

## Práctica 3. Pseudocódigo.

Hacer una calculadora que haga lo siguiente:

1. Sumar dos números
2. Restar dos números
3. Multiplicar dos números
4. Dividir dos números
5. Salir



A, B: Números ingresados  
A,B son números reales

R: Resultado de la operación  
R es un número real

O: {  
1: Suma  
2: Resta  
3: Multiplica  
4: Divide  
5: Salir

## Práctica 3. Pseudocódigo.

Algoritmo

1. Inicio
2. Definir A,B,O y R
3. Mientras O sea diferente de 5 hacer
4. Si O es 1  $R = A + B$
5. Si O es 2  $R = A - B$
6. Si O es 3  $R = A * B$
7. Si O es 4  $R = A / B$
8. Mostrar R
9. Fin

# Algoritmo calculadora

## Repetir

```
    Escribir "Calculadora"
    Escribir "1.- Suma"
    Escribir "2.- Resta"
    Escribir "3.- Multiplicacion"
    Escribir "4.- Division"
    Escribir "5.- Salir"
    Escribir "Favor de seleccionar una opcion:"
    Leer seleccion
```

## Segun seleccion Hacer

```
1:
    Escribir "Ingrese los valores a sumar"
    Leer valor1, valor2
    resultado <- valor1 + valor2
    Escribir valor1, "+", valor2, "=", resultado

2:
    Escribir "Ingrese los valores a restar"
    Leer valor1, valor2
    resultado <- valor1 - valor2
    Escribir valor1, "-", valor2, "=", resultado

3:
    Escribir "Ingrese los valores a multiplicar"
    Leer valor1, valor2
    resultado <- valor1 * valor2
    Escribir valor1, "*", valor2, "=", resultado

4:
    Escribir "Ingrese los valores a dividir"
    Leer valor1, valor2
    Si valor2 != 0 Entonces
        resultado <- valor1 / valor2
        Escribir valor1, "/", valor2, "=", resultado
    SiNo
        Escribir "No se puede realizar una division entre cero"
    Fin Si

5:
    Escribir "Gracias por usar mi programa. Hasta pronto!"
De Otro Modo:
    Escribir "Opcion no valida, favor de seleccionar una opcion valida"
```

## Fin Segun

Hasta Que seleccion=5

## FinAlgoritmo

# Programación básica

---

## TEMA II. DIAGRAMAS DE FLUJO

# Diagramas de flujo

## Introducción

Una vez que un problema ha sido analizado y se ha diseñado un algoritmo que lo resuelva de manera eficiente, se procede a la etapa de codificación del algoritmo.

Para que la solución de un problema (algoritmo) pueda ser codificada, se debe generar una representación del mismo. Una representación algorítmica elemental gráfica son los diagramas de flujo.

Un digagrama de flujo es la representación gráfica de un algoritmo, es decir, muestra en forma gráficos los pasos a seguir para solucionar un problema.

# Objetivo del tema

---

El alumno escribirá diagramas de flujo a partir de un algoritmo.

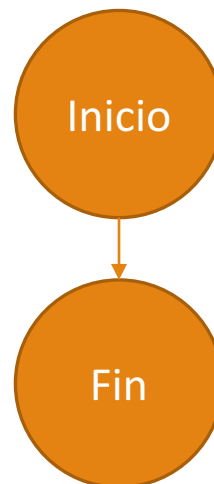


# Construcción de un diagrama de flujo

---

El diagrama cuenta con diferentes elementos gráficos y reglas para conectarlos.

1. Alcance del programa: Todo diagrama de flujo está limitado por un Inicio y un Fin. Dentro de estos límites se deben colocar todos los elementos gráficos que representan el algoritmo del programa
2. Existen gráficos determinados que representan acciones o instrucciones en el diagrama. También cuentan con reglas para conectar a los elementos.
3. Se recomienda usar una plantilla o un programa que ayude a dibujar los elementos.



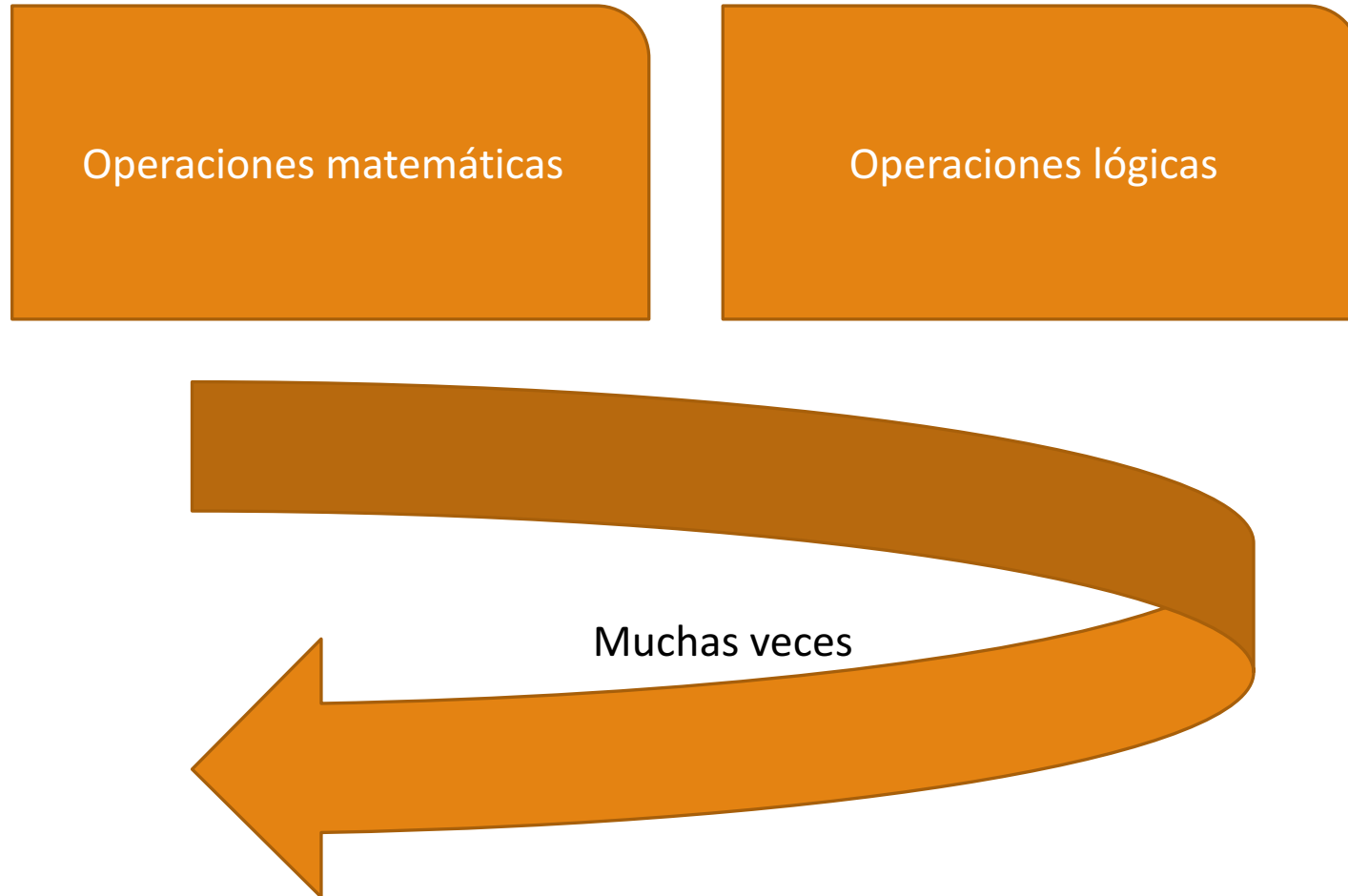


El diagrama de flujo es una forma de preparar un algoritmo para ser posteriormente codificado en algún lenguaje de programación



# Acciones en un diagrama de flujo

---



# **CARACTERÍSTICAS DEL DIAGRAMA DE FLUJO**

Estructuras de control de flujo.

Las estructuras de control de flujo permiten la ejecución y la repetición de un conjunto de instrucciones:

- Estructura de control secuencial.
- Estructuras de control selectivas. (SI -ENTONCES, DE LO CONTRARIO, SELECCIÓN DE CASOS)
- Estructuras de control iterativas o repetitivas. (MIENTRAS, y HACER - MIENTRAS)

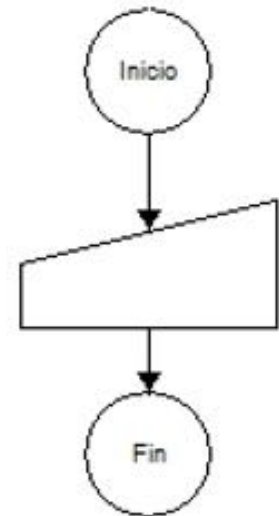
## Escribir. Mandar un mensaje

```
1  Algoritmo prueba
2      Escribir "Hola Mundo";
3  FinAlgoritmo
```



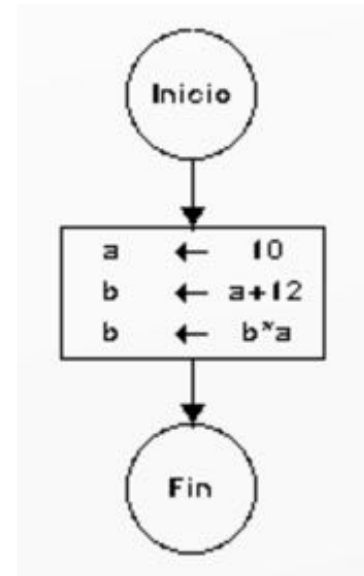
## Leer. Ingresar información por el usuario

```
1  Algoritmo prueba
2      Escribir "Favor de ingresar un número";
3      Leer x;
4  FinAlgoritmo
```



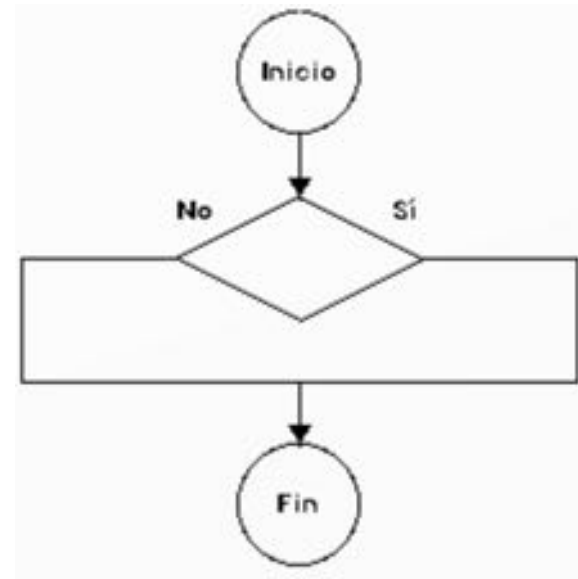
## Asignar. Asignar un valor a un identificador

```
1  Algoritmo prueba
2    Escribir "Favor de ingresar un número";
3    Leer x;
4    Escribir "El valor ingresado fue: ",x
5     $x \leftarrow -x + 20$ 
6    Escribir "El valor asignado es: ",x
7  FinAlgoritmo
```



## Si-Entonces. Operaciones lógicas de control

```
1  Algoritmo prueba
2    Escribir "Favor de ingresar un número";
3    Leer x;
4    Escribir "El valor ingresado fue: ",x
5     $x \leftarrow -x + 20$ 
6    Escribir "El valor asignado es: ",x
7    Si  $x > 40$  Entonces
8      Escribir "El valor asignado es mayor a 40: "
9    SiNo
10     Escribir "El valor asignado es menor o igual a 40"
11   Fin Si
12 FinAlgoritmo
```



## Ejemplo 1.

**Problema:** Suma de dos números enteros

**Restricciones:** Los números deben ser diferentes de cero



A, B: Números enteros a sumar  
A,B son diferentes de cero

S: La suma de los dos números  
S es un número entero

### Algoritmo

1. Inicio
2. Definir A,B y S
3. Si A y B son enteros y diferentes de cero continuar, si no ir al paso 2
4. Entonces  $S = A+B$
5. Mostrar S
6. Fin

## Ejemplo 2.

**Problema:** Calcular el volumen de una esfera

**Restricciones:** El radio debe ser diferente de cero



R: Radio de la esfera

R es un real diferente de cero y positivo

V: Volumen de la esfera

V es un real diferente de cero y positivo

### Algoritmo

1. Inicio
2. Definir R, V
3. Si R es real diferente de cero y positivo continuar si no ir a 2
4. Entonces  $V = (3/4) * (3.1416 * R * R * R)$
5. Mostrar V
6. Fin

## Ejemplo 3.

**Problema:** Multar a un automóvil que rebasa el límite de 50 km/h

**Restricciones:** La velocidad del automóvil no puede ser negativa



V: Velocidad del automóvil en km/h

V es un real positivo

M: Si se multa o no

### Algoritmo

1. Inicio
2. Definir V, M
3. Si V es real y positivo continuar si no ir a 2
4. Si V es mayor a 50
5. Entonces M=1
6. Si no M=0
7. Mostrar M
8. Fin

M: {  
0: No se multa  
1: Se multa



## Ejemplo 4.

**Problema:** Obtener el mayor de dos números enteros dados

**Restricciones:** Los números de entrada deben ser diferentes



A, B: Números a comparar  
A,B son números reales diferentes

S: El mayor de los números

Algoritmo

1. Inicio
2. Definir A,B y S
3. Si A y B son números reales y diferentes, si no ir al paso 2
4. Si  $A > B$
5. Entonces  $S =$
6. Fin

S: { A  
B

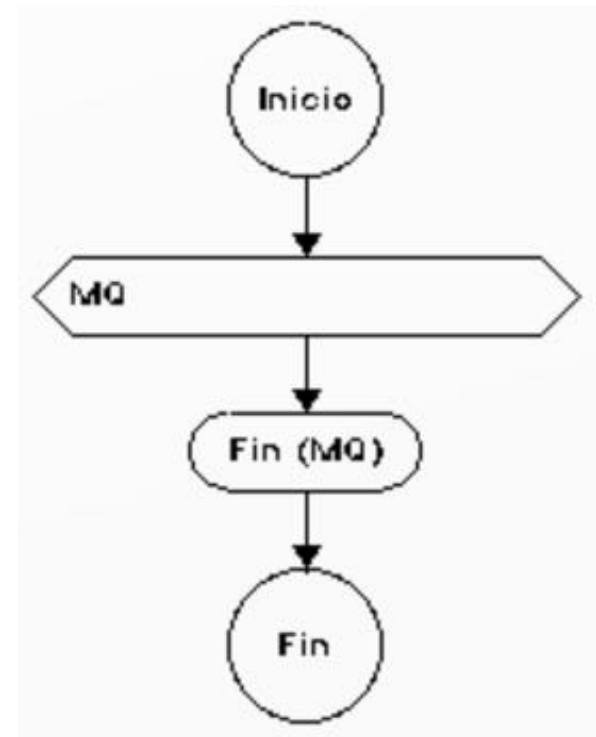
## Mientras. Operaciones lógicas de control

Mientras expresion\_logica Hacer  
secuencia\_de\_acciones  
Fin Mientras



## Repetir. Operaciones lógicas de control

Repetir  
secuencia\_de\_acciones  
Hasta Que expresion\_logica



## Ejemplo 6.

**Problema:** Hacer un contador regresivo para el lanzamiento de un cohete, el usuario ingresará el valor desde donde comenzará la cuenta regresiva. La cuenta decrementará hasta llegar a cero, Se tendrá que mostrar la cuenta regresiva y cuando llegue a cero se mandará el mensaje de “Despega!”

Restricciones: El número deberá ser mayor o igual a 10 y ser entero.



N: Número inicial entero mayor a cero

N: Cuenta regresiva de  $N=N-1$  hasta 0

M: Mensaje “Despega!”

### Algoritmo

1. Inicio
2. Definir N y M = “Despega!”
3. Si  $N \geq 10$  continua si no ir a 2
4. Repetir paso 5 y 6 hasta que N sea 0
5.  $N = N - 1$
6. Mostrar N
7. Mostrar M
8. Fin



## Práctica 3. Pseudocódigo.

Hacer una calculadora que haga lo siguiente:

1. Sumar dos números
2. Restar dos números
3. Multiplicar dos números
4. Dividir dos números
5. Salir



A, B: Números ingresados  
A,B son números reales

O: {  
1: Suma  
2: Resta  
3: Multiplica  
4: Divide  
5: Salir

## Práctica 3. Pseudocódigo.

Algoritmo

1. Inicio
2. Definir A,B,O y R
3. Mientras O sea diferente de 5 hacer
4. Si O es 1  $R = A + B$
5. Si O es 2  $R = A - B$
6. Si O es 3  $R = A * B$
7. Si O es 4  $R = A / B$
8. Mostrar R
9. Fin

[goo.gl/aNnHcP](https://goo.gl/aNnHcP)

