

PSEUDOCÓDIGO

El pseudocódigo es una herramienta de programación en la que las instrucciones se escriben en palabras similares al inglés o español, que facilitan tanto la escritura como la lectura de programas.

En esencia, el **pseudocódigo** se puede definir como un lenguaje de **especificaciones** de algoritmos.

Se considera un primer borrador, dado que el pseudocódigo tiene que traducirse posteriormente a un lenguaje de programación.

El pseudocódigo no puede ser ejecutado por una computadora.

La ventaja del **pseudocódigo** es que en su uso, en la planificación de un programa, el programador se puede concentrar en la lógica y en las estructuras de control y **no preocuparse de las reglas de un lenguaje específico**.





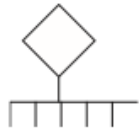









UN DIAGRAMA DE FLUJO

Un diagrama de flujo (flowchart) es una de las técnicas de representación de algoritmos más antigua

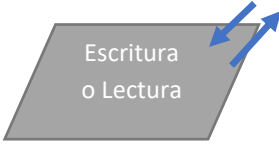
Un diagrama de flujo es un diagrama que utiliza los símbolos (cajas) y que tiene los pasos de algoritmo escritos en esas cajas unidas por flechas, denominadas líneas de flujo, que indican la secuencia en que se debe ejecutar.

Los símbolos más utilizados representan: Proceso, Decisión, conectores, fin, Entrada/salida dirección del flujo

SÍMBOLOS PRINCIPALES DE LOS DIAGRAMAS DE FLUJO

Símbolos principales	Función
	Terminal (representa el comienzo, “inicio”, y el final, “fin” de un programa. Puede representar también una parada o interrupción programada que sea necesario realizar en un programa).
	Entrada/Salida (cualquier tipo de introducción de datos en la memoria desde los periféricos, “entrada”, o registro de la información procesada en un periférico, “salida”).
	Proceso (cualquier tipo de operación que pueda originar cambio de valor, formato o posición de la información almacenada en memoria, operaciones aritméticas, de transferencia, etc.).
	Decisión (indica operaciones lógicas o de comparación entre datos —normalmente dos— y en función del resultado de la misma determina cuál de los distintos caminos alternativos del programa se debe seguir; normalmente tiene dos salidas —respuestas SÍ o NO— pero puede tener tres o más, según los casos).
	Decisión múltiple (en función del resultado de la comparación se seguirá uno de los diferentes caminos de acuerdo con dicho resultado).
	Conector (sirve para enlazar dos partes cualesquiera de un organograma a través de un conector en la salida y otro conector en la entrada. Se refiere a la conexión en la misma página del diagrama).
	Indicador de dirección o línea de flujo (indica el sentido de ejecución de las operaciones).
	Línea conectora (sirve de unión entre dos símbolos).
	Conector (conexión entre dos puntos del organograma situado en páginas diferentes).
	Llamada a subrutina o a un proceso predeterminado (una subrutina es un módulo independientemente del programa principal, que recibe una entrada procedente de dicho programa, realiza una tarea determinada y regresa, al terminar, al programa principal).
	Pantalla (se utiliza en ocasiones en lugar del símbolo de E/S).
	Impresora (se utiliza en ocasiones en lugar del símbolo de E/S).
	Teclado (se utiliza en ocasiones en lugar del símbolo de E/S).
	Comentarios (se utiliza para añadir comentarios clasificadores a otros símbolos del diagrama de flujo. Se pueden dibujar a cualquier lado del símbolo).

SALIDA Y ENTRADA DE DATOS

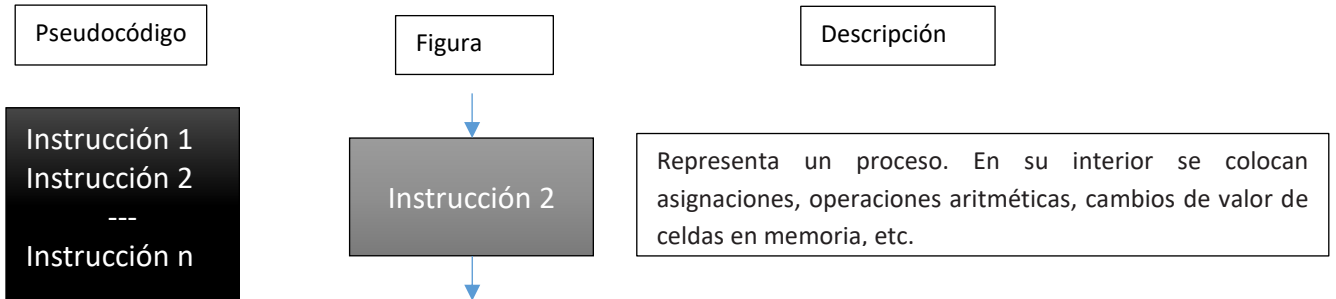
Pseudocódigo	Figura	Descripción
ESCRIBIR . . . LEER		Se utiliza para introducir los datos de entrada o Mostrar a la Salida (pantalla) datos. Expresa lectura o escritura ¹ .

¹ Dependiendo de la bibliografía consultada los símbolos para escritura y lectura pueden variar, aquí se muestran los más comúnmente usados.

ESTRUCTURAS DE CONTROL

SECUENCIA O PROCESO

Las instrucciones se ejecutan en el orden indicado por los índices; de arriba hacia abajo y una después de la otra

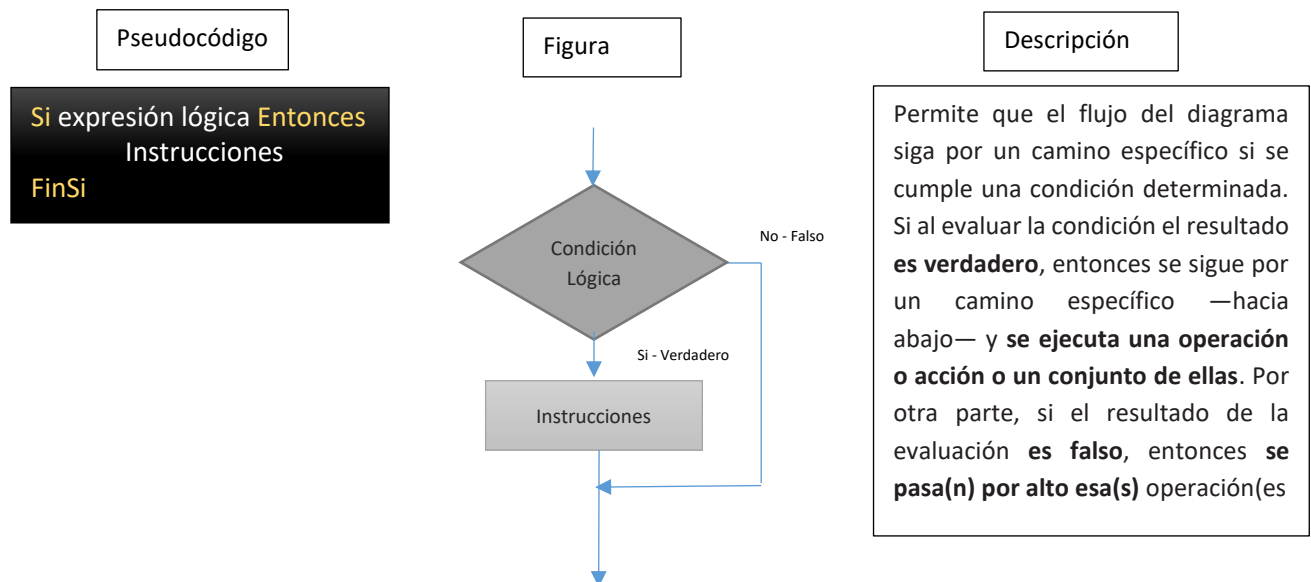


ESTRUCTURA SELECTIVA

Las **estructuras selectivas** permiten expresar las elecciones que se hacen durante la resolución del problema. Hay varios tipos de estructuras de este tipo

1. Selectiva simple
2. Selectiva doble (alternativa).
3. Selectiva múltiple.
4. Selectiva casos.

SELECTIVA SIMPLE



SELECTIVA DOBLE (ALTERNATIVA).

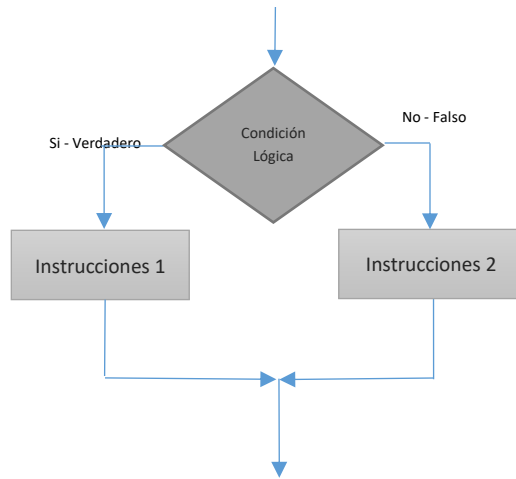
Pseudocódigo

```

SI expresión lógica ENTONCES
    Instrucciones 1
SINO
    Instrucciones 2
FINSI

```

Figura



Descripción

Permite que el flujo del diagrama se bifurque por dos ramas diferentes en el punto de la toma de decisión. Si al evaluar la condición el resultado es verdadero, entonces se sigue por un camino específico —el de la izquierda— y se ejecuta una acción determinada o un conjunto de ellas. Por otra parte, si el resultado de la evaluación es falso, entonces se sigue por otro camino -el de la derecha- y se realiza(n) otra(s) acción(es)

SELECTIVA MÚLTIPLE..

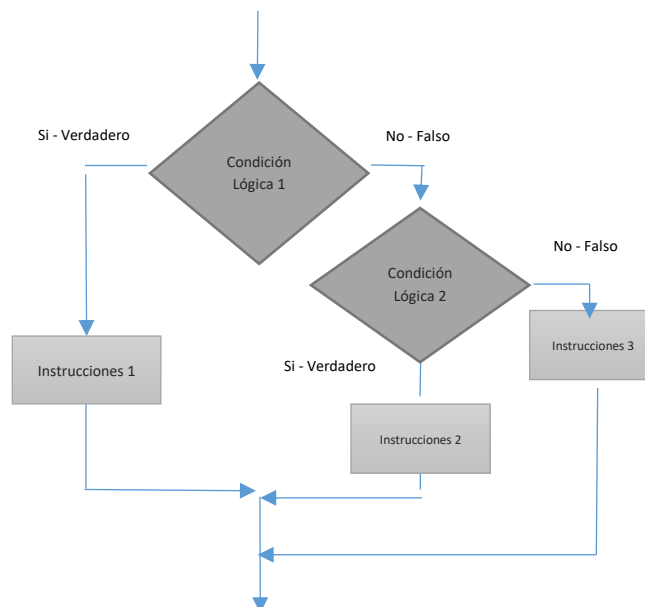
Pseudocódigo

```

SI expresión lógica 1 ENTONCES
    Instrucciones 1
SINO expresión lógica 2 ENTONCES
    Instrucciones 2
SINO expresión lógica 3 ENTONCES
    Instrucciones 3
SINO
    Instrucciones n
Fin si

```

Figura



Descripción

La **estructura selectiva múltiple** es usada para anidar condiciones lógicas mutuamente excluyentes

SELECTIVA CASOS

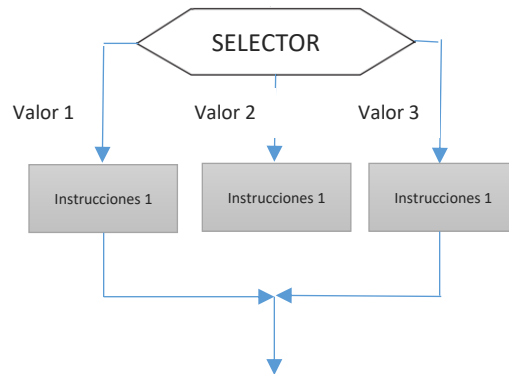
La **estructura selectiva múltiple-casos** se usa para comparar un mismo valor con varios valores.

Pseudocódigo

```

Seleccionar expresión
CASO valor1
    Instrucciones1
CASO valor2
    Instrucciones2
CASO valor3
    Instrucciones3
    .
    .
EN OTRO CASO
    Instruccionesn
Fin seleccionar
  
```

Figura



Descripción

Permite que el flujo del diagrama se bifurque por varias ramas en el punto de la toma de decisión. La elección del camino a seguir depende del contenido de la variable conocida como *selector*, **la cual puede tomar valores de un conjunto previamente establecido**

ESTRUCTURA ITERATIVA

BUCLE MIENTRAS

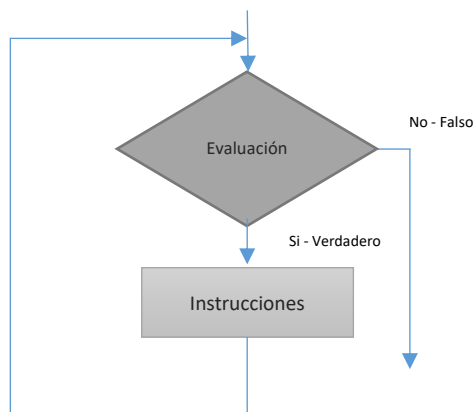
El bucle se repite mientras la condición *P sea cierta*, si al llegar por primera vez al bucle la condición es falsa, el cuerpo del bucle no se ejecuta *ninguna vez*

Pseudocódigo

```

MIENTRAS expresión lógica
entonces HACER
    Instrucciones
FIN MIENTRAS
  
```

Figura



Descripción

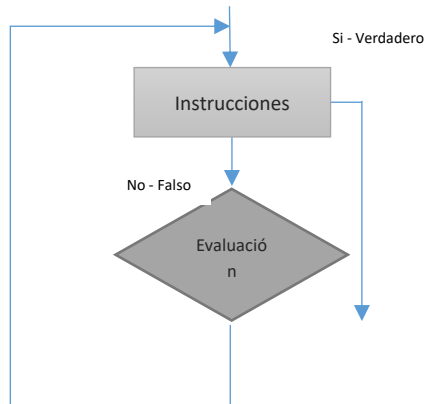
Permite repetir un conjunto de instrucciones. Sin embargo, el número de veces que se debe repetir depende de las proposiciones que contenga el ciclo. Cada vez que corresponde iniciar el ciclo se evalúa una condición, si ésta es verdadera (diferente de cero) se continúa con la ejecución, de otra forma se detiene

BUCLE REPETIR

Pseudocódigo

REPETIR
Instrucciones
HASTA QUE expresión lógica

Figura



Descripción

La estructura es adecuada cuando no sabemos el número de veces que se debe repetir un ciclo, pero conocemos que se debe ejecutar por lo menos una vez

Se ejecuta el conjunto de instrucciones una vez, y luego cada vez que corresponde iniciar nuevamente el ciclo se evalúa las condiciones, siempre al final del conjunto de instrucciones.

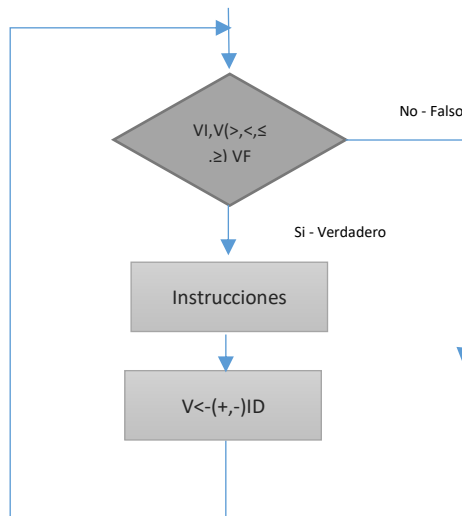
BUCLE PARA (CADA)

La estructura iterativa “para”, toma en cuenta la noción de **variable-contador**.

Pseudocódigo

PARA *i* de inicio hasta fin [*paso p*]
HACER
Instrucciones
FIN PARA

Figura



Descripción

La *i* recibe el valor inicio y se ejecutan las instrucciones; luego, *i* se incrementa el valor de *p* (el paso) y se reejecutan las instrucciones, si el valor de *i* es menor que fin

En resumen, las instrucciones se ejecutan 0 o varias veces, dependiendo de los valores de inicio, fin, y el paso

El valor de la variable *i* se puede usar al interior de las instrucciones, pero no puede ser modificado.

V Variable de control del ciclo
VI valor inicial
VF Valor Final
ID Incremento o decremento

BIBLIOGRAFÍA

Luis Joyanes Aguilar (2008), FUNDAMENTOS DE PROGRAMACIÓN Algoritmos, estructura de datos y objetos, cuarta edición, McGRAW-HILL

Osvaldo Cairo (2006) Fundamentos de Programación piensa en C, Prentice-Hall, primera edición.