

# DESARROLLO DE APLICACIONES MULTIPLATAFORMA PROGRAMACIÓN

## TEMA 1 METODOLOGÍA DE LA PROGRAMACIÓN

# CONTENIDOS

- 1.1. Pseudocódigo. Elementos.
- 1.2. El entorno.
- 1.3. Constantes, Variables y Asignación.
- 1.4. Entrada y salida.
- 1.5. Sentencias de control: alternativas.
- 1.6. Operadores: Comparación, Aritméticos y Lógicos.
- 1.7. Ejemplo I.
- 1.8. Alternativas concatenadas.
- 1.9. Sentencias de control: bucles.
- 1.10. Ejemplo II.
- 1.11. Teorema de la programación estructurada.

## 1.1. PSEUDOCÓDIGO. ELEMENTOS.

- El pseudocódigo se puede definir como un lenguaje intermedio entre el lenguaje natural y el lenguaje de programación que usaremos posteriormente para generar nuestro programa.
- Es un lenguaje, también llamado *falso lenguaje*, que permite generar algoritmos dando soluciones a problemas planteados.
- Reglas:
  - Las sentencias se ejecutan de arriba abajo y una a una. Una sentencia es una “frase” del algoritmo en la que se lleva a cabo una acción concreta o se realiza alguna operación.
  - Una sentencia comenzará a ejecutarse cuando la que se encuentra justo antes ha finalizado.
  - Las sentencias finalizarán con el signo punto y coma (;).
  - Definiremos lo que se llama entorno, que agrupa variables y constantes.
  - Usaremos las palabras “Inicio” y “Fin” para limitar el algoritmo.
  - Tipos de sentencias: de asignación, de entrada/salida de datos y sentencias de control (alternativas y bucles).

## 1.2. EL ENTORNO.

- El entorno es un grupo de objetos que usaremos para almacenar información de algún tipo. Estos objetos serán *variables* o *constantes*.
- Una variable es un contenedor donde guardaremos un valor. Este contenedor será referenciado por un identificador o nombre. El valor de este contenedor puede variar durante la ejecución del algoritmo.
- Una constante es similar a una variable en el contexto de que almacenan un valor y es referenciado por un nombre, pero este valor no variará a lo largo de la ejecución del algoritmo.
- Una asignación permite almacenar un valor en una variable o constante.
- Reglas:
  - Comenzaremos con la palabra “Entorno” seguida de dos puntos (:).
  - Los tipos de datos de las variables se escriben en minúsculas.
  - Las palabras reservadas, como “constante”, se escriben en minúsculas.
  - Los nombres de constantes se escriben con todas sus letras en mayúsculas.
  - Los nombres de variables se escriben en minúsculas menos la primera letra.

## 1.3. CONSTANTES, VARIABLES Y ASIGNACIÓN.

Declaración de una constante:

*constante tipo\_dato nombre\_constante=valor;*

Declaración de una variable:

*tipo\_dato nombre\_variable;*  
*tipo\_dato nombre\_variable=valor;*

Entorno:

**constante tipo\_dato nombre\_constante=valor;**  
**tipo\_dato nombre;**  
**tipo\_dato nombre=valor;**

Entorno:

constante real PI=3.1416  
entero radio=2;  
real area;

## 1.4. ENTRADA Y SALIDA.

- Los valores de entrada en un programa real serán insertados a través de la línea de comandos, cuadros de texto de formularios, listas desplegables, etc., usando dispositivos periféricos estándar de entrada, tales como el teclado y el ratón, o incluso, un fichero de datos.
- Los datos de salida serán aquellos generados por las operaciones que el propio algoritmo realice a partir de los datos de entrada. La salida estándar será la pantalla, pero también, una impresora o un fichero.

Lectura:

*Leer nombre\_variable*

Escritura:

*Escribir valor*

Entorno:

```
constante real PI=3.1415;  
int radio;
```

Inicio:

```
Escribir "Introduce el radio";  
Leer radio;
```

Fin;

## 1.5. SENTENCIAS DE CONTROL: ALTERNATIVAS.

- En una instrucción alternativa se establece una condición o regla. El código a ejecutar dependerá de si la condición establecida se cumple o no.
  - Alternativas simples. Son aquellas en las que se ejecutarán un conjunto de instrucciones solo si la condición establecida es cierta.
  - Alternativas dobles. Se especifica qué hacer en caso de que la condición sea verdadera o falsa, es decir, si la regla establecida es verdadera se ejecutarán una serie de instrucciones y si es falsa, otras.
  - Alternativas múltiples. Aquí la expresión o regla a comprobar no devuelve los valores verdadero o falso, devuelve un valor incluido en un rango establecido y según este se realiza una acción u otra.

## 1.5. SENTENCIAS DE CONTROL: ALTERNATIVAS.



*Si CONDICION entonces*  
    *Instrucción 1;*  
    *Instrucción 2;*  
    ...  
    *Instrucción n;*  
*FinSi;*

## 1.5. SENTENCIAS DE CONTROL: ALTERNATIVAS.

Entorno:

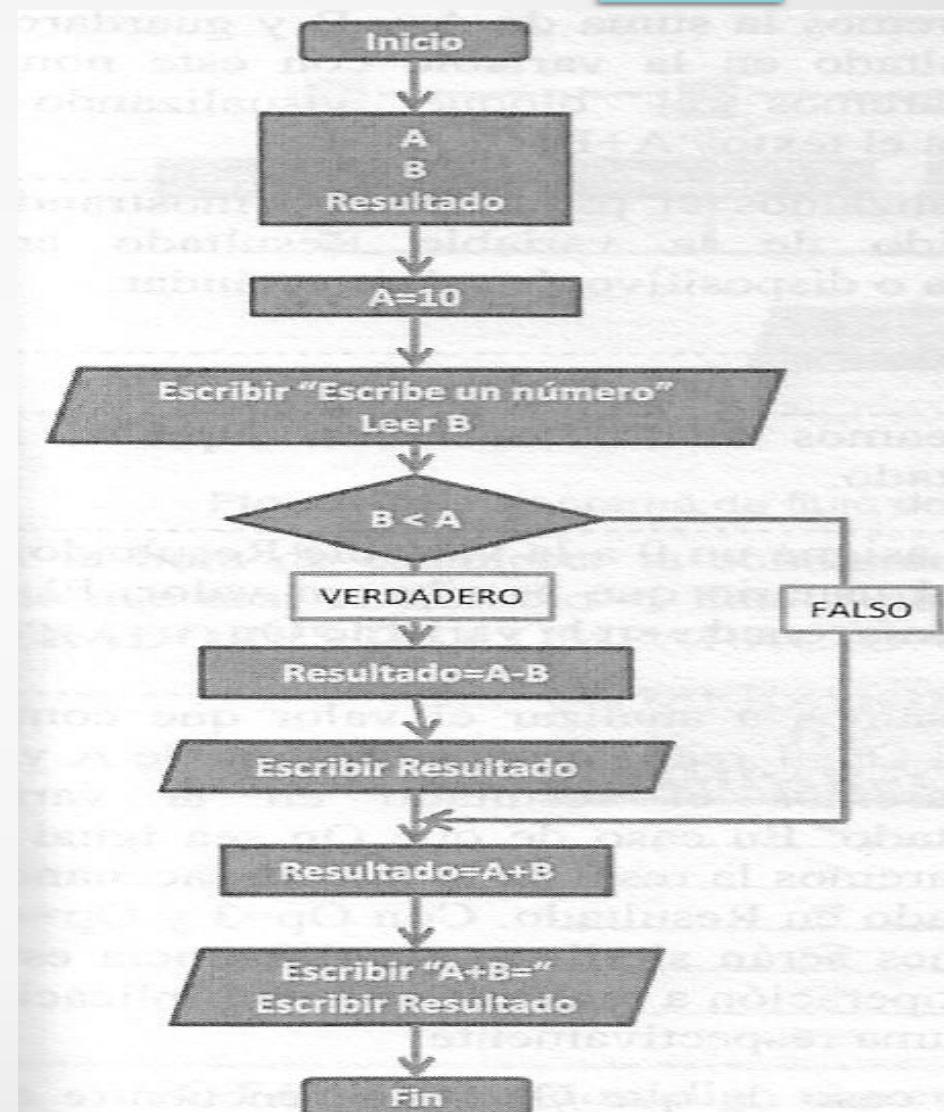
```
    entero A, B, Resultado;
```

Inicio:

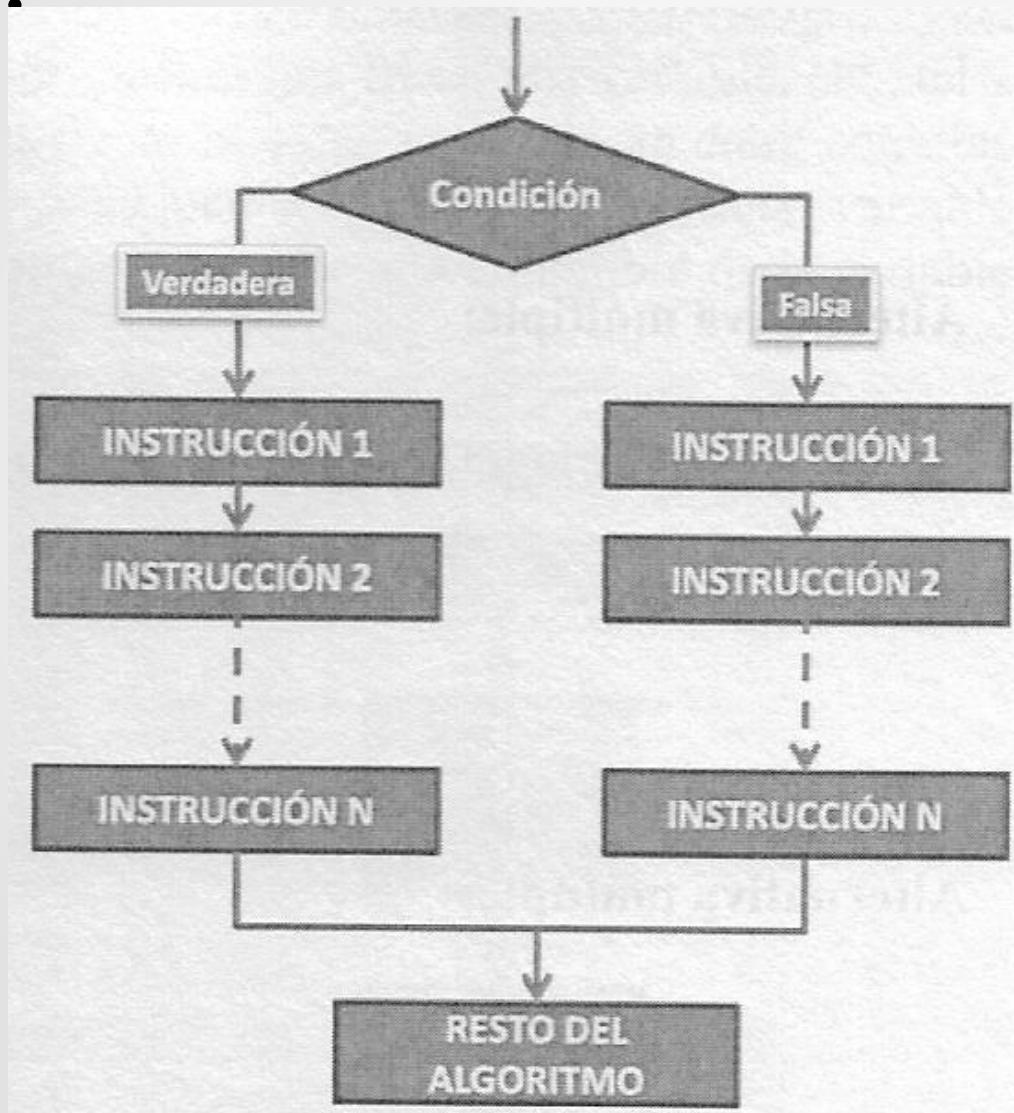
```
    A=10;  
    Escribir "Escribe un  
        número";  
    Leer B;
```

```
Si B < A entonces  
    Resultado=A-B;  
    Escribir Resultado;  
FinSi;
```

```
    Resultado=A+B;  
    Escribir "A+B=";  
    Escribir Resultado;  
Fin;
```



## 1.5. SENTENCIAS DE CONTROL: ALTERNATIVAS.



*Si CONDICIÓN entonces*

*Instrucción 1;*

*...*

*Instrucción n;*

*Sino*

*Instrucción 1;*

*...*

*Instrucción n;*

*FinSi;*

## 1.5. SENTENCIAS DE CONTROL: ALTERNATIVAS.

Entorno:

    entero A, B, Resultado;

Inicio:

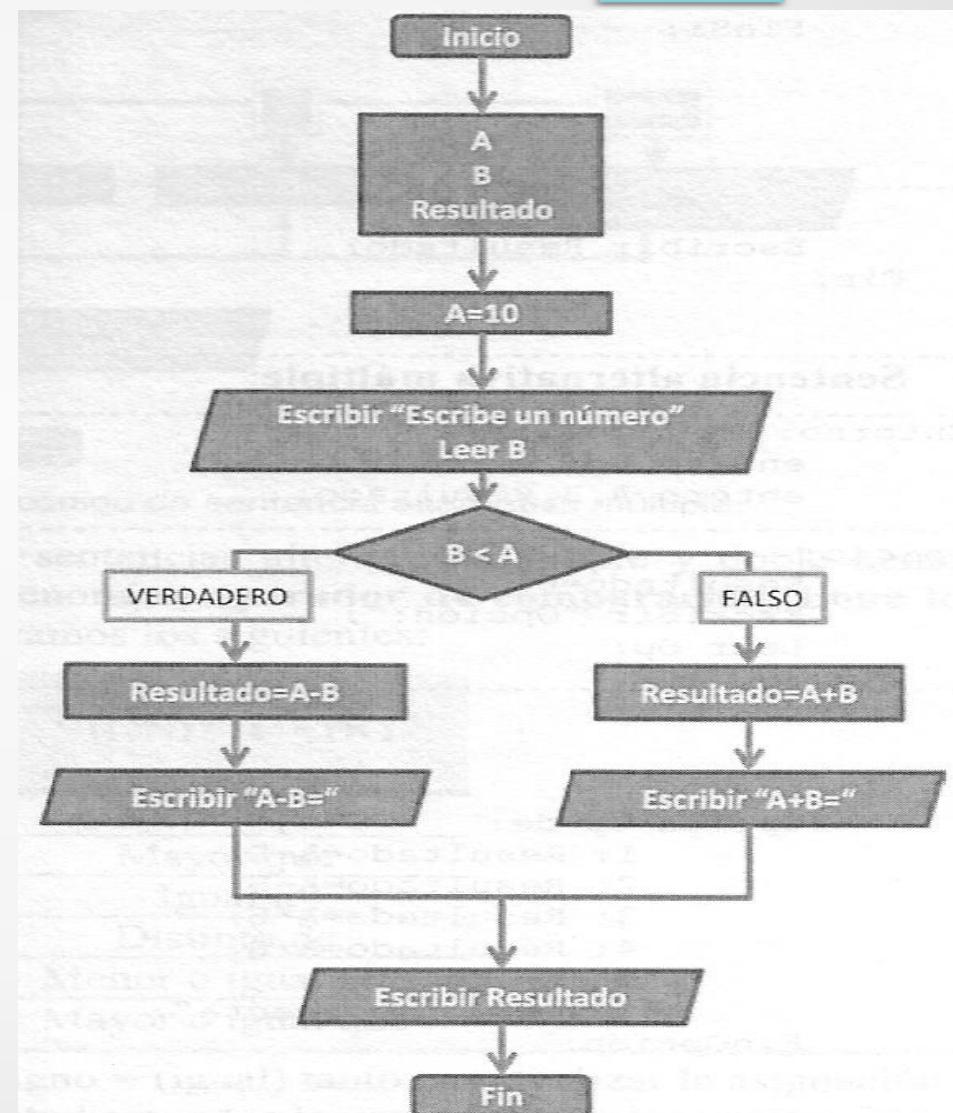
    A=10;

    Escribir "Escribe un  
    número";

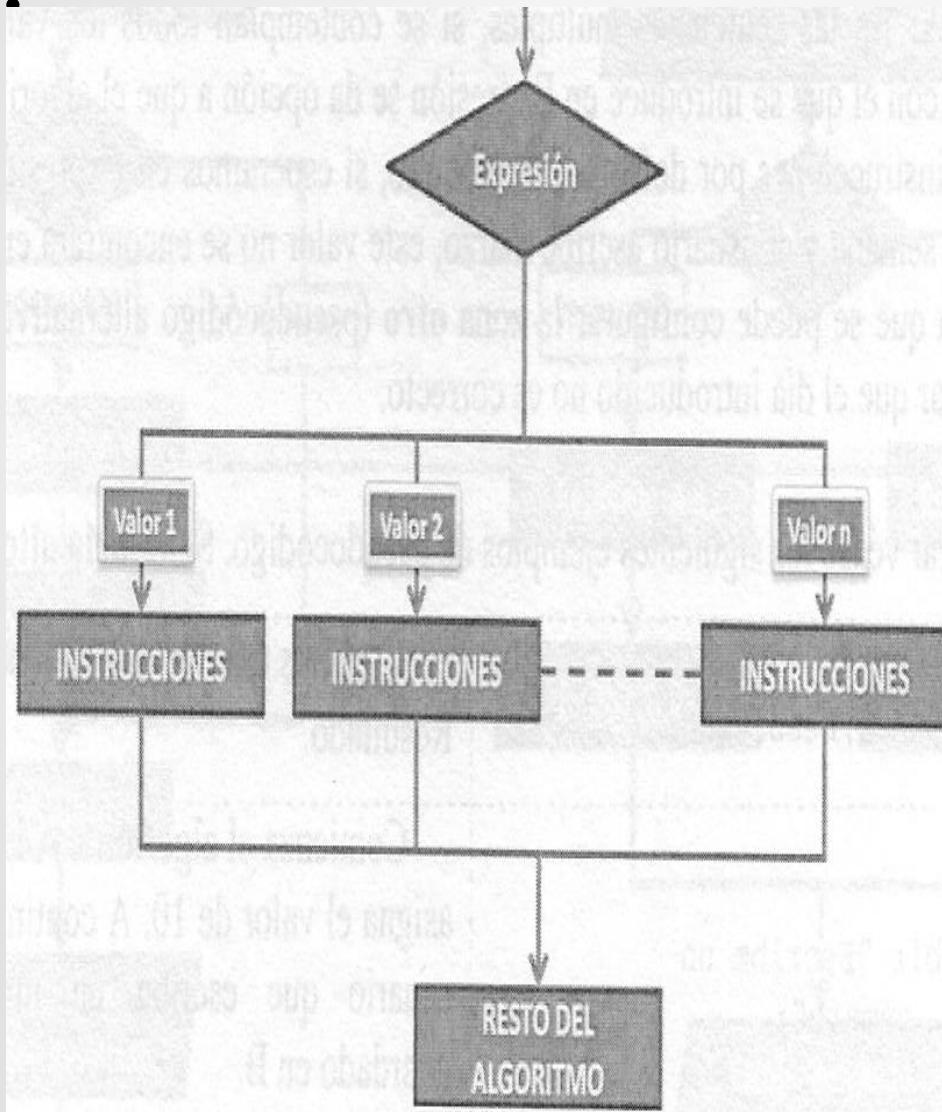
    Leer B;

    Si B < A entonces  
        Resultado=A-B;  
        Escribir "A-B=";  
    Sino  
        Resultado=A+B;  
        Escribir "A+B=";  
    FinSi;

    Escribir Resultado;  
Fin;



## 1.5. SENTENCIAS DE CONTROL: ALTERNATIVAS.



*Opcion EXPRESIÓN de*

*valor 1: Intrucción/Instrucciones(1);*

*valor 2: Intrucción/Instrucciones(2);*

*...*

*valor n: Intrucción/Instrucciones(n);*

*otro: Intrucción/Instrucciones por defecto;*

*FinOpcion;*

## 1.5. SENTENCIAS DE CONTROL: ALTERNATIVAS.

Entorno:

```
entero Op;  
entero A, B, Resultado;
```

Inicio:

```
Resultado=0;  
Escribir "Opción:";  
Leer Op;
```

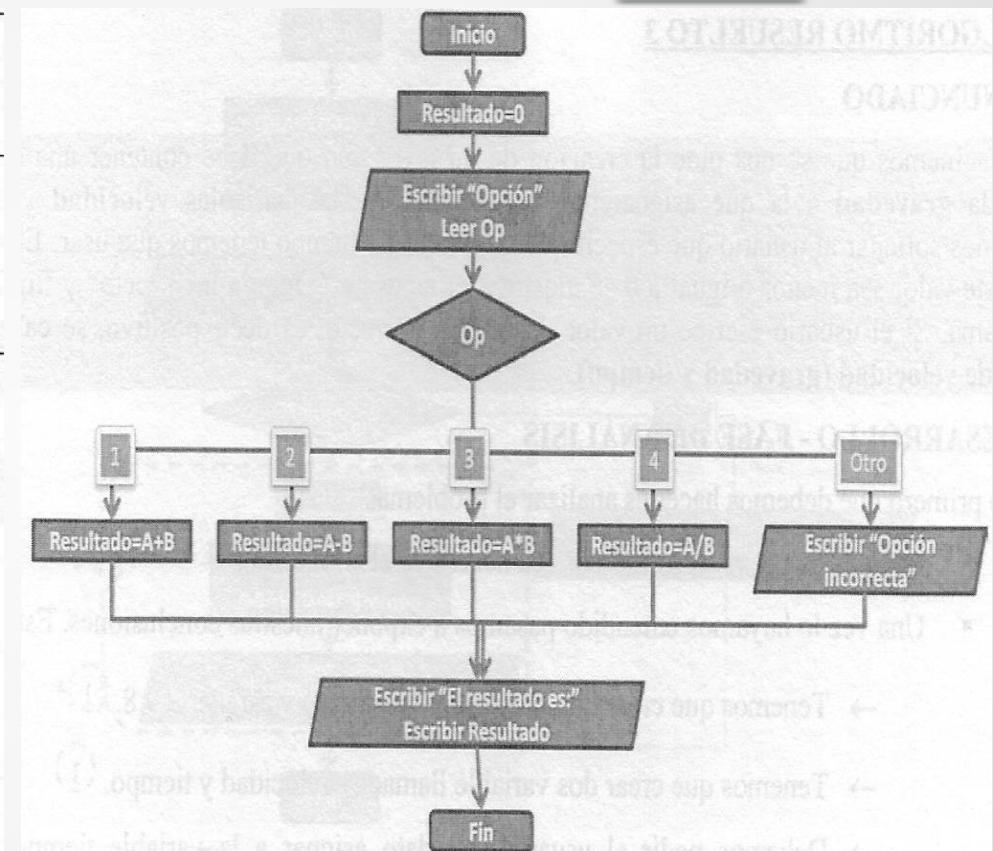
Opcion Op de:

```
1: Resultado=A+B;  
2: Resultado=A-B;  
3: Resultado=A*B;  
4: Resultado=A/B;
```

```
Otro: Escribir  
"Opción incorrecta";
```

FinOpcion;

```
Escribir "El resultado  
es:";  
Escribir Resultado;
```



## 1.6. OPERADORES.

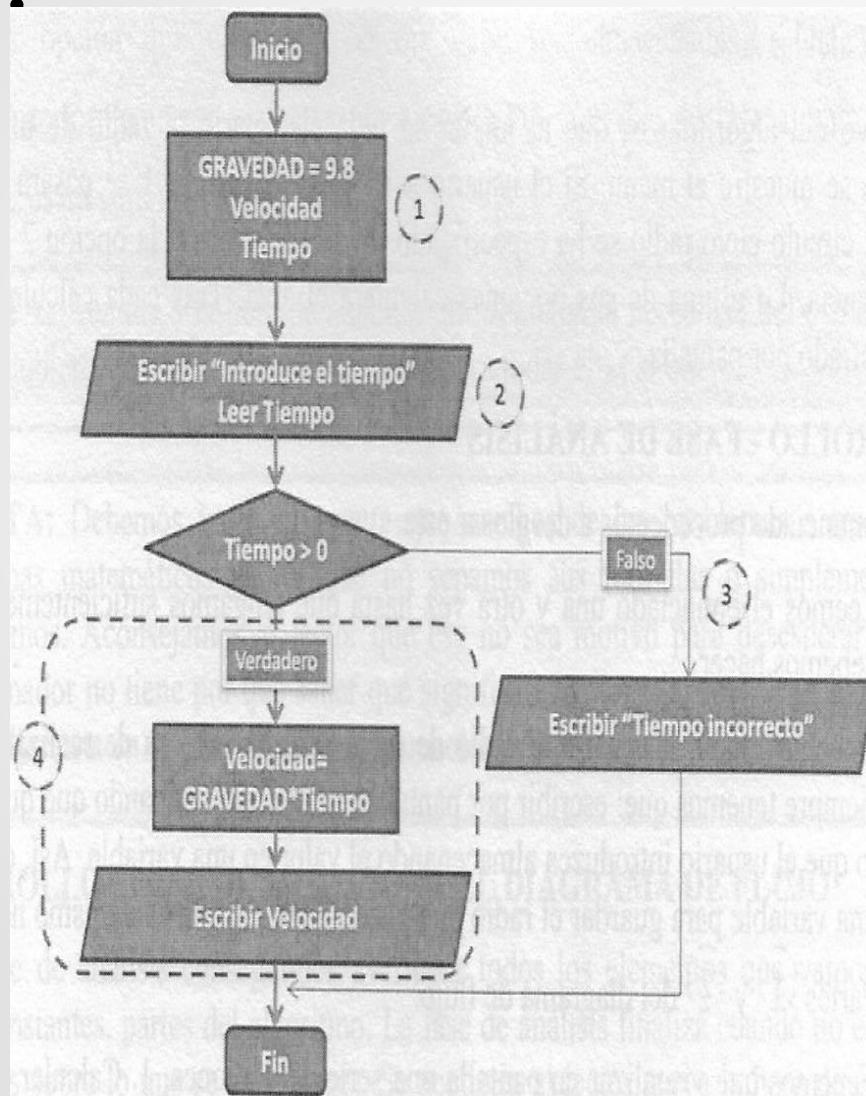
### COMPARACIÓN, ARITMÉTICOS Y LÓGICOS.

OPERADORES DE COMPARACIÓN	SIGNIFICADO
<	Menor que
>	Mayor que
=	Igual a
◊	Distinto de
<=	Menor o igual que
>=	Mayor o igual que

OPERADORES ARITMÉTICOS	SIGNIFICADO
+	Suma
-	Resta
*	Multiplicación
/	División

OPERADORES LÓGICOS	SIGNIFICADO
Y	La condición final será cierta si todas las reglas que la forman lo son
O	La condición final será cierta si al menos una de las reglas que la forman lo es
No	Invierte el significado de la condición. Si la regla es cierta se convertirá en falsa y viceversa.

## 1.7. EJEMPLO I.



Entorno:

```
/*Zona (1)*/
constante real GRAVEDAD;
real Velocidad;
entero Tiempo;
```

Inicio:

```
/*Las siguientes dos líneas forman la zona (2)*/
Escribir "Introduce el tiempo";
Leer Tiempo;
Si Tiempo > 0 entonces
    /*Zona (4)*/
    Velocidad=GRAVEDAD*Tiempo;
    Escribir Velocidad;
Sino
    /*Zona (5)*/
    Escribir "Tiempo incorrecto";
FinSi;
```

Fin;

## 1.8. ALTERNATIVAS CONCATENADAS.

*Si CONDICIÓN entonces*

*Instrucciones;*

***Sino***

*Instrucciones;<sup>5</sup>*

*Si CONDICIÓN entonces*

*Instrucciones;*

*Si CONDICIÓN entonces*

*...*

***Sino***

*...*

***FinSi;***

***Sino***

*Instrucciones;*

*Si CONDICIÓN entonces*

*...*

*...*

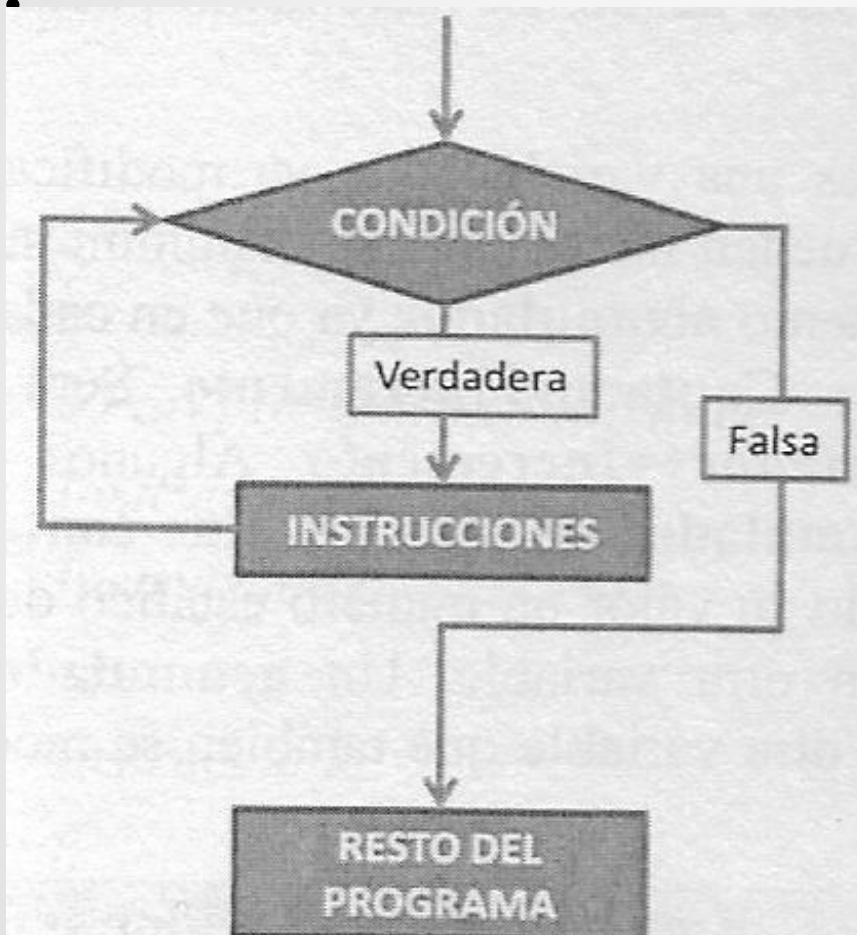
***FinSi;***

***FinSi;***

## 1.9. SENTENCIAS DE CONTROL: BUCLES.

- Un bucle es un tipo de sentencia de control que hará que una o varias instrucciones se repitan tantas veces como permita la configuración de dicha sentencia.
- Así, un bucle debe contener al menos una instrucción y estar controlado por una condición que determine el número de repeticiones.
- La condición de control debe ser modificada de algún modo en el interior del bucle para garantizar que este finalice en un momento determinado.
  - Mientras. La condición se evalúa al comenzar, de tal forma que pueda darse en caso en que no llegue a ejecutarse ninguna instrucción.
  - Hacer ... Mientras. La condición se evalúa al final, de tal forma que al menos una vez se ejecutan las instrucciones.
  - Repetir ... Hasta. Similar a la anterior.
  - Para. El conjunto de instrucciones se ejecuta un número concreto de veces. Se usan “contadores” o “acumuladores” para controlar el número de ejecuciones del mismo, además de un “valor inicial” y un “valor final”.

## 1.9. SENTENCIAS DE CONTROL: BUCLES.



**Mientras CONDICIÓN hacer**  
    Instrucción1;  
    Instrucción2;  
    ...  
    Instrucciónn;  
    **modificar CONDICIÓN;**  
**FinMientras;**

### Uso del bucle Mientras

- (1) Entorno:
- (2)     entero Contador;
- (3) Inicio:
- (4)     Contador=0;
- (5)     Mientras Contador < 5 hacer
- (6)         Escribir "Hola Mundo";
- (7)         Contador=Contador+1;
- (8)     FinMientras;
- (9) Fin;

## 1.9. SENTENCIAS DE CONTROL: BUCLES.



**Hacer:**

*Instrucción1;*

...

*Instrucciónn;*

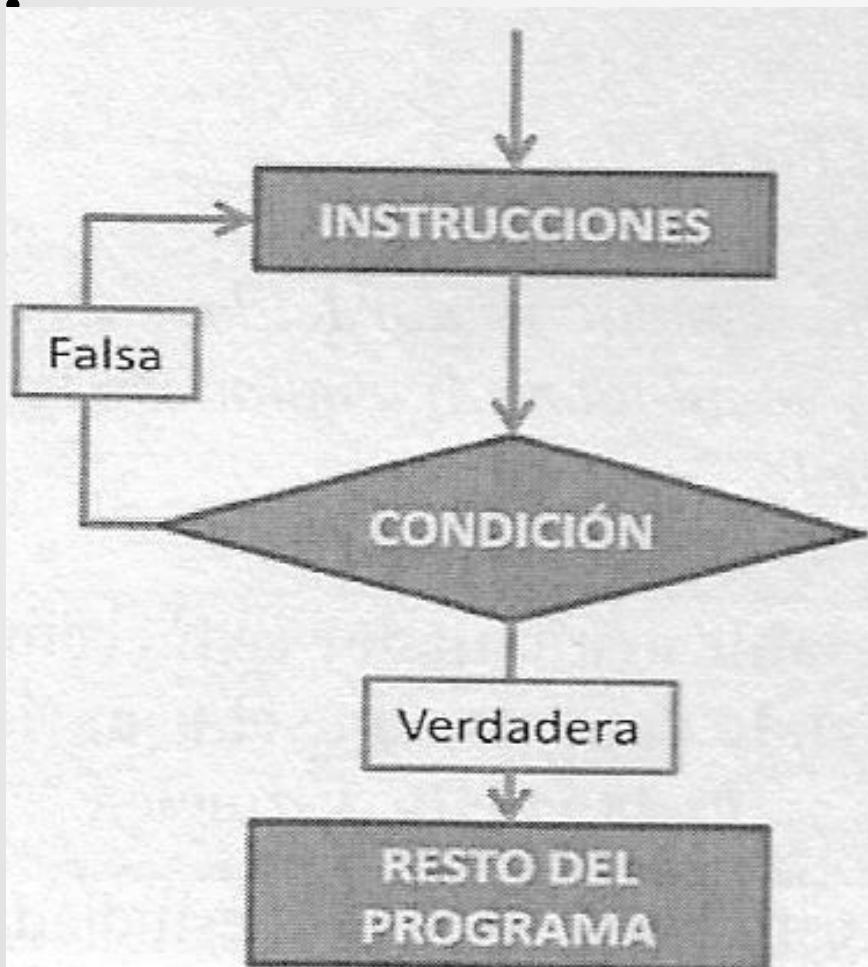
***modificar CONDICIÓN;***

***Mientras CONDICIÓN;***

### Uso del bucle Hacer...Mientras

- (1) Entorno:
- (2) entero Contador;
- (3) Inicio:
- (4) Contador=0;
- (5) Hacer
- (6)                   Escribir "Homa Mundo";
- (7)                   Contador=Contador+1;
- (8)                   Mientras Contador < 5;
- (9) Fin;

## 1.9. SENTENCIAS DE CONTROL: BUCLES.



*Repetir*

*Instrucción1;*

*Instrucción2;*

...

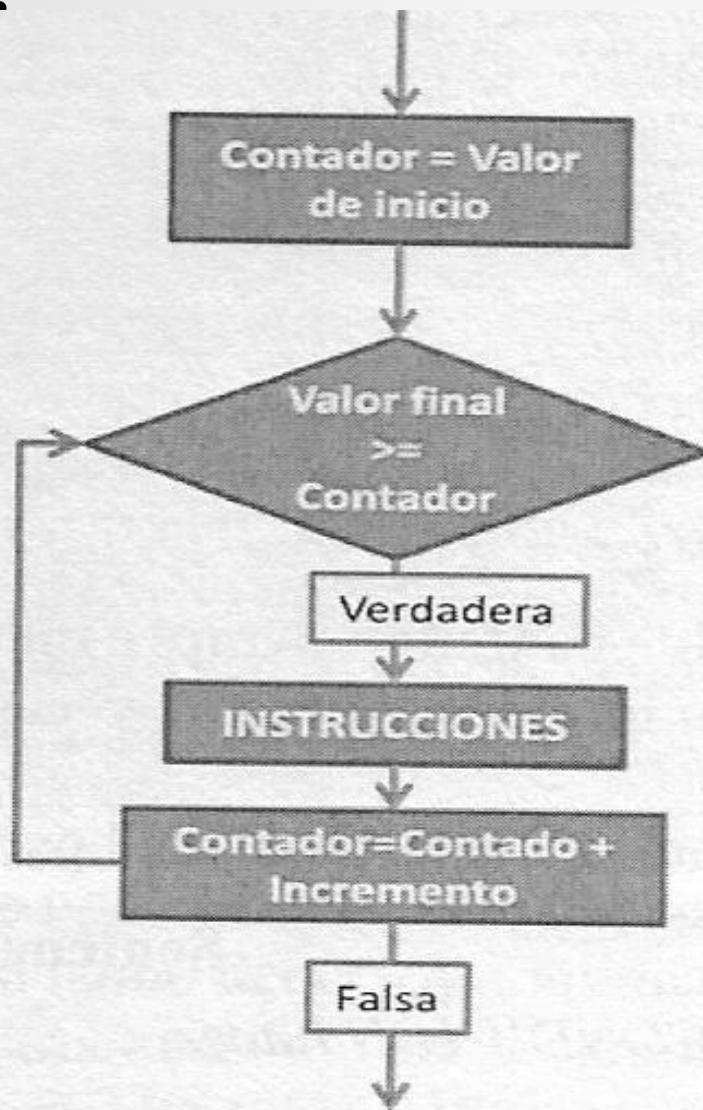
*Instrucciónn;*

*Hasta CONDICIÓN;*

### Uso del bucle Repetir hasta

- (1) Entorno:
- (2)    entero Contador;
- (3) Inicio:
- (4)    Contador=0;
- (5)    Repetir
- (6)                Escribir "Hola Mundo";
- (7)                Contador=Contador+1;
- (8)                Hasta Contador >= 5;
- (9) Fin;

## 1.9. SENTENCIAS DE CONTROL: BUCLES.

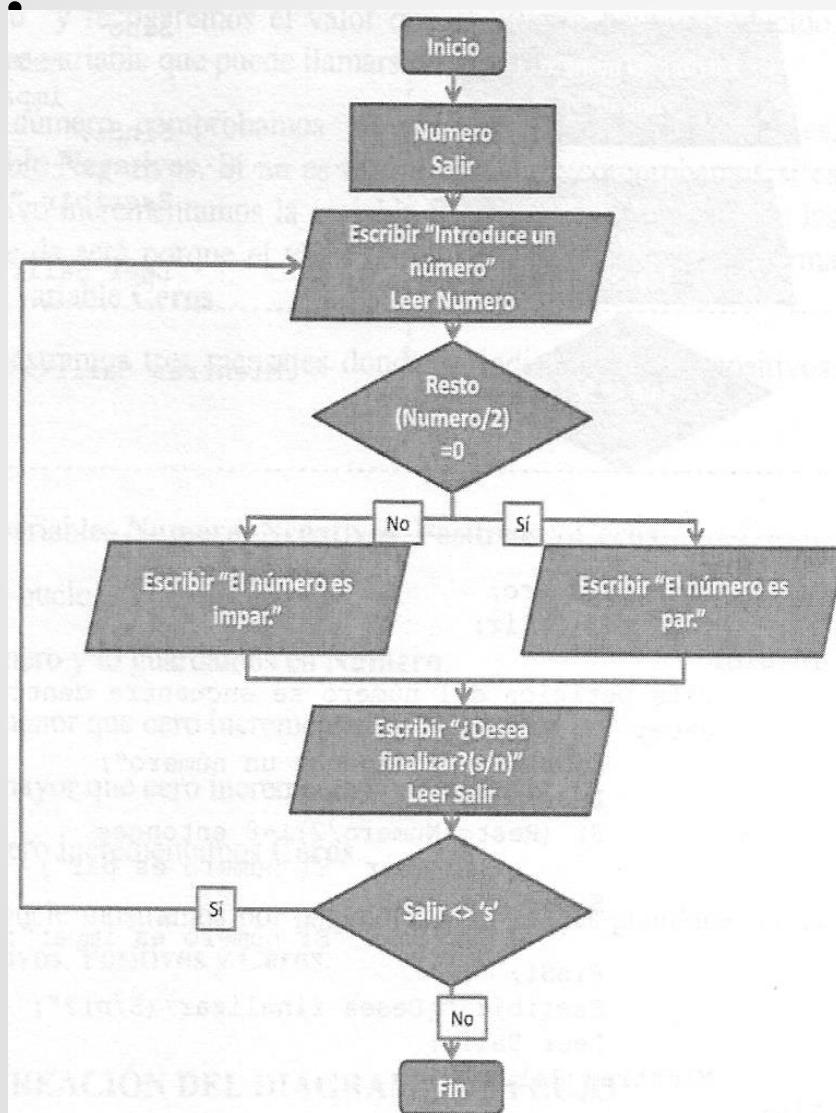


**Para** c=Vi a Vf **incremento Inc** **hacer**  
**Instrucción1;**  
**Instrucción2;**  
**...**  
**Instrucciónn;**  
**FinPara:**

## Uso del bucle Para

- (1) Entorno:
  - (2) entero Contador;
  - (3) Inicio:
  - (4) Para Contador=0 a 5 incremento 1 hacer
  - (5) Escribir "Hola Mundo";
  - (6) FinPara;
  - (7) Fin;

## 1.10. EJEMPLO II.



Entorno:

entero Número;  
carácter Salir;

Inicio:

/\*La petición del número se encuentra dentro del bucle\*/  
Hacer

Escribir "Introduce un número";  
Leer Número;  
Si (Resto (Número/2))=0 entonces  
    Escribir "El número es par";  
Sino

    Escribir "El número es impar";  
FinSi;  
Escribir "¿Desea finalizar (S/n)?";  
Leer Salir;  
Mientras Salir<>'s';

Fin;

## 1.11. TEOREMA DE LA PROGRAMACIÓN ESTRUCTURADA.

- El teorema de la programación estructurada establece que todo programa puede ser implementado mediante el uso de tres tipos de estructuras de control:
  - *Secuencial.*
  - *Alternativa*
  - *Repetitiva.*
- Existen lenguajes, sobre todo en el pasado, que usaban estructuras GOTO, que equivalen a un salto de línea, es decir, cada vez que veíamos un GOTO x, el programa saltaba a la línea con número x.
- Este teorema establece que este tipo de sentencias no son necesarias y pueden ser sustituidas por un bucle, una sentencia alternativa simple, doble o múltiple, o bien mediante una secuencia de instrucciones.
- Hoy día, prima la programación orientada a objetos, donde se usan los elementos vistos en este tema pero para la ejecución de acciones o métodos de unos componentes denominados objetos.

# TEMA 1

## METODOLOGÍA DE LA PROGRAMACIÓN

FIN