

INF (ICI) 116 Fundamentos de Programación

Tips - Unidad 1

Operadores y Representaciones (Diagrama de flujo y Pseudocódigo)

Operadores Aritméticos

Operador	Operación	Uso	Resultado
+	Adición	4 + 6	10
-	Sustracción	45 - 102	-57
*	Multiplicación	9 * 17	153
/	División (real)	32 / 3	10.666666666666666
//	División (entera)	32 // 3	10
**	Potencia	5 ** 3	125
MOD	Resto de la división entera	32 MOD 3	2
		¿(N MOD 2) = 0?	
DIV	Cociente de la división entera	32 DIV 3	10

Operadores Relacionales

Se evalúa la relación entre dos elementos o expresiones. La evaluación proporciona un valor **VERDADERO** o **FALSO**

Operador	Descripción	Uso	Resultado
<	Menor que	1 < 0	Falso (F)
<=	Menor o igual que	-1 <= 0	Verdadero (V)
>	Mayor que	1 > 0	Verdadero (V)
>=	Mayor o igual que	-1 >= 0	Falso (F)
<>	distinto	0 <> 0	Falso (F)
=	igual	-1 = -1	Verdadero (V)

Operadores Lógicos

Un operador lógico es un componente que permite crear proposiciones lógicas. El orden de precedencia es como sigue:

NO	Negación
Y	Conjunción
O	Disyunción

Recordar la tabla de verdad de p , q para Y, O y NO.

p	q	$(p \text{ Y } q)$	$(p \text{ O } q)$	NO p
V	V	V	V	F
V	F	F	V	
F	V	F	V	V
F	F	F	F	

Proposición	Resultado
“Está lloviendo Y NO está mojada la calle”	F
falso Y NO está mojada la calle	
falso Y NO falso	
falso Y verdadero	
falso	

Precedencia de operadores

A. Aritméticos

1. ()
2. potencias
3. *, /
4. MOD, DIV
5. +, -

B. Relacionales

1. ()
2. <, <=, >, >=
3. =, <>

C. Lógicos

1. ()
2. NO
3. Y
4. O

Ejercicios

Para $N = 2$

$$15 + 59 * 75 / 9 < 2 ** 3 ** 2 \text{ Y } (15 + 59) * 75 \text{ MOD } N = 1$$

Resolución

Ayuda: $2 ** 3 ** 2$ es equivalente a 2^{3^2} es equivalente a $2^{(3^2)} = 512$




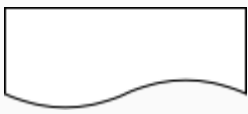
```
15 + 59 * 75 / 9 < 2 ** 3 ** 2 Y (15 + 59) * 75 MOD N = 1
15 + 59 * 75 / 9 < 2 ** 3 ** 2 Y      74      * 75 MOD N = 1
15 + 59 * 75 / 9 < 2 ** 9      Y      74      * 75 MOD N = 1
15 + 59 * 75 / 9 <      512      Y      74      * 75 MOD N = 1
15 +      4425 / 9 <      512      Y      74      * 75 MOD N = 1
15 + 491.6666666 <      512      Y      74      * 75 MOD N = 1
15 + 491.6666666 <      512      Y      5550      MOD N = 1
15 + 491.6666666 <      512      Y      5550      MOD 2 = 1
15 + 491.6666666 <      512      Y              0      = 1
      506.6666666 <      512      Y              0      = 1
      Verdadero      Y              0      = 1
      Verdadero      Y      Falso
      Falso
```

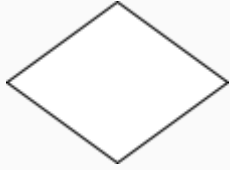
Para N = -1

```
-15 ** 2 * 132 DIV 11 <> N 0 -2 ** 3 > (32 MOD 22)
```

Simbología de diagramas de flujo.

Para la representación de soluciones, utilizando Diagrama de Flujos, debemos conocer la simbología:

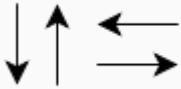
Símbolo	Representación
	Inicio o fin del diagrama
	Entrada de datos (lectura, interacción de entrada)
	Operación/Expresión
	Mensaje (escritura, interacción de salida)



Condición
(sólo tiene dos posibles valores: V o F)



Selector de múltiples opciones



Flujos

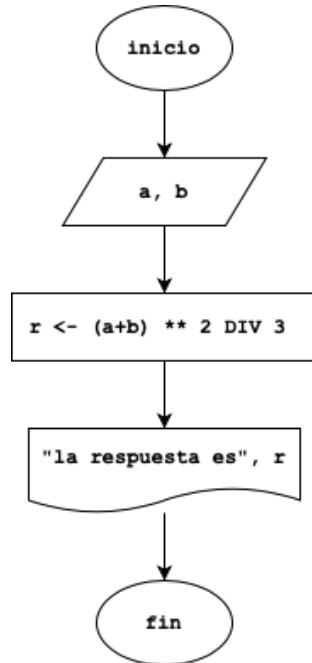
Palabras reservadas en Pseudocódigo.

Para la representación de soluciones, utilizando Pseudocódigo, debemos conocer las palabras reservadas:

Palabra	Descripción	Uso
hacer	Detalla una operación.	hacer <code>a <- 5 ** 4</code>
leer	Permite "capturar" datos del usuario (interacción de entrada)	leer(<code>algo</code>)
escribir	Permite mostrar datos al usuario (interacción de salida)	escribir(" <code>algo</code> ")
si-entonces	Representa una condición a evaluar.	si <code>a > 10</code> entonces ... <i>hacer algo</i> fin si
si-entonces-sino	Describe una alternativa cuando la condición a evaluar es falsa	si <code>a > 10</code> entonces ... <i>hacer algo</i> sino ... <i>hacer otra cosa</i> fin si
mientras-hacer	Ciclo indeterminado con una condición a evaluar. Termina cuando la condición evaluada es falsa.	mientras <code>a > 10</code> hacer ... <i>hacer algo mientras a sea mayor que 10</i> fin mientras
para-desde-hasta-hacer	Ciclo determinado. Termina cuando se alcanza el valor final.	para <code>a</code> desde <code>1</code> hasta <code>10</code> hacer ... <i>hacer algo 10 veces</i> fin para

Ejemplos de representaciones: Diagrama de flujo y Pseudocódigo

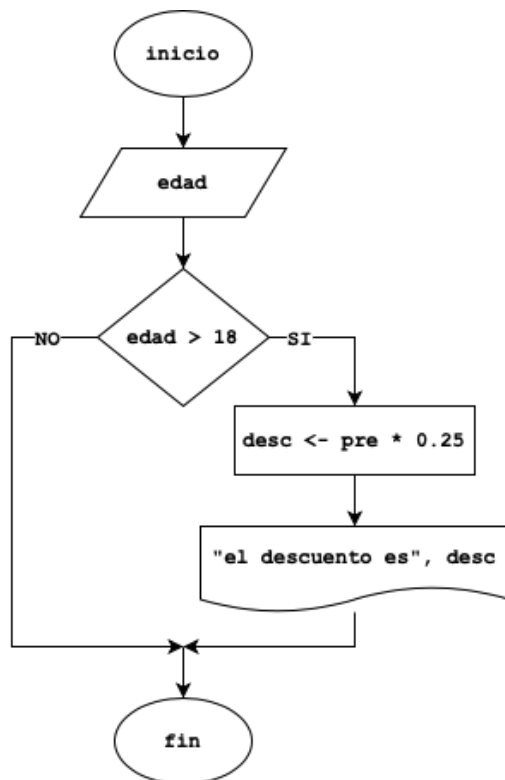
Estructura general de las representaciones



```

programa calculo_simple
{... comentarios del programa ...}
{... definición de variables ...}
inicio
  leer(a, b)
  hacer r <- (a+b) ** 2 DIV 3
  escribir("la respuesta es", r)
fin
  
```

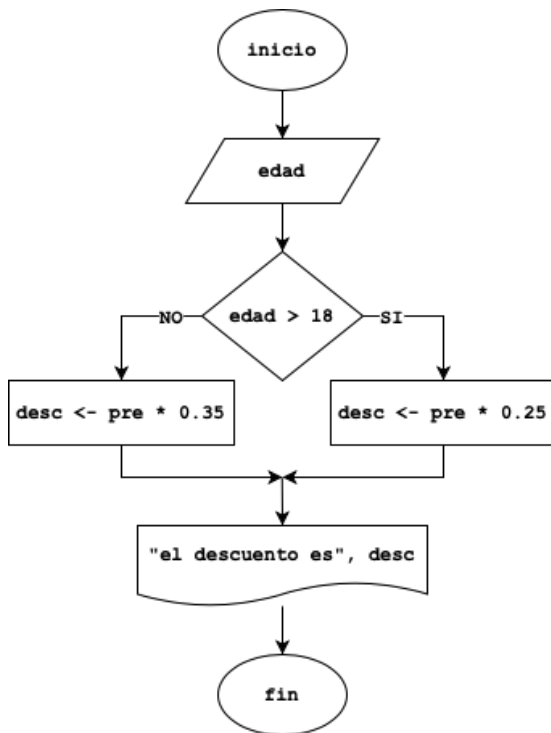
Estructura selectiva simple



```

programa mayoria_edad_1
{... comentarios del programa ...}
{... definición de variables ...}
inicio
  leer(edad)
  si edad > 18 entonces
    hacer desc <- pre * 0.25
    escribir("el descuento es", desc)
  fin si
fin
  
```

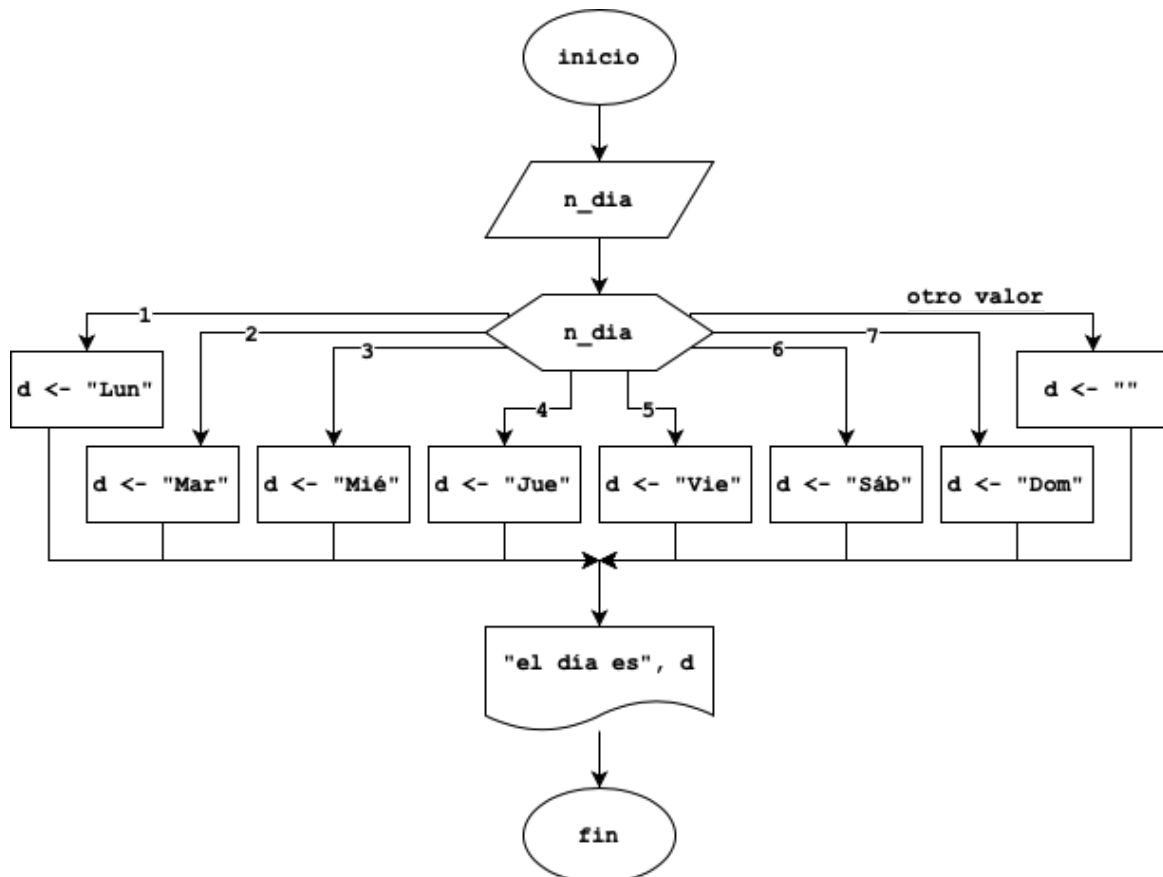
Estructura selectiva doble



```

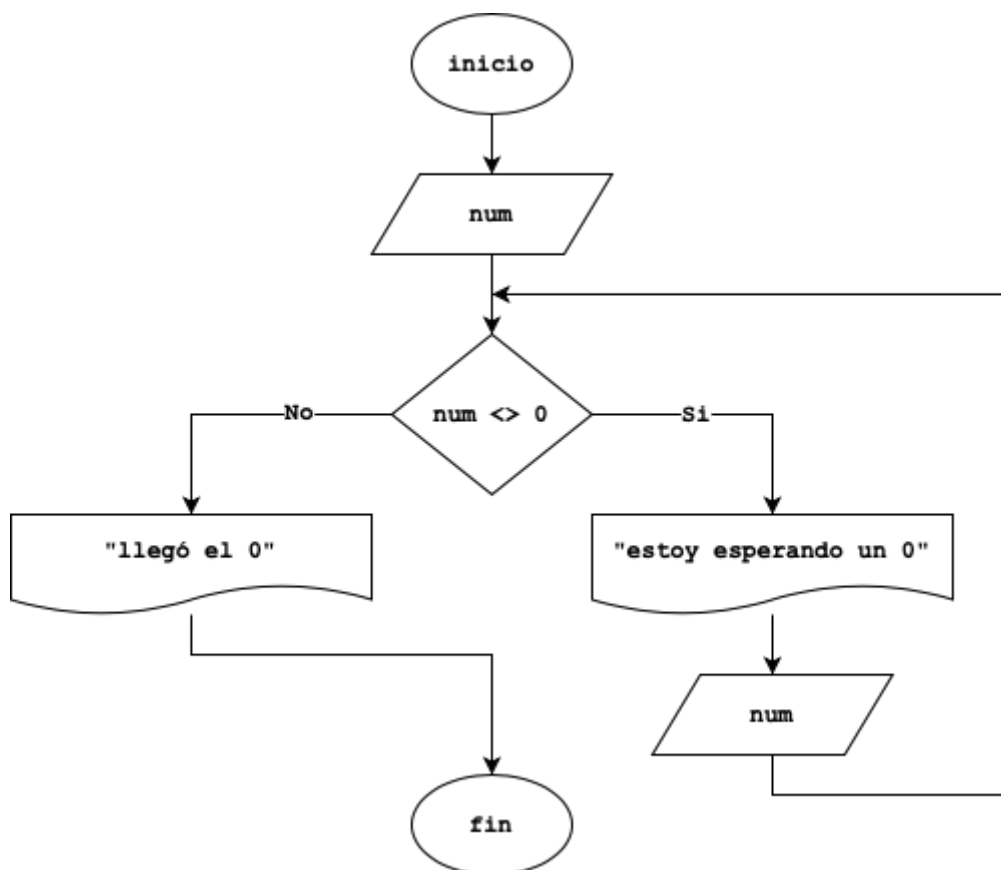
programa mayoria_edad_2
{... comentarios del programa ...}
{... definición de variables ...}
inicio
  leer(edad)
  si edad > 18 entonces
    hacer desc <- pre * 0.25
  sino
    hacer desc <- pre * 0.35
  fin si
  escribir("el descuento es", desc)
fin
  
```

Estructura selectiva múltiple



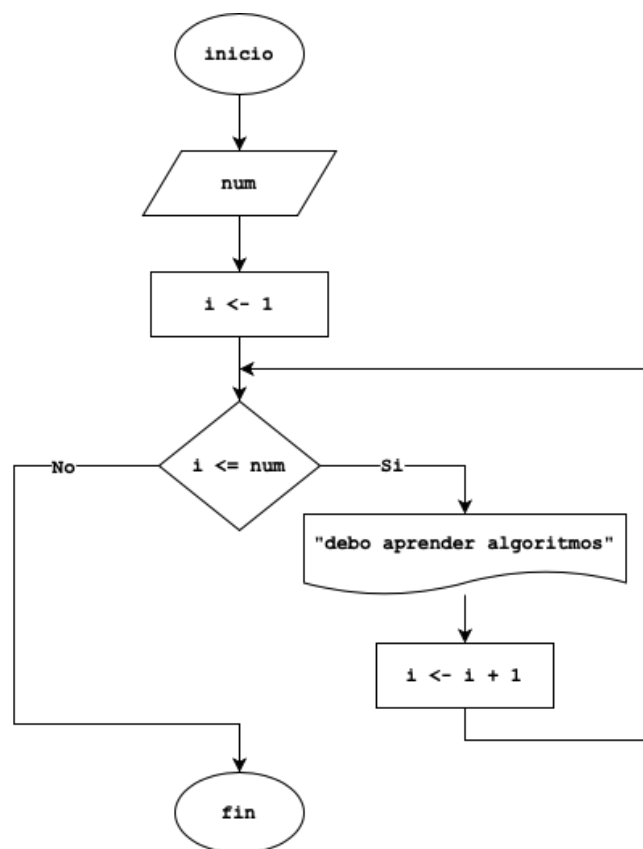
```
programa dias_semana
{... comentarios del programa ...}
{... definición de variables ...}
inicio
  leer(n_dia)
  si n_dia igual
    1. hacer d <- "Lun"
    2. hacer d <- "Mar"
    3. hacer d <- "Mie"
    4. hacer d <- "Jue"
    5. hacer d <- "Vie"
    6. hacer d <- "Sáb"
    7. hacer d <- "Dom"
  sino
    hacer d <- ""
  fin si
  escribir("el día es", d)
fin
```

Estructura repetitiva indeterminada (mientras)



```
programa dias_semana
{... comentarios del programa ...}
{... definición de variables ...}
inicio
  leer(num)
  mientras num <> 0 hacer
    escribir("estoy esperando un 0")
    leer(num)
  fin mientras
  escribir("llegó el 0")
fin
```

Estructura repetitiva determinada (para)



```
programa dias_semana
{... comentarios del programa ...}
{... definición de variables ...}
inicio
  leer(num)
  hacer i <- 1
  para i desde 1 hasta num hacer
    escribir("debo aprender a programar")
    hacer i <- i + 1
  fin para
fin
```