

INTRODUCCIÓN A FORTRAN 77

Las computadoras son dispositivos simples, que pueden realizar operaciones sencillas, de acuerdo a las instrucciones que nosotros somos capaces de darles, a través de un *programa*. Una vez que la máquina ejecuta la serie de instrucciones se obtienen los resultados.

El conjunto de instrucciones que la persona puede utilizar para definir un problema capaz de ser entendido por la computadora, recibe el nombre de “lenguaje de programación”. Al principio, estos lenguajes eran muy complicados para el hombre, y con el tiempo fueron evolucionando para que sean más próximos a la aplicación que va a diseñarse. Estos lenguajes se conocen, actualmente, como “lenguajes de alto nivel”. Éste es el caso del lenguaje FORTRAN (FORmula TRANslation): el programador escribe las instrucciones en un lenguaje de alto nivel y la computadora las traduce a su propio lenguaje para luego ejecutarlas.

El programa que se escribe en lenguaje FORTRAN se llama “lenguaje fuente” y el programa que se obtiene luego de la compilación (conversión de las instrucciones en lenguaje de la máquina) se llama “lenguaje objeto”. Durante el proceso de compilación se obtienen mensajes de diagnóstico, que nos ayudan a detectar posibles errores en el programa generado.

Se denomina “línea” de un programa FORTRAN a una sucesión de 72 caracteres. Cada carácter ocupa o se escribe en una columna y éstas se enumeran consecutivamente de izquierda a derecha.

Una “línea de comentario” es aquella que contiene en la columna 1 un **C** o un **!** (este último puede incluirse en cualquier columna) y después del símbolo el comentario que el programador quiere agregar en el código, el cual no es tenido en cuenta por el compilador.

Las columnas desde la 1 a la 5 pueden contener una etiqueta para las sentencias FORTRAN, si se las debe referir en otra parte del código. La etiqueta es un número entero, y no debe repetirse para diferentes sentencias. En cualquiera de las columnas de la 7 a la 72 se escribe, en toda la línea, la sentencia FORTRAN (una sentencia por línea), con o sin espacios en el medio.

1. Constantes

El término *constante* designa un valor específico y determinado que se define al hacer un programa y que no cambia a lo largo del mismo. Los tipos de constantes que pueden utilizarse en un programa FORTRAN se describen en la Tabla 1.

Constantes	Ejemplo
Enteras	3; -57; +28
Reales básicas	9,43; -67,32; 0,3; ,3; 24
Reales con exponente	34,5E2; -,86E-3
Enteras con exponente	875E-2; -32E9
Lógicas	.TRUE.;.FALSE.
Caracter	'AMIGO';'X';'Y'
Doble precisión	43,2D-3; ,23D+04

Tabla 1: Constantes

2. Variables

Una *variable* en FORTRAN es un nombre simbólico con el que se designa un dato que puede tomar diversos valores durante la ejecución del programa. Las variables se forman empleando caracteres alfanuméricos (el primero siempre debe ser alfabético). Para la computadora, la variable hace referencia a una determinada zona o localización de la memoria donde se archiva o almacena el dato correspondiente. Los tipos de variables que pueden utilizarse se describen en la Tabla 3.

Variables	Ejemplo
Enteras	INTEGER I,NUM,LONG
Reales	REAL RADIO,VEL,X,Y
Lógicas	LOGICAL UP
Caracter	CHARACTER*4 IDENT
Doble precisión (DP)	DOUBLE PRECISION POS

Tabla 2: Variables

3. Sentencias

Una *sentencia* o instrucción es una secuencia de caracteres que siguen reglas sintácticas y que tiene significado para el programador y para el compilador. Las sentencias se clasifican en “ejecutables”, las cuales especifican acciones, y “no ejecutables”, que describen características. Las sentencias FORTRAN se forman mediante *palabras clave* que tienen un significado para el lenguaje.

Un programa es un conjunto formado por una serie de sentencias y, opcionalmente, líneas de comentarios.

3.1. Operadores

3.1.1. Operadores aritméticos

Operador	Significado
+	Suma
-	Resta
*	Multiplicación
/	División
**	Exponenciación

Tabla 3: Variables

3.1.2. Operadores y expresiones de relación

Se denomina “operador de relación” a un operador tal que al ser aplicado sobre dos elementos del mismo tipo, datos numéricos o *character*, produce un resultado lógico, es decir, *cierto* o *falso*. Los operadores se resumen en la Tabla 4.

Operador	Significado
.GT.	Mayor que ($>$)
.GE.	Mayor o igual que (\geq)
.LT.	Menor que ($<$)
.LE.	Menor o igual que (\leq)
.EQ.	Igual que ($=$)
.NE.	Distinto de (\neq)

Tabla 4: Operadores de relación

3.1.3. Operadores lógicos

Para formar expresiones lógicas más complejas, el lenguaje FORTRAN dispone de operadores especiales, denominados *operadores lógicos*. Con ellos se amplían las posibilidades del álgebra lógica. Éstos son cinco, y se resumen en la Tabla 5.

Operador	Significado
.NOT.	Negación
.AND.	Inserción
.OR.	Unión
.EQV.	Equivalente
.NEQV.	No equivalente

Tabla 5: Operadores lógicos

3.2. Expresiones aritméticas

Se forman utilizando operandos, es decir, constantes y/o variables, operadores y paréntesis, de manera que agrupados indiquen la ejecución de un determinado cálculo aritmético. Si todos los datos de la expresión son enteros, se dice que la expresión es *entera*; si todos los datos son reales, la expresión es *real*; si hay de ambos tipos, la expresión se llama *mixta*. La Tabla 6 muestra el resultado que se obtiene cuando se combinan diferentes tipos de datos.

+, -, *, /, **	Dato entero	Dato real	Dato DP
Dato entero	Resultado entero	Resultado real	Resultado DP
Dato real	Resultado real	Resultado real	Resultado DP
Dato DP	Resultado DP	Resultado DP	Resultado DP

Tabla 6: Resultado obtenido al combinar datos de distinto tipo

3.3. Sentencias PROGRAM, STOP y END

Cuando se escribe un programa en FORTRAN, la primera sentencia puede ser **PROGRAM** (**PROGRAM** nombre).

Para indicar la terminación de la ejecución de un programa, se emplea la sentencia **STOP**.

La última sentencia del programa tiene que ser **END**, y le indica al compilador que ya no hay más sentencias en el programa fuente para ser convertidas en programa objeto.

PROGRAM es una sentencia no ejecutable, mientras que **STOP** y **END** sí lo son.

3.4. Sentencia GO TO

En los programas las instrucciones se ejecutan de manera secuencial. Existen situaciones en las cuales es necesario volver a repetir la ejecución de una instrucción anterior o producir un salto a otra sentencia de más adelante, ignorando las que están en el medio. Estos saltos se denominan *bifurcaciones*. Las instrucciones que sirven para gobernar y dirigir la realización o ejecución de otras instrucciones se denominan *secuencias de control*, y pertenecen a la familia de las ejecutables.

3.4.1. GO TO incondicional

GO TO n

donde n es la etiqueta de la instrucción siguiente a realizar.

3.4.2. GO TO calculado

GO TO(n1,n2,n3,...,nm),i

donde i es una variable, y n1, n2, n3 son etiquetas. Según el valor que adquiera i, se ejecutará la instrucción con la etiqueta que corresponda. Por ejemplo, si i es 2, se salta a la línea con etiqueta n2.

3.5. Sentencia IF

Esta sentencia también se utiliza para generar bifurcaciones según ciertas condiciones.

3.5.1. IF aritmético

La sintaxis es:

IF(x)n1,n2,n3

Entonces, si x es menor, igual o mayor que cero, se ejecuta la sentencia con etiqueta n1, n2 ó n3, respectivamente.

3.5.2. IF lógico

Permite el uso de las expresiones lógicas para poder efectuar bifurcaciones dentro de un programa FORTRAN. Su sintaxis es:

```
IF(expresión)sentencia
```

es decir, si la expresión entre paréntesis es verdadera, se ejecuta la sentencia indicada a continuación. Si es falsa, se continua con la ejecución del programa.

3.5.3. Bloque IF

En primer lugar, permite comprobar si una condición es cierta o falsa. Dependiendo de esta condición, se ejecuta uno u otro grupo de sentencias, finalizando el proceso en un punto común. La sintaxis es como sigue:

```
IF(expresión)THEN
  :
ELSE
  :
END IF
```

3.6. Bloque DO

Sirve para ejecutar repetidamente una serie de sentencias en bucle. Una posible sintaxis es:

```
DO i=n1,n2,n3
  Sentencia1
  Sentencia2
  :
END DO
```

donde i es una variable, generalmente entera, y $n1$, $n2$ y $n3$ son constantes o variables que indican, respectivamente, un valor inicial, un valor final y un incremento. Si $n3$ es 1, puede omitirse.

3.7. Sentencia CONTINUE

Esta sentencia es ejecutable, pero no genera ninguna instrucción al ser traducida al código de máquina por el compilador.

3.8. Sentencia PAUSE

Indica una parada temporal en la ejecución de un programa.

4. Ejercicios

1. Dados tres valores X , Y , Z , permutarlos circularmente.
2. Dadas tres cantidades, ingresadas por pantalla, ordenarlas de manera creciente.
3. Conociendo tres cantidades X , Y , Z , y sabiendo que $X < Y < Z$, dada otra cantidad A , ubicarla en la serie y escribir los cuatro números de manera creciente.
4. Cálculo de factoriales
 - a) Calcular el factorial de todos los números menores o iguales que 20, utilizando el bloque **DO**, y mostrar los resultados por pantalla.
 - b) Calcular el factorial de todos los números menores o iguales que 20, utilizando la sentencia **GO TO**, y mostrar los resultados por pantalla. Ayuda: será necesario utilizar la sentencia **IF**.
 - c) Modificar el programa para guardar los resultados en un archivo de texto.
5. Calcular el producto escalar de dos vectores de tres componentes, $V1$ y $V2$.
6. Calcular el producto vectorial de dos vectores de tres componentes, $V1$ y $V2$.
7. Escribir un programa que encuentre las raíces de ecuaciones cuadráticas del tipo $y(x) = Ax^2 + Bx + C$, cuyas constantes sean ingresadas por el usuario desde el teclado. Si la resolución es posible, comunicar el resultado por pantalla. Si el discriminante de la ecuación es negativo, avisar al operador sobre la imposibilidad de resolución en el conjunto de los reales.
8. Graficar una función cuadrática en alguna zona de su dominio, utilizando un programa escrito en lenguaje FORTRAN, y el graficador GNUPLOT.
9. Graficar la misma función elegida en el ítem anterior, pero haciendo uso sólo del programa graficador.