

LENGUAJE FORTRAN. UTILIZACIÓN DE FORMATOS

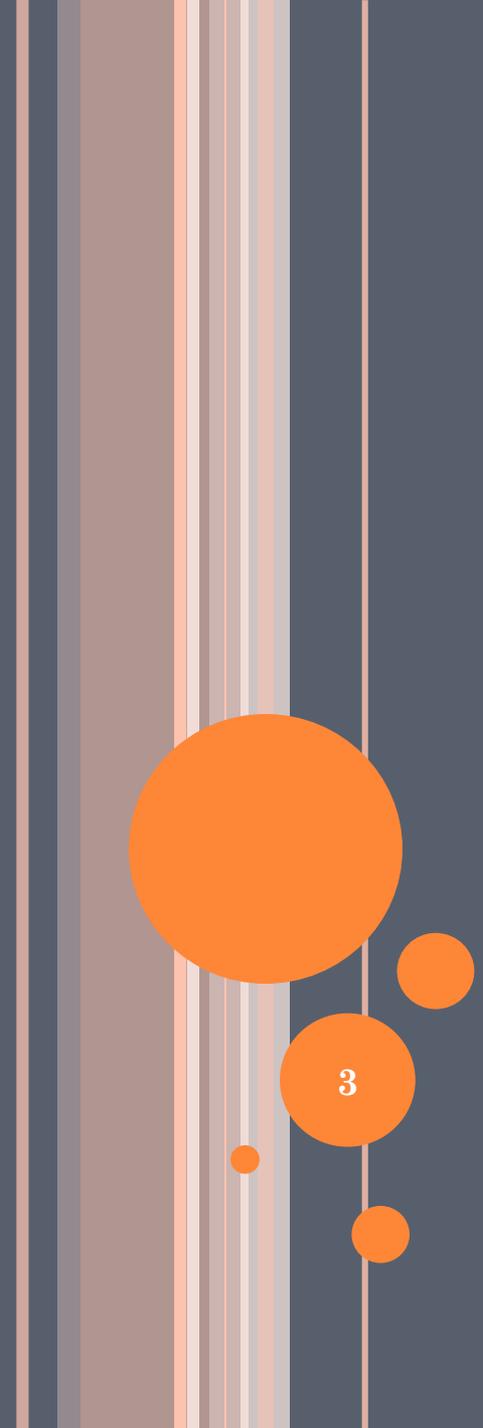
Programación en Fortran

Valentín Moreno

ÍNDICE

1. Formatos
2. Sentencias de salida con formato
3. Sentencias de entrada con formato

1. FORMATOS

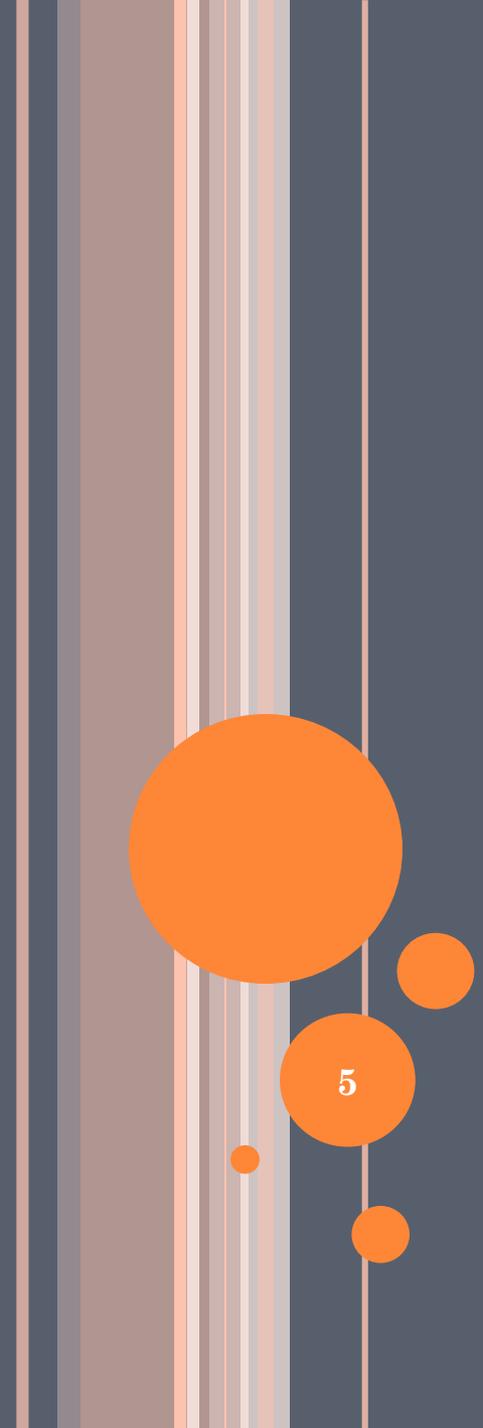


3

1. FORMATOS

Formatos		Significado
Iw	Iw.m	Constantes y variables enteras
Bw	Bw.m	Datos enteros en formato binario
Ow	Ow.m	Datos enteros en formato octal
Zw	Zw.m	Datos enteros en formato hexadecimal
Fw.d		Datos reales con notación decimal
Ew.d	Ew.dEe	Datos reales con notación exponencial
ESw.d	ESw.dEe	Datos reales con notación científica
ENw.d	ENw.dEe	Datos reales con notación ingenieril
Gw.d	Gw.dEe	Datos reales con formato adaptable
A	Aw	Caracteres
...	'*...*'	Cadenas de caracteres
Lw		Datos lógicos
Tc	TLn TRn	Tabuladores
nX		Espacios en horizontal
/		Espaciado vertical
:		Control para lectura de caracteres

- w n° entero positivo que indica la anchura del campo (por ejemplo, 6 espacios)
- m n° entero igual o mayor que cero que indica un n° mínimo de dígitos a leer o imprimir
- d n° entero positivo que indica el número de cifras a la derecha del punto decimal
- e n° entero igual o mayor que cero que indica el n° de cifras en un exponente
- * un carácter
- c n° entero positivo que representa la posición de un carácter
- n n° entero positivo que indica un n° de caracteres



2. SENTENCIAS DE SALIDA CON FORMATO

5

2. SENTENCIAS DE SALIDA CON FORMATO

La forma general de la sentencia PRINT es la siguiente:

PRINT formato, lista_de_constantes_y_variables
donde *formato* es una de las opciones siguientes:

- Un asterisco (*): En este caso se trata de un formato libre, donde las variables de la lista se imprimen de un modo “razonable”, dependiente del compilador utilizado.
- Una constante o variable de tipo CHARACTER (o una expresión de este tipo, o bien un vector de caracteres), que especifica los formatos con los que se imprimirán las variables de la lista.

2. SENTENCIAS DE SALIDA CON FORMATO

- Espaciado horizontal:
 - nX → Deja n espacios en blanco al principio de la línea.
 - Tm → Indica que el siguiente dato se imprimirá en esa línea a partir de la posición m .
- Factor de repetición. Anteponiendo un número a un descriptor de formato entero o real permite imprimir ese número de enteros o reales con el formato indicado. También se puede aplicar a un conjunto de descriptores utilizando los paréntesis.
- Salto de línea /. Se pueden encadenar varios (///) o utilizar un factor de repetición (3/).

2. SENTENCIAS DE SALIDA CON FORMATO

- Se deben combinar los distintos descriptores cuando se quieren imprimir en la misma línea distintas variables o constantes.
- Ejemplo

```
Print ' (A, F5.2) ', "Temperatura:", temp
```



- Nótese que es necesario indicar un formato por cada elemento a imprimir.

Ejemplos:

- I6 → Se imprimirá un entero ocupando 6 espacios:
 - Si el entero es positivo y tiene menos de 6 cifras se ajusta por la derecha y se completa con blancos.
 - Si es negativo debe tener menos de 6 cifras para que se imprima completo ya que debemos tener en cuenta el espacio que ocupa el signo.
 - Si la representación precisa mas de 6 espacios se imprimirán 6 asteriscos para indicar que se ha producido un error. (*****)

2.1 NÚMEROS ENTEROS

Ejemplos (cont.):

- Sentencia:

```
Print ' (I3) ', 10
```

Salida

```
| 10
```

- Sentencia:

```
Print ' (I3) ', -10
```

Salida

```
| -10
```

- Sentencia:

```
Print ' (I3) ', 307
```

Salida

```
| 307
```

- Sentencia:

```
Print ' (I3) ', 1004
```

Salida

```
| ***
```

2.1 NÚMEROS REALES

Ejemplos:

- F6.2 → Se imprimirá un real ocupando 6 espacios en total, ocupando 2 la parte decimal (el punto siempre ocupa un espacio):
 - Parte entera: al igual que en el caso entero se ajusta a la derecha y se rellena de espacios a la izquierda teniendo en cuenta el signo en caso negativo.
 - Parte decimal: Si tiene menos de 2 cifras decimales se completa con ceros a la derecha.
 - Si la parte entera precisa mas espacios se imprimirán 6 asteriscos para indicar que se ha producido un error. (*****)

2.1 NÚMEROS REALES

Ejemplos (cont.):

- Sentencia:

```
Print '(F8.3)', 100.6438
```

Salida

```
| 100.644
```

- Sentencia:

```
Print '(F8.3)', -100.64
```

Salida

```
|-100.640
```

- Sentencia:

```
Print '(F8.3)', -1000.64
```

Salida

```
|*****
```

2.3 CADENAS DE CARACTERES

- El descriptor A se puede utilizar de dos formas:
 - $A \rightarrow$ se imprime la cadena de caracteres ocupando el espacio que necesite
 - $An \rightarrow$ la cadena se imprimirá ocupando n espacios:
 - *Si la cadena tiene menos caracteres se ajusta por la derecha y se rellena de blancos por la izquierda*
 - *Si la cadena tiene más de n caracteres se imprimen los n primeros caracteres empezando por la izquierda*

2.3 CADENAS DE CARACTERES

Ejemplos:

- Sentencia:

```
Print '(A)', "Hola ¿Cómo estas?"
```

Salida

```
| Hola ¿Cómo estas?
```

- Sentencia:

```
Print '(A20)', "Hola ¿Cómo estas?"
```

Salida

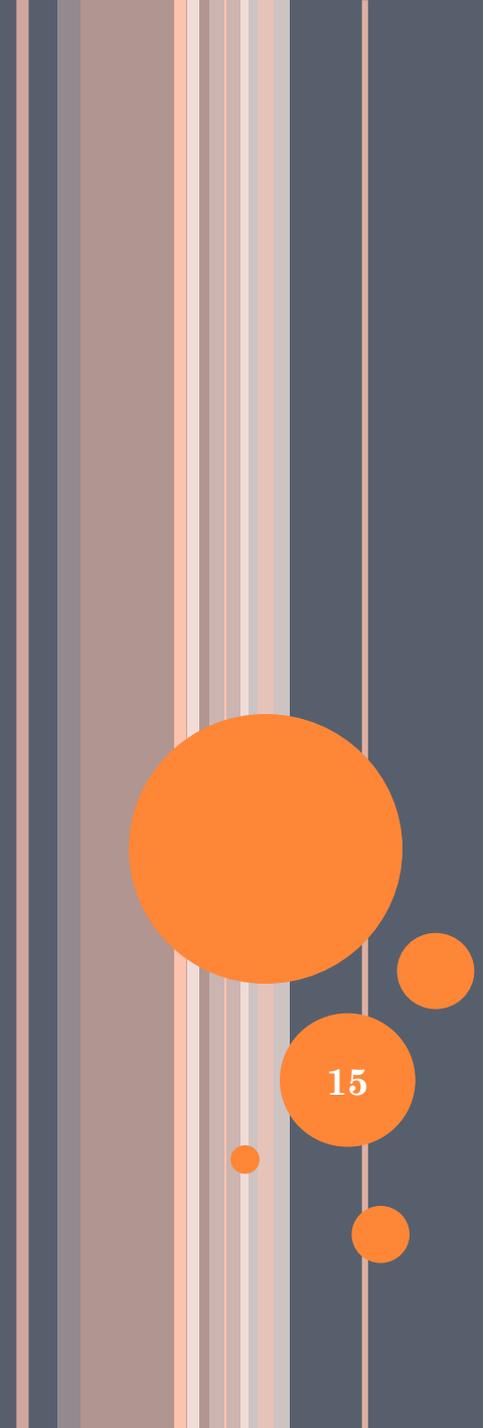
```
|   Hola ¿Cómo estas?
```

- Sentencia:

```
Print '(A10)', "Hola ¿Cómo estas?"
```

Salida

```
| Hola ¿Cómo
```



3. SENTENCIAS DE ENTRADA CON FORMATO

15

3. SENTENCIAS DE ENTRADA CON FORMATO

La forma general de la sentencia READ es la siguiente:

READ formato, lista_de_variables

donde *formato* es muy similar a lo visto anteriormente a excepción de que aquí no pueden aparecer constantes con cadenas de caracteres y que la presencia de comas como separador se ignora.

3. SENTENCIAS DE ENTRADA CON FORMATO

- Salto de caracteres. Se pueden utilizar los descriptores de posición horizontal X y T para ignorar caracteres en la línea de entrada.
 - Ejemplos
 - **Read** ' (2X,I2,3X,I3) ' , numero1,numero2
se guardara en la variable numero1 lo que aparezca a partir del carácter 2 y en numero2 lo que aparezca a partir del carácter 3 después de haber leído la numero1.
Si escribimos
n=10 m=354
en numero1 guardaremos el 10 y en numero2 el 354.
 - **Read** ' (T2,I2,T8,I3) ' , numero1,numero2
se guardara en la variable numero1 lo que aparezca a partir del carácter 2 y en numero2 lo que aparezca a partir del carácter 8 contado desde el inicio de línea. En el ejemplo anterior devolvería lo mismo
- Líneas de entrada múltiples. Si se encuentra el descriptor salto de línea / se comienza a leer una nueva línea.

3.1 VARIABLES ENTERAS

- La entrada de datos se produce en una sola línea.
- Los datos pueden prepararse con o sin separadores (blancos), pero en cualquier caso deben ajustarse exactamente a la anchura de campo prevista.
- Ejemplo

Integer I, J, K

READ '(I5,I10,I5)', I, J, K

- Entrada: 12345123456789054321
Valores: I=12345, J=1234567890, K=54321
- Entrada: 12345 123456789 5432
Valores: I=12345, J= 123456789, K= 5432

2.1 VARIABLES REALES

- La entrada de datos se produce en una sola línea.
- Los valores pueden prepararse con o sin puntos decimales. Si no se introducen puntos decimales, los puntos se suponen exactamente donde lo indica el formato.
- Si se definen los puntos en los valores de entrada, esta definición tiene prioridad sobre la indicada en los formatos.

- Ejemplo:

Real I, J, K

READ '(F6.2,F5.3,F3.1)', I, J, K

- Por ejemplo, para leer los valores **123.56**, **7.901** y **2.4** algunas posibilidades de entrada son:
 - 123.567.9012.4
 - 12356 7901 24
 - 123.56 79012.4

2.3 CADENAS DE CARACTERES

- Se leen tantos caracteres de la línea de entrada como longitud o n° de caracteres tienen las variables correspondientes.

- Ejemplo

READ ' (A3 , A5 , A3 , A7) ' , A , B , C , D

- Entrada: “La casa es azul”
Valores: A=“La ”, B=“casa ”, C=“es ”, D=“azul ”
- Entrada: “Las casas son azules”
Valores: A=“Las”, B=“ casa ”, C=“s s”, D=“on azul”
- Entrada: “Lascasasonazules”
Valores: A=“Las”, B=“casas”, C=“son”, D=“azules”

BIBLIOGRAFÍA

- Javier García de Jalón, Francisco de Asís de Ribera, Junio 2004. *“Aprenda Fortran 8.0 como si estuviera en primero”*. Escuela Técnica Superior de Ingenieros Industriales. Universidad Politécnica de Madrid