

Resolución de problemas con computadora

El proceso de solución de un problema con una computadora conduce a la escritura de un programa y a su ejecución en la misma. Aunque el proceso de diseñar programas es esencialmente un proceso creativo, se puede considerar una serie de pasos comunes.

Los pasos para la resolución de un problema son:

- Análisis del problema.
- Diseño del algoritmo.
- Codificación.
- Compilación y ejecución.
- Verificación y depuración.

Los dos primeros pasos conducen a un diseño detallado escrito en forma de algoritmo. Durante el tercer paso (codificación) se implementa el algoritmo en un código escrito en un lenguaje de programación, en nuestro caso Qbasic.

La fase de compilación y ejecución traduce y ejecuta el programa. En las fases de verificación y depuración el programador busca errores de las etapas anteriores y los elimina.

Antes de conocer las tareas a realizar en cada paso, vamos a considerar el concepto y significado de algoritmo.

Un algoritmo es un método para resolver un problema mediante una serie de pasos precisos, definidos y finitos.

Características de un algoritmo:

- Preciso (indicar el orden de realización en cada paso).
- Definido (si se sigue dos veces, se obtiene el mismo resultado cada vez).
- Finito (tiene fin; un numero determinado de pasos)

Análisis del problema

Como primer paso se debe analizar el problema. Este paso requiere una clara definición, donde se contemple exactamente lo que debe hacer el programa y el resultado o solución deseada.

Dado que se busca una solución por computadora, se precisan especificaciones detalladas de entrada y salida.

Para poder definir un problema es conveniente responder a las siguientes preguntas:

- ¿Qué entradas se requieren (tipo y cantidad)?
- ¿Cuál es la salida deseada (tipo y cantidad)?
- ¿Qué método produce la salida deseada?

Diseño del algoritmo

En la etapa de análisis del proceso de programación se determina **que** hace el programa.

En la etapa de diseño se determina **cómo** hace el programa la tarea solicitada. Para ello vamos a utilizar una herramienta de programación llamada Diagrama de Flujo (flowchart).

Diagramas de flujo

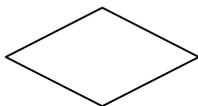
Un diagrama de flujo es una representación grafica de un algoritmo. Los símbolos más utilizados se detallan a continuación:



Símbolo terminal, con las palabras "INICIO" o "FIN" identifica el comienzo o el final del algoritmo. También se puede escribir el nombre del algoritmo.



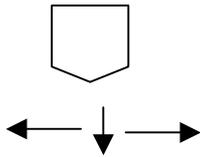
Símbolo de procesamiento, indica que la computadora lleva a cabo algún proceso de la información.



Símbolo de decisión, indica una decisión de la computadora que debe elegir entre dos o más alternativas del algoritmo.



Símbolo de entrada/ salida, sirve para indicar una operación de entrada o salida de datos.



Símbolo conector, representa la unión entre dos o más partes de un diagrama de flujo.

Las flechas se utilizan para conectar los símbolos anteriores en el orden en el que se deban ejecutarse.

Codificación de un programa

Codificación es la escritura en un lenguaje de programación de la representación del algoritmo desarrollada en las etapas anteriores. Dado que el diseño de un algoritmo es independiente del lenguaje de programación utilizado para su implementación, el código puede ser escrito con facilidad en un lenguaje o en otro.

Compilación y ejecución de un programa

Una vez que el algoritmo se ha convertido en un programa fuente, es preciso introducirlo en la memoria mediante el teclado y almacenarlo posteriormente en un disco. Esta operación se realiza con un editor, posteriormente el programa fuente se convierte en un archivo de programa que se graba en disco.

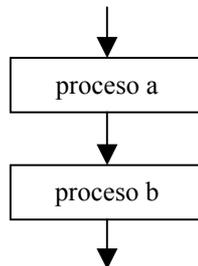
El programa fuente debe ser traducido a lenguaje máquina. Este proceso se realiza con el compilador y el sistema operativo que se encarga prácticamente de la compilación. Si tras la compilación se encuentran errores, es preciso volver a editar el programa, corregir los errores y compilar de nuevo.

Estructuras de control

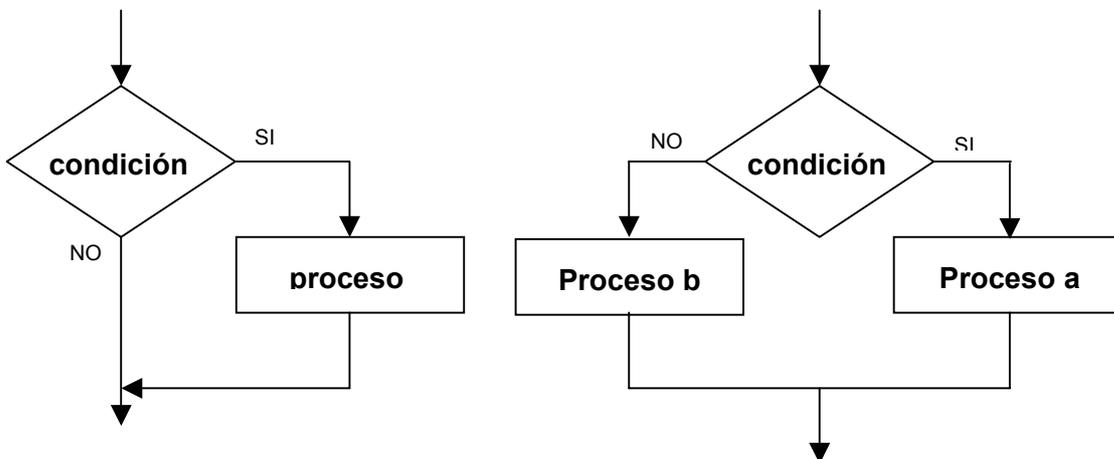
Las estructuras de control son métodos de especificar el orden en que las instrucciones de un algoritmo se ejecutaran. El orden de ejecución de las instrucciones determina el flujo de control.

Se puede demostrar que cualquier algoritmo o programa se puede realizar como combinación de tres estructuras de control fundamentales, estas estructuras son:

Estructura secuencial: Implica la ejecución secuencial de procedimientos o instrucciones, uno después de otro. En un programa con estructura secuencial, el orden lógico de las instrucciones coincide con su orden físico.

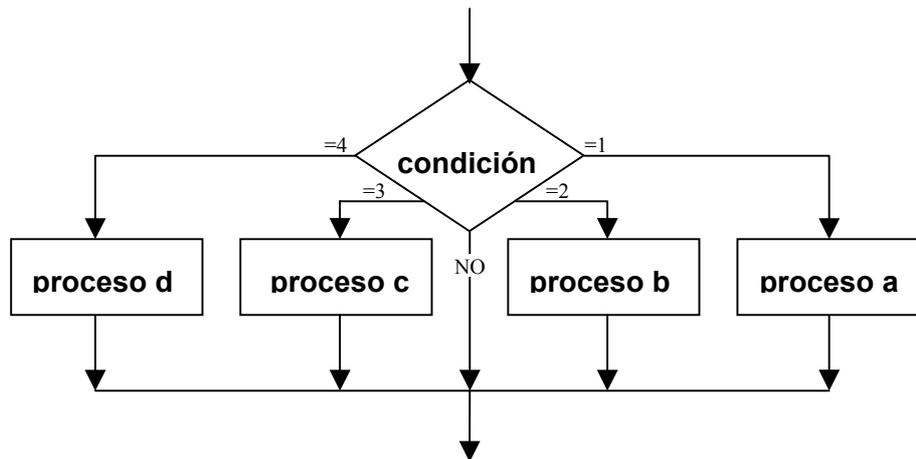


Estructura Selectiva: Muestra exactamente la elección de una entre dos o más "ramas" diferentes (proceso o conjuntos de instrucciones). La rama que se elige depende del valor de una "condición".



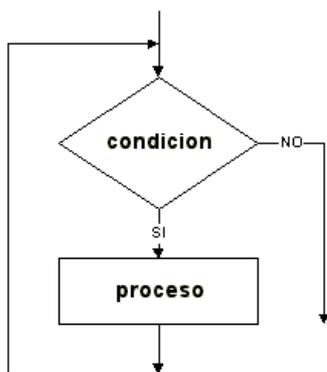
Estructura selectiva simple

Estructura selectiva doble.

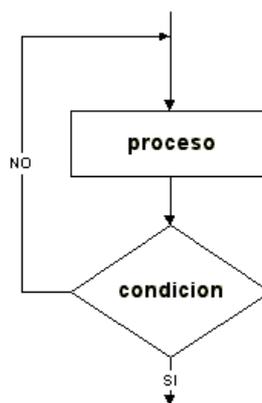


Estructura selectiva múltiple.

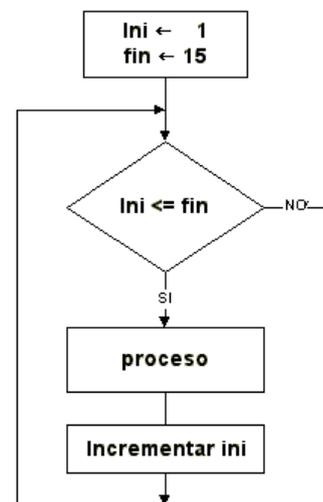
Estructura repetitiva: esta estructura se utiliza cuando un proceso o grupo de instrucciones tienen que repetirse un cierto número de veces o hasta que se cumpla una condición.



Estructura repetitiva del tipo “hacer mientras”.



Estructura repetitiva del tipo “hacer hasta”.



Estructura repetitiva del tipo FOR.

Reglas de selección de bucles:

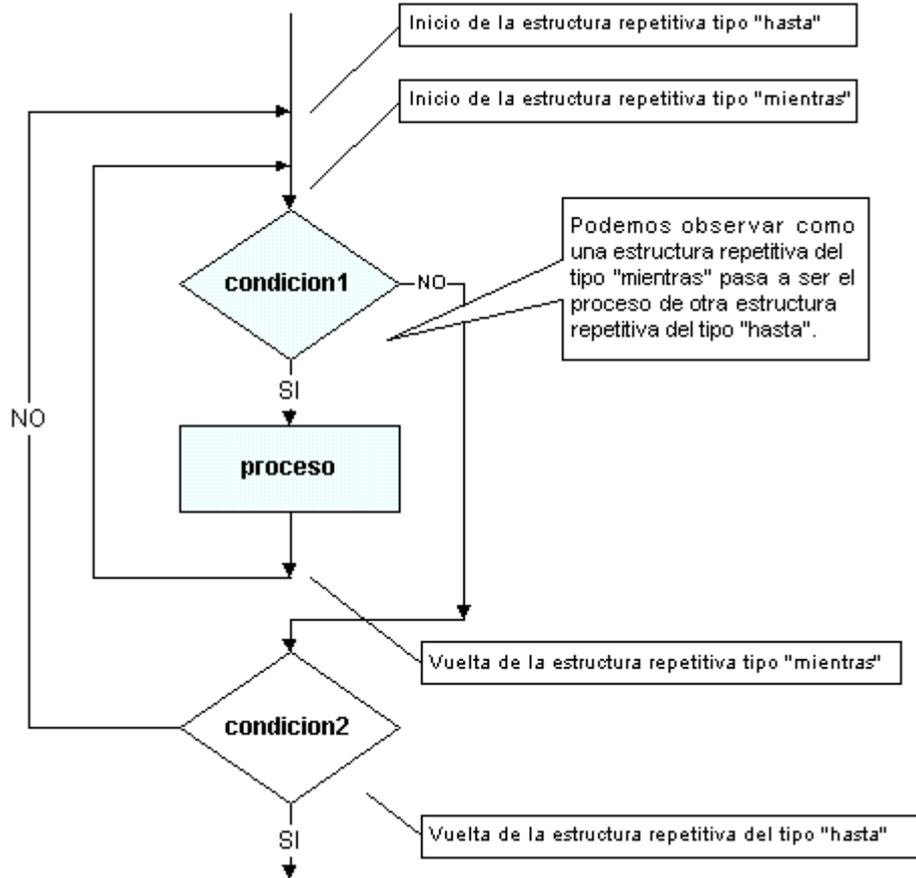
1. Si el número de iteraciones se conoce por adelantado, se debe utilizar **FOR**.
2. Si el número de iteraciones es indeterminado y el bucle no se debe ejecutar cuando sea falsa la primera vez, se debe usar **hacer mientras**.
3. Si el número de iteraciones es indeterminado y el bucle se debe ejecutar siempre al menos una vez, se utiliza **hacer hasta**.

Bucles infinitos:

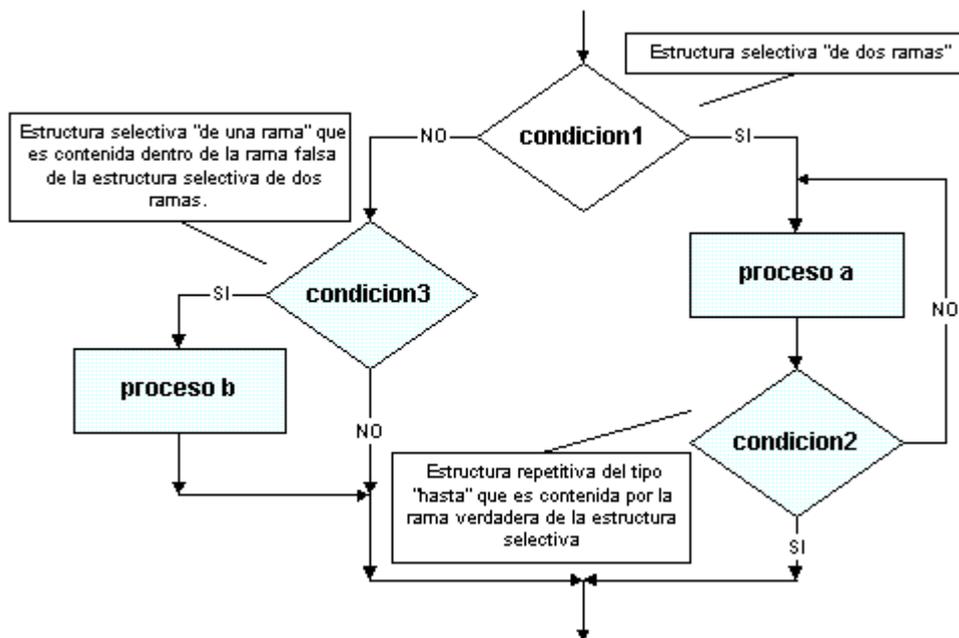
Un bucle se dice que es infinito cuando no termina nunca su ejecución. Esta situación se provoca cuando la condición o expresión lógica de salida no se cumple nunca y es una situación que se debe evitar.

Estructuras repetitivas y selectivas anidadas

El cuerpo de las estructuras de control selectivas y repetitivas puede contener cualquier tipo de sentencias secuenciales (simples o compuestas), como también otras estructuras de control selectivas y/o repetitivas.



Estructuras repetitivas anidadas



Estructura selectiva anidada con otra estructura repetitiva y selectiva.