

Notas (breves) de FORTRAN 90.

IFormatos, ciclos DO y archivos.

Una diferencia que veremos aquí se refiere al uso de formatos. En Fortran 90 no utilizamos etiquetas, con lo cual los formatos no se colocan en una sentencia `FORMAT`, sino que se colocan como segundo argumento en el `WRITE` o en el `READ`, por ejemplo:

```
WRITE(*,"(I3,2x,F5.2)") K, A
```

En algunos casos, cuando la expresión del formato es muy larga, puede ser engorroso colocarlo en el `READ` o en el `WRITE`. Esto podemos subsanarlo utilizando una variable de tipo carácter adecuadamente definida, a la que le asignamos la cadena de caracteres que define el formato. Nuestro ejemplo anterior, ahora quedaría escrito de la siguiente manera:

```
...  
CHARACTER(30) :: FORM  
...  
FORM = "(I3,2X,F5.2)"  
WRITE(*,FORM) K, A
```

En cuanto a las estructuras de repetición (Ciclos DO) en Fortran 90 no hay diferencias con el uso o la sintaxis del Fortran 77.

Con respecto al manejo de archivos, es posible asignar un valor variable al número de unidad lógica, y

en forma similar asignarle un nombre variable.

```
N = 15  
DO I=10,N  
    OPEN(I, FILE = NOM(I), STATUS = "OLD")  
ENDDO
```

Así abriremos 6 archivos con número de unidad entre 10 y 15, con los respectivos nombres almacenados en el vector `NOM`. De acuerdo con la cláusula `STATUS` los archivos son existentes, como habíamos visto en Fortran 77.

Además, al igual que en Fortran 77, existe la cláusula `ACTION`. Ésta nos permite que, si un archivo es sólo de lectura, no podamos escribir en él, evitando accidentes de sobrescritura. De igual manera, si es de escritura, el programa no nos permitirá leerlo. Por ejemplo, queremos que *Arch.dat* contenga los datos para el programa (lectura) y *Arch.sal* contenga los resultados (escritura), entonces escribiremos:

```
OPEN(10, FILE = "Arch.dat", STATUS = "OLD", &  
    ACTION = "READ")  
OPEN(11, FILE = "Arch.sal", STATUS = "OLD", &  
    ACTION = "WRITE")
```