

Un ordinograma debe reflejar:

- El comienzo del programa.
- Las operaciones.
- La secuencia en que se realizan.
- El final del programa.

En la representación de ordinogramas, es conveniente seguir las siguientes reglas:

- El comienzo del programa figurará en la parte superior del ordinograma.
- Los símbolos de comienzo y fin deberán aparecer una única vez.
- El flujo de operaciones será, siempre que sea posible, de arriba a abajo y de izquierda a derecha.
- Se debe guardar una cierta simetría en la representación de bifurcaciones y bucles, así como en el conjunto total del ordinograma.
- Se evitarán siempre los cruces de líneas utilizando conectores.
- El uso de comentarios estará restringido al mínimo imprescindible.
- Si en un ordinograma se ha separado una parte de otra por medio de un conector, las posibles conexiones que puedan aparecer desde esta última a la anterior, se harán igualmente con conectores, evitando el uso de líneas de flujo directas.

SÍMBOLOS BÁSICOS PARA LOS DIAGRAMAS DE FLUJO

Símbolos de operación

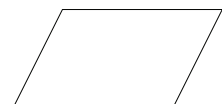
Terminal

Indica el principio o fin de un diagrama de flujo. En su interior debe introducirse las palabras **INICIO** o **FIN**. En todo diagrama de flujo deben aparecer únicamente dos símbolos de este tipo, uno indicando el principio del mismo, y otro indicando el fin.



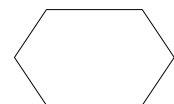
Datos

Indica el proceso de declaración de objetos dentro de un programa o de una rutina.



Preparación

Indica el proceso de inicialización de variables, dentro de un programa o de una rutina.



Proceso

Indica un cualquier proceso en general. En su interior se pueden englobar un conjunto de instrucciones.



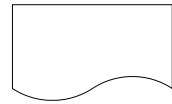
Proceso Predefinido

Proceso con un nombre, una subrutina, un módulo o en un subprograma.

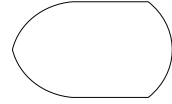


Documento Impreso

Representa cualquier documento que pueda ser leído por las personas, como puede ser un documento de impresora.

Pantalla

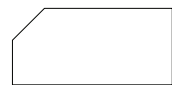
Representa la información que se visualiza por pantalla para ser leída por una persona.

Entrada manual

Representa la introducción manual de datos, ya sea mediante el teclado o mediante un lector de código de barras.

Tarjeta

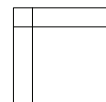
Se escriben datos mediante tarjetas, por ejemplo, tarjetas perforadas o formularios con marcado electro sensible.

**Almacenamiento de datos**Datos almacenados

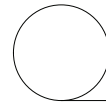
Cualquier tipo de dato almacenado, independientemente del soporte del mismo.

Almacenamiento interno

Cualquier dato almacenado en un dispositivo interno.

Datos secuenciales

Representa a los datos almacenados dentro de un dispositivo secuencial, como puede ser una cinta streamer

Datos de acceso directo

Representa a los datos almacenados dentro de un dispositivo de acceso directo o aleatorio, como puede ser un disco magnético u óptico.

**Símbolo de comentario**Comentario

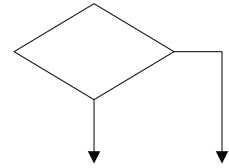
Dentro de los algoritmos podemos introducir comentarios como documentación interna de los mismos. No hay que abusar excesivamente de los mismos, pero es recomendable introducirlos en ocasiones.

Aquí dentro se introduce el comentario o aclaración del usuario

Símbolos de decisión

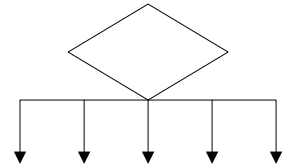
Bifurcación doble

Representa a una instrucción condicional con solo dos posibles salidas. En cada una de las salidas se debe especificar las palabras **SI** o **NO**.



Bifurcación múltiple

Representa a una instrucción de comparación múltiple. En cada una de las salidas se debe especificar el valor que representa la misma.



Líneas de flujo

Líneas de control

Representa como va evolucionando el programa durante su ejecución. Es decir, como se va pasando el control del programa entre los distintos bloques del mismo.



Símbolos de conexión

Reagrupamiento

Permite reagrupar un conjunto de líneas de control.



Conector dentro de la misma página

Permite romper las líneas de control para impedir que estas se crucen y obtener mayor claridad en el diagrama. Dentro del conector se especifica un número natural.



Conector entre distintas páginas

Cuando un diagrama no es posible representarlo todo en una hoja, y debemos continuar en otra utilizaremos este tipo de conector. Dentro del conector se especifica un natural.



PARTES PRINCIPALES DE UN PROGRAMA

Dentro del bloque de instrucciones de un programa podemos diferenciar tres partes fundamentales:

1. Entrada de datos.
2. Procesamiento de datos.
3. Salida de resultados.

En algunos casos, estas tres partes están perfectamente delimitadas; pero, en la mayoría, sus instrucciones quedan entremezcladas a lo largo del programa, si bien mantienen una cierta localización geométrica impuesta por la propia naturaleza de las mismas.

ENTRADA DE DATOS

La constituyen todas aquellas instrucciones que toman datos de un dispositivo externo, almacenándolos en la memoria central para que puedan ser procesados.

También se consideran dentro de este apartado las instrucciones de depuración de los datos de entrada, es decir, aquellas que se encargan de comprobar la corrección de los mismos.

PROCESO O ALGORITMO

Esta formada por las instrucciones que modifican los objetos a partir de su estado inicial, hasta el estado final, dejando estos disponibles en la memoria central.

SALIDA DE RESULTADOS

Conjunto de instrucciones que toman los datos finales de la memoria central y los envían a los dispositivos externos.

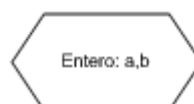
CLASIFICACION DE LAS INSTRUCCIONES

Las instrucciones pueden clasificarse en:

1. Instrucciones de declaración.
2. Instrucciones primitivas.
3. Instrucciones compuestas.
4. Instrucciones de control.

INSTRUCCIONES DE DECLARACION

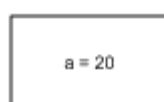
Se utilizan para indicar al procesador que reserve espacio en la memoria central para un objeto del programa, indicando asimismo su nombre, tipo y características.



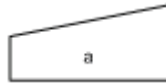
INSTRUCCIONES PRIMITIVAS

Son aquellas que ejecuta el procesador de modo inmediato. Las principales son:

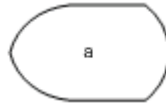
- Asignación.



- Entrada de datos.



- Salida de datos.



INSTRUCCIONES COMPUESTAS

Son aquellas que el procesador no puede ejecutar directamente, sino que realiza una llamada a un subprograma, subrutina, etc. Es decir, una serie de instrucciones simples encerradas en un único bloque.



INSTRUCCIONES DE CONTROL

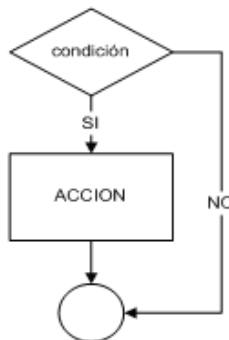
Son aquellas destinadas de controlar la ejecución de otras instrucciones. Podemos diferenciar entre:

- Instrucciones alternativas.
- Instrucciones repetitivas.

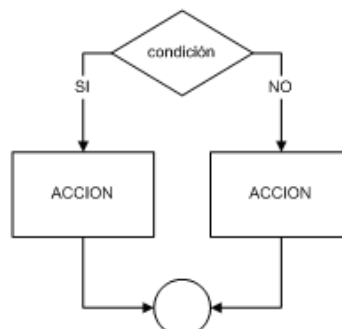
Instrucciones alternativas

Controla la ejecución de unas y otras instrucciones según una condición.

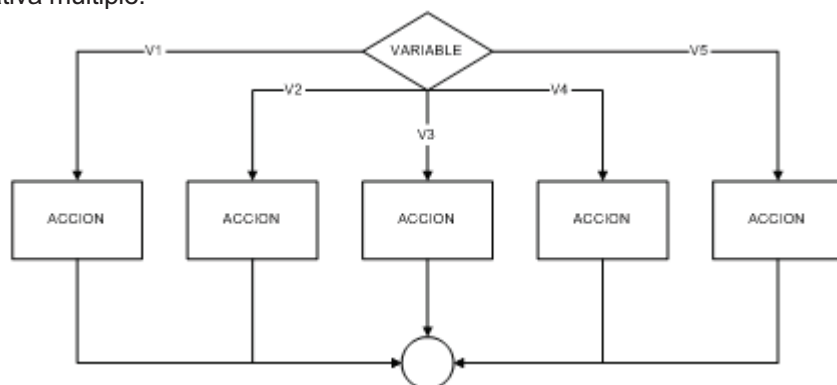
- Alternativa simple:



- Alternativa doble:



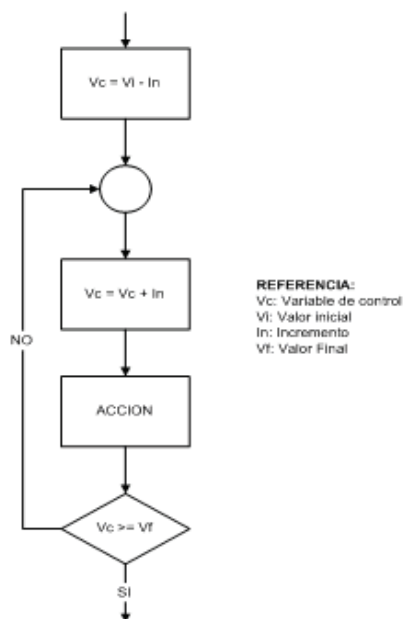
- Alternativa múltiple:



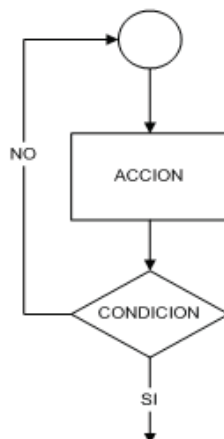
Instrucciones de repetición

Hace que se repita una o varias instrucciones un número determinado o indeterminado de veces.

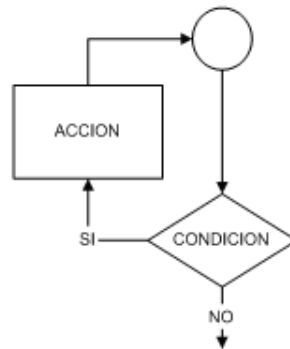
- El bucle PARA:



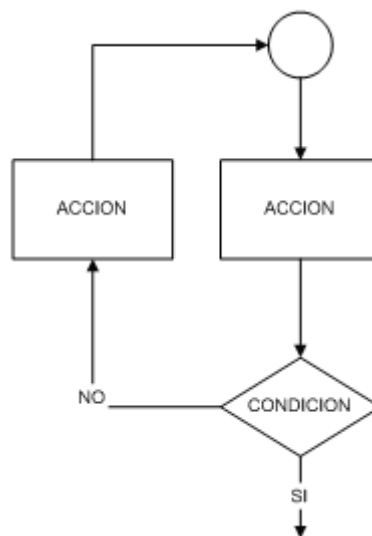
- El bucle REPETIR



- El bucle MIENTRAS



- El bucle ITERAR



ELEMENTOS AUXILIARES DE UN PROGRAMA

Son variables que realizan funciones específicas dentro de un programa, y por su gran utilidad, frecuencia de uso y peculiaridades, conviene hacer un estudio separado de las mismas. Las más importantes son: contadores, acumuladores e interruptores.

CONTADORES

Un contador es un campo de memoria cuyo valor se incrementa en una cantidad fija, positiva o negativa, generalmente asociado a un bucle.

Se utiliza en los siguientes casos:

- Para contabilizar el número de veces que es necesario repetir una acción.
- Para contar un suceso particular solicitado por el enunciado del problema.