FUNDAMENTOS DE PROGRAMACIÓN

INTRODUCCIÓN A LA PROGRAMACIÓN

Ing. Elba Ma. Bodero Poveda, PhD

Introducción a la Programación

Computadora

Es un aparato electrónico que recibe datos (entrada), los procesa (instrucciones denominado programa) y devuelve información (salida), también conocido como ordenador o PC (Personal Computer). En la actualidad existe una variedad de computadoras para diferentes propósitos.

Servidores

Computadora personal

Computadora portátil

PDA

Arquitectura de una computadora

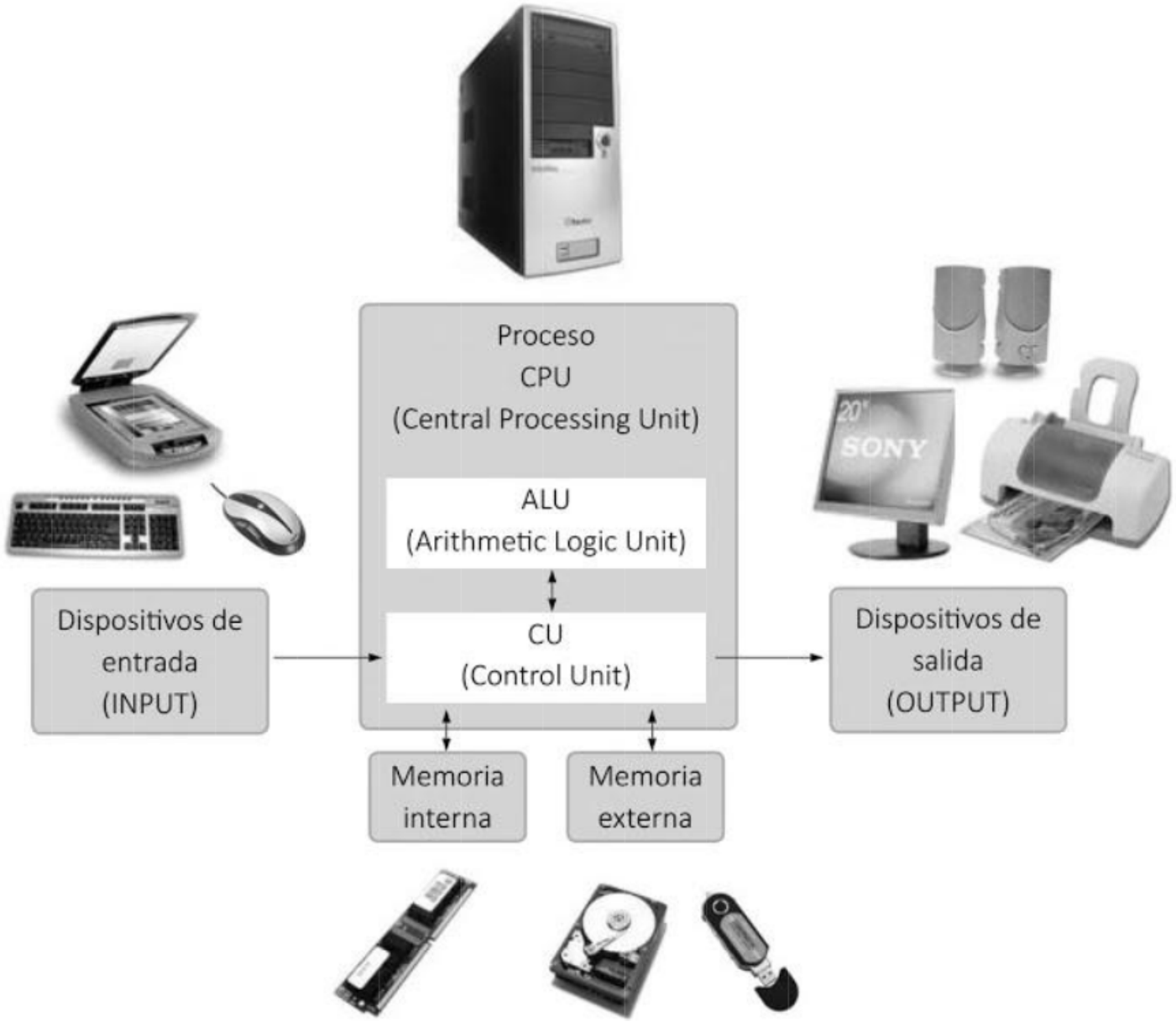
Hardware: *Hard* (duro) – *ware* (componente), representa la parte física de la computadora.

Arquitectura de una computadora

Software: *Soft* (blando) – *ware* (componente), representa la parte lógica de la computadora (los programas); estos se encuentran almacenados en los componentes físicos de la computadora, tales como memorias RAM, ROM, Discos Duros (*Hard Disk*), entre otros.

Programas internos tales como:

- BIOS
- CMOS
- SetUp


Proceso
CPU
(Central Processing Unit)
ALU
(Arithmetic Logic Unit)
CU
(Control Unit)
Dispositivos de entrada (INPUT)
Dispositivos de salida (OUTPUT)
Memoria interna
Memoria externa
SONY

Unidades de medida de almacenamiento

La memoria interna (RAM) y las memorias externas (disco duro) almacenan información. La información que se guarda y entiende la PC está en formato binario (0- 1).

BIT (BInary DigiT): El bit representa la unidad mínima de información que almacena una computadora.

BYTE: Está compuesto por 8 bit (01110011), entonces existe $2^8 = 256$ combinaciones diferentes (tabla de código ASCII).

Por lo general, la información se representa por caracteres y cada carácter (número, letra, símbolo, etc.) es un byte. Para medir la información se utilizan múltiplos de bytes.

Byte	1 B		8 bits
Kilobyte	1 KB	2^{10} bytes	1024 bytes
Megabyte	1 MB	2^{20} bytes	1024 KB
Gigabyte	1 GB	2^{30} bytes	1024 MB
Terabyte	1 TB	2^{40} bytes	1024 GB

Unidades de medida de almacenamiento

La memoria interna (RAM) y las memorias externas (disco duro) almacenan información. La información que se guarda y entiende la PC está en formato binario (0- 1).

BIT (BInary DigiT): El bit representa la unidad mínima de información que almacena una computadora.

BYTE: Está compuesto por 8 bit (01110011), entonces existe $2^8 = 256$ combinaciones diferentes (tabla de código ASCII).

Por lo general, la información se representa por caracteres y cada carácter (número, letra, símbolo, etc.) es un byte. Para medir la información se utilizan múltiplos de bytes.

Byte	1 B		8 bits
Kilobyte	1 KB	2^{10} bytes	1024 bytes
Megabyte	1 MB	2^{20} bytes	1024 KB
Gigabyte	1 GB	2^{30} bytes	1024 MB
Terabyte	1 TB	2^{40} bytes	1024 GB

Programa vs programación

Los programas o ***software*** son un conjunto de instrucciones ordenadas para ejecutarse de forma rápida y precisa en una computadora. El ***software*** se divide en dos grupos: ***software*** de **sistema operativo** y ***software* de aplicaciones**.

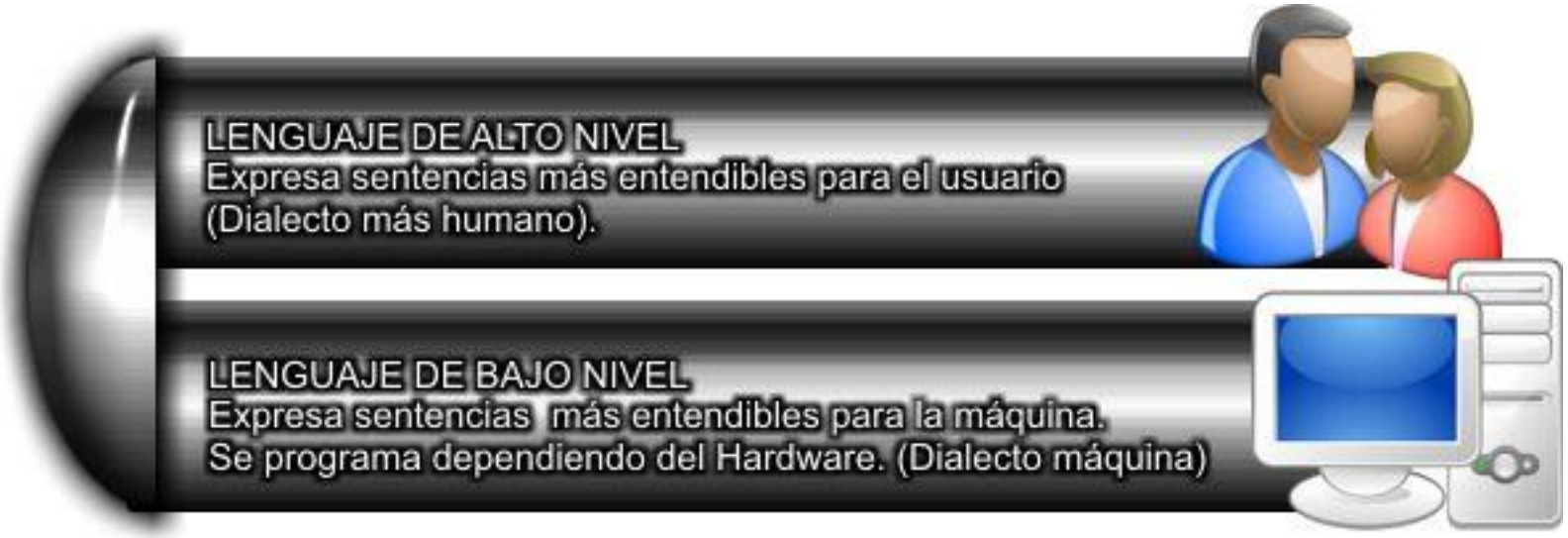
El proceso de escribir un programa se denomina **programación**, y el conjunto de instrucciones que se utilizan para escribir un programa se llama **lenguaje de programación**.

Lenguaje /s de programación

Sirve para escribir programas y permite la comunicación usuario (programador) versus máquina (PC).

Existen tres tipos de lenguajes de programación:

- **Lenguaje de máquina**
- **Lenguaje de bajo nivel (ensamblador)**
- **Lenguaje de alto nivel**

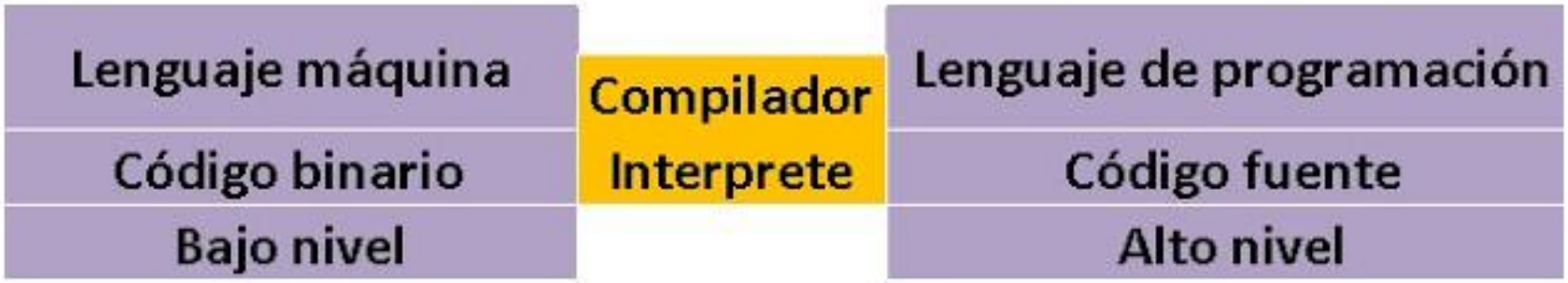


Traductores del lenguaje de programación

Son programas que traducen los **códigos fuentes** (programas escritos en un lenguaje de alto nivel) a **código máquina**.

Los traductores se dividen en:

- **Intérpretes**
- **Compiladores**



Etapas de desarrollo de un programa

REQUERIMIENTO → **ANÁLISIS** → **DISEÑO** → **IMPLEMENTACIÓN** → **PRUEBAS** → **DESPLIEGUE**

Etapas de desarrollo de un programa

- **Requerimiento:** Enunciado del problema a resolver.
- **Análisis:** ¿Qué? (Entender el problema – entrada – proceso – salida).
- **Diseño:** ¿Cómo? (Resolver el problema – algoritmo – diagrama de flujo – diseño de interfaz de usuario).
- **Implementación:** ¿Hacerlo? (Codificación / Programar).
- **Pruebas:** ¿Funciona? (Verificar / Comprobar).
- **Despliegue:** ¿Instalar? (Distribuir el programa).

FUNDAMENTOS DE PROGRAMACIÓN

LÓGICA DE PROGRAMACIÓN

Ing. Elba Ma. Bodero Poveda, PhD

Algoritmo

Método que describe la solución de un problema computacional mediante una serie de pasos precisos, definidos y finitos.

- **Preciso**
- **Definido**
- **Finito**

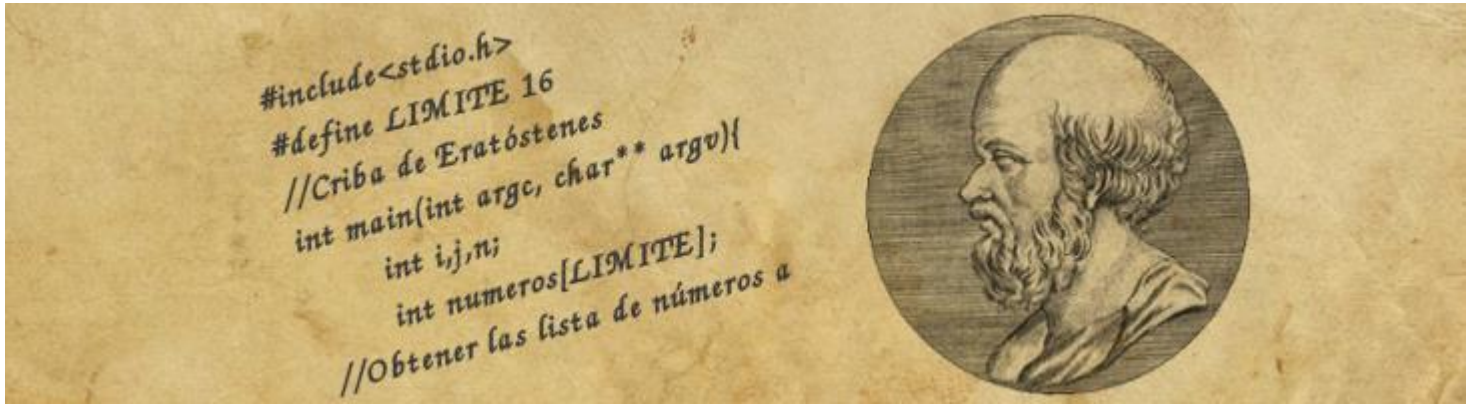
La solución de un algoritmo debe describir tres partes:

- **Entrada:** Datos que se necesitan para poder ejecutarse.
- **Proceso:** Acciones y cálculos a realizar.
- **Salida:** Resultado esperado.

Algoritmo

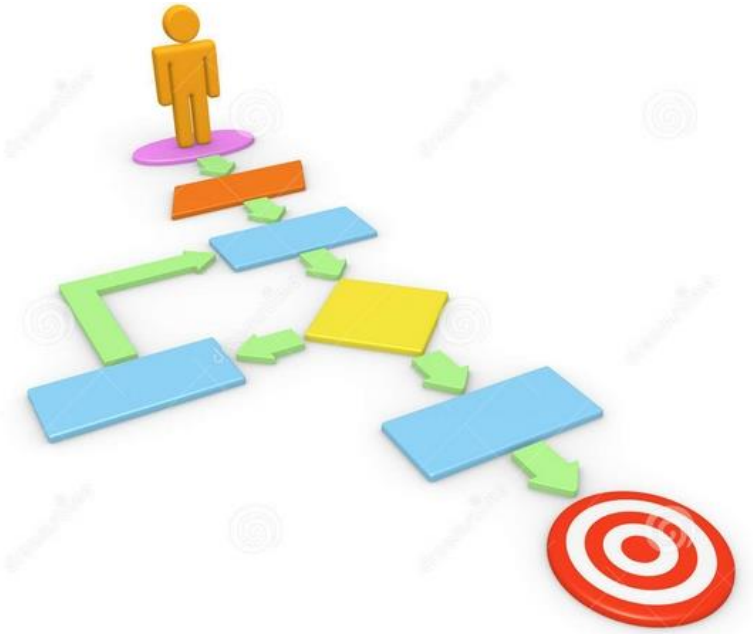
Los algoritmos son la base de la programación de ordenadores, ya que los **programas de ordenador** se pueden entender como algoritmos escritos en un código especial, entendible por un ordenador.

La desventaja del diseño de algoritmos radica en que no podemos escribir lo que deseemos; el lenguaje ha utilizar no debe dejar posibilidad de duda, debe recoger todas las posibilidades.



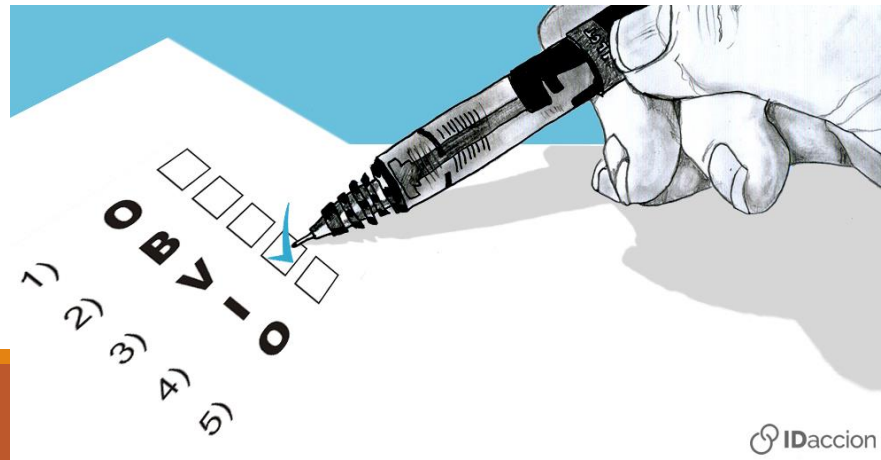
Características que deben cumplir los algoritmos obligatoriamente

- **Un algoritmo debe resolver el problema para el que fue formulado.**
- **Los algoritmos son independientes del lenguaje de programación.**
- **Los algoritmos deben ser precisos.**
- **Los algoritmos deben ser finitos.**
- **Los algoritmos deben poder repetirse.**



Características aconsejables para los algoritmos

- **Validez:** Un algoritmo es válido si carece de errores. Un algoritmo puede resolver el problema para el que se planteó y, sin embargo, no ser válido debido a que posee errores.
- **Eficiencia:** Un algoritmo es eficiente si obtiene la solución al problema en poco tiempo. No lo es si tarda en obtener el resultado.
- **Óptimo:** Un algoritmo es óptimo si es el más eficiente posible y no contiene errores. La búsqueda de este algoritmo es el objetivo prioritario del programador. No siempre podemos garantizar que el algoritmo hallado sea el óptimo, a veces sí.



Fases en la generación de algoritmos

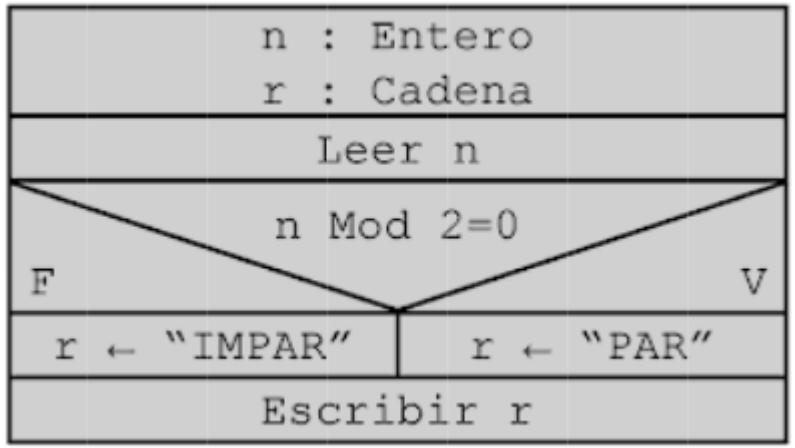
Hay tres fases en la elaboración de un algoritmo:

a. **Análisis:** En esta se determina cuál es exactamente el problema a resolver. Qué datos forman la entrada del algoritmo y cuáles deberán obtenerse como salida.
b. **Diseño:** Elaboración del algoritmo.
c. **Prueba:** Comprobación del resultado. Se observa si el algoritmo obtiene la salida esperada para todas las entradas.

Herramientas de un algoritmo: Pseudocódigo

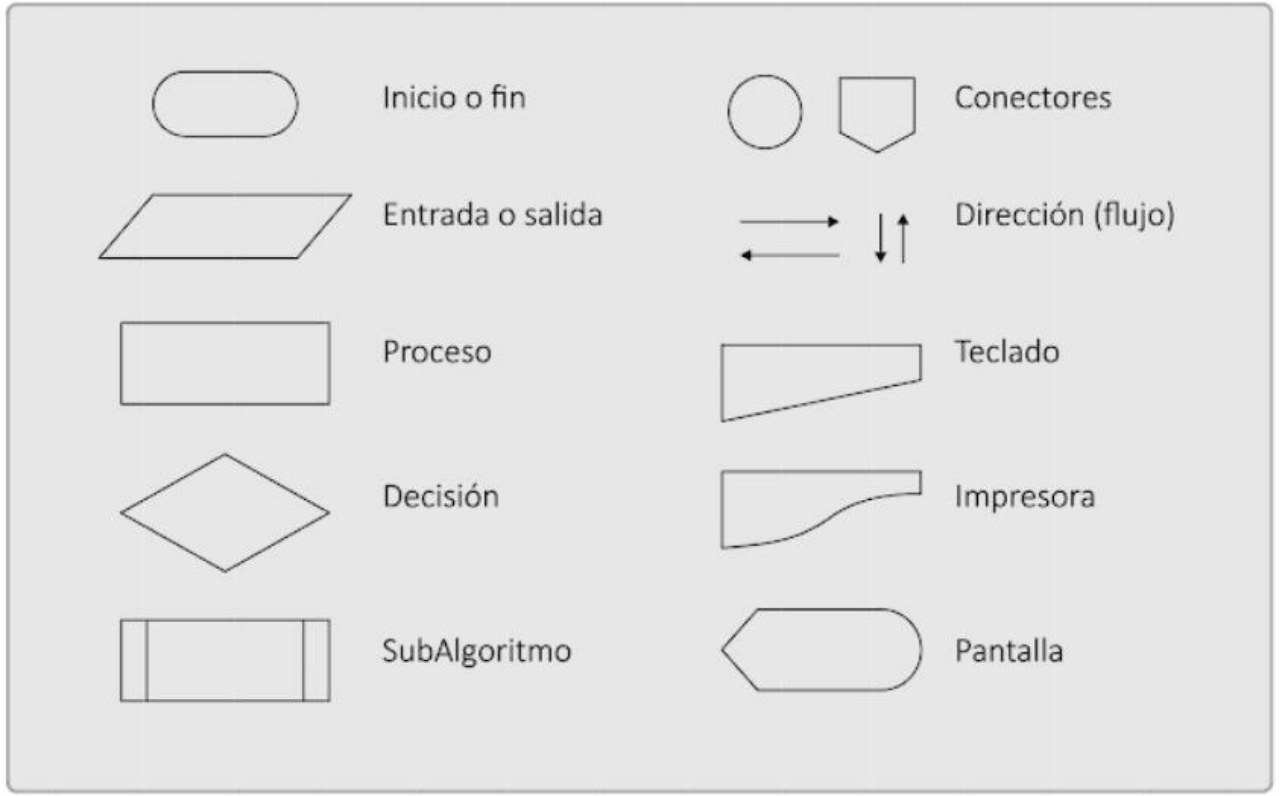
```
Inicio

   //Variables
   n : Entero
   r : Cadena

   //Entrada
   Leer n

   //Proceso
   Si n Mod 2 = 0 Entonces
       r ← "PAR"
   SiNo
       r ← "IMPAR"
   Fin Si

   //Salida
   Escribir r

Fin
```

Herramientas de un algoritmo: Nassi Scheneiderman (N-S)



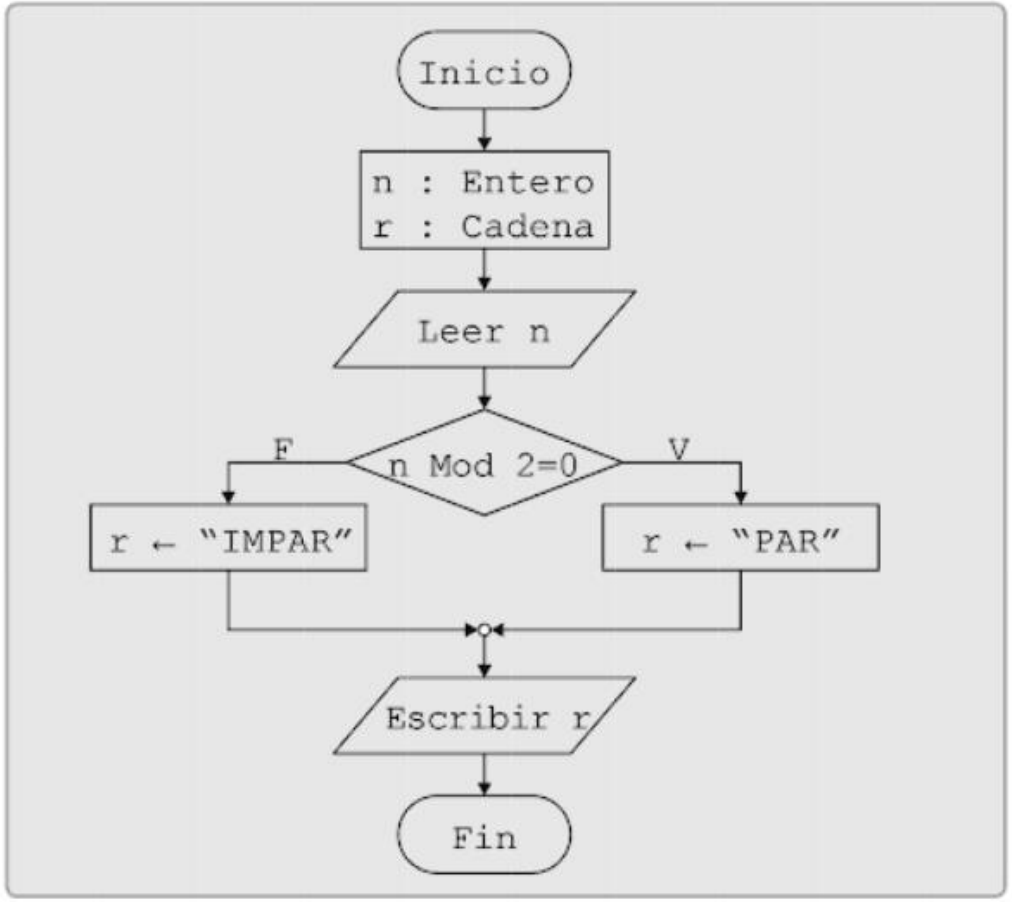
2

Herramientas de un algoritmo: Diagrama de Flujo



3

Herramientas de un algoritmo: Diagrama de Flujo



3

Instrucciones

Instrucción de inicio/ fin:

Diagrama de flujo	Pseudocódigo
	Inicio
	Fin

PROGRAMACIÓN

INSTRUCCIONES·DE·USO

Instrucciones

Instrucción de asignación:

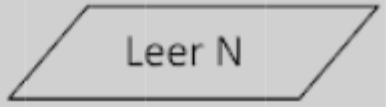
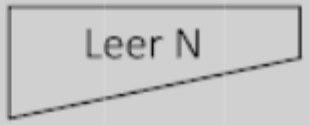
Diagrama de flujo	Pseudocódigo
N ← 10	N ← 10
N = 10	N = 10

PROGRAMACIÓN

INSTRUCCIONES·DE·USO

Instrucciones

Instrucción de lectura:

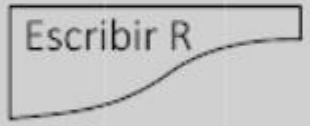
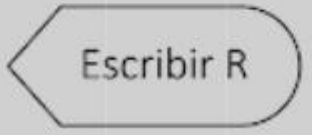
Diagrama de flujo	Pseudocódigo
	Leer N
	Leer N

PROGRAMACIÓN

INSTRUCCIONES·DE·USO

Instrucciones

Instrucción de escritura:

Diagrama de flujo	Pseudocódigo
	Escribir R
	Escribir R
	Escribir R

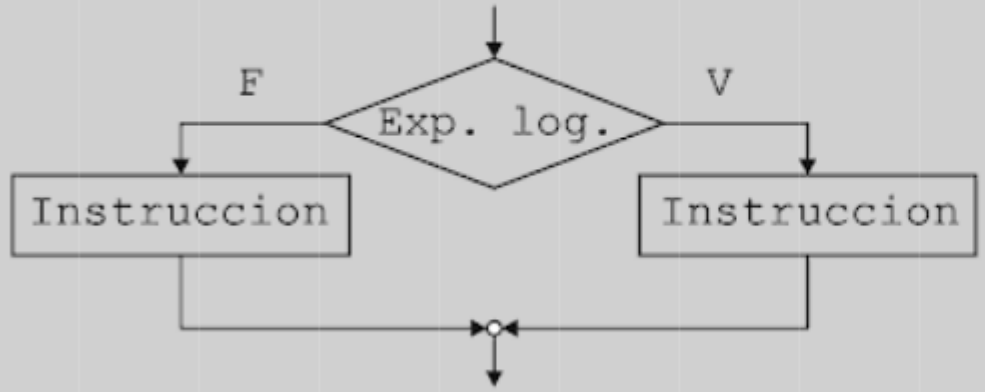
PROGRAMACIÓN

INSTRUCCIONES·DE·USO

Instrucciones

Instrucción de bifurcación:

Diagrama de flujo



Pseudocódigo

Si <Exp. Log> **Entonces**

<Instrucción>

SiNo

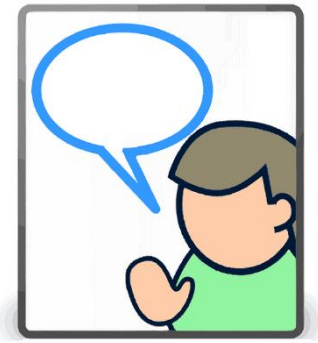
<Instrucción>

Fin Si

PROGRAMACIÓN

INSTRUCCIONES·DE·USO

Comentarios



Permiten describir y explicar, además sirve como ayuda para recordar y entender las operaciones que se van a ejecutar.

```
//Variables
N : Entero
```

```
'Variables
Dim N As Integer
```

Palabras Reservadas

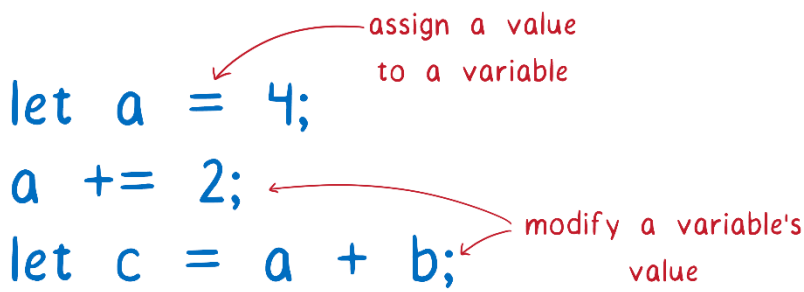
```
Dim, Integer, Long, Single, Double, If, For, Select, Me ...
```

Identificadores

```
// Identificador de Variable es Dim
// y esta es palabra reservada
Dim Dim As Integer
```

1. Deben comenzar por una letra.
2. No debe coincidir con palabras reservadas del lenguaje de programación que está utilizando.

Variables




```
//Variables
N : Entero
```

```
//Asignar un valor
N ← 10
//Cambiar su valor
N ←50
```

```
'Variables
Dim N As Integer
```

```
'Asignar un valor
N = 10
'Cambiar su valor
N = 50
```

$$2^k + k - 1$$

Constantes

```
//Constantes
PI ← 3.14159 : Real
//Error ya no puede modificarlo
PI ← 3.14
```

```
'Constantes
Const PI As Integer = 3.14159
'Error ya no puede modificarlo
PI = 3.14
```

Tipo de datos simples: Entero


```
//Crear la variable
//(identificador y tipo de dato)
N : Entero

//Asignar un valor
//(identificador, operador de asignación y valor)
N ← 15
```

```
'Entero corto
Dim N As Integer
'Asignar un valor (error de desbordamiento)
'Sobrepaso su limite (rango)
```

```
N = 45000
'Entero largo
Dim N As Long
'Asignar un valor
N = 4500099
```

Tipo de datos simples: Real


```
//Crear la variable
//(identificador y tipo de dato)
N : Real
//Asignar un valor
//(identificador, operador de asignación y valor)
N ← 15.75
```

Tipo de datos simples: Real


```
'Precisión simple
Dim N As Single
'Se redondea a 15.123457
N = 15.12345678
'Precisión doble
Dim N As Double
'Lo almacena sin redondear 15.12345678
N = 15.12345678
```

Tipo de datos simples: Caracter

```
//Crear la variable
R : Caracter
//Asignar un valor
R ← 'A'
R ← '9'
R ← '*'
```

```
'Crear la variable
Dim R As String
'Asignar un valor
R = "A"
R = "9"
R = "*"
```

A	B	$c.11$
V	V	F
V	F	V
F	V	F
F	F	V

Tipo de datos simples: Lógico

```
//Crear la variable
L : Logico
//Asignar un valor
L ← VERDADERO
L ← FALSO
```

```
'Crear la variable
Dim L As Boolean
'Asignar un valor
L = True
L = False
```

Tipo de datos complejos: Cadena

```
//Crear la variable
R : Cadena
//Asignar un valor
R ← "ricardomarcelo@hotmail.com"
```

Operadores y expresiones: Aritméticos

Operador	Descripción
+	Suma
-	Resta
*	Multiplicación
/	División
\	División entera
^	Exponenciación
Mod	Módulo (resto de una división)

Operadores y expresiones: Aritméticos

Expresiones aritméticas

8 x 3	Equivale a	8*3 = 24
$8 \div 3$ o $\frac{8}{3}$	Equivale a	8/3 = 2.666666 8\3 = 2
8^2	Equivale a	8^2 = 64
$\sqrt{9}$	Equivale a	9^(1/2) = 3
$9 \mid 4$ $\underline{\quad}$ $\textcircled{1} \quad 2$	Equivale a	9 Mod 4 = 1

Operadores y expresiones: Relacionales

Operador	Descripción
=	Igualdad
>	Mayor que
>=	Menor o igual que
<	Menor que
<=	Menor o Igual que
<>	Diferente a

Operadores y expresiones: Relacionales

Expresiones lógicas (condiciones)

8 = 3	Falso
8 > 3	Verdadero
8 <= 3	Verdadero
8 <> 8	Falso

Operadores y expresiones: Lógicos

Operador	Descripción
Y	Y lógico
O	O lógico
No	No lógico

Operadores y expresiones: Lógicos

«Y» lógico: Si p y q son valores lógicos, ambos deben ser verdaderos para que **Y** devuelva verdadero.

Expresiones lógicas (condiciones)

8 > 4	Y	3 = 6	Falso
7 <> 5	Y	5>=4	Verdadero

Operadores y expresiones: Lógicos

«O» lógico: Si p y q son valores lógicos, uno de ellos debe ser verdadero para que **O** devuelva verdadero.

Expresiones lógicas (condiciones)

8 > 4	O	3 = 6	Verdadero
7 <> 5	Y	5>=4	Verdadero

«No» lógico: Si p es un valor lógico, el operador **No** invierte su valor.

Expresiones lógicas (condiciones)

NO (8 > 4)	Falso
NO (7 <> 7)	Verdadero

Operadores y expresiones: Cadena

Operador	Descripción
+	Unir cadenas
&	Unir cadenas

Expresiones de cadena

“Ricardo” + “ “ + “Marcelo”	Ricardo Marcelo
“ricardomarcelo” & “@” & “hotmail.com”	ricardomarcelo@hotmail.com

Control de Flujo

- Estructura secuencial
- Estructura selectiva simple y doble
- Estructura selectiva múltiple
- Estructura repetitiva mientras
- Estructura repetitiva para