

Guía práctica de estudio 08: Estructuras de selección



Elaborado por:

Ing. Jorge A. Solano Gálvez
Guadalupe Lizeth Parrales Romay

Revisado por:

M.C. Edgar E. García Cano

Autorizado por:

M.C. Alejandro Velázquez Mena

Guía práctica de estudio 08:

Estructuras de selección

Objetivo:

Elaborar programas en lenguaje FORTRAN que incluyan las estructuras de selección para la resolución de problemas básicos.

Actividades:

- Utilizar las estructuras *if* e *if-else*.
- Utilizar los tipos de datos y expresiones lógicas que se pueden manejar en las estructuras de selección.

Introducción

Las estructuras de control de flujo en un lenguaje especifican el orden en que se realiza el procesamiento de datos.

Las estructuras de selección (o condicionales) permiten realizar una u otra acción con base en una expresión lógica. Las acciones posibles a realizar son mutuamente excluyentes, es decir, solo se puede ejecutar una a la vez dentro de toda la estructura.

Lenguaje FORTRAN posee dos estructuras de selección: la estructura *if-else* y la estructura *select-case*.

Licencia GPL de GNU

El software presente en esta guía práctica es libre bajo la licencia GPL de GNU, es decir, se puede modificar y distribuir mientras se mantenga la licencia GPL.

```
/*
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see <http://www.gnu.org/licenses/>.
 *
 * Author: Jorge A. Solano
 */
```

Estructura de control selectiva if

La estructura de control de flujo más simple es la estructura condicional **if**, su sintaxis es la siguiente:

```
if (expresión_lógica) ! línea de código a ejecutar
```

En esta estructura se evalúa la expresión lógica y, si se cumple (si la condición es verdadera), se ejecuta la instrucción de la línea. Si no se cumple la condición, se continúa con el flujo normal del programa.

Si el bloque de código a ejecutar consta de más de una línea de código la sintaxis es la siguiente:

```
if (expresión_lógica) then
  Bloque de código a ejecutar
endif
```

En este caso, se evalúa la expresión lógica y, si se cumple (si la condición es verdadera), se ejecuta el bloque de código. Si no se cumple la condición, se omite el bloque de código y se continúa con el flujo normal del programa.

NOTA 1: La expresión lógica evaluada regresará como resultado un valor lógico. Dentro de las estructuras de control `.FALSE.` indica que la expresión lógica es falsa y `.TRUE.` indica que la expresión lógica es verdadera.

Código (estructura de control selectiva if)

```

program ifSimple

c Este programa valida si el número almacenado en la variable 'a'
c es mayor al número almacenado en la variable 'b'

integer a, b
parameter (a = 3, b = 2)

if (a .GT. b) write (*,*) 'a es mayor que b'

write (*,*) 'El programa sigue su flujo ...'

stop
end

```

Código (estructura de control selectiva if)

```

program ifLogical

c Este programa comprueba las condiciones
c .FALSE. -> falso
c .TRUE. -> verdadero

if (.FALSE.) then
  write (*,*) 'Este bloque de código nunca se ejecuta'
  write (*,*) 'porque la condición siempre es falsa'
endif

if(.TRUE.) then
  write (*,*) 'Este bloque de código siempre se ejecuta'
  write (*,*) 'porque la condición siempre es verdadera'
endif

stop
end

```

Estructura de control selectiva if-else

La sintaxis de la estructura de control de flujo **if-else** es la siguiente:

```

    if (expresión_lógica) then
c     bloque de código a ejecutar
c     si la condición es verdadera
    else
c     bloque de código a ejecutar
c     si la condición es falsa
    endif

```

Esta estructura evalúa la expresión lógica y si la condición es verdadera se ejecutan las instrucciones del bloque que se encuentra entre las primeras llaves, si la condición es falsa se ejecuta el bloque de código que está después de la palabra reservada 'else'. Al final de que se ejecute uno u otro código (mutuamente excluyente), se continúa con el flujo normal del programa.

Código (estructura de control selectiva if-else)

```

    program ifElse

c Este programa permite validar si un
c número entero es par o impar.
c El número se lee desde la entrada
c estándar (el teclado).

    integer num

    write (*,*) 'Ingrese el numero: '
    read (*,*) num

    if((num/2) .EQ. 0) then
        write (*,*) 'El numero ', num, ' es par'
    else
        write (*,*) 'El numero ', num, ' es impar'
    endif

    stop
end

```

Es posible *anidar* varias estructuras if-else, es decir, dentro de una estructura if-else tener una o varias estructuras if-else. También es posible validar varias expresiones lógicas en las estructuras else, es decir:

```

    if (expresión_lógica) then
c     bloque de código a ejecutar
c     si la condición_lógica es verdadera
    else (expresión_lógica2)
c     bloque de código a ejecutar
c     si la condición_lógica2 es verdadera
    else
c     bloque de código a ejecutar si
c     las condiciones anteriores son falsas
    endif

```

Código (estructura de control selectiva if-else anidada)

```

program anidadoIfElse

c Este programa ordena en forma descendente
c tres valores enteros dados. Los valores
c se leen desde la entrada estándar (el teclado).

integer uno, dos, tres

write (*,*) 'Ingrese 3 numeros'

read (*,*) uno, dos, tres

if(uno .GT. dos) then
    if(dos .GT. tres) then
        write (*,*) uno,'es mayor que',dos,'que es mayor a',tres
    elseif(uno .GT. tres) then
        write (*,*) uno,'es mayor que',tres,'que es mayor a',dos
    else
        write (*,*) tres,'es mayor que',uno,'que es mayor a',dos
    endif
elseif(dos .GT. tres) then
    if(tres .GT. uno) then
        write (*,*) dos,'es mayor que',tres,'que es mayor que', uno
    else
        write (*,*) dos,'es mayor que',uno,'que es mayor a',tres
    endif
else
    write (*,*) tres,'es mayor que',dos,'que es mayor a',uno
endif

stop
end

```

Estructura de control selectiva select-case

La sintaxis de la estructura select-case es la siguiente:

```
select case (opcion_a_evaluar){  
  case (valor1)  
c    Código a ejecutar  
  case (valor2)  
c    Código a ejecutar  
  ...  
  case (valorN)  
c    Código a ejecutar  
  case default  
c    Código a ejecutar
```

La estructura *select-case* evalúa la variable que se encuentra entre paréntesis después de las palabras reservadas *select case* y la compara con los valores constantes que posee cada caso (case). Los tipos de datos que puede evaluar esta estructura son enteros, caracteres y lógicos.

Si la opción a evaluar no coincide dentro de algún caso, entonces se ejecuta el bloque por defecto (default). El bloque por defecto normalmente se escribe al final de la estructura, pero se puede escribir en cualquier otra parte.

Código (estructura de control selectiva select-case)

```
program charCase

c Este programa permite elegir una opción del
c menú a partir del carácter ingresado. La
c opción se lee desde la entrada estándar (el teclado).

character op

write (*,*) 'Menú:'
write (*,*) 'Elegir la opción deseada'
write (*,*) 'a) Ingresar'
write (*,*) 'b) Registrarse'
write (*,*) 'c) Salir'
read (*,*) op

select case (op)

case default
  write (*,*) 'Opción no válida'
case ('a')
  write (*,*) 'Se seleccionó Ingresar'
case ('b')
  write (*,*) 'Se seleccionó Registrarse'
case ('c')
  write (*,*) 'Se seleccionó Salir'
end select

stop
end
```


Código (estructura de control selectiva select-case)

```
program intCase

c Este programa permite elegir una opción del menú a
c partir del entero ingresado. La opción se lee desde
c la entrada estándar (el teclado)

integer op, valor

parameter (valor = 3)

write (*,*) 'Menú'
write (*,*) 'Elegir la opción deseada'
write (*,*) '1) Ingresar'
write (*,*) '2) Registrarse'
write (*,*) '3) Salir'

read (*,*) op

select case (op)

case (valor-2)
write (*,*) 'Se seleccionó Ingresar'
case (valor-1)
write (*,*) 'Se seleccionó Registrar'
case (valor)
write (*,*) 'Se seleccionó Salir'
case default
write (*,*) 'Opción no válida'

end select

stop
end
```

Código (estructura de control selectiva select-case)

```

program logicalCase

c Este programa permite validar una opción a partir del
c valor lógico ingresado. La opción se lee desde
c la entrada estándar (el teclado)

logical op

write (*,*) '¿Desea continuar?'
write (*,*) 'T) Sí'
write (*,*) 'F) No'

read (*,*) op

select case (op)

case (.TRUE.)
write (*,*) 'Se seleccionó Sí'
case (.FALSE.)
write (*,*) 'Se seleccionó No'
case default
write (*,*) 'Opción no válida'
end select

stop
end

```

Bibliografía

- Oracle (2010). Fortran 77 Lenguaje Reference. Consulta: Julio de 2015. Disponible en: <http://docs.oracle.com/cd/E19957-01/805-4939/>
- Stanford University (1995). Fortran 77 Tutorial. Consulta: Julio de 2015. Disponible en: http://web.stanford.edu/class/me200c/tutorial_77/