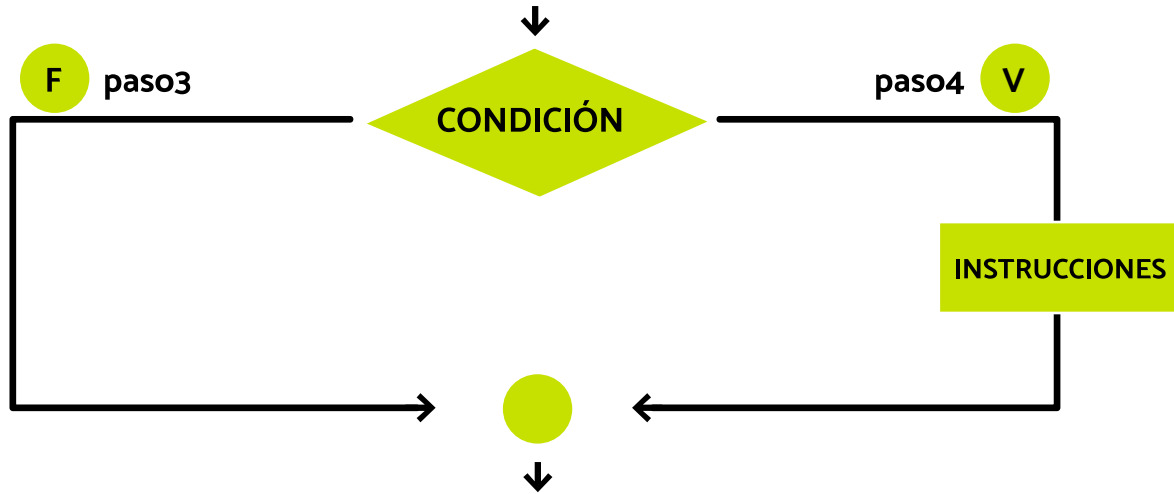


estructura de decisión si SIMPLE /IF

Se utiliza para evaluar una condición, si esta se cumple se ejecuta un determinado grupo de instrucciones, en caso de que no se cumpla termina el bloque condicional.

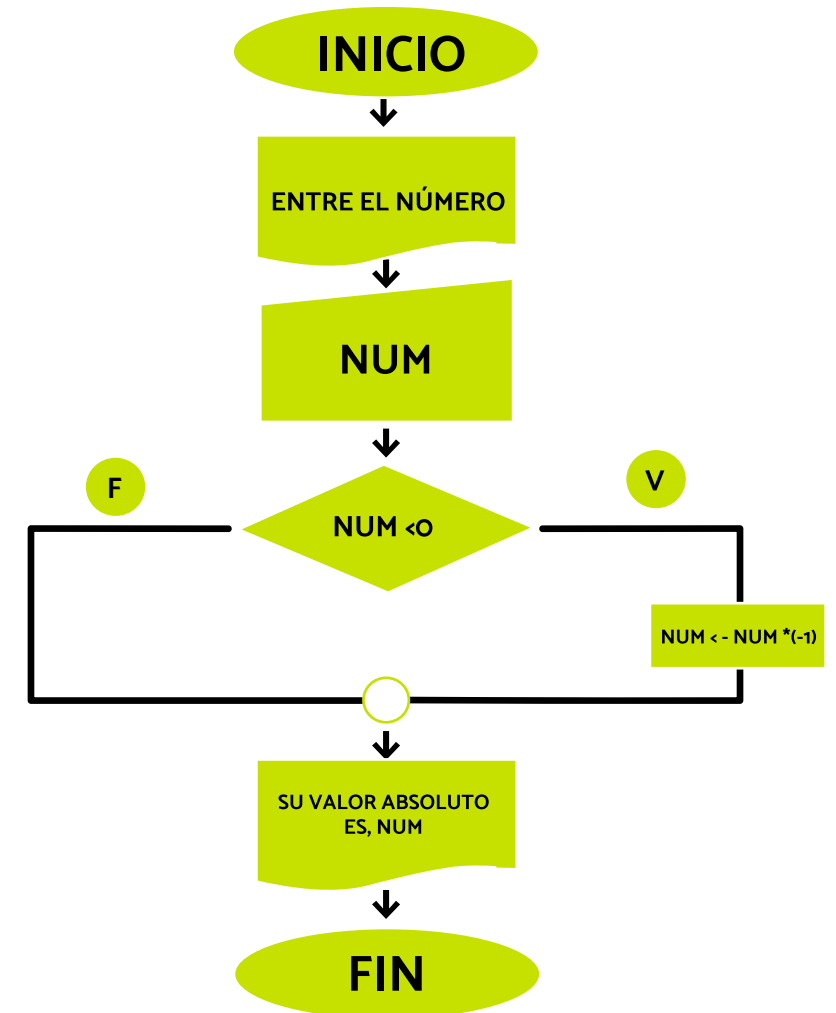


Paso 1, evalúa la condición, en caso de que sea verdadera va al paso #2; en caso de que la condición sea falsa va al paso #3.

Paso 2, ejecuta las instrucciones de la Parte Verdadera y sigue el flujo del algoritmo.

Paso 3, al no cumplir la condición, termina la estructura condicional y continua secuencia del programa

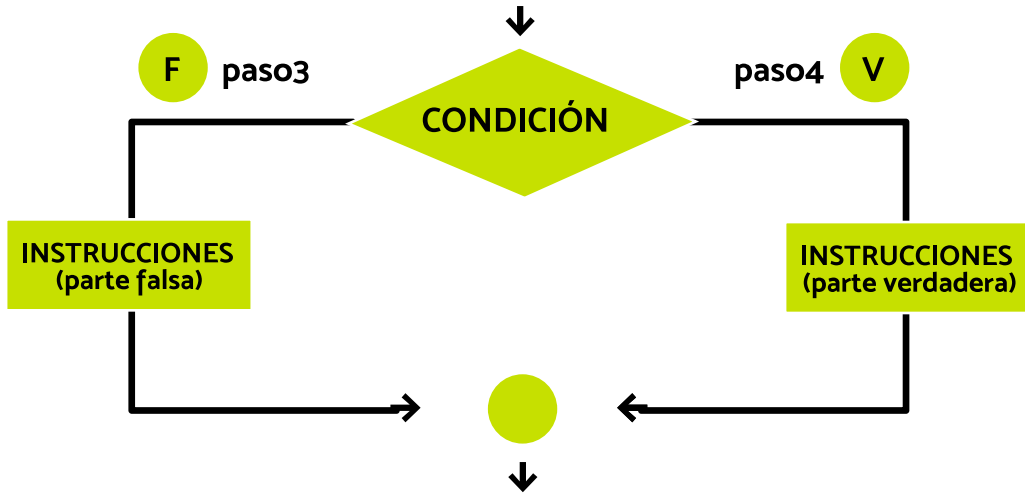
diagrama de flujo





**estructura de decisión si
COMPUESTA IF/ELSE**

*** Se utiliza para evaluar una condición, si ésta se cumple se ejecuta un determinado grupo de instrucciones, en caso que no se cumpla termina el bloque condicional.**

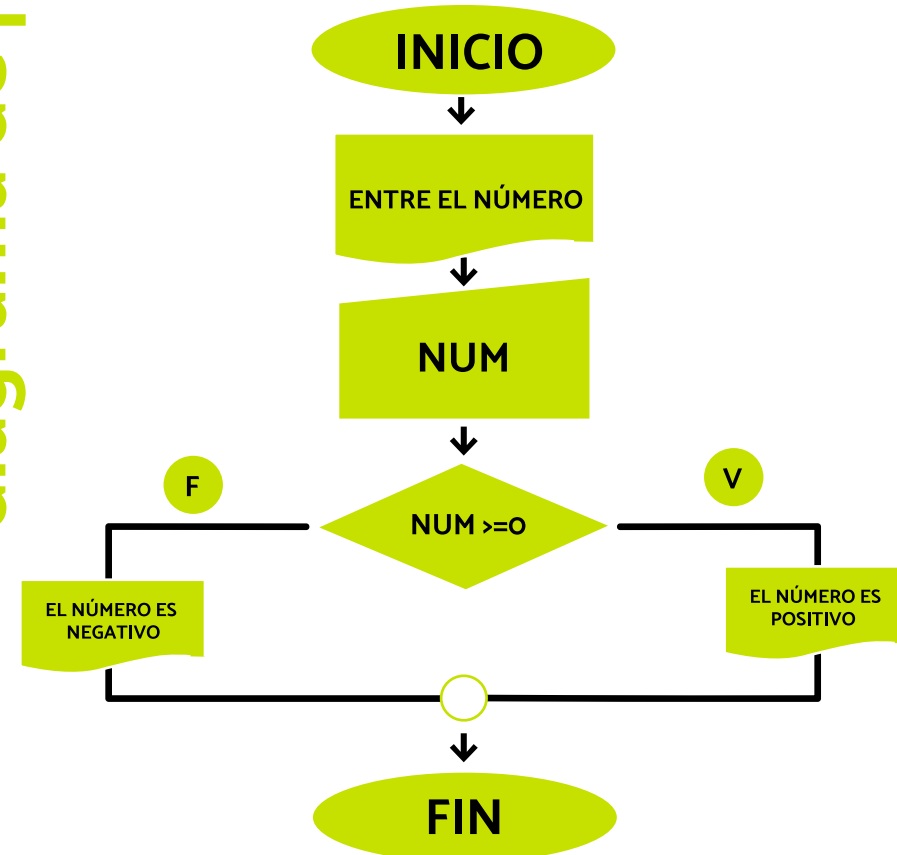


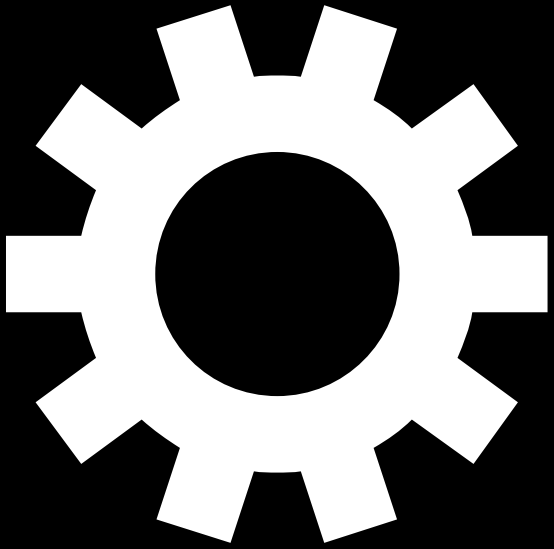
Paso 1, evalúa la condición, en caso de que sea verdadera va al paso #2; en caso de que la condición sea falsa va al paso #3.

Paso 2, ejecuta las instrucciones de la Parte Verdadera y sigue el flujo del algoritmo.

Paso 3, ejecuta las instrucciones de la Parte Falsa y sigue el flujo del algoritmo.

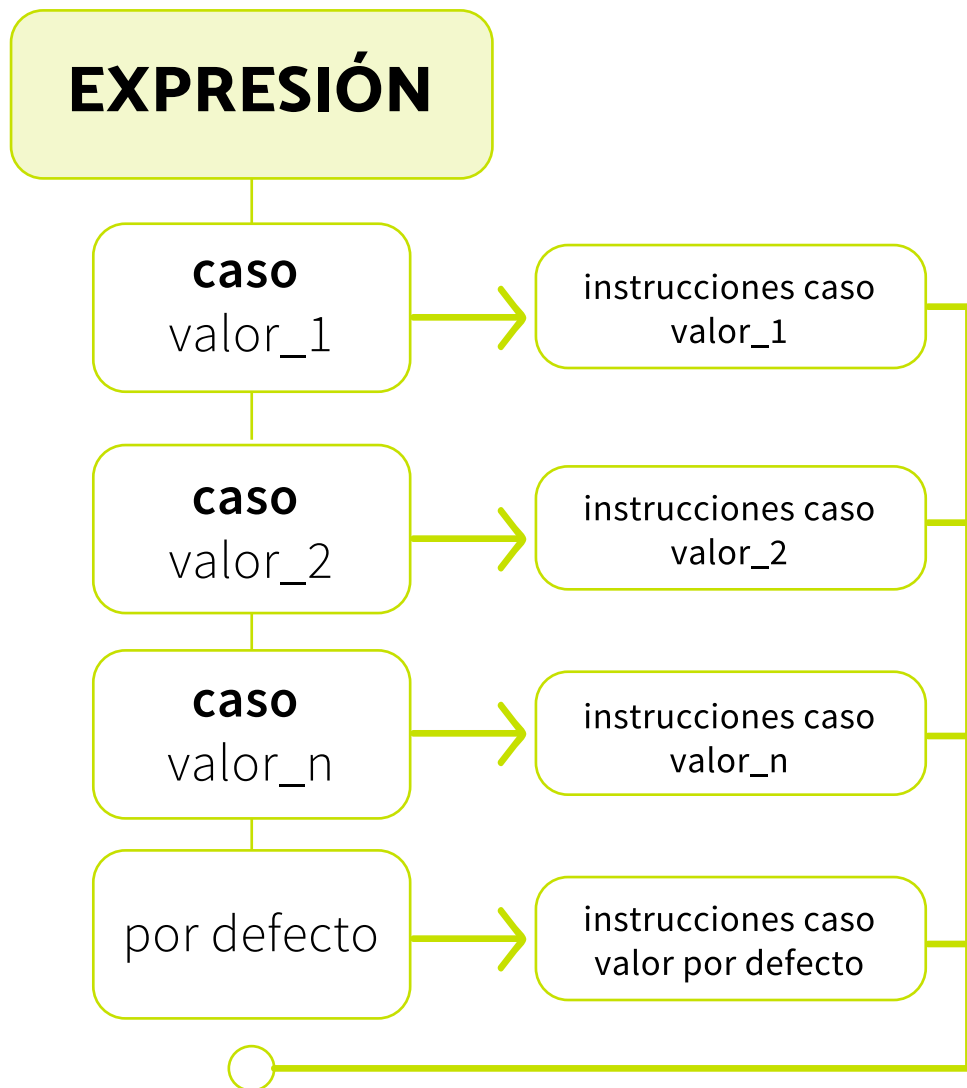
diagrama de flujo





**estructura de selección
por casos (Switch)**

⚙️ Se utiliza para agilizar la toma de decisiones múltiples, trabaja de la misma manera que lo harían sucesivos if anidados.



Evalúa una expresión de acuerdo a un dato ingresado y lo compara en distintas instancias (CASE), ejecuta un bloque de instrucciones de acuerdo a el cumplimiento de dicha condición. De no cumplir en ninguno de los casos, sale del bloque **CASE**.

Evalúa una expresión de acuerdo a un dato ingresado y lo compara en distintas instancias (CASE), ejecuta un bloque de instrucciones de acuerdo al cumplimiento de dicha condición. De no cumplir en ninguno de los casos, ejecuta las instrucciones por defecto, termina la estructura Switch y continua la secuencia del programa.

La acción **default** es usada para los valores que no correspondieron en casos anteriores, y puede aparecer sin "accionD()", e incluso, con el break al final.

Diseñamos el siguiente diagrama, que responde al algoritmo anterior.

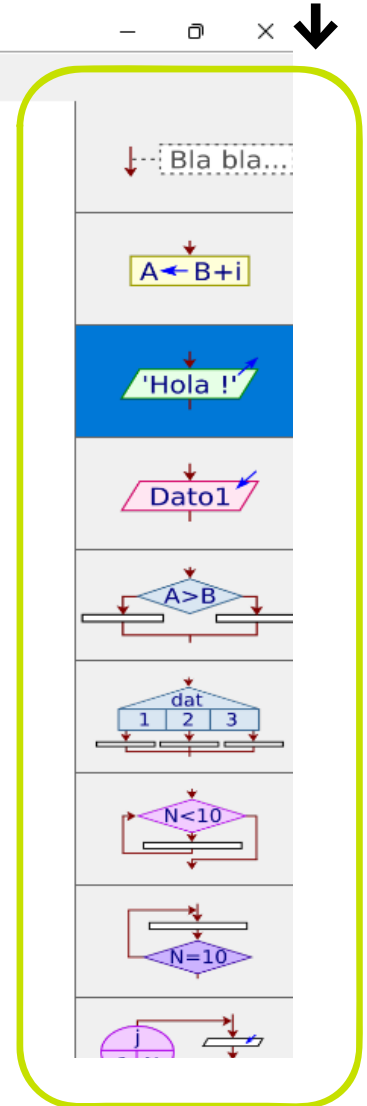
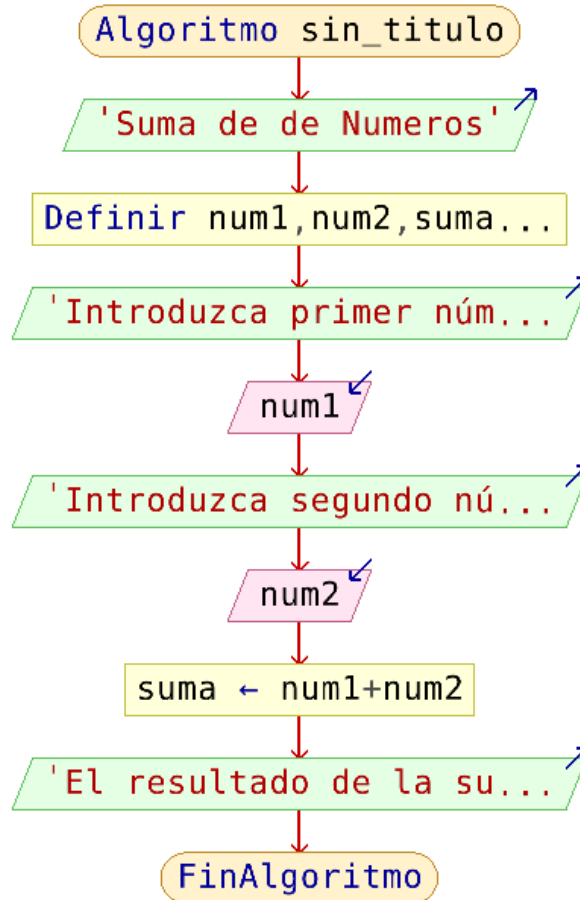
Lado derecho es donde elegimos el símbolo para agregar a nuestro diagrama.



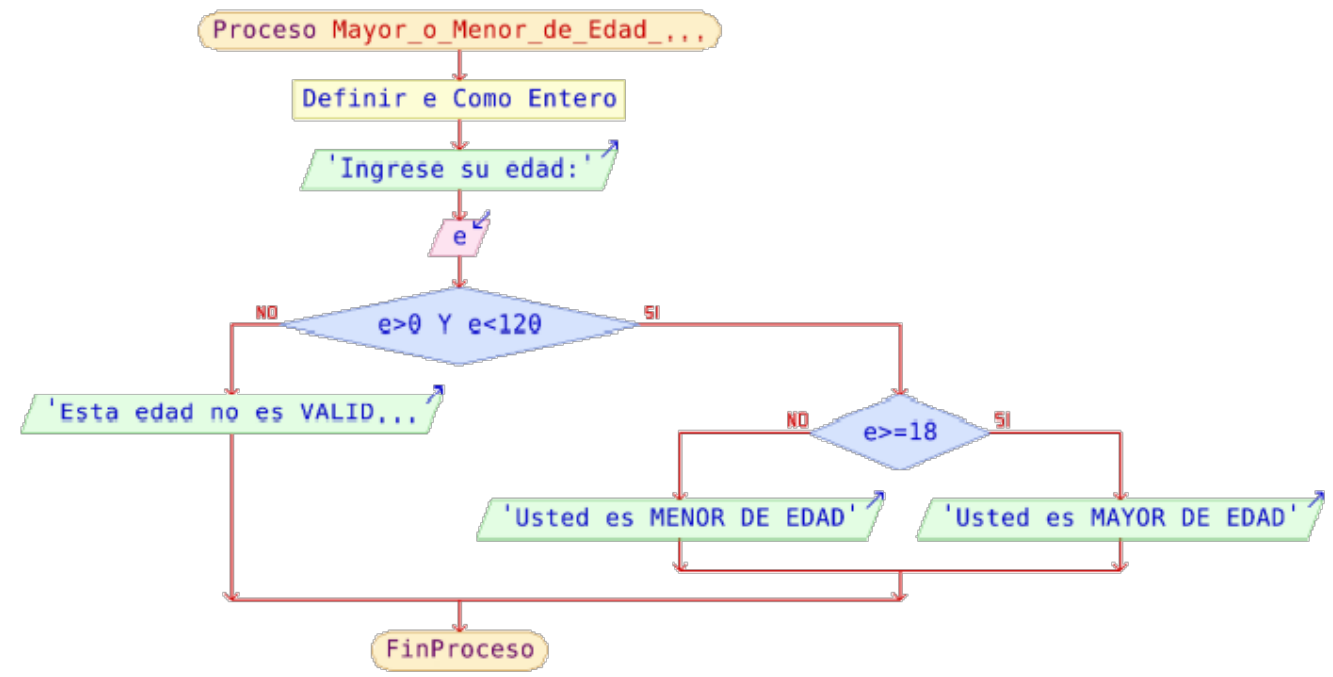
Una vez terminado lo ejecutamos

```
PSelnt - Ejecutando proceso SIN_TITULO
*** Ejecución Iniciada. ***
Suma de de Numeros
Introduzca primer número (entero):
> 5
Introduzca segundo número (entero):
> 6
El resultado de la suma es: 11
*** Ejecución Finalizada. ***
```

Vemos como aparece una ventana donde nos pide números enteros y luego nos da un resultado.



⚙ Diseñar un algoritmo que permita determinar si una persona es mayor de edad o menor de edad, determine y muestre a través de un mensaje.
Validar la entrada a través de un mensaje, sabiendo que una persona no puede tener menos de 0 años o más de 120.



⚙ Desarrolle un algoritmo que permita leer tres valores y almacenarlos en las variables num1, num2, num3 respectivamente. El algoritmo debe imprimir cual es el mayor .
Recuerde constatar que los tres valores introducidos por el teclado sean valores distintos



PSelnt

lógica de programación

Acciones repetitivas o interactivas

Botón Mientras Permite realizar cierta acción determinada mientras se cumpla dicha condición.

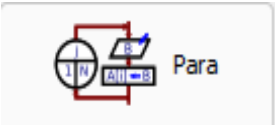


Ej. Mientras num != 0 Hacer

Leer num; cant_num<-cant_num+1;

FinMientras

Botón Para

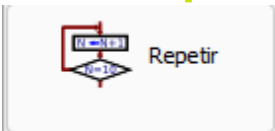


Ejecutar un bloque de código y lo repite una serie de veces, según la condición expresada inicialmente.

Ej. Para i <- 1 Hasta 10 Con Paso 1 Hacer

Escribir i; FinPara

Botón Repetir



Repite una series de acciones hasta que se de cierta condición.

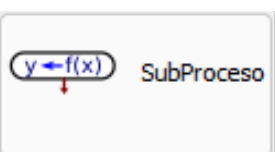
Dicha acción la determina el operador =, que hace que salga del bucle.

Ej. Repetir Leer num;

cant_num<-cant_num+1;

Hasta Que num = 0

Botón Sub Procesos



Permite añadir Funciones/SubProcesos al programa.

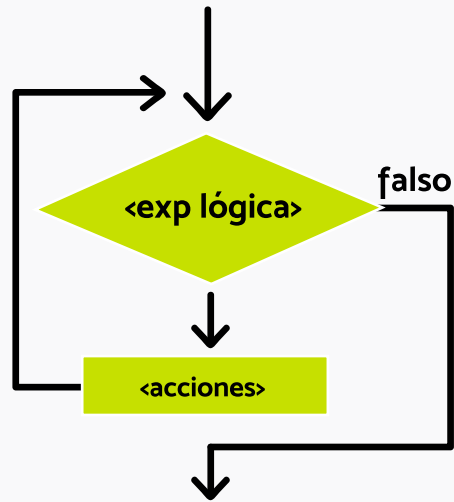
Ej. SubProceso ImprimirResultado(x)

Escribir "El resultado es: ",x;

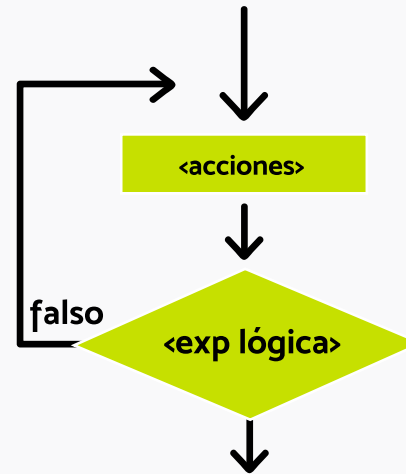
FinSubProceso

	WHILE	DO WHILE	FOR
USO	cuando no sabemos el número de interacciones	cuando no sabemos el número de interacciones	cuando si sabemos el número de interacciones
NÚMERO DE EJECUCIONES	0 o más veces	1 o más veces	0 o más veces
APLICACIONES	cuando la condición de terminación no es numérica, por ejemplo, una función boolean , tambien es útil para lectura de archivos	mostrar un menú, mostrar mensajes de error mientras el usuario no seleccione una opción válida	contar números, recorrer estructuras de datos (veremos esto más adelante)
FRECUENCIA DE USO	frecuente	poco frecuente	muy frecuente

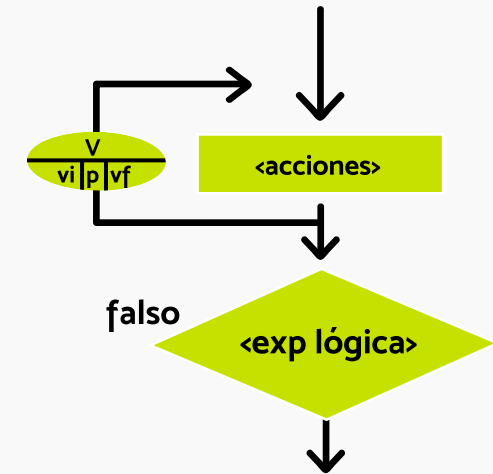
mientras - hacer



repetir - hasta



para



```

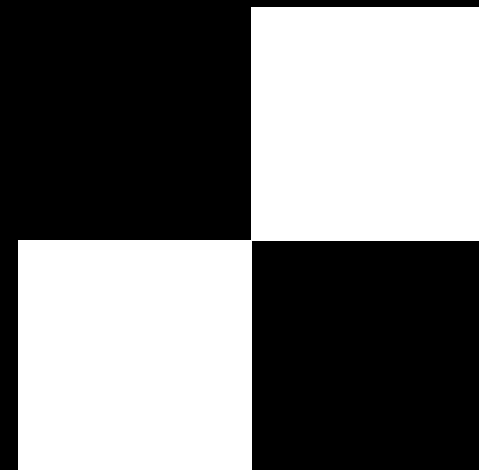
1  Proceso cliclo_repita_y_repita
2    Definir n,1 Como Entero;
3    Escribir "Cantidad de repeticiones:";
4    Leer n;
5    i=1;
6    Mientras i<=n Hacer
7      |   Escribir "Repetición número: ",1;
8      |   i=i+1;
9    Fin Mientras
10 FinProceso
  
```

```

1  Proceso cliclo_repita_y_repita
2    Definir i,n Como Entero;
3    i=0;
4    Escribir "Cantidad de repeticiones:";
5    Leer n;
6    Repetir
7      |   i=i+1;
8      |   Escribir "Repetición número: ",i;
9    Hasta Que i>=n
10 FinProceso
  
```

```

1  Proceso cliclo_repita_y_repita
2    Definir n,1 Como Entero;
3    Escribir "Cantidad de repeticiones:";
4    Leer n;
5    Para i<=1 Hasta n Con Paso 1 Hacer
6      |   Escribir "Repetición número: ",1;
7    Fin Para
8  FinProceso
  
```



tipos de datos

- El objetivo de cualquier programa es procesar datos. Estos datos tienen que ser de un tipo.



TIPO DE DATO		CARACTERÍSTICAS	EJEMPLO
numérico	entero	sin decimales - positivos y negativos	5, -1000.800, -2
	real	con decimales - positivos y negativos	8.5, -9.865, 3.1416
alfa numérico	caracter	puede ser un dígito, una letra o un símbolo especial no se pueden realizar operaciones aritméticas con ellos	L, R, 5, \$
	cadena de caracteres	combinación de caracteres alfabéticos especiales y numéricos deben ser delimitados por comillas	“Calle #23 - A”
lógicos	booleanos	utilizados para dar alternativas a ciertas condiciones	verdadero falso



variables

Una variable es donde se guardan (y se recupera) datos que se utilizan en un programa, y puede cambiar en el transcurso de dicho programa.

Cuando escribimos código, las variables se utilizan para:

- Guardar datos y estados.
- Asignar valores de una variable a otra.
- Representar valores dentro de una expresión matemática.
- Mostrar valores por pantalla.
- Todas las variables deben ser de un tipo de datos, ya sea un dato de tipo primitivo, como un número o texto, o un dato abstracto, como un objeto que se ha creado.

El valor que le damos a las variables se llama asignación.

clasificación

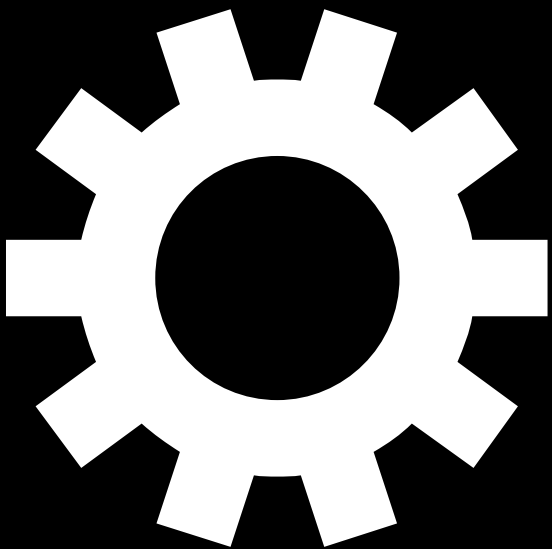
Simple: consiste en pasar un valor constante a una variable. Dos ejemplos: $a = 14$; $a = 15$.

Contador: sirve para verificar el número de veces que se realiza un proceso. Dos ejemplos: $a = a + 1$; $a = a + 1$

Acumulador: se utiliza como un sumador en un proceso. Dos ejemplos: $a = a + b$; $a = a + b$

De trabajo: puede recibir el resultado de una operación matemática que involucre muchas variables.

Dos ejemplos: $a = c + b * 2 / 4$; $a = c + b * 2 / 4$



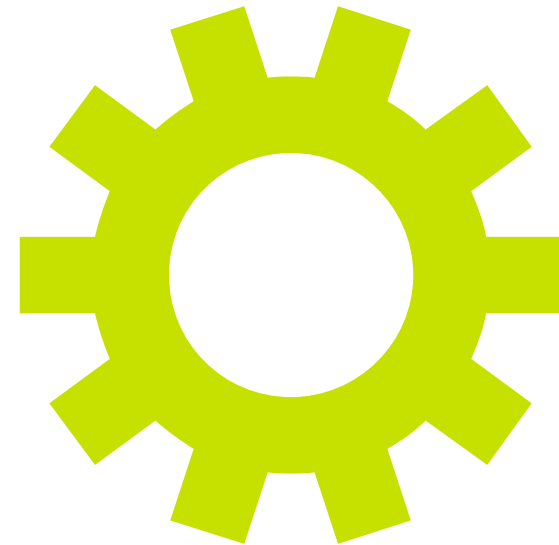
constantes

Es un valor que no puede ser alterado/modificado durante la ejecución de un programa, únicamente puede ser leído.

Una constante corresponde a una longitud fija de un área reservada en la memoria principal del ordenador, donde el programa almacena valores fijos.

No se puede cambiar durante la ejecución del programa, es decir, en tiempo de ejecución.

clasificación

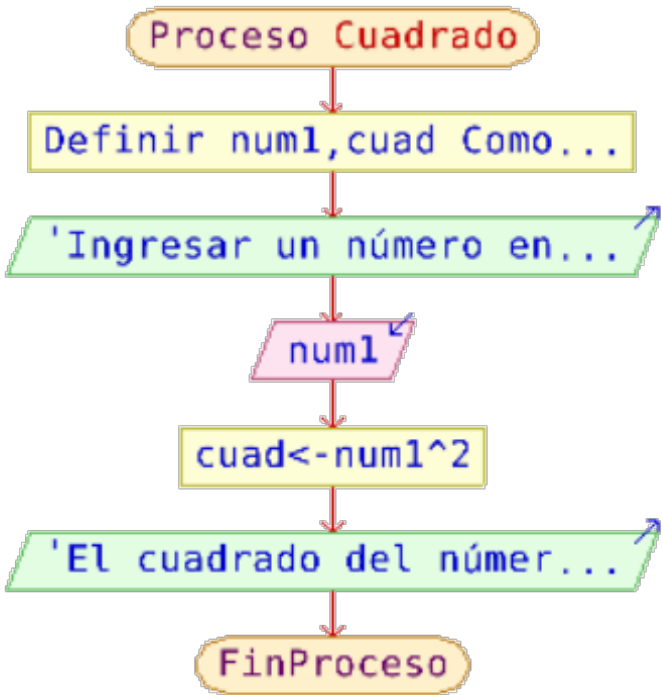




ejercicios PSeInt

🌐 Diseñar un algoritmo que permita ingresar un valor entero y calcule y muestre su cuadrado:

operadores algebraicos	+ (suma)
	- (resta)
	* (multiplicación)
	/ (división)
	^ (potencia)
	MOD (resto/modulo)
operadores lógicos	Y (conjunción)
	O (disyunción)
	NO (negación)
operadores relacionados	= (igual)
	<> (distinto)
	!= (distinto)
	< (menor)
	<= (menor o igual)
	> (mayor)
	>= (mayor o igual)



```
1 Algoritmo Cuadrado
2   Definir num1, cuad Como Entero;
3   Escribir "Ingresar un número entero:";
4   Leer num1;
5   cuad=num1^2;
6   Escribir "El cuadrado del npumero ingresado es igual a ",cuad;
7   FinAlgoritmo
```

funciones matemáticas

abs (valor absoluto)
trunc (valor truncado)
redon (valor redondeado)
rc (raiz cuadrada)
sen (seno)
cos (coseno)
tan (tangente)
asen (arcoseno)
acos (arcocoseno)
atan (arcotangente)
ln (logaritmo natural)
exp (func. exponencial)
azar (numero aleatorio)

funciones de cadena

Longitud
SubCadena
Concatenar
ConvertirANumero
ConvertirATexto
Mayusculas
Minusculas

- 🌐 Diseñar un algoritmo que permita ingresar un tiempo en segundos y permita determinar a cuantos minutos y segundos restantes equivale.

- 🌐 Diseñar un algoritmo que permita ingresar una temperatura en grados Celsius y convertirlos en grados Fahrenheit.

celsius/fahrenheit conversion

$$^{\circ}\text{C} = \frac{5}{9}^{\circ}\text{F} - 32$$

$$^{\circ}\text{F} = \frac{9}{5}^{\circ}\text{C} + 32$$

- 🌐 Desarrolle un algoritmo que le permita leer un valor de un radio de un círculo en centímetros, calcule y muestre el área.

$$A = \pi r^2$$