

Control FPWIN Pro

Software Según el Estándar IEC 61131-3

| | |
|----------------|------------------------------------|
| Título | Históricos de Datos. Arrays |
| Versión | 1.000 |

Definición de Array

Se define un Array como un conjunto de datos **consecutivos y del mismo tipo** (BOOL, INT, STRING, etc.), que tienen el **mismo nombre** y se diferencian por su **índice**. Una definición más corta es que un Array es una **tabla de datos** del mismo tipo.

Cada dato almacenado se le denomina **Elemento del Array**. A cada elemento del array le corresponde un número exclusivo llamado **Índice**.

Las variables del tipo Array han de ir siempre acompañadas de corchetes [] y en el interior de dicho corchete se incluye el índice.

La definición de una array se realiza de la siguiente forma:

NOMBRE [INI..FIN] OF TYPE

Donde,

INI es el valor entero inicial del bloque (ejemplo → 10)

FIN es el valor entero final del bloque. (Ejemplo → 122). Siempre INI<FIN

TYPE es el tipo de dato (INT, BOOL, REAL, DINT, STRING,...)

En este caso, el array está entre 10 y 122 elementos (ambos inclusive). Esto implica que el tamaño del array es de 113 elementos.

Si se define el siguiente array de datos booleanos comenzando en el R0:

```
VAR_GLOBAL g_abArray [R0] ARRAY [0..2] OF BOOL [3(FALSE)]
```

Nota. Es el número de ELEMENTOS y no las posiciones de memoria (DT, R, FL...) que ocupa.

En este caso:

```
g_abArray[0] → R0 //Primer elemento del array
g_abArray[1] → R1 //Segundo elemento del array
g_abArray[2] → R2 //Tercer Elemento...
```

Entre los corchetes se puede insertar una variable del tipo entero (llamada **Índice o Puntero**) y que indica la posición de la tabla de acceso (ej.: g_abArray [iPuntero]).

```
Si iPuntero=0 → g_abArray [iPuntero]=R0
Si iPuntero=0 → g_abArray [iPuntero]=R1
Si iPuntero=0 → g_abArray [iPuntero]=R2
```

Mediante funciones de incrementar o decrementar sobre el puntero, el programador se puede desplazar por la tabla y seleccionar el elemento que considere en cada momento.

Los arrays pueden ser de hasta 3 dimensiones respetando la nomenclatura anterior, separando cada dimensión por comas.

ARRAY [INI1..FIN1,INI2..FIN2,INI3..FIN3] OF TYPE

Se verá el direccionamiento de los arrays sobre los siguientes ejemplos:

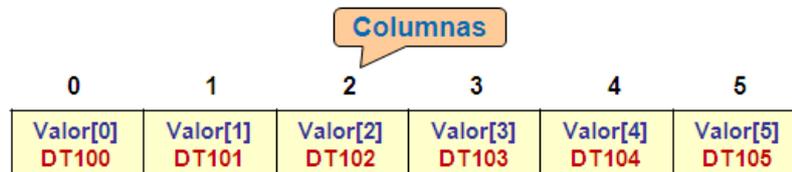
| | | | |
|------------|-------------------------|-------|-------------------------------|
| VAR_GLOBAL | g_aArray_Unidimensional | DT100 | ARRAY [0..5] OF INT |
| VAR_GLOBAL | g_aArray_bidimensional | DT110 | ARRAY [0..5,0..2] OF INT |
| VAR_GLOBAL | g_aArray_tridimensional | DT130 | ARRAY [0..5,0..2,1..4] OF INT |

Nota: Al indicarle al FPWIN PRO que cree una variable del tipo Array siempre crea una variable de una dimension con el indice **0..2**, es decir, de 3 elementos. Estos valores se han de modificar mediante teclado para adaptarlos a la aplicación.

Array Unidimensional:

| | | | |
|------------|-------------------------|-------|---------------------|
| VAR_GLOBAL | g_aArray_Unidimensional | DT100 | ARRAY [0..5] OF INT |
|------------|-------------------------|-------|---------------------|

Un array unidimensional es una fila.
Es una matriz de **1 x C**, donde **C** es el número de columnas.



En el siguiente ejemplo, se escribe en un array unidimensional valores de 10 en 10:

```

(* Introducir valores en todos los elementos
del array de tal manera que se incrementen de
10 en 10. El primer elemento valdrá 10 *)

iValor:=10;

For iIndice := 0 To 5 Do
  g_aiTabla_Datos[iIndice]:=iValor;
  iValor:=iValor+10;
End_for;

```

g_aiTablaDatos =

| | | | | | |
|----|----|----|----|----|----|
| 10 | 20 | 30 | 40 | 50 | 60 |
|----|----|----|----|----|----|

iValor =

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|

Array Bidimensional:

| | | | |
|------------|------------------------|-------|--------------------------|
| VAR_GLOBAL | g_aArray_bidimensional | DT110 | ARRAY [0..5,0..2] OF INT |
|------------|------------------------|-------|--------------------------|

Un array bidimensional es una tabla.

Es una matriz de **F x C**, donde **F** es el número de filas y **C** el número de columnas.

| | 0 | 1 | 2 |
|---|---------------------|---------------------|---------------------|
| 0 | Valor[0,0] DT110 | Valor[0,1] DT111 | Valor[0,2] DT112 |
| 1 | Valor[1,0] DT113 | Valor[1,1] DT114 | Valor[1,2] DT115 |
| 2 | Valor[2,0] DT116 | Valor[2,1] DT117 | Valor[2,2] DT118 |
| 3 | Valor[3,0] DT119 | Valor[3,1] DT120 | Valor[3,2] DT121 |
| 4 | Valor[4,0] DT122 | Valor[4,1] DT123 | Valor[4,2] DT124 |
| 5 | Valor[5,0] DT125 | Valor[5,1] DT126 | Valor[5,2] DT127 |

Columnas (0, 1, 2)

Filas (0, 1, 2, 3, 4, 5)

En el siguiente ejemplo, se inicializan a 0 los elementos de una tabla de datos:

| | | | | | |
|---|-----|----------|--------------------------|---------|------------------------------------|
| 0 | VAR | iFila | INT | 0 | Puntero para recorrer las filas |
| 1 | VAR | iColumna | INT | 0 | Puntero para recorrer las columnas |
| 2 | VAR | aiDatos | ARRAY [1..5,0..3] OF INT | [20(0)] | Array de 2 dimensiones |

```
(*Puntero iFila para recorrer las filas*)
(*Puntero iColumna para recorrer las filas*)

For iFila := 1 To 5 By 1 Do
  (* BY es el incremento del puntero*)
  (* Si se omite es 1 *)
  For iColumna := 0 To 3 Do
    aiDatos[iFila,iColumna]:=0;
  End_for;
End_for;
```

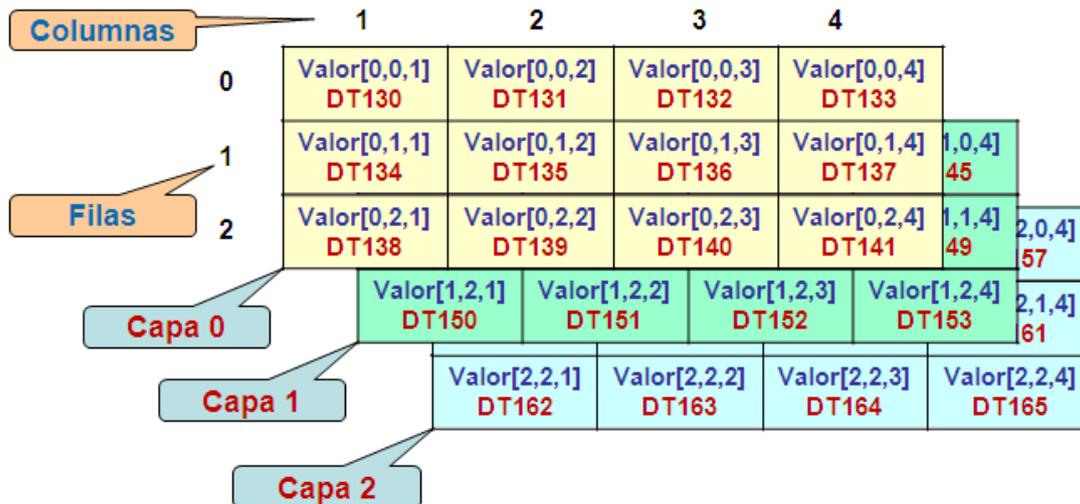
| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | 0 | 0 |

Array Tridimensional:

```
VAR_GLOBAL g_aArray_tridimensional DT130 ARRAY [0..5,0..2,1..4] OF INT
```

Un array tridimensional es como un cubo.

Es un matriz de **c x F x C**, donde **c** es la capa del cubo, **F** es el número de filas y **C** el número de columnas.



Sobre Este Documento

Este documento no tiene carácter oficial ni se podrá responsabilizar a Panasonic Electric Works España por las erratas o información errónea contenida en el mismo, declinando toda responsabilidad por su utilización.