



UNIVERSIDAD
SURCOLOMBIANA

LENGUAJES DE PROGRAMACIÓN.

MILIA ESTEFANY MONTOYA SERRATO.

CÓDIGO: U20201189533.



UNIVERSIDAD
SURCOLOMBIANA

¿QUE ES?

Es un conjunto de símbolos y reglas sintáctica que definen su estructura, es utilizado para controlar el comportamiento físico y lógico de una máquina , los lenguajes de programación facilitan la tarea de programar, ya que disponen de formas adecuadas que permiten ser leídas y escritas por personas.



CARACTERÍSTICAS.

Para utilizar un lenguaje de programación, de manera efectiva, debemos estudiarlo y comprenderlo desde tres perspectivas:

- ❖ **Sintaxis:** el conjunto de símbolos y reglas para formar sentencias.
- ❖ **Semántica:** las reglas para transformar sentencias en instrucciones lógicas.
- ❖ **Pragmática:** utilizando las construcciones particulares del lenguaje.

En español, las letras forman palabras que forman oraciones. En los lenguajes de programación, los caracteres forman sentencias que en conjunto forman instrucciones.



ELEMENTOS ENSENCIALES PARA TODO LENGUAJE DE PROGRAMACIÓN.

- ❖ Operadores.
- ❖ Variables.
- ❖ Condicionantes.
- ❖ Bucles.
- ❖ Funciones.
- ❖ Sintaxis.



TIPOS DE LENGUAJES DE PROGRAMACIÓN.

Podríamos decir que lenguaje de programación es el conjunto de símbolos y palabras que permiten al usuario de una computadora darle instrucciones órdenes para que la computadora los ejecute.

Los tipos de lenguajes de programación se dividen en:

- ❖ Lenguaje de Bajo Nivel.
- ❖ Lenguaje de Alto Nivel.



LENGUAJE DE BAJO NIVEL.

Assembly (en español, lenguaje ensamblador)

```

;nasm 2.11.08
section .data
    hello:    db 'Hola Mundo!',10    ; 'Hello world!' plus a linefeed character
    helloLen: equ $-hello           ; Length of the 'Hello world!' string
section .text
    global _start
_start:
    mov eax,4    ; The system call for write (sys_write)
    mov ebx,1    ; File descriptor 1 - standard output
    mov ecx,hello ; Put the offset of hello in ecx
    mov edx,helloLen ; helloLen is a constant, so we don't need to say
                    ; mov edx,[helloLen] to get it's actual value

    int 80h     ; Call the kernel

    mov eax,1    ; The system call for exit (sys_exit)
    mov ebx,0    ; Exit with return code of 0 (no error)
    int 80h;

```

Lenguaje de Máquina

Ejemplo:

Código de operación	Dirección	Significado
00010101	10000001	(a) Cargar contenido de la dir. 129 en Acumulador
00010111	10000010	(b) Sumar contenido de la dir 130 al Acumulador
00010110	10000011	(c) Almacenar contenido del Acumulador en la dir. 131

CÓDIGO BINARIO

A	01000001	N	01001110	a	01100001	n	01101110
B	01000010	O	01001111	b	01100010	o	01101111
C	01000011	P	01010000	c	01100011	p	01110000
D	01000100	Q	01010001	d	01100100	q	01110001
E	01000101	R	01010010	e	01100101	r	01110010
F	01000110	S	01010011	f	01100110	s	01110011
G	01000111	T	01010100	g	01100111	t	01110100
H	01001000	U	01010101	h	01101000	u	01110101
I	01001001	V	01010110	i	01101001	v	01110110
J	01001010	W	01010111	j	01101010	w	01110111
K	01001011	X	01011000	k	01101011	x	01111000
L	01001100	Y	01011001	l	01101100	y	01111001
M	01001101	Z	01011010	m	01101101	z	01111010

LENGUAJE DE ALTO NIVEL.

Este tipo de lenguajes de programación son independientes de la máquina, los podemos usar en cualquier computador con muy pocas modificaciones o sin ellas, son muy similares al lenguaje humano, pero precisan de un programa interprete o compilador que traduzca este lenguaje de programación de alto nivel a uno de bajo nivel como el lenguaje de máquina que la computadora pueda entender.

Listado de un programa en BASIC

```
3LOAD TWO SQUARES
3LIST

30 HOME : HGR
35 PRINT CHR$(4);"BLOAD SHAPE 1,A#0300"
36 POKE 232,0: POKE 233,03
37 POKE - 16302,0
40 ST$ = "T W O S Q U A R E S "
60 A = 90:B = 5
70 HCOLOR= 7: ROT= 0
80 FOR S = 1 TO 40
90 SCALE= S
100 DRAW 2 AT A,B
101 DRAW 2 AT A + 1,B + 1
105 HPLLOT 90,5 TO 10,86
110 NEXT S
120 HCOLOR= 5
130 X = 260:Y = 75
140 ROT= 5
150 FOR S = 60 TO 1 STEP - 1
160 SCALE= S
170 DRAW 2 AT X,Y: DRAW 2 AT X + 1,Y + 1
180 NEXT S
185 PRINT CHR$(4);"BLOAD SHAPE ALPHABET,A#6000"
186 VT = 21:HT = 2
187 SCALE= 2: ROT= 0
190 GOSUB 5000
195 REM THIS ADDS LINES TO FRAME 1
200 FOR S = 200 TO 175 STEP - 4
```

Beginner's
All-Purpose
Symbolic
Instruction
Code

LENGUAJES DE PROGRAMACIÓN IMPERATIVOS

Lenguaje cobol.

```

EDIT      MTH.COBOL.SRCLIB (EVALALSO) - 01.00
Command ==>
=COLS>  ----+-----1-----+-----2-----+-----3-----+-----4-----+-----
*****  ***** Top of Data *****
000100      IDENTIFICATION DIVISION.
000200      PROGRAM-ID. EVALALSO.
000300      ENVIRONMENT DIVISION.
000400      DATA DIVISION.
000500      WORKING-STORAGE SECTION.
000600      01 AGE                                PIC 9(03).
000700      01 GENDER                              PIC X(01).
000800      PROCEDURE DIVISION.
000900          ACCEPT AGE.
001000          ACCEPT GENDER.
001100          EVALUATE TRUE ALSO TRUE
001200              WHEN AGE > 018 ALSO GENDER = 'M'
  
```

Lenguaje pascal.

```

File Edit Search Run Compile Debug Tools Options Window Help
ABOUT.PAS
program aboutTurboPascal;
uses crt;
BEGIN
  TextBackground(White);
  TextColor(Black);
  writeln('About Turbo Pascal (With DOSBox) Dialog Ver 1.5 Bulid 732');
  writeln(' Copyright (C) 2018-2019 Luu Nguyen Thien Hau ');
  clrscr;
  writeln('          About Turbo Pascal (With DOSbox) ');
  writeln('-----');
  writeln(' Turbo Pascal (With DOSBox) 7.3.2 ');
  writeln(' (Turbo Pascal 7.0), (DOSBox 0.74-2, Reported DOS version 5.0) ');
  writeln(' Copyright (C) 2017-2019 Luu Nguyen Thien Hau ');
  writeln(' Turbo Pascal (With DOSBox) is free and open source Under GNU GPL');
  writeln(' Website: tpwdb.weebly.com ');
  writeln('-----');
  writeln(' This program Uses, With Permissions, the folloing copyights materials
  writeln(' DOSBox version 0.74-2 ');
  writeln(' Copyright 2002-2018 DOSBox Team, Pubilished Under GNU GPL');
  writeln(' ');
  writeln(' ');
  
```

Lenguaje C.

```

C:\c\tabla-de-multiplicar-de-un-numero.exe

1. Calcular el doble de un número entero.
2. Calcular la mitad de un número entero.
3. Calcular el cuadrado de un número entero.
4. Salir.

Introduzca opción <1-4>: 3

Introduzca un número entero: 16

El cuadrado de 16 es 256

1. Calcular el doble de un número entero.
2. Calcular la mitad de un número entero.
3. Calcular el cuadrado de un número entero.
4. Salir.

Introduzca opción <1-4>: 1

Introduzca un número entero: 19

El doble de 19 es 38

1. Calcular el doble de un número entero.
2. Calcular la mitad de un número entero.
3. Calcular el cuadrado de un número entero.
4. Salir.

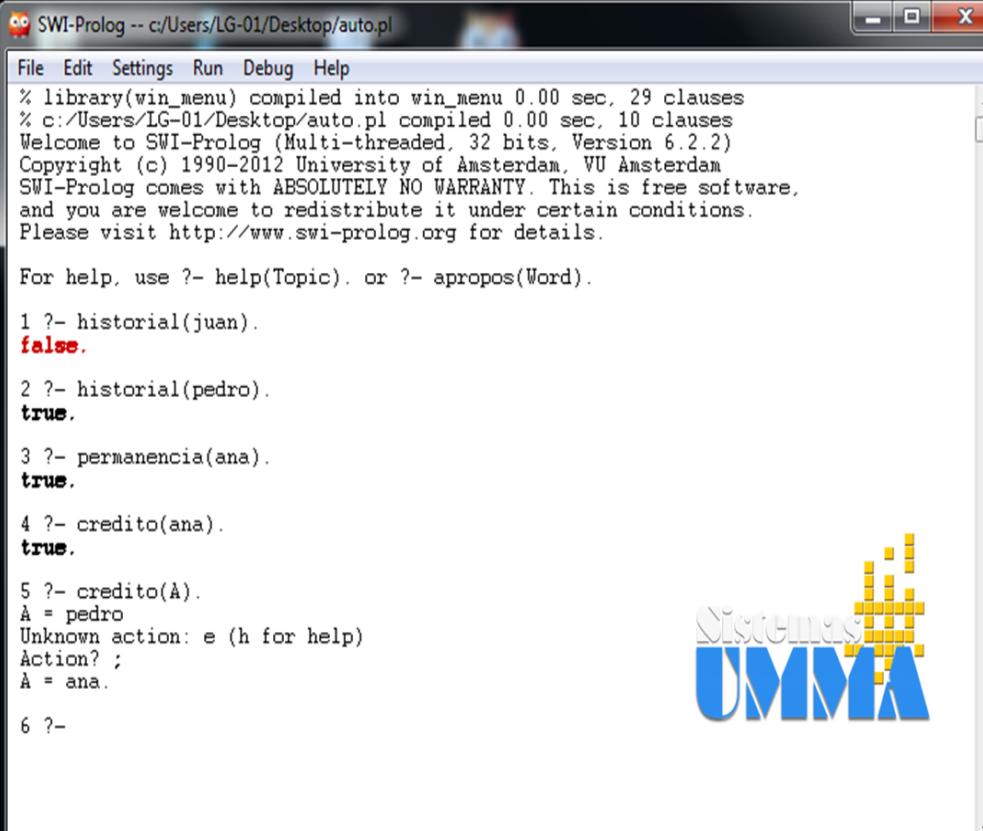
Introduzca opción <1-4>: 4
  
```


LENGUAJES DE PROGRAMACIÓN DECLARATIVOS.

Lenguaje lisp.

```
1 %lisp
2 >; Guardar valores como una lista de caracteres
3 >(define (SumarSiguiente V)
4   (cond ((null V) (progn (print "Suma=") 0))
5         (T (+ ( SumarSiguiente (cdr V)) (car V) ) ) )
6 SUMARSIGUIENTE
7 >; Crear vector de valores de entrada
8 (defun ObtenerEntrada(f c)
9   (cond ((eq c 0) nil)
10         (T (cons (read f) (ObtenerEntrada f (- c 1))))))
11 OBTENERENTRADA
12 >(defun Hazlo()
13   (progn
14     (setq archivoent (open "lisp.data"))
15     (setq arreglo (ObtenerEntrada archivoent (read archivoent)))
16     (print arreglo)
17     (print (SumarSiguiente arreglo)))
18 HAZLO
19 >Hazlo
20
21 (1 2 3 4)
22 "Suma="
23 10
24 10
```

Lenguaje prolog.



```
SWI-Prolog -- c:/Users/LG-01/Desktop/auto.pl
File Edit Settings Run Debug Help
% library(win_menu) compiled into win_menu 0.00 sec, 29 clauses
% c:/Users/LG-01/Desktop/auto.pl compiled 0.00 sec, 10 clauses
Welcome to SWI-Prolog (Multi-threaded, 32 bits, Version 6.2.2)
Copyright (c) 1990-2012 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?- historial(juan).
false.
2 ?- historial(pedro).
true.
3 ?- permanencia(ana).
true.
4 ?- credito(ana).
true.
5 ?- credito(A).
A = pedro
Unknown action: e (h for help)
Action? ;
A = ana.
6 ?-
```

LENGUAJES DE PROGRAMACIÓN ORIENTADOS A OBJETOS

Lenguaje C++.

```
"C:\DOCUMENTS AND SETTINGS\OWNER\DESKTOP\130 PROGRAMAS EN VISUAL C++ USANDO I... - □ x
```

```
Cantidad de billetes de un dolar: 4
Cantidad de billetes de cinco dolares: 2
Cantidad de billetes de diez dolares: 7
Cantidad de billetes de veinte dolares: 2
Cantidad de billetes de cincuenta dolares: 5
Cantidad de billetes de cien dolares: 1
Cantidad de monedas de un centavo: 154
Cantidad de monedas de cinco centavos: 44
Cantidad de monedas de diez centavos: 34
Cantidad de monedas de veinticinco: 4

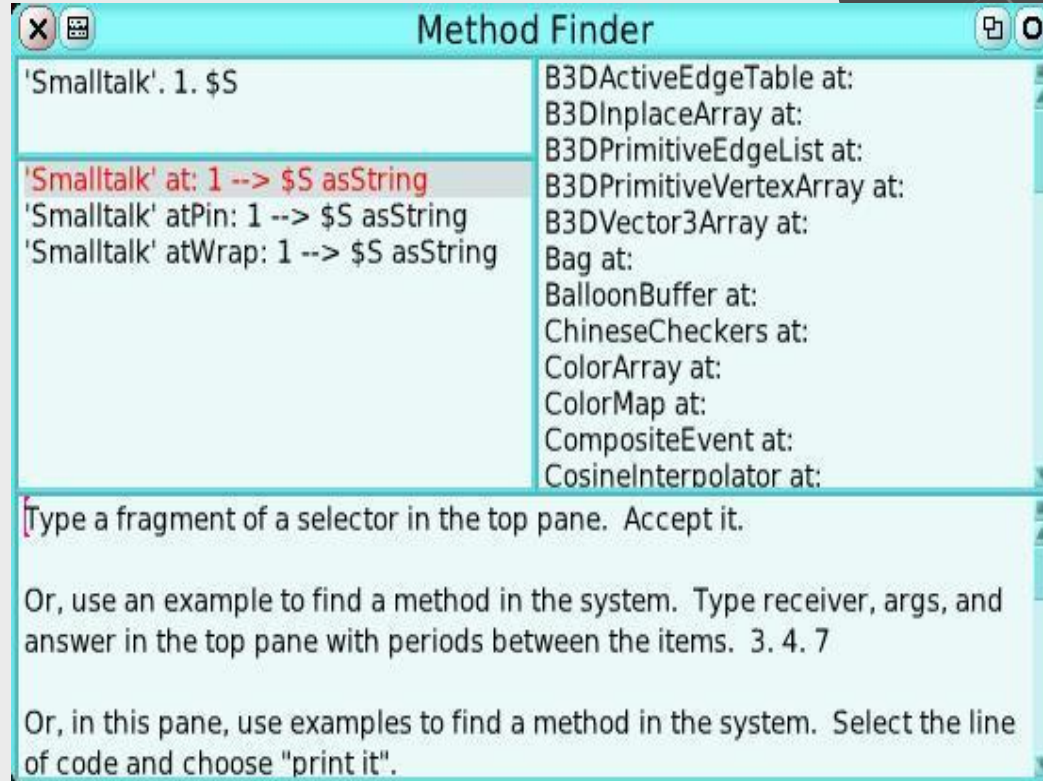
LA CANTIDAD DE DINERO QUE HAY EN BILLETES ES: $474.00
LA CANTIDAD DE DINERO QUE HAY EN MONEDAS ES: $8.14
LA CANTIDAD TOTAL DE DINERO QUE HAY ES: $482.14

MONEDA      CODIGO
Quetzal     1
Lempira     2
Euro        3

Introduzca el codigo de la moneda a la que desea convertir los $482.14: 1
$482.14 EQUIVALE A 3834.31 QUETZALES

SI DESEA SALIR PRESIONE 1 O DE LO CONTRARIO PRESIONE OTRO NUMERO: _
```

Lenguaje Smalltalk.



Method Finder

'Smalltalk'. 1. \$S

'Smalltalk' at: 1 --> \$S asString

'Smalltalk' atPin: 1 --> \$S asString

'Smalltalk' atWrap: 1 --> \$S asString

B3DActiveEdgeTable at:
B3DInplaceArray at:
B3DPrimitiveEdgeList at:
B3DPrimitiveVertexArray at:
B3DVector3Array at:
Bag at:
BalloonBuffer at:
ChineseCheckers at:
ColorArray at:
ColorMap at:
CompositeEvent at:
CosineInterpolator at:

Type a fragment of a selector in the top pane. Accept it.

Or, use an example to find a method in the system. Type receiver, args, and answer in the top pane with periods between the items. 3. 4. 7

Or, in this pane, use examples to find a method in the system. Select the line of code and choose "print it".