



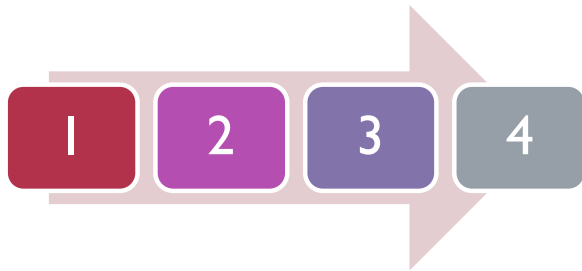
INSTRUCCIONES BÁSICAS DE UN ALGORITMO

ING. GIANKARIS G. MORENO R., M.SC

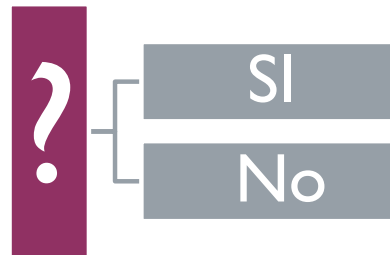


INSTRUCCIONES BÁSICAS DE UN ALGORITMO

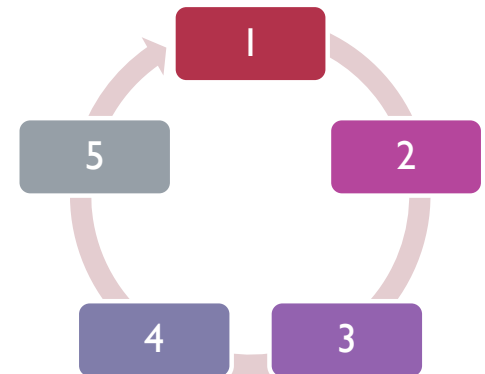
- Son instrucciones cuyo objetivo es controlar la ejecución de otras instrucciones o alterar el orden de ejecución normal de las instrucciones de un programa o algoritmo.
- Un algoritmo puede manejar:



Estructuras de Secuencia



Estructuras Alternativas



Estructuras Repetitivas

ESTRUCTURAS DE SECUENCIA

- Las estructuras de secuencia son todas aquellas sentencias o instrucciones, que son ejecutadas en el orden en que aparecen en el algoritmo o programa.
- Ejemplo:
 - Declaraciones de variables y constantes
 - Asignaciones
 - Entradas y salidas de datos

OPERADORES RELACIONALES Y LÓGICOS

- Operadores Relacionales
- Los operadores relacionales nos permiten comparar expresiones de tipos compatibles, devolviendo un resultado de tipo lógico: la comparación puede ser **cierta (C)** o **falsa (F)**.
- Estructura: **expresión_1 operador_relacional expresión_2**
- Los operadores relacionales son:

Descripción	Operador	Estructura	Ejemplo	
Menor que	<	(a < b)	7 < 9	C
Mayor que	>	(a > b)	2 > 15	F
Menor o igual que	<=	(a <= b)	8 <= 8	C
Mayor o igual que	>=	(a >= b)	4 >= 1	C
Igual que	==	(a == b)	'm' == 'M'	F
Distinto que	!=	(a != b)	'S' != 'N'	C

OPERADORES RELACIONALES Y LÓGICOS

- Operadores Lógicos
- Los operadores lógicos, se utilizan para establecer relaciones entre valores lógicos (C o F). Estos valores pueden ser resultado de una expresión relacional, devolviendo un resultado de tipo lógico (C o F).
- Los operadores relacionales son:

Descripción	Operador	Ejemplo
Negación	NO	NO (7 < 9) F
Y (AND)	Y	(2 < 15) Y (6 >= 1) C
O (OR)	O	(8 != 8) O (5 < 10) C

OPERADORES RELACIONALES Y LÓGICOS

- Operadores Lógicos

Tabla de la verdad del Y

A	B	A Y B
Cierto	Cierto	Cierto
Cierto	Falso	Falso
Falso	Cierto	Falso
Falso	Falso	Falso

Tabla de la verdad del O

A	B	A O B
Cierto	Cierto	Cierto
Cierto	Falso	Cierto
Falso	Cierto	Cierto
Falso	Falso	Falso

Tabla del No

A	NO (A)
Cierto	Falso
Falso	Cierto

OPERADORES RELACIONALES Y LÓGICOS

- Ejemplos de Operadores Lógicos

- $(90 > 0) \text{ Y } (1 == 1)$

C Y C
C

- $\text{NO}('S' == 's') \text{ Y } (12 < 4)$

C Y F
F

- $(35 < 12) \text{ O } (15 > 5)$

F O C
C

- $(10 < 10) \text{ O } (6 != 6)$

F O F
F

ESTRUCTURAS ALTERNATIVAS

- Son las estructuras que condicionan la ejecución de las sentencias / instrucciones, dependiendo del cumplimiento o no, de una condición.
- Esto quiere decir que puede haber más de un camino a seguir, todo va a depender de si se cumple o no la condición evaluada.
- Pueden ser simples, dobles y múltiples



ESTRUCTURAS ALTERNATIVAS

- Las condiciones:
 - Pueden tomar dos posibles valores: **cierto** o **falso**.
 - Pueden manejar/evaluar tipos de datos **entero**, **real** o **carácter**.
 - Se forman empleando los operadores **relacionales** y **lógicos**.
- Ejemplos de una condición:

(valor \geq 71)

(edad < 25 Y sexo == 'M')

ESTRUCTURAS ALTERNATIVAS SIMPLES

- Son aquella que solo evalúan la parte cierta de una condición.
- Formato:

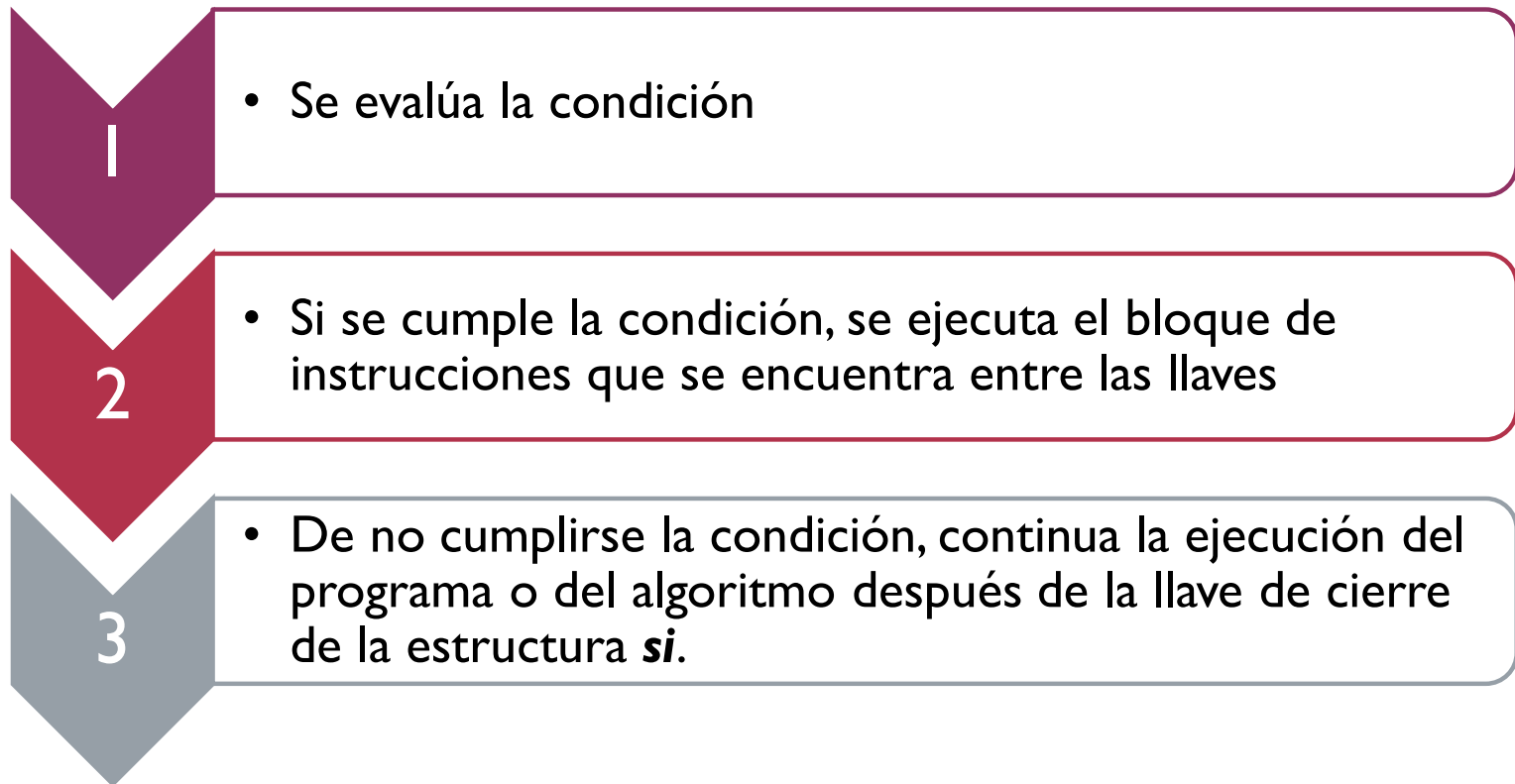
```
si (condición)  
{  
    //bloque de instrucciones  
}
```

- Ejemplo:

```
si (numero > 0)  
{  
    imprimir (“El número es positivo”);  
}
```

ESTRUCTURAS ALTERNATIVAS SIMPLES

■ ¿Cómo trabaja la estructura alternativa simple?



ESTRUCTURAS ALTERNATIVAS DOBLES

- Son aquella que permiten evaluar una condición y si la condición se cumple se ejecutan las sentencias dentro del bloque de instrucciones del **si**. De no cumplirse la condición, se ejecutan las instrucciones que se encuentran en el bloque de instrucciones del **de otro modo**.
- Formato:

```
si (condición)
{
    //bloque de instrucciones
}
de otro modo
{
    //bloque de instrucciones
}
```

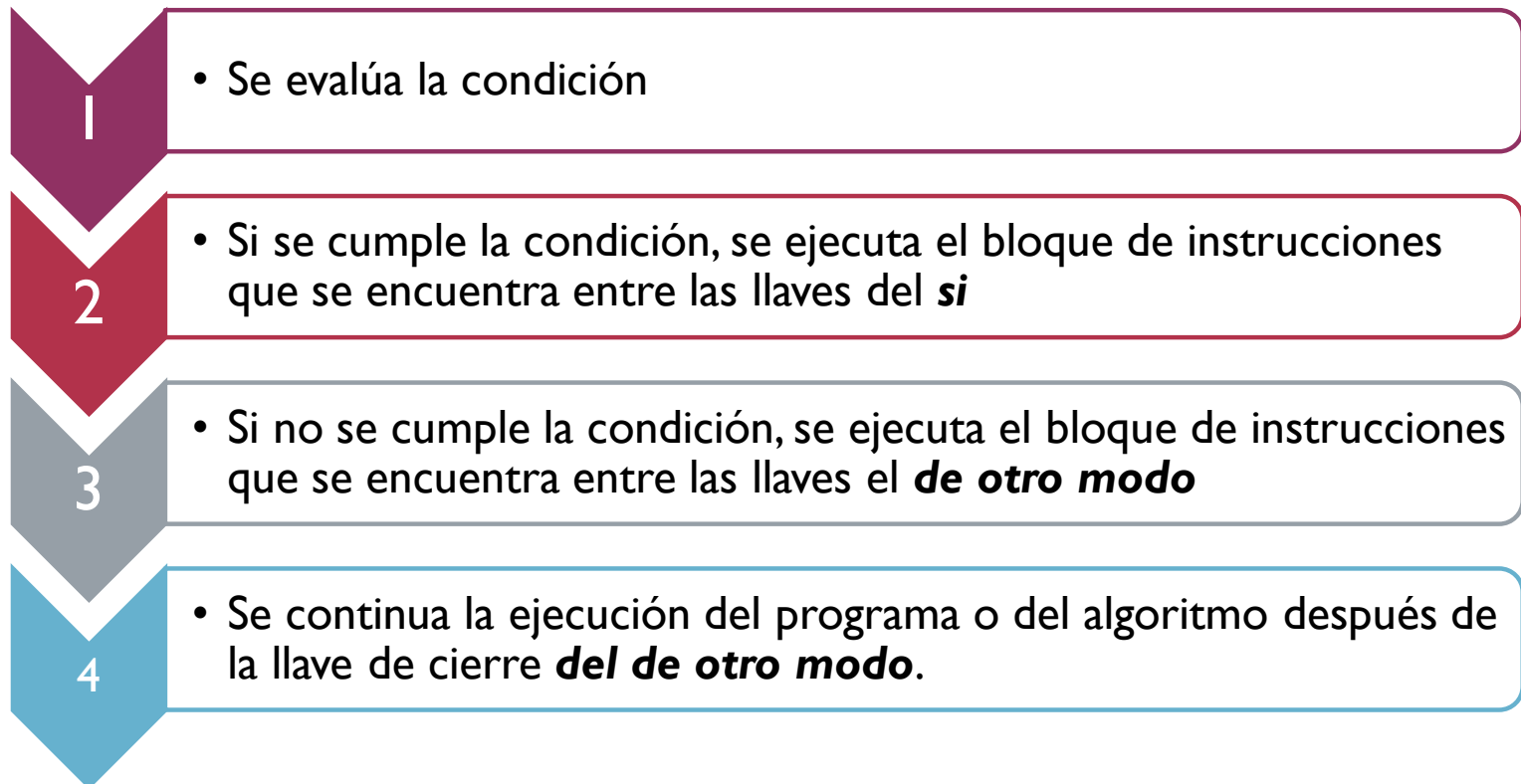
ESTRUCTURAS ALTERNATIVAS DOBLES

- Ejemplo:

```
si (numero > 0)  
{  
    imprimir (“El número es positivo”);  
}  
de otro modo  
{  
    imprimir (“El número es negativo”);  
}
```

ESTRUCTURAS ALTERNATIVAS DOBLES

- ¿Cómo trabaja la estructura alternativa dobles?



ESTRUCTURAS ALTERNATIVAS MÚLTIPLES

- Son aquellas que permiten evaluar más de una condición para dar respuestas a un problema.
- Formato:

```
si (condición 1)
{
    //bloque de instrucciones
}
de otro modo
    si (condición2)
    {
        //bloque de instrucciones
    }
de otro modo
{
    //bloque de instrucciones
}
```

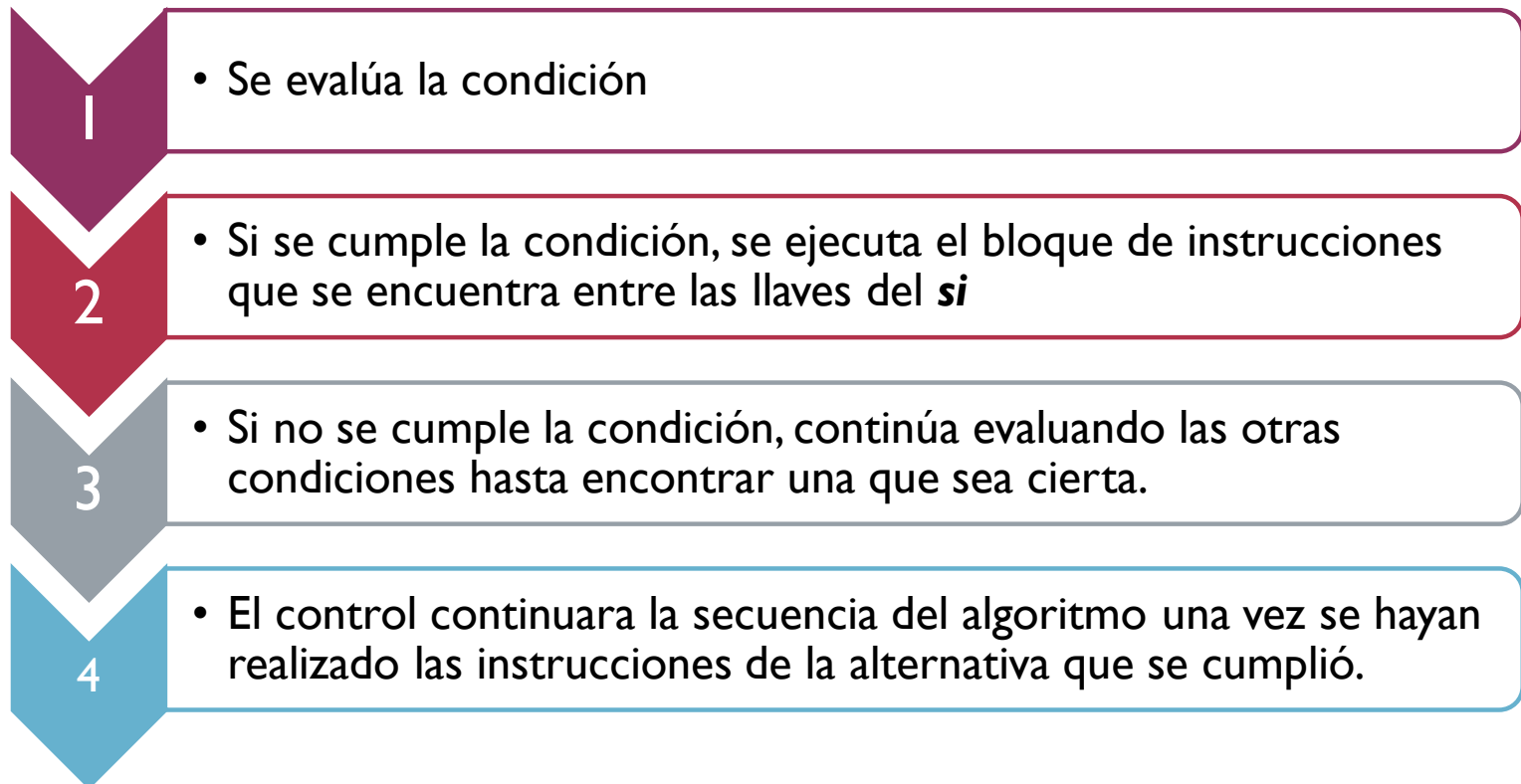
ESTRUCTURAS ALTERNATIVAS MÚLTIPLES

- Ejemplo:

```
si (numero > 0)
{
    imprimir ("El número es positivo");
}
de otro modo
    si (numero < 0)
    {
        imprimir ("El número es negativo");
    }
de otro modo
    {
        imprimir ("El número es cero");
    }
```


ESTRUCTURAS ALTERNATIVAS MÚLTIPLES

■ ¿Cómo trabaja la estructura alternativa dobles?





¿PREGUNTAS?



I. Elabore un programa que imprima un mensaje cuando una persona es mayor de edad.

Algoritmo MayorDeEdad

```
{  
    //Bloque Declarativas de Variables  
    entero edad;  
  
    //Bloque de Instrucciones  
    imprimir ("Introduzca su edad: ");  
    leer (edad);  
    si (edad >= 18)  
    {  
        imprimir ("Es mayor de edad");  
    }  
}
```

I. Elabore un programa que dada la calificación de un estudiante le indique con un mensaje si ha aprobado o reprobado el curso.

Algoritmo Aprobado

```
{  
    //Bloque Declarativas de Variables  
    entero calif;  
  
    //Bloque de Instrucciones  
    imprimir (“Introduzca la calificación obtenida ”);  
    leer (calif);  
    si (calif >= 71)  
    {  
        imprimir (“Ha aprobado el curso”);  
    }  
    de otro modo  
    {  
        imprimir (“Ha reprobado el curso”);  
    }  
}
```

I. Elabore un programa que permita determinar el mayor de tres números dados por el usuario. El programa debe imprimir cuál es el mayor.

Algoritmo NumeroMayor

```
{  
    //Bloque Declarativas de Variables  
    entero num1, num2, num3;  
  
    //Bloque de Instrucciones  
    imprimir ("Introduzca tres números enteros:");  
    leer (num1, num2, num3);  
    si (num1 > num2 Y num1 > num3)  
    {  
        imprimir ("El mayor es: ", num1);  
    }  
    de otro modo  
        si (num2 > num1 Y num2 > num3)  
        {  
            imprimir ("El mayor es: ", num2);  
        }  
        de otro modo  
        {  
            imprimir ("El mayor es: ", num3);  
        }  
}
```



¿PREGUNTAS?

