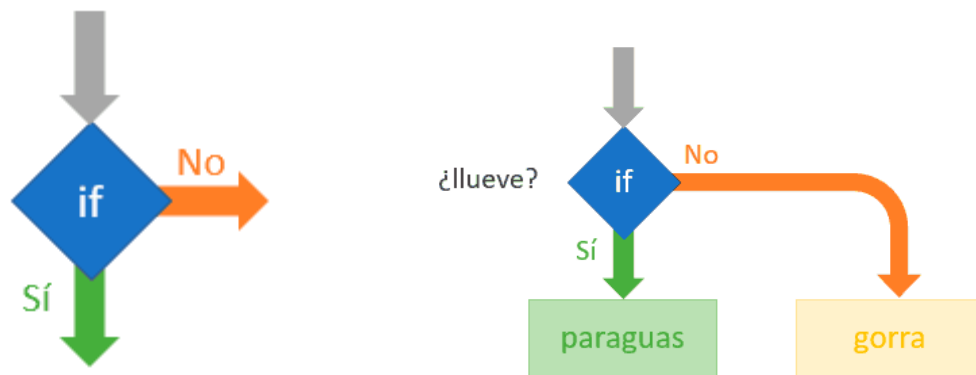


La **lógica condicional** es un concepto fundamental en la programación que nos permite tomar decisiones y ejecutar diferentes bloques de código según si se cumple o no una determinada condición. En pseudocódigo, usamos estructuras condicionales para representar esta lógica de manera clara y sencilla, sin tener que preocuparnos por la sintaxis específica de un lenguaje de programación real. Esto hace que el pseudocódigo sea una excelente herramienta para planificar y diseñar algoritmos antes de empezar a codificarlos.

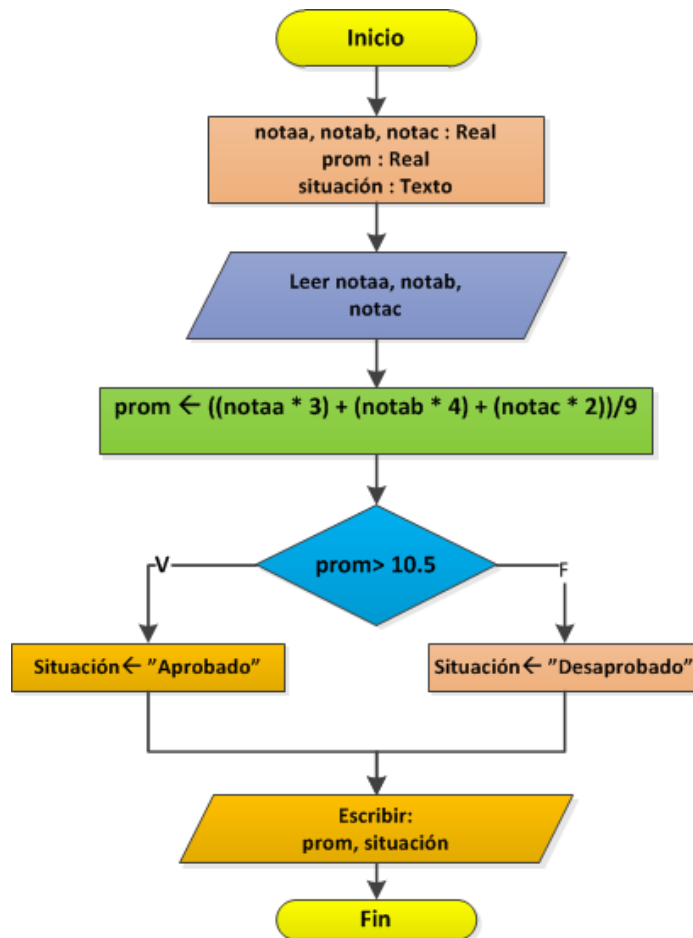
Ventajas de las estructuras condicionales

Las estructuras condicionales, como el si-entonces-sino (if-else) y el según-sea (switch-case), nos permiten crear programas más flexibles y dinámicos. Sus principales ventajas son:

- **Toma de decisiones:** Habilitan a un programa para responder de manera diferente a diferentes entradas. Por ejemplo, un programa podría verificar la edad de un usuario y mostrar un mensaje distinto si es mayor o menor de 18 años.
- **Control del flujo:** Permiten controlar el orden en el que se ejecutan las instrucciones, haciendo que el programa siga un camino u otro según la condición.
- **Validación de datos:** Ayudan a verificar que los datos ingresados por el usuario sean válidos antes de procesarlos, evitando errores.



Hacer un programa en pseudocódigo que permita ingresar tres notas para un alumno, la nota A que tiene una importancia de 3 créditos, la nota B tiene 4 Créditos, y la nota C tiene sólo 2 créditos; se debe de calcular el promedio teniendo en cuenta las notas y al final si el alumno tiene un promedio igual o mayor a 10.5 mostrar “Aprobado”, en caso contrario “Desaprobado”.



Estructuras condicionales en pseudocódigo

Condicional si-entonces-sino (if-else)

Esta es la estructura más común y sencilla. Se utiliza cuando hay que elegir entre dos posibles caminos: uno si la condición es verdadera y otro si es falsa.

Sintaxis:

si (condición) Entonces

// Bloque de código a ejecutar si la condición es verdadera

Sino

// Bloque de código a ejecutar si la condición es falsa

FinSi

Ejemplo:

Supongamos que queremos determinar si un número es positivo o negativo.

```
1  Algoritmo VerificarNumero
2      Definir numero Como Entero
3
4      Escribir "Ingrese un numero:"
5      Leer numero
6
7      Si (numero > 0) Entonces
8          Escribir "El numero es positivo."
9      Sino
10         Escribir "El numero es negativo o cero."
11     FinSi
12
13 FinAlgoritmo
```

Condicional según-sea (switch-case)

Esta estructura es ideal para manejar múltiples opciones de forma más organizada que anidando varios si-entonces-sino. Se utiliza cuando una variable puede tener varios valores discretos y para cada valor se debe ejecutar un bloque de código diferente.

Sintaxis:

Según sea (variable) Hacer

Caso valor1:

```
// Bloque de código para valor1
```

Caso valor2:

```
// Bloque de código para valor2
```

De otro modo:

```
// Bloque de código si la variable no coincide con ningún caso
```

FinSegun

Ejemplo: Imagina que un programa debe mostrar un mensaje diferente según el día de la semana que ingrese el usuario.

```
1  Algoritmo DiaDeLaSemana
2      Definir dia Como Caracter
3
4      Escribir "Ingrese un dia de la semana (L, M, X, J, V, S, D):"
5      Leer dia
6
7      Segun (dia) Hacer
8          'L':
9              Escribir "Hoy es Lunes."
10         'M':
11             Escribir "Hoy es Martes."
12         'X':
13             Escribir "Hoy es Miercoles."
14         'J':
15             Escribir "Hoy es Jueves."
16         'V':
17             Escribir "Hoy es Viernes."
18         'S':
19             Escribir "Hoy es Sabado."
20         'D':
21             Escribir "Hoy es Domingo."
22  De otro modo:
23      Escribir "Ese no es un dia valido."
24  FinSegun
25
26  FinAlgoritmo
--
```

Ejercicios de estructuras condicionales

El presente documento contiene 9 enunciados para practicar estructuras condicionales en pseudocódigo. Los primeros 4 están resueltos, transcriben el código en PseInt mientras analizan la lógica de la solución y el resto deben darle solución para que practiquen.

Enunciados

1. Escribe un algoritmo que lea tres números enteros y determine cuál de ellos es el menor. El algoritmo debe mostrar el número menor.

```
1  Algoritmo numeroMenor
2      Definir numUno, numDos, numTres como real;
3
4      Escribir "Ingrese un primer numero: ";
5      Leer numUno;
6
7      Escribir "Ingrese un segundo numero: ";
8      Leer numDos;
9
10     Escribir "Ingrese un tercer numero: ";
11     Leer numTres;
12
13     si numUno < numDos y numUno < numTres Entonces
14         Escribir " El numero menor es: ",numUno;
15     SiNo
16         si numDos < numTres Entonces
17             Escribir "El numero menor es: ",numDos;
18         SiNo
19             Escribir " El numero menor es: ",numTres;
20         FinSi
21     FinSi
```

2. Diseña un algoritmo que calcule el sueldo semanal de un trabajador, dado el número de horas trabajadas y el pago por hora. Si el trabajador trabaja más de 40 horas, las

horas extras se pagan a 1.5 veces la tarifa normal. El algoritmo debe mostrar el sueldo semanal.

```
1  Algoritmo sueldoSemanal
2      Definir numHoras, pagoHoras, total, calHorasExtras como real;
3
4      Escribir "Ingrese el numero de horas ";
5      Leer numHoras;
6
7      Escribir "Ingrese el valor de la hora";
8      Leer pagoHoras;
9
10     si numHoras > 40 Entonces
11         |   calHorasExtras = (numHoras - 40);
12         |   total = 40*pagoHoras + calHorasExtras*pagoHoras*1.5;
13     SiNo
14         |   total = numHoras * pagoHoras;
15     FinSi
16     Escribir "Total a pagar en la semana: $", total;
17
18 FinAlgoritmo
```

3. **Determinar si un número es múltiplo de 3 y de 5:** Crea un algoritmo que lea un número entero y determine si es múltiplo de 3 y de 5 simultáneamente. El algoritmo debe mostrar un mensaje indicando el resultado.

```

1  Algoritmo multiplos
2      Definir num como real;
3
4      Escribir "Ingrese un numero: ";
5      Leer num;
6
7      si num % 3 == 0 y num % 5 == 0 Entonces
8          Escribir "El numero es multiplo de 3 y de 5";
9      SiNo
10         Escribir "El numero no es multiplo de 3 y de 5";
11     FinSi
12
13 FinAlgoritmo
14

```

4. Convertir una calificación numérica a letra (A, B, C, D, F):

Escribe un algoritmo que lea una calificación numérica (0-100) y la convierta a una calificación en letra según la siguiente escala: 90-100: Excelente, 80-89: Sobresaliente, 70-79: Aceptable, 60-69: Insuficiente, 0-59: Deficiente. El algoritmo debe mostrar la calificación en letra.

```

1  Algoritmo calificacion
2      Definir nota como real;
3
4      Escribir "Ingresa calificacion (0-100)";
5      Leer nota;
6
7      si nota > 0 y nota ≤ 59 Entonces
8          Escribir "Deficiente";
9      SiNo
10         si nota ≥ 60 y nota ≤ 69 Entonces
11             Escribir "Insuficiente";
12         SiNo
13             si nota ≥ 70 y nota ≤ 79 Entonces
14                 Escribir "Aceptable";
15             SiNo
16                 si nota ≥ 80 y nota ≤ 89 Entonces
17                     Escribir "Sobresaliente";
18                 SiNo
19                     si nota ≥ 90 y nota ≤ 100 Entonces
20                         Escribir "Excelente";
21                     SiNo
22                         Escribir "ERROR:calificación fuera de rango";
23                     FinSi
24                 FinSi
25             FinSi
26         FinSi
27     FinSi
28
29 FinAlgoritmo

```

5. Crea un algoritmo que pida un número entero y realice las siguientes verificaciones:

1. Primero, determina si el número es positivo, negativo o cero.
2. Si el número es diferente de cero, verifica si es par o impar.

El algoritmo debe mostrar un mensaje para cada uno de los siguientes casos:

- El número es positivo y par.
- El número es positivo e impar.
- El número es negativo y par.
- El número es negativo e impar.

- El número es cero.
6. Crea un algoritmo que lea el monto de una compra y calcule el descuento aplicable. Si el monto es mayor o igual a 1000, se aplica un 10% de descuento. El algoritmo debe mostrar el monto final con el descuento aplicado.
 7. Escribe un algoritmo que lea las longitudes de los tres lados de un triángulo y determine si el triángulo es equilátero (tres lados iguales), isósceles (dos lados iguales) o escaleno (todos los lados diferentes). El algoritmo debe mostrar el tipo de triángulo.
 8. Desarrolla un algoritmo que lea el código de un producto y su precio, y calcule el descuento aplicable según el siguiente esquema: código A: 10%, código B: 20%, código C: 30%. El algoritmo debe mostrar el precio final con el descuento aplicado.
 9. Diseña un algoritmo que calcule el salario de un empleado, dado su salario base y sus años de antigüedad. Si tiene 10 años o más de antigüedad, recibe una bonificación del 10% sobre el salario base; si tiene menos de 10 años, la bonificación es del 5%. El algoritmo debe mostrar el salario final con la bonificación incluida.