



# REPRESENTACIÓN DE ALGORITMOS

**Ing. Moisés Álvarez Huamán**

# ALGORITMO

- ⦿ Un **Algoritmo** constituye una lista completa de los pasos y una descripción de los datos que son necesarios para resolver un determinado problema en el ámbito de un método.
  - ⦿ Un algoritmo se desarrolla para un determinado método y que su definición tiene dos partes esenciales:
    - > Una lista de pasos que deben ser ejecutados
    - > Una descripción de los datos que son manipulados por estos pasos
-

# CARACTERÍSTICAS

- Descripción de los pasos a ser ejecutados.
- Descripción de los datos que son ejecutados.
- Un algoritmo debe ser preciso, indicando el orden de realización de cada paso.
- Todo algoritmo debe ser finito.
- Un algoritmo debe estar definido.
- Un algoritmo puede o no tener datos de entrada.
- Un algoritmo producirá uno o más datos de salida.
- Los datos de entrada y salida deben almacenarse en una estructura de datos.
- El resultado que se obtenga debe satisfacer a los requerimientos de la persona interesado.
- Debe ser estructurado. Es decir, debe ser fácil de leer, entender, usar y cambiar si es preciso.

# HERRAMIENTA PARA REPRESENTAR ALGORITMOS

## HERRAMIENTA: PSEUDOCÓDIGO

- Un **Pseudocódigo** permite expresar un algoritmo con palabras en castellano que son semejantes a las sentencias de un lenguaje de programación.

```
1  Proceso CuentaParImpar
2      Definir N,NP,NI Como Entero;
3      N<-1;
4      NP<-0;
5      NI<-0;
6      Mientras N>0 Hacer
7          Escribir "Ingrese un numero";
8          Leer N;
9          SI N%2=0 Entonces
10             NP<-NP+1;
11         Sino
12             NI<-NI+1;
13         FinSi
14     FinMientras
15     Escribir "NumerosPares=",NP;
16     Escribir "NumerosImpares=",NI;
17 FinProceso
```

# PAUTAS BÁSICAS:

- ⦿ Todo algoritmo debe tener un nombre, el cual deberá comenzar con una letra mayúscula.
- ⦿ Es necesario que se determine los datos de entrada y la información de salida.
- ⦿ Para asignar un valor a la variable "x" se utiliza el signo igual.

```
N<-1;
```

- ⦿ Para declarar una variable "x" se deberá determinar que tipo de dato se almacenara.

```
Definir N, NP, NI Como Entero;
```

- ⦿ Para indicar que la computadora lea un valor desde un dispositivo externo y lo almacene en la variable "Z", se utiliza:

```
Leer N;
```

- ⦿ Para indicar que la computadora escriba hacia un destino externo:

- > Para escribir un mensaje: `Escribir "Ingrese un numero";`

- > Para escribir el valor de una variable: `Escribir "NumerosPares=", NP;`

- > Para escribir el resultado de una expresión: `ESCRIBIR x+2`

# ESTRUCTURA BÁSICA DE UN PSEUDOCÓDIGO

Nombre del algoritmo

Inicio

```
Proceso CuentaParImpar
  Definir N, NP, NI Como Entero;
  N<-1;
  NP<-0;
  NI<-0;
  Mientras N>0 Hacer
    Escribir "Ingrese un numero";
    Leer N;
    SI N%2=0 Entonces
      NP<-NP+1;
    Sino
      NI<-NI+1;
    FinSi
  FinMientras
  Escribir "NumerosPares=", NP;
  Escribir "NumerosImpares=", NI;
FinProceso
```

Fin

# PROGRAMA PARA SUMAR DOS NÚMEROS

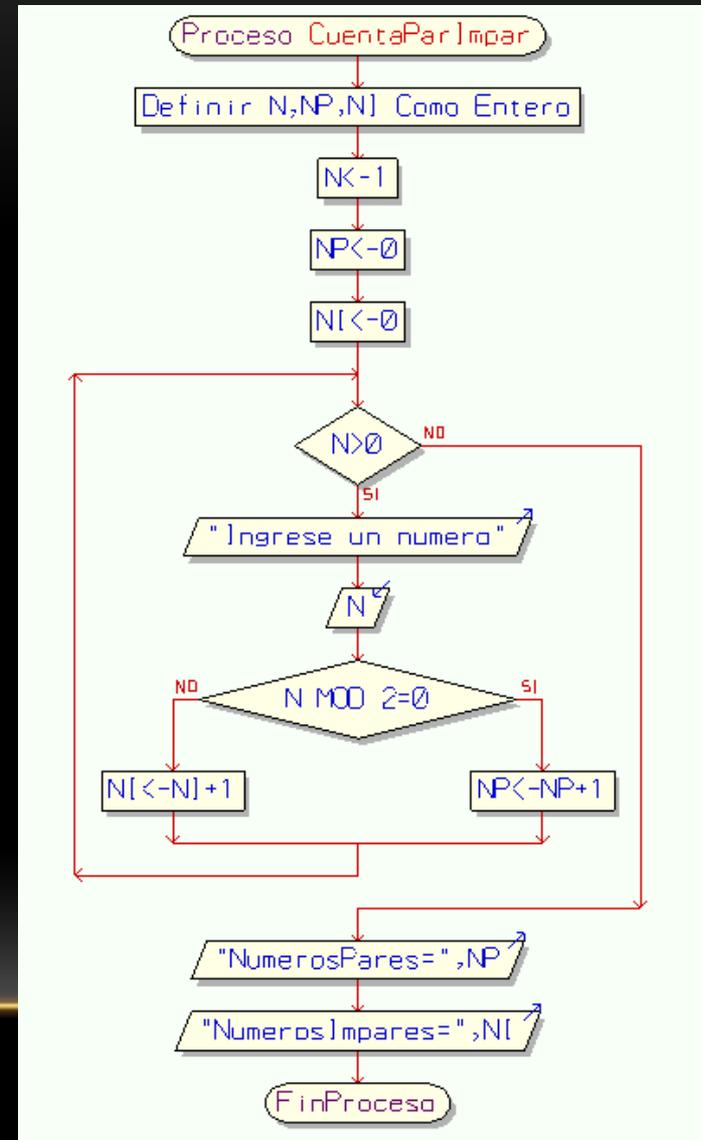
1. Proceso SUMA
  2. Definir N1,N2 como enteros;
  3. Leer N1,N2;
  4.  $S \leftarrow N1+N2$ ;
  5. Escribir S;
  6. FinProceso
-

# REALIZAR UN PROGRAMA QUE LEA EL RADIO DE UNA CIRCULO Y CALCULE EL ÁREA DE LA MISMA

1. Proceso AreaCirculo
  2. Definir Area Como Real;
  3. Definir Radio Como Entero;
  4. Escribir "Ingrese Radio";
  5. Leer Radio;
  6.  $Area \leftarrow \pi * Radio^2$ ;
  7. Escribir "Area=",Area;
  8. FinProceso
-

# HERRAMIENDA DIAGRAMA DE FLUJO

- Un **Diagrama de Flujo** permite ilustrar la secuencia de pasos de un algoritmo por medio de símbolos especializados y líneas de flujo describe la lógica para la solución de un problema (algoritmo). **El Diagrama de Flujo** es la representación gráfica de un algoritmo.



# ESTA REPRESENTACIÓN GRÁFICA DEBE TENER LAS SIGUIENTES CUALIDADES

- ◎ **Sencillez** en su construcción.
- ◎ **Claridad** en su comprensión.
- ◎ **Normalización** en su diseño.
- ◎ **Flexibilidad** en su modificación

En la práctica se suele utilizar indistintamente los términos **diagrama de flujo, organigrama y ordinograma** para referenciar cualquier representación gráfica de los flujos de datos o de las operaciones de un programa.