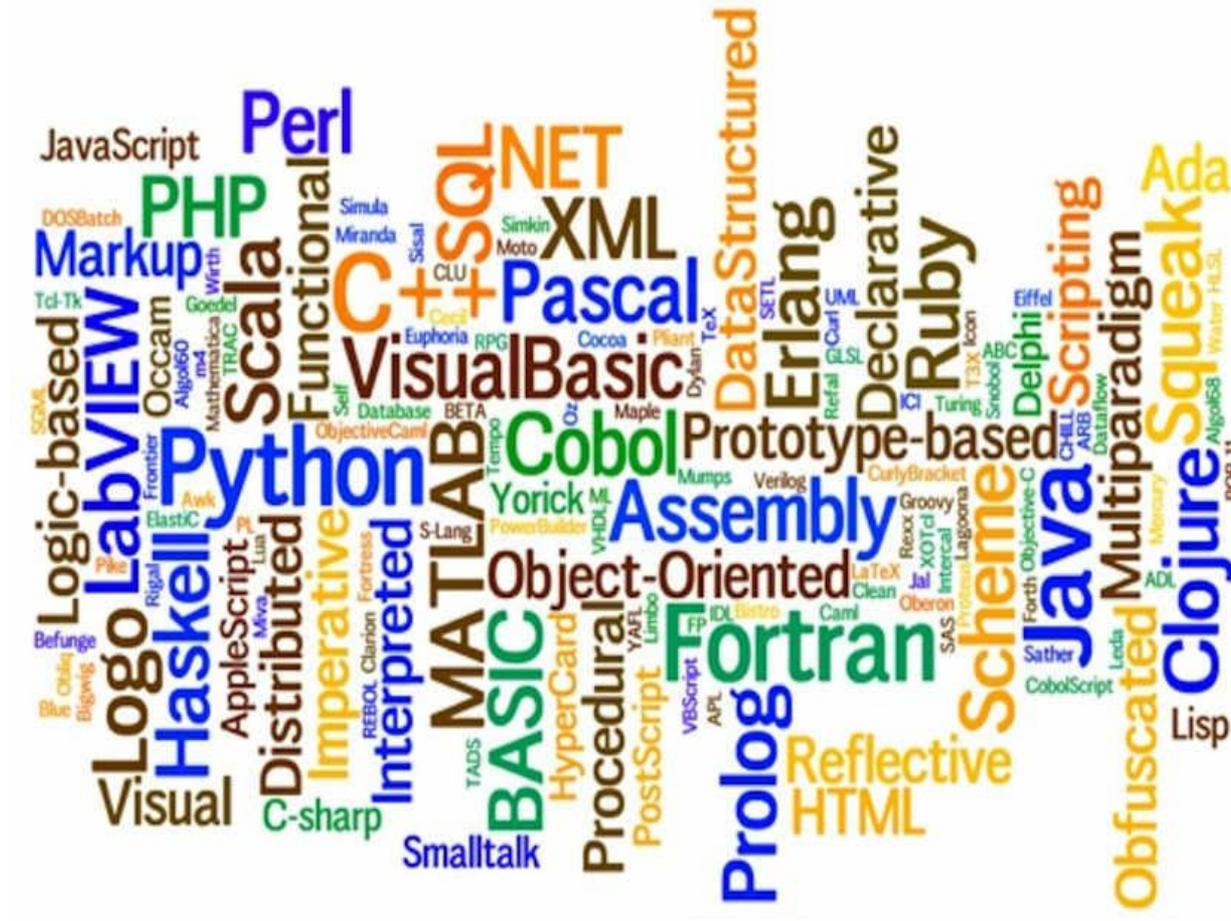


# LENGUAJES DE PROGRAMACIÓN



Nouhath Jusseth Galvis Plazas

# LENGUAJE DE PROGRAMACIÓN

Es un conjunto de símbolos y reglas sintácticas 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.

# ELEMENTOS ESENCIALES DE TODO LENGUAJE DE PROGRAMACIÓN

▶ **OPERADORES:** Son todos aquellos símbolos que representan enlaces entre cada uno de los argumentos que intervienen en una operación y se utilizan para construir expresiones.

▶ **VARIABLES:** Se clasifican como contenedores de datos y por ello se diferencian según el tipo de dato que son capaces de almacenar. En la mayoría de lenguajes de programación se requiere especificar un tipo de variable para guardar un dato concreto.

▶ **CONDICIONANTES:** Son estructuras de código que se indican para que cierta parte del programa se ejecute, deben cumplirse ciertas premisas; por ejemplo: que dos valores sean iguales, que un valor exista, que un valor sea mayor que otro... Estos condicionantes por lo general solo se ejecutan una vez a lo largo del programa.

▶ **BUCLES:** Son parientes cercanos de los condicionantes, pero ejecutan constantemente un código mientras se cumpla una determinada condición.

▶ **FUNCIONES:** Se usan para evitar que se repitan fragmentos de código. Una función podría considerarse como una variable que encierra código dentro de si.

▶ **SINTAXIS:** A la forma visible de un lenguaje de programación se le conoce como sintaxis.

# LENGUAJE DE BAJO NIVEL

Son lenguajes totalmente dependientes de la máquina, es decir que el programa que se realiza con este tipo de lenguajes no se pueden migrar o utilizar en otras máquinas.

# LENGUAJE DE MAQUINA

- ▶ Código binario (1 y 0).
- ▶ Este lenguaje es mucho más rápido que los lenguajes de alto nivel.
- ▶ Es larga y difícil para la corrección de errores.

```
11001010 00010111 11110101 00101011
00010111 11110101 00101011 00101011
11001010 00010111 11110101 00101011
00010111 11110101 00101011 00101011
11001010 11110101 00101011 00101011
11001010 11001010 11110101 00101011
11001010 11110101 00101011 00101011
11001010 00010111 11110101 00101011
00010111 11110101 00101011 00101011
11001010 11110101 00101011 00101011
```

# LENGUAJE EMSAMBLADOR

- ▶ Serie de números y letras.
- ▶ Es una traducción directa del código maquina.
- ▶ Los códigos son mas cortos y gracias a esto se necesita menos memoria.

## Raiz cuadrada:

```
org 100h
siguiente:
inc r
mov ax, r
mul ax
cmp ax, m
jbe siguiente ;(r*r) MENOR O IGUAL a m
dec r
ret

;Variables
m dw 146
r dw 0 ;Al final debe ser r=sqrt(m).
```

## Potencia:

```
name "potencia" ;8 chars DOS
org 100h ;counter to 100h
mov cx, num2
mov ax, num1
inicio:
mov bx,num1
mul bx ;ax = ax * bx
loop inicio ;c--

mov num3,ax ;copiamos el resultado
ret

;Variables "db" para byte y "dw" para word
num1 dw 0Ah
num2 dw 03h
num3 dw 0h
```

# LENGUAJE DE ALTO NIVEL

Este tipo de lenguajes de programación son independientes de la máquina, 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.

# LENGUAJES DE PROGRAMACIÓN IMPERATIVOS

- ▶ Programa mediante una serie de comandos, agrupados en bloques y compuesto de ordenes condicionales.
- ▶ Tenemos el Cobol, Pascal, C , Ada, Basic, Etc....

```
JLOAD TWO SQUARES
JLIST

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
```

**B**eginner's  
**A**ll-Purpose  
**S**ymbolic  
**I**nstruction  
**C**ode

# LENGUAJES DE PROGRAMACIÓN DECLARATIVOS

- ▶ Programación lógica y funcional. Los problemas se representan por medio de lógica matemática.
- ▶ Lenguajes de dominio específico (DSLs). Lenguajes descriptivos para un propósito específico, tales como HTML, CSS y SQL.
- ▶ Como el Lisp, el Prolog, Haskell, Etc....

```
Warning: changing directory causes all loaded modules to be unloaded,
because the search path has changed.
Prelude> :load "ejemplo2.hs"
[1 of 1] Compiling Main                ( ejemplo2.hs, interpreted )
Ok, modules loaded: Main.
*Main> mitad 5
2.5
*Main> mitad 3.1
1.55
*Main> cuadrado 2.2

<interactive>:1:10:
  No instance for (Fractional Int)
    arising from the literal `2.2'
  Possible fix: add an instance declaration for (Fractional Int)
  In the first argument of `cuadrado', namely `2.2'
  In the expression: cuadrado 2.2

  In an equation for `it': it = cuadrado 2.2
*Main> triangulo (3.2,6)
9.6
*Main>
```

```
Archivo Edición Formato Ver Ayuda
triple :: Int->Int
triple x=x*3

cuadrado :: Int->Int
cuadrado l=1+l+l+l

mitad :: Float->Float
mitad y=y/2

tercera :: Int->Int
tercera a=a^3

triangulo :: (Float,Float)->Float
triangulo (b,h) = b*h/2
```

Sistemas  
UMMA  
<http://sistemasumma.wordpress.com>

# LENGUAJES DE PROGRAMACIÓN ORIENTADOS A OBJETOS

- ▶ Fomenta la reutilización y extensión del código.
- ▶ Permite la creación de sistemas mas complejos.
- ▶ Tales como el Smalltalk el C++, Java. Etc....

```
Public Class Form1
    Dim player1 As New Persons.Players
    Dim player2 As New Persons.Players
    Private Sub Form1_Load(ByVal sender
        With player1
            .FirstName = "Wyrn"
            .LastName = "Garcia"
            .Number = 13
            .Position = "Shortstop"
        End With
        With player2
            .FirstName = "Joe"
            .LastName = "Black"
            .Number = 51
            .Position = "Catcher"
        End With
    End Sub
End Class
```

```
public class Lanzador {
    public static void main(String[] args) {
        Granja g = new Granja("granja1");

        Cerdo cerdo = new Cerdo("Cerdo Manolo");
        Vaca vaca = new Vaca("Vaca Marisa");
        g.añadirAnimal(cerdo);
        g.añadirAnimal(vaca);

        System.out.println("Vemos los animales que tenemos en la granja");
        g.muestraAnimales();

        cerdo.imprimeNombre();
        vaca.imprimeNombre();
    }
}
```

▶ <https://www.youtube.com/watch?v=DCwMcs1QGy0>

# RESUMEN

