Algoritmos y estructuras de programación

5.5.4. Repetición indexada : FOR / PARA
5.5.5. Repetición condicional : WHILE / MIENTRAS
5.5.6. Rupturas de ciclo : BREAK / CONTINUE
5.5.7. Elección de casos : SWITCH / CASO

5.5.4 Estructura de repetición indexada: FOR

Este tipo de estructura permite implementar la repetición de un cierto conjunto de instrucciones un número pre-determinado de veces.

Para ello se utiliza una variable de control del bucle, llamada también `índice`, que va recorriendo un conjunto pre-fijado de valores en un orden determinado. Para cada valor del `índice` en dicho conjunto, se ejecuta una vez el mismo conjunto de instrucciones.

En la Figura 5.8 se han representado la forma de escribir esta estructura en MATLAB y el organigrama correspondiente: el bloque de instrucciones se ejecuta una vez para cada valor del `indice`, que va tomando sucesivamente el valor de cada componente del vector `V`, de longitud `N`.

```
...
for indice=V
   instrucciones
end
...
```

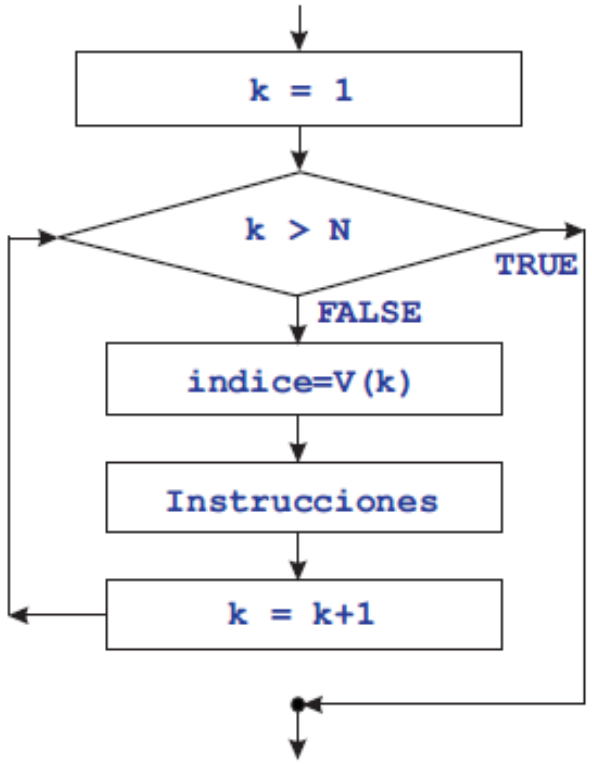


Figura 5.8: Repetición indexada: sintaxis MATLAB y diagrama de flujo.
El indice del bucle recorre los valores de un vector V de longitud N.

Como ejemplo de utilización de la estructura FOR, véanse los Algoritmos 5.8 y 5.9 para calcular la suma de los n primeros números impares.

Nota 5.7

a) El valor de la variable de control indice puede ser utilizado o no dentro del conjunto de instrucciones que forman parte del cuerpo del FOR, pero no debe ser modificado.

b) El conjunto de valores que debe recorrer el indice puede ser **vacío** (N=0). En ese caso, el bloque de instrucciones no se ejecuta ninguna vez.

c) Las estructuras FOR e IF pueden "anidarse", es decir, incluir una dentro de la otra, con la restricción (de sentido común) de que la interior tiene que estar completamente contenida en uno de los bloques de instrucciones de la otra. Véase, como ejemplo, el Algoritmo 5.10.

Algoritmo 5.8 Dado un entero, n, calcular la suma de los n primeros números impares.

```
Inicio
   LEER n
   HACER suma=0
   Para i= 1, 3, 5, ..., 2*n-1
     HACER suma=suma+i
   Fin Para
   IMPRIMIR 'La suma vale : ', suma
Fin
```

Algoritmo 5.9 Dado un entero, n, calcular

$$\sum_{k=0}^{n} \left(\frac{1}{2}\right)^k = 1 + \frac{1}{2} + \frac{1}{2^2} + \cdots + \frac{1}{2^n}$$

```
Inicio
   LEER n
   HACER suma=1
   HACER ter=1
   Para k= 1, 2, ..., n
     HACER ter=ter/2
     HACER suma=suma+ter
   Fin Para
   IMPRIMIR 'La suma vale : ', suma
Fin
```

Algoritmo 5.10 Dado un número natural, n, imprimir la lista de sus divisores, en orden decreciente.

```
Inicio
   LEER n
   IMPRIMIR ' Lista de divisores del numero: ', n
   Para i=ParteEntera(n/2) hasta 2 (incremento -1)
      Si resto(n/i)=0
         IMPRIMIR i
      Fin Si
   Fin Para
   IMPRIMIR 1
Fin
```

5.5.5 Estructura repetitiva condicional: WHILE

Permite implementar la repetición de un mismo conjunto de instrucciones mientras que se verifique una determinada condición: el número de veces que se repetirá el ciclo no está definido *a priori*.

El diagrama de flujo descriptivo de esta estructura se muestra en la Figura 5.9.

```
...
while expresión-lógica
   instrucciones
end
...
```

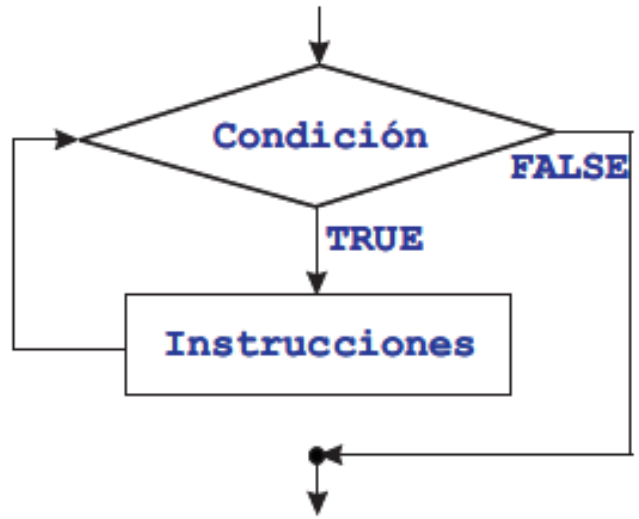


Figura 5.9: Estructura repetitiva WHILE: sintaxis MATLAB y diagrama de flujo.

Su funcionamiento es evidente, a la vista del diagrama:

1. Al comienzo de cada iteración se evalúa la expresión-lógica.
2. Si el resultado es VERDADERO, se ejecuta el conjunto de instrucciones y se vuelve a iterar, es decir, se repite el paso 1.
3. Si el resultado es FALSO, se detiene la ejecución del ciclo WHILE y el programa se sigue ejecutando por la instrucción siguiente al END.

El Algoritmo 5.11 es un ejemplo de utilización de esta estructura.

Algoritmo 5.11 Imprimir de forma ascendente los 100 primeros números naturales.

```
Inicio
   i=1
   Mientras que i ≤ 100
      IMPRIMIR i
      HACER i=i+1
   Fin Mientras
Fin
```

5.5.6 Ruptura de ciclos de repetición: BREAK y CONTINUE

En ocasiones es necesario interrumpir la ejecución de un ciclo de repetición **en algún punto interno del bloque de instrucciones que se repiten**. Lógicamente, ello dependerá de que se verifique o no alguna condición.

La interrupción puede hacerse de dos formas:

1. **Abandonando el ciclo de repetición definitivamente.**

2. **Abandonando la iteración en curso, pero comenzando la siguiente.**

Las instrucciones para poner esto en práctica tienen nombres diversos en los distintos lenguajes de programación. En MATLAB, la primera opción se implementa con la instrucción `BREAK` y la segunda con la instrucción `CONTINUE`. Ambas pueden utilizarse tanto para romper un ciclo `FOR` como un ciclo `WHILE`. Cuando se utiliza la orden `BREAK` dentro de un ciclo `FOR`, el `índice` del bucle conserva, fuera del mismo, el último valor que tomó.

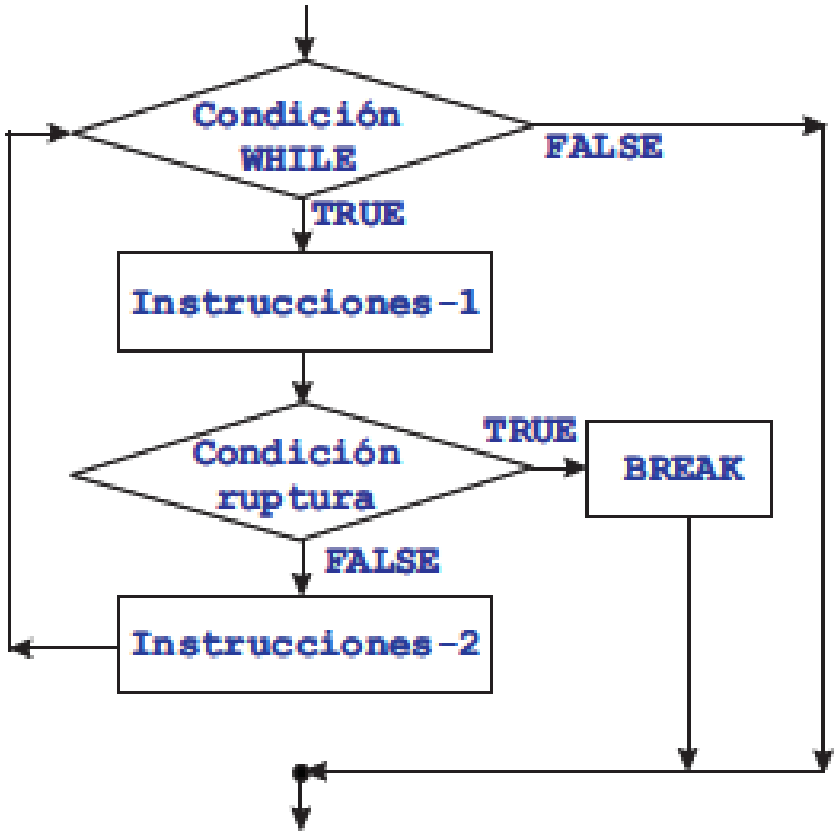


Figura 5.10: Diagrama de flujo de la ruptura de ciclo BREAK.

Algoritmo 5.10 Dado un número natural, n, imprimir la lista de sus divisores, en orden decreciente.

```
Inicio
   LEER n
   IMPRIMIR ' Lista de divisores del numero: ', n
   Para i=ParteEntera(n/2) hasta 2 (incremento -1)
      Si resto(n/i)=0
         IMPRIMIR i
      Fin Si
   Fin Para
   IMPRIMIR 1
Fin
```

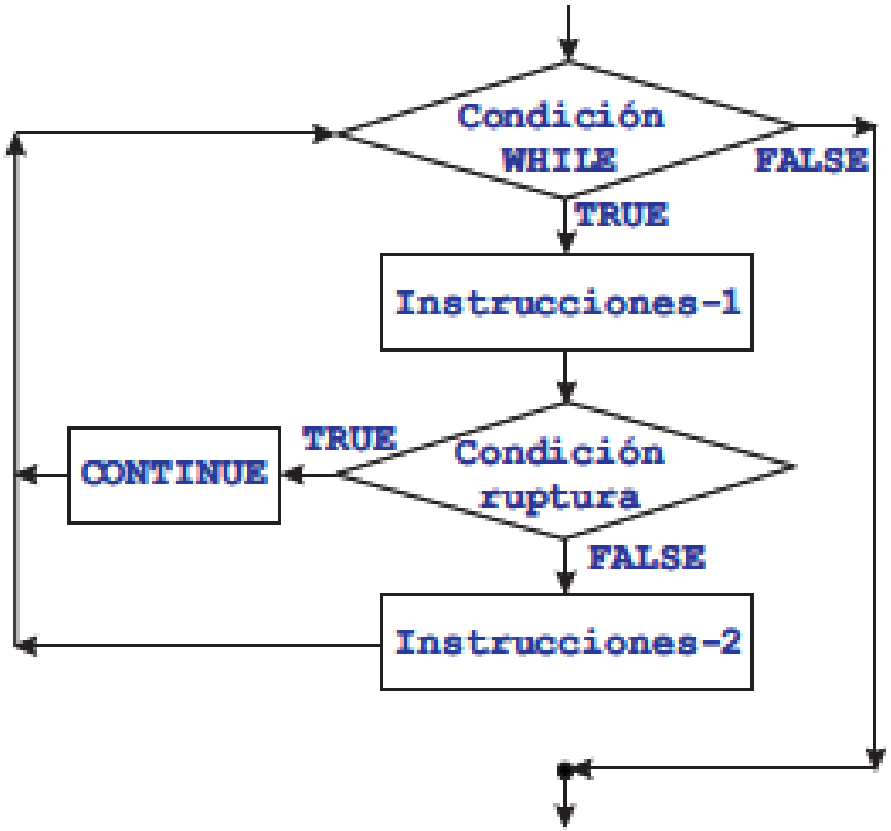


Figura 5.11: Diagrama de flujo de la ruptura de ciclo CONTINUE.

Algoritmo 5.11 Imprimir de forma ascendente los 100 primeros números naturales.

```
Inicio
    i=1
    Mientras que i ≤ 100
        IMPRIMIR i
        HACER i=i+1
    Fin Mientras
Fin
```

5.5.7 Estructura de elección entre varios casos: SWITCH

Este tipo de estructura permite decidir entre varios caminos posibles, en función del valor que tome una determinada instrucción.

El diagrama de flujo correspondiente a una de estas estructuras (con cuatro casos) se presenta en la Figura 5.12.

```
switch expresión
  case valor-1
    instrucciones caso 1
  case valor-2
    instrucciones caso 2
  ...
  case {valores...}
    instrucciones caso k
  otherwise
    instrucciones
end
```

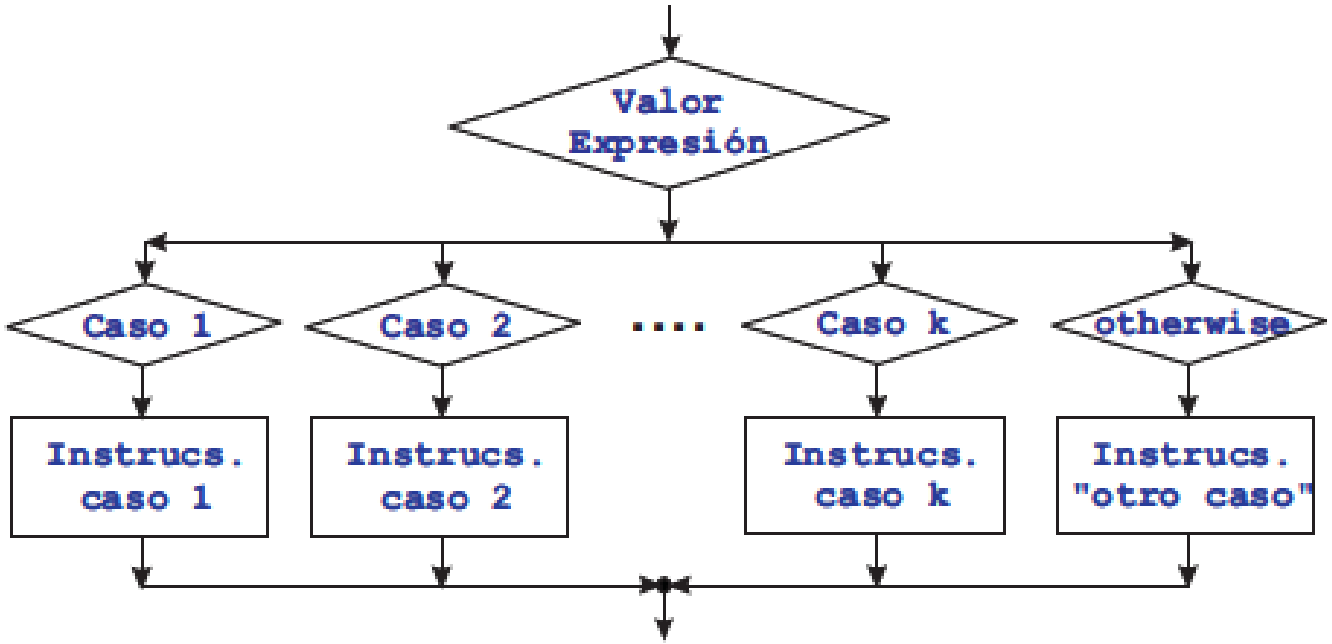


Figura 5.12: Estructura de elección de caso SWITCH: sintaxis MATLAB y diagrama de flujo.

En cada uno de los casos, el `valor` correspondiente puede ser o bien un sólo valor, o bien un conjunto de valores, en cuyo caso se indican entre llaves. La cláusula `OTHERWISE` y su correspondiente conjunto de instrucciones puede no estar presente.

El funcionamiento es el siguiente:

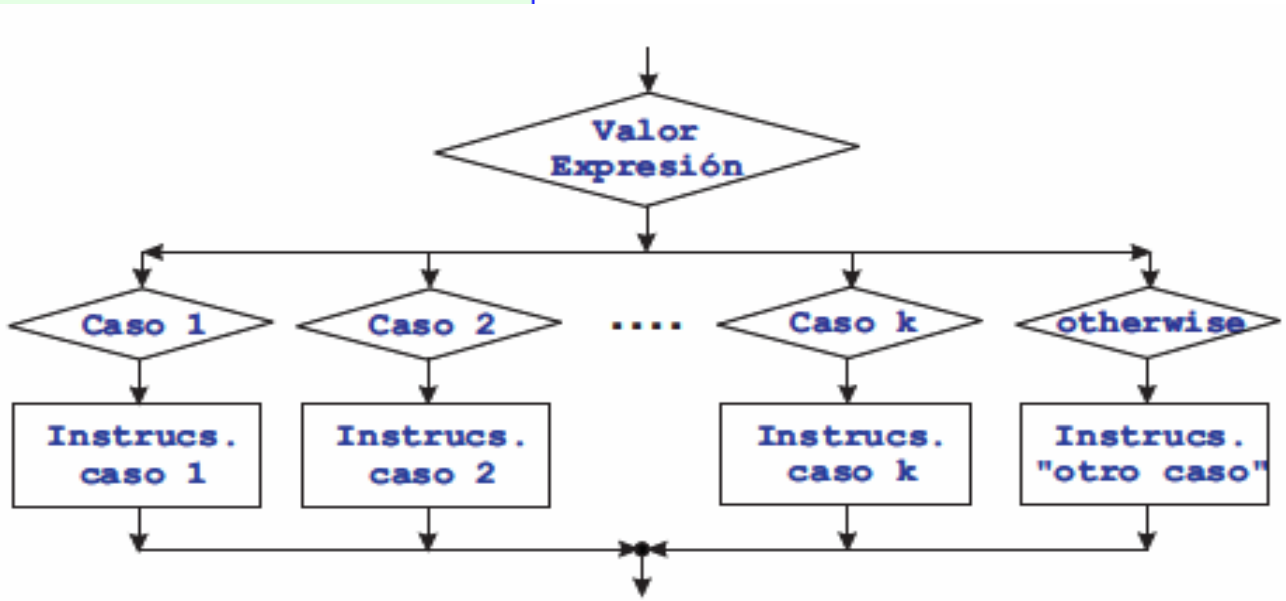
1. Al comienzo se evalúa la `expresión`.
2. Si `expresión` toma el valor (ó valores) especificados junto a la primera cláusula `CASE`, se ejecuta el conjunto de instrucciones de este caso y después se abandona la estructura `SWITCH`, continuando por la instrucción siguiente al `END`.
3. Se repite el procedimiento anterior, de forma ordenada, para cada una de las cláusulas `CASE` que siguen.
4. Si la cláusula `OTHERWISE` está presente y la `expresión` no ha tomado ninguno de los valores anteriormente especificados, se ejecuta el conjunto de instrucciones correspondiente.

Obsérvese que se ejecuta, **como máximo** el conjunto de instrucciones de uno de los casos, es decir, una vez que se ha verificado un caso y se ha ejecutado su conjunto de instrucciones, no se testea el resto de casos, ya que se abandona la estructura. Obviamente, si la cláusula `OTHERWISE` no está presente, puede ocurrir que no se dé ninguno de los casos.

Como ejemplo de utilización, se presenta el mismo proceso del Algoritmo 5.6, pero utilizando la sentencia SWITCH.

Algoritmo 5.12 Dados dos números reales, a y b, y el símbolo, S (carácter), de un operador aritmético (+, -, *, /), imprimir el resultado de la operación a S b

```
LEER a , b , S
Elegir caso S
  Caso '+'
     IMPRIMIR 'El resultado es =', a+b
  Caso '-'
     IMPRIMIR 'El resultado es =', a-b
  Caso '*'
     IMPRIMIR 'El resultado es =', a*b
  Caso '*'
     IMPRIMIR 'El resultado es =', a*b
  Caso '/'
     Si a≠0
        IMPRIMIR 'El resultado es =', a/b
     Si no, si b≠0
        IMPRIMIR 'El resultado es =', Inf (infinito)
     Si no
        IMPRIMIR 'El resultado es =', NaN (indeterminación)
     Fin Si
  En otro caso
     IMPRIMIR 'El operador no se reconoce'
  Fin Elegir caso
```



Practica-1

1.- ACTIVACION DE CALEFACTOR

Escriba un algoritmo y realice el diagrama de flujo en Word para el siguiente problema:

- Tenemos un calefactor sin termostato conectado al ordenador que lo puede activar o desactivar: calefactor =ON/OFF.
- También tenemos un termómetro que toma la temperatura cada 5 minutos y la deja disponible en "temp_actual" en grados Celsius.
- Necesitamos un algoritmo que cada 5 minutos, pida la temperatura deseada "temp_objetivo" y realice las acciones necesarias para mantenerla, hasta el próximo chequeo. Puedes usar la instrucción Espera(5seg) para regular el algoritmo.
- Si la puerta o la ventana estan abiertas, no se activa el calefactor (puerta_abierta=ON/OFF), ventana_abierta = ON/OFF).

Entrega estos ejercicios en el formato que te indique el tutor.