, Asignación.**
- Ejemplos.

variables

Operadores

tipos

¿cómo dar valor a una variable?

lectura

asignación

¡cómo visualizar el valor de una variable!

escritura

Lectura

separadas por coma

Readln (lista de variables)

Separados por blanco o comas



2.5 0.9 12

Los valores ingresan por teclado y se almacenan en las variables de la lista según el orden de ingreso

A,B :real;

C :integer;

Readln(A, B, C)

A \leftarrow 2.5
B \leftarrow 0.9
C \leftarrow 12

Importante: Las variables booleanas no se leen



Asignación

Es una operación que permite almacenar un valor en una variable

Variable := <expresión>

la **Variable** de la izquierda recibe el resultado de la **expresión** (constante, variable)

A:= 3;

B:= A * 10;

C:= C + B;

A	B	C
5	-1	2
3		
	30	
		32

El tipo de la expresión debe ser igual al de la variable, salvo que la variable es real y la expresión es entera

real ← entero



Escritura

variables y/o constantes
y/o expresiones
separadas por coma

`Write(lista de salida)`

`Writeln(lista de salida)`

escribe los elementos de la lista y queda posicionado en la misma línea o en la siguiente

`Write (A, B, C)`  2.51 0.9 12 _

`Writeln(A, B, C)`  2.51 0.9 12
_

`Write('A=', A, ' B=', B, ' Suma=', A+B)`  A=2.51 B=0.9 Suma=3.4 _

La orden Write(A, B, C) produciría la siguiente salida

Write (A, B, C)  0.251000E01 0.900000E00 12

los reales se visualizan
con notación científica

Para una interpretación más clara se utiliza escritura con formato

Para los reales se debe especificar también, la cantidad de *decimales* deseada .

Writeln(elemento real : Ancho: Decimales);

Cantidad de columnas, abarca signo,
partes entera, punto y decimales

	1	2	3	4	5	6	7	8	9	10
Write(A: 8:3);				2	.	5	1	0		
Write(A:3:1) ;	2	.	5							
Write(C:10) ;									1	2

ERRORES de PROGRAMA

Sintácticos provienen de no respetar la sintaxis del lenguaje, por ejemplo

A: int ;

Tipo de dato no existe

Writeln (A, es el resultado) ;

Un comentario
va entre ''

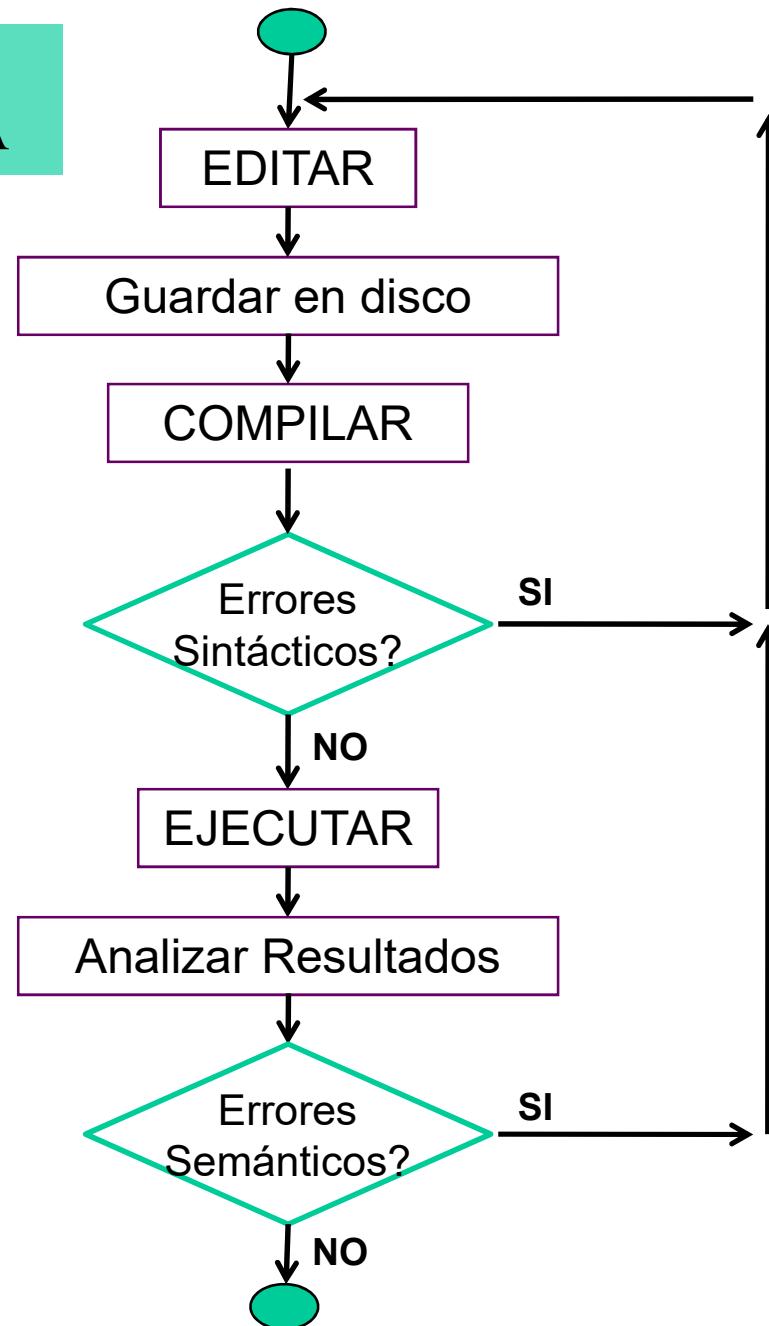
Semánticos provienen de la lógica de la solución, por ejemplo si se desea promediar tres valores

prom := A + B + C / 3

Faltan ()



CICLO DE VIDA DE UN PROGRAMA



PROGRAM Identificador → Encabezamiento

USES

Identificador/es

LABEL

Declaración de etiquetas

CONST

Definición de constantes

TYPE

Declaración de tipos

VAR

Declaración de variables

PROCEDURE

Declaración de procedimientos

FUNCTION

Declaración de funciones

BEGIN

Sentencias ejecutables

END.

Tipo de
Datos

Bloque

Partes de un programa

Program Uno;

Const

 Factor = 2;

Var

 Largo, Ancho, Per, Sup : real;

Begin

 Writeln('Ingrese largo y ancho de un rectángulo');

 Readln (Largo, Ancho);

 Per:= Factor*(Largo + Ancho);

 Sup:= Largo * Ancho;

 Writeln('El perímetro es ', Per :8:2, 'y la superficie es ',Sup:8:2);

End.



¿Qué proceso realiza?

¿Con que declaración comienza un programa?

¿Cómo y donde se declaran las variables?

¿Con que carácter se indica que termina una instrucción?

¿Cómo se indica el comienzo y el fin de la sección ejecutable

Program Dos; // Obtiene la distancia entre dos puntos

Var

X1 , Y1 , X2 , Y2 : real;

Begin

{ingreso de datos}

Comentarios para aclaraciones del programador, son ignorados por el procesador

Writeln('Ingrese coordenadas del primer punto'); Readln (X1, Y1);

Writeln('Ingrese coordenadas del segundo punto'); Readln (X2, Y2);

(* impresión del resultado*)

Writeln('La distancia entre ambos puntos es 'sqrt(sqr(X2 - X1) + sqr(Y2 - Y1)));

End.

(X1, Y1)

(X2, Y2)

Confucio (551-478 a. C.)

Lo escuche y lo olvidé;
lo leí y lo entendí;
lo hice y lo aprendí