

Laboratorio No. 2

Programación de computadoras 2
Maynor David Salguero Guillén





¿Qué es Programación?



Se refiere a la acción de crear programas o aplicaciones a través del desarrollo de un **código fuente**, que se basa en el conjunto de instrucciones que sigue el ordenador para ejecutar un programa. Pero la programación no es solo el trabajo de escritura del código, sino todo un conjunto de tareas que se deben cumplir, a fin de que el código que se escribió resulte correcto y robusto, y cumpla con el objetivo o los objetivos para los que fue creado.



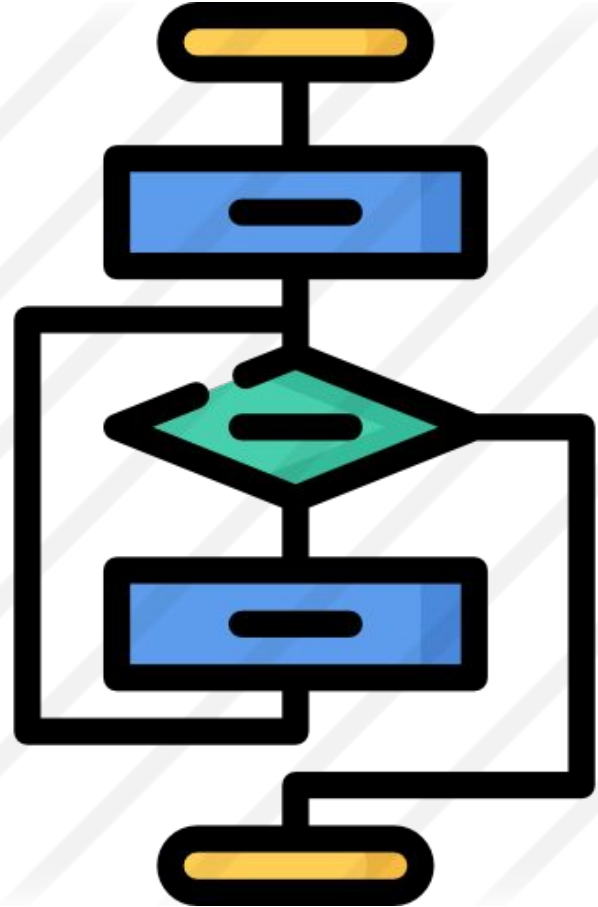
Algoritmo

Es una lista bien definida, ordenada y finita de operaciones, que permite encontrar la solución a un problema determinado



Un algoritmo se puede expresar como:

- Lenguaje Natural
- Pseudocódigo
- Diagramas de Flujo
- Programas





¿Cómo se implementa un algoritmo?

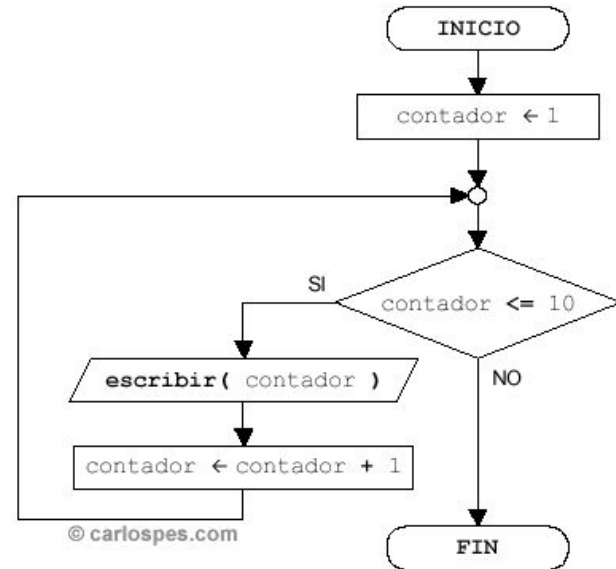
Descripción de
alto nivel

Descripción
Formal

Implementación

Ejemplo: Imprimir 1 al 10

Imprimir un número empezando en 1, luego imprimir el 2 y así sucesivamente hasta que se imprima el número 10.



```
Module Program
    Oreferencias
    Sub Main(args As String())

        Dim contador As Integer = 1
        While contador <= 10
            Console.WriteLine(contador)
            contador += 1
        End While

    End Sub
End Module
```




Consola de depuración de Microsoft Visual Studio

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```



Historia de la programación

Lenguaje de Máquina

```
01101000 01101111
01101100 01100001
00100000 01101101
01110101 01101110
01100100 01101111
```

Ensamblador

```
1 section .text
2     global _start
3
4     _start:
5         mov     edx,len
6         mov     ecx,msg
7         mov     ebx,1
8         mov     eax,4
9         int     0x80
10
11        mov     eax,1
12        int     0x80
13
14 section .data
15 msg db 'Hola, mundo!', 0xa
16 len equ $ - msg
```

Lenguaje de alto nivel

```
Sub Main(args As String())
    Console.WriteLine("Hola Mundo")
End Sub
```



Elementos de un programa

Palabras reservadas: Palabras que dentro del lenguaje significan la ejecución de una instrucción determinada, por lo que no pueden ser utilizadas con otro fin.

Operadores: Símbolos que indican la aplicación de operaciones lógicas o matemáticas.

Variables: Datos que pueden variar durante la ejecución del programa.

Constantes: Datos que no varían durante la ejecución del programa.

Identificadores: Nombre que se le da a las diferentes variables para identificarlas.



VARIABLES

TIPO DE DATOS	MEMORIA REQUERIDA	RANGO DE VALORES	EQUIVALENCIA CON USADAS EN PSEUDOCÓDIGO	OBSERVACIONES
Integer (Entero)	2 bytes ó 4 bytes según versión	- 32768 a 32767 ó -2.147.483.648 a 2.147.483.647 según versión	Entero	Uso en contadores, control de bucles etc.
Long (Entero largo)	4 bytes ó 8 bytes según versión	- 2147483648 a 2147483647 ó -9,2E+18 a 9,2E+18 según versión	Entero	Igual que integer pero admite un rango más amplio
Single (Decimal simple)	4 bytes	- $3,4 \cdot 10^{38}$ a $3,4 \cdot 10^{38}$	Real	Hasta 6 decimales o más según versión. También admite enteros
Double (Decimal doble)	8 bytes	- $1,79 \cdot 10^{308}$ a $1,79 \cdot 10^{308}$	Real	Hasta 14 decimales o más según versión. También admite enteros
Boolean (Booleano)	2 bytes	True o False	Booleano	False es el valor predeterminado y va asociado a cero
String	10 bytes + 1 byte por cada carácter	0 a $2 \cdot 10^9$ caracteres	Alfanumérica	Texto de longitud variable
Variant (Object en otras versiones)	Variable	Los del tipo double para números o los del tipo string para texto	No tiene	Admite números enteros, decimales o texto



OPERADORES Y TIPOS DE OPERADORES



Operador de asignación:

El único operador disponible en esta categoría es el operador =.
Permite asignar un valor a una variable. El mismo operador se utiliza sea cual sea el tipo de la variable (numérica, cadena de caracteres...).

Aritméticos, Relacionales, Lógicos y de concatenación.

Operadores Aritméticos	Operadores de comparación	Operaciones de concatenación	Operadores lógicos
Operador ^	< (Menor que)	&	Operador And
Operador *	<= (Menor o igual que)	+	Operador Eqv
Operador /	> (Mayor que)		Operador Imp
Operador \	>= (Mayor o igual que)		Operador Not
Operador Mod	= (Igual a)		Operador Or
Operador +	<> (Distinto de)		Operador Xor
Operador -	Is		
	Like		



PARTE PRACTICA



DUDAS