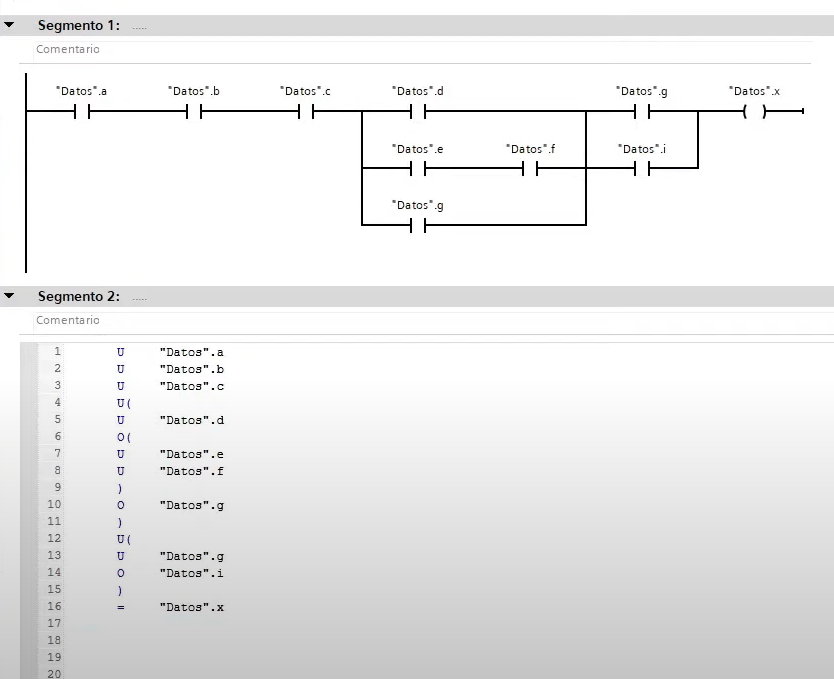
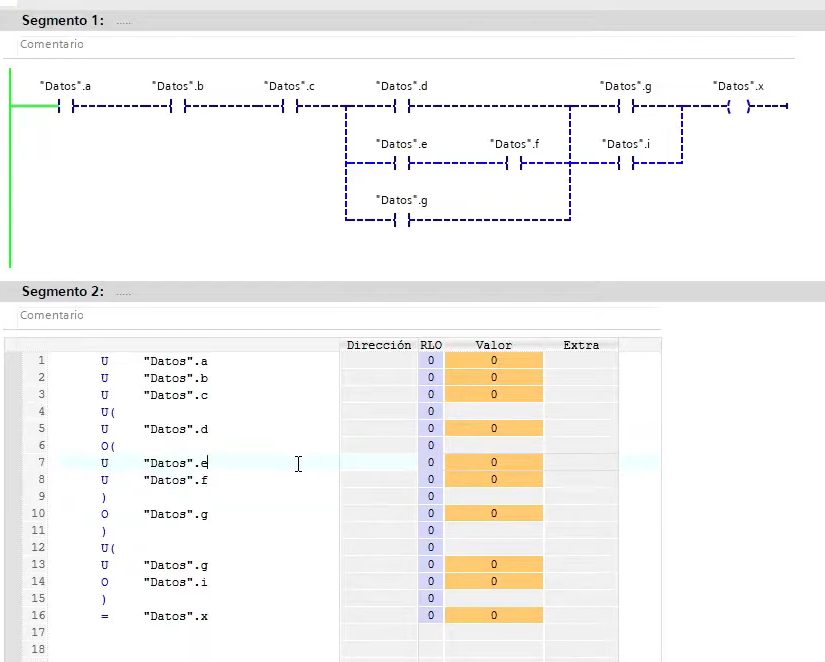
# Lógica con Bits

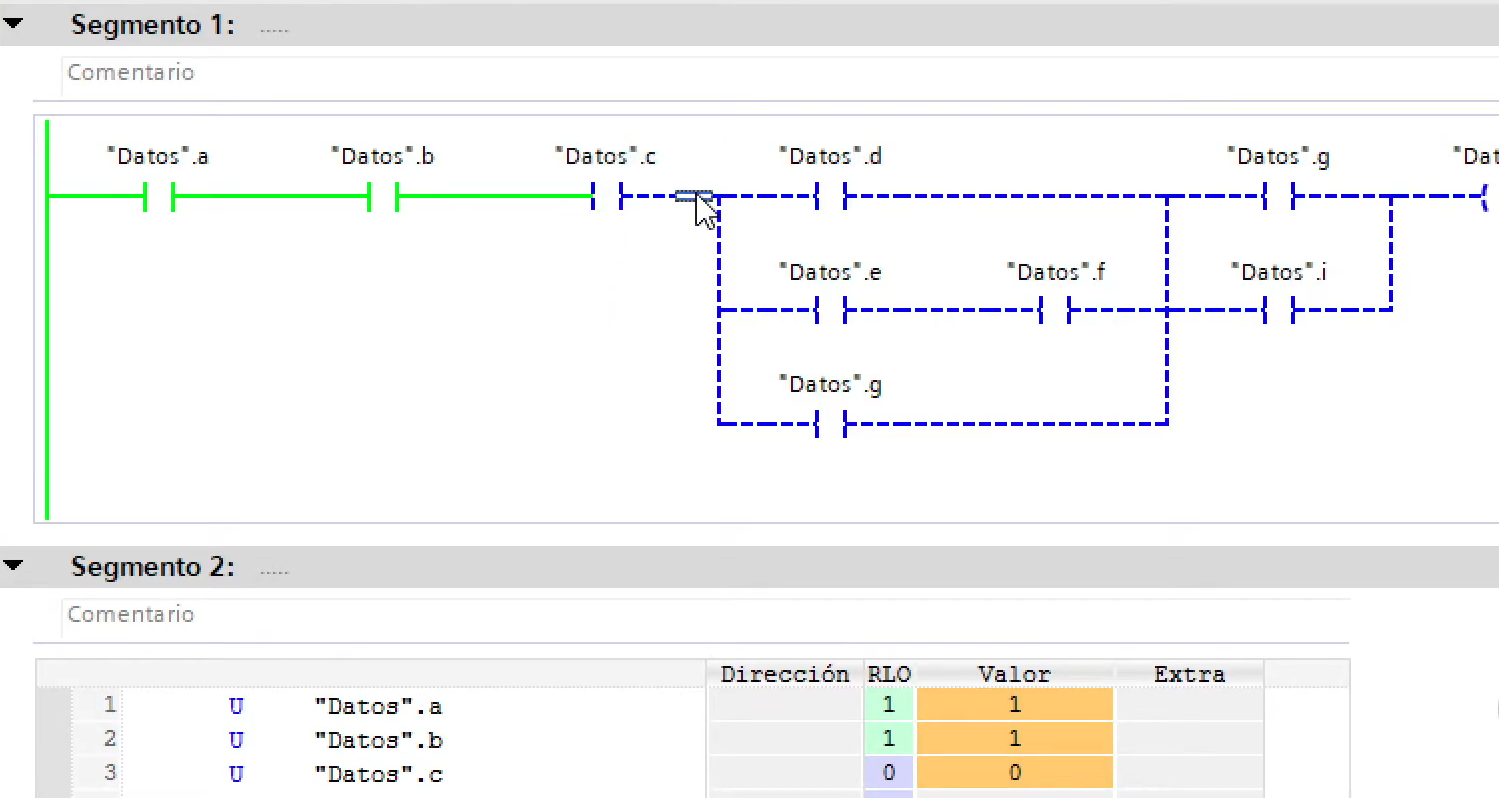
## RLO

El RLO es el estado resultande de las operaciones que hay desde la primera condición hasta la ocndición deseada.

Aunque existe, sólo se muestra de manera direta en AWL. En KOP de manera indirecta iluminando el contacto.







## Saltos en AWL

Los saltos son similares a una instrucción if else. Bajo una condición, podemos programar para que la ejecucción del programa se dirija a un bloque u otro.

SPB: salta si RLO vale 1

SPBN: salta si RLO vale 0

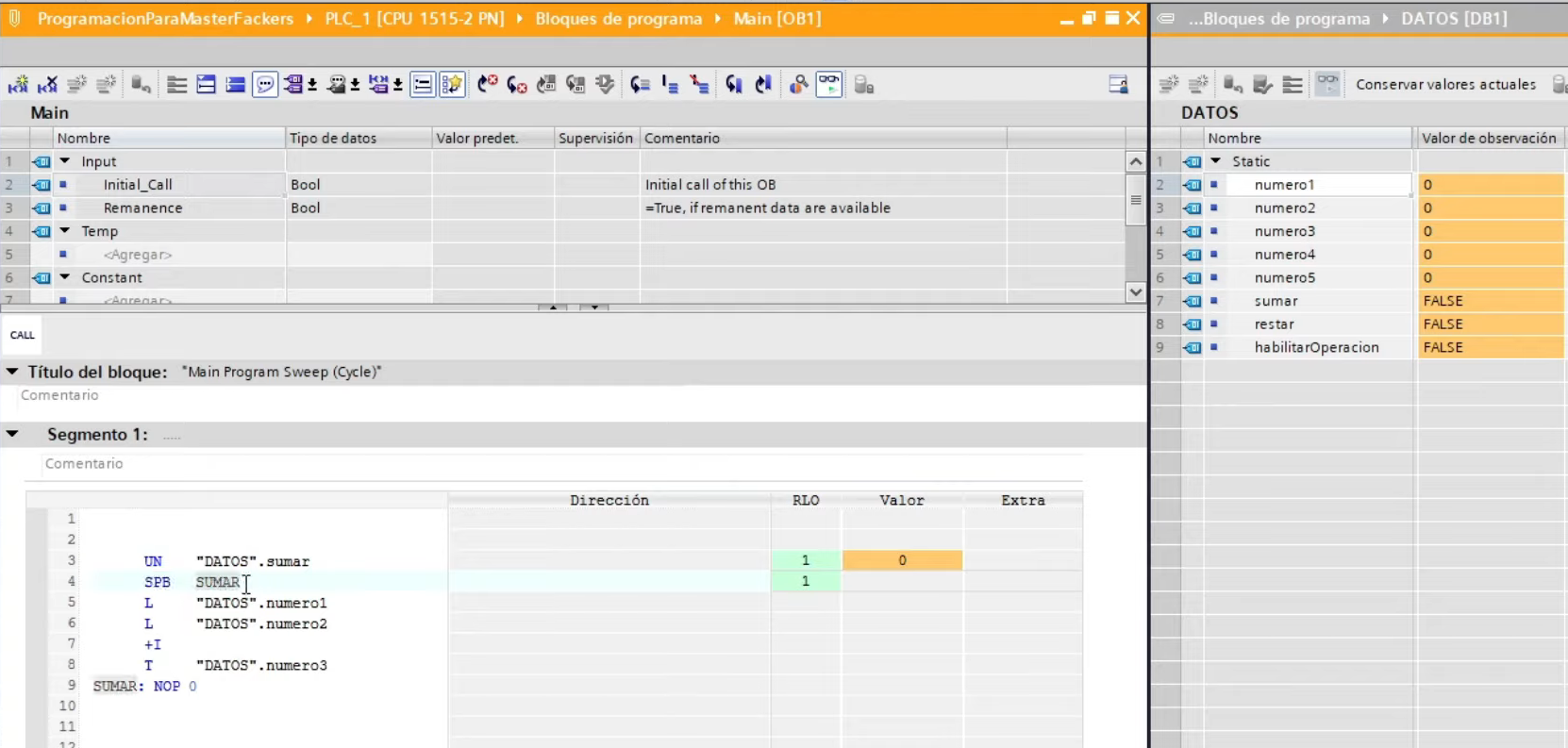
// codigo

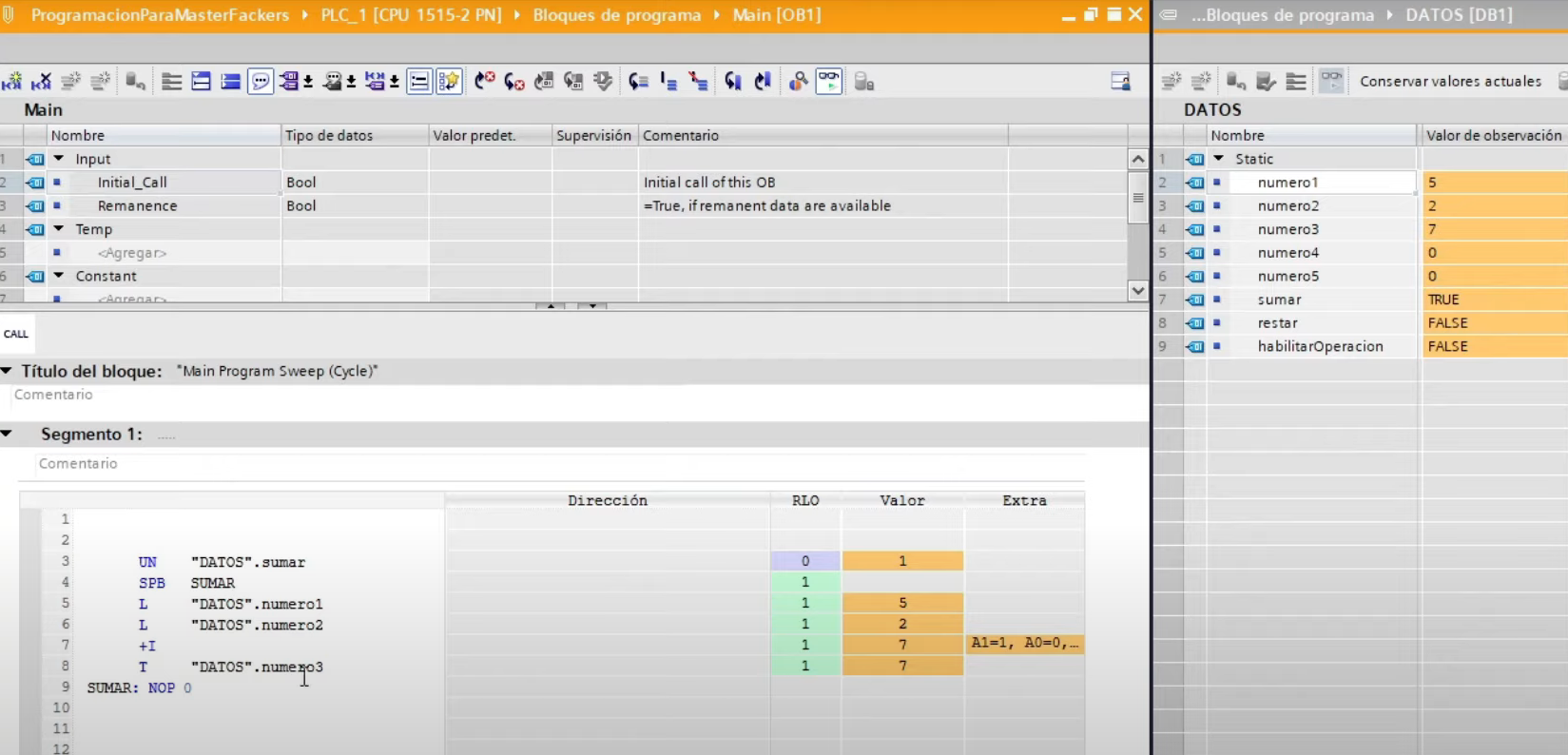
SPB <ediqueta>

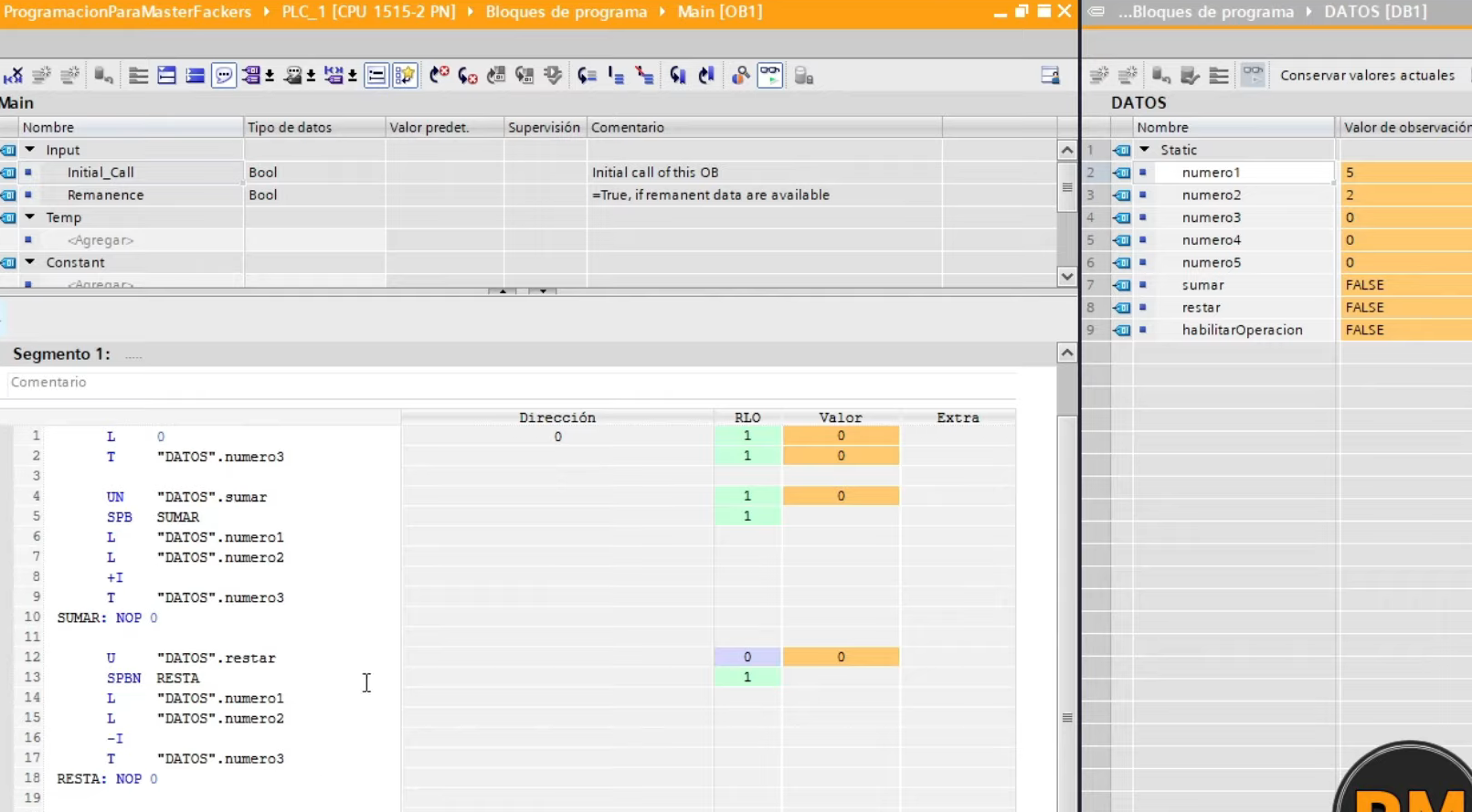
<etiqueta> : NOP 0 // NOP 0 indica instrucción nula.

// codigo aqui

Ejemplo







## Bucles en AWL

Para hacer bucles en AWL se necesita usar saltos.

|  |  |
| --- | --- |
| SCL | AWL |
| num := 0;  FOR #i := 0 to 10 DO  num := Num + 1;  END\_FOR; |  |

## Instrucción AUF

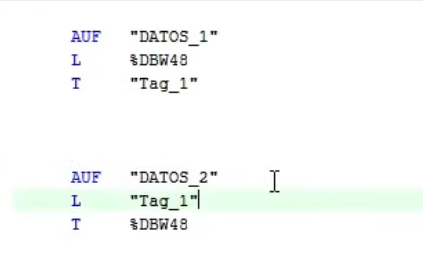
Acceder a un DB

AUF “DB\_Data1” // Accediendo al DB\_Data1

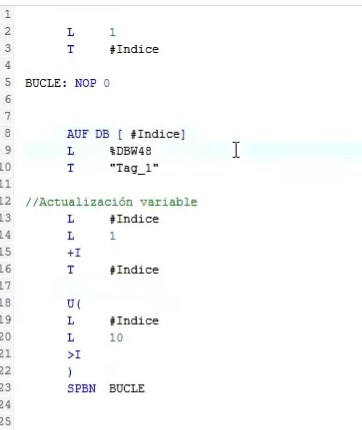
L %DBW48 // Cargamos el valor de la posición (offset) 48 del DB

T “DB\_Data2”.Value1 // Transferimos ese dato a valor Value1 del DB Data2

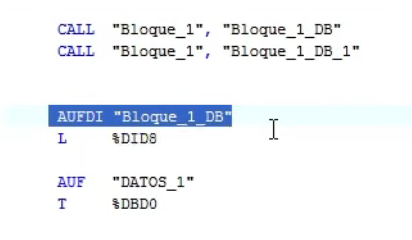
Otra manera



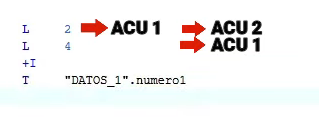
Acceder a varios DBs de forma dinámica: AUF DB [ #indice]



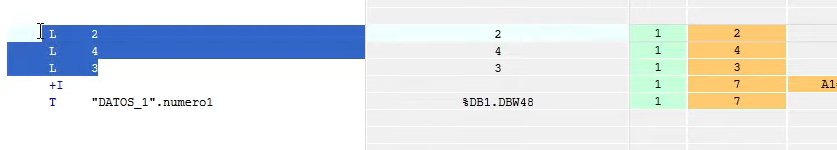
AUFDI (DB de instancia de una función)



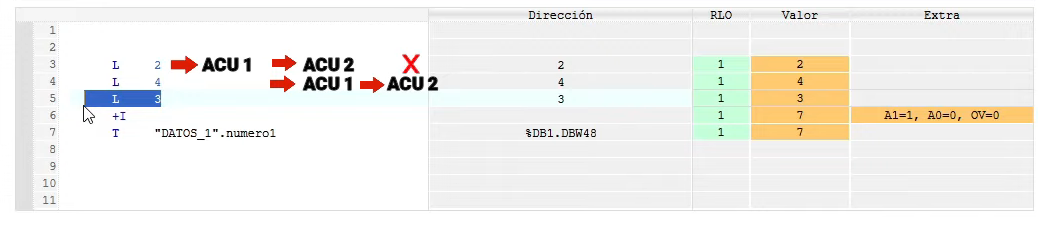
## Acumuladores



Sólo hay 2 acumuladores. En el ejemplo siguiente el L 2 se ignora:

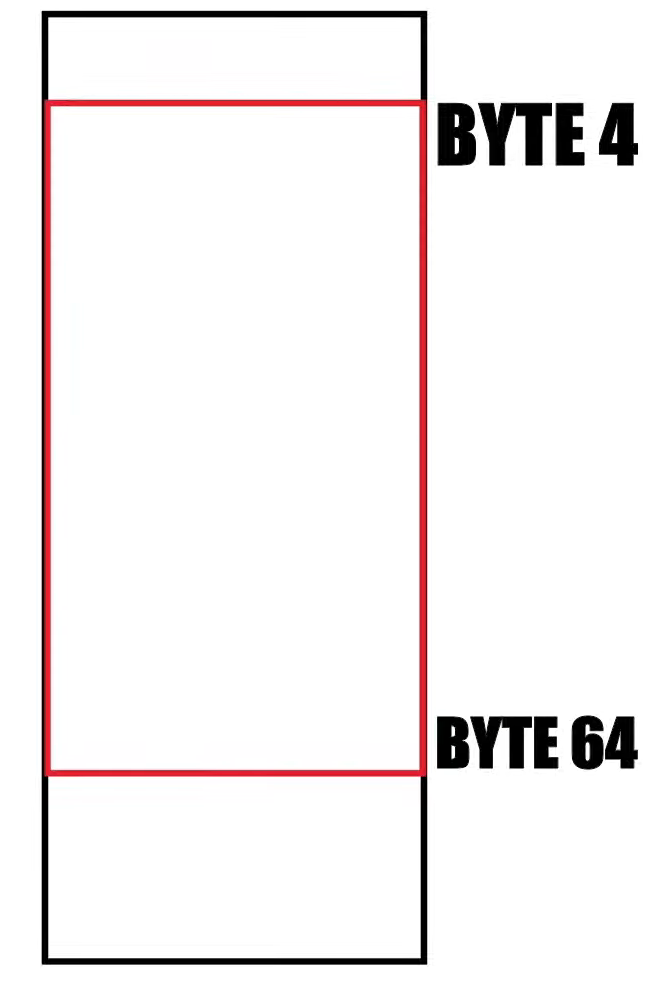


El Acu1 de la primera instrucción se pierde:



## Punteros

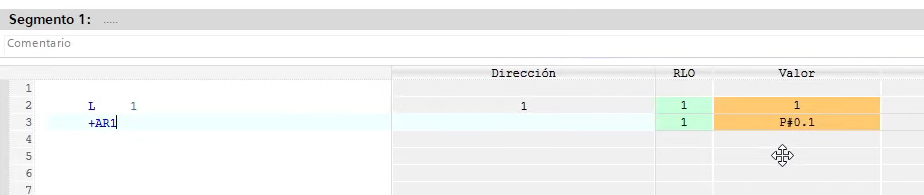
Un putero es la infomación sobre el área de memoria sobre un elemento (DB, variable ...)

Puntero que de un DB del Byte 4 al Byte 64

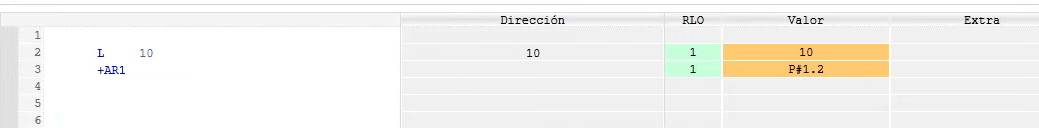
## Registros AR1 y AR2 en AWL

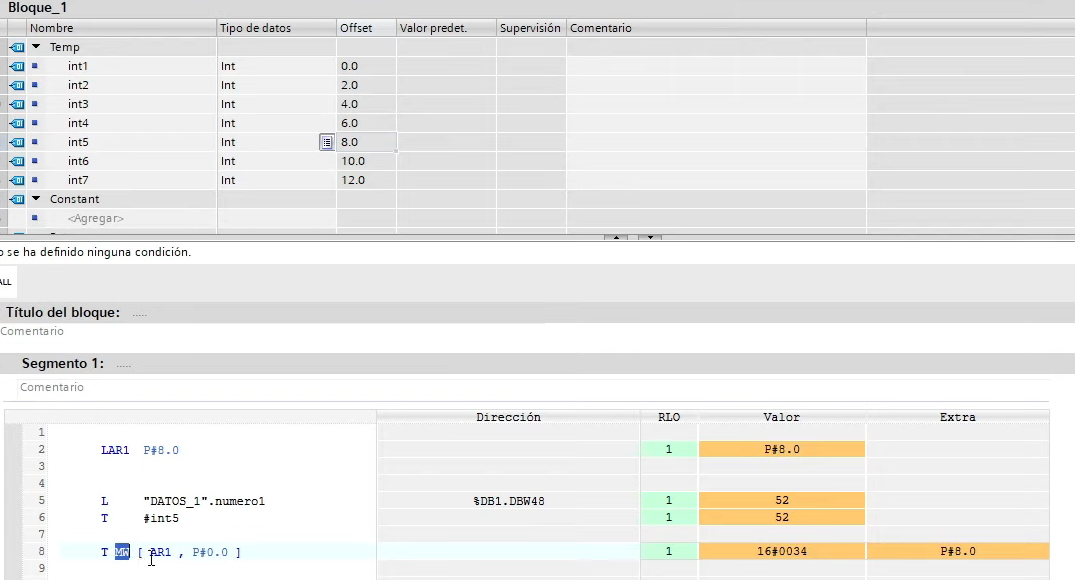
Los registros no almacenan valores, sino direcciones de memoria.

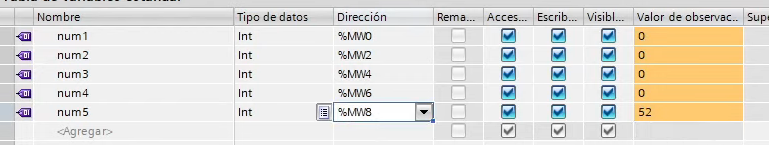
En el Puntero P#0.1 (dirección de memoria) estará almacenado el valor de 1:

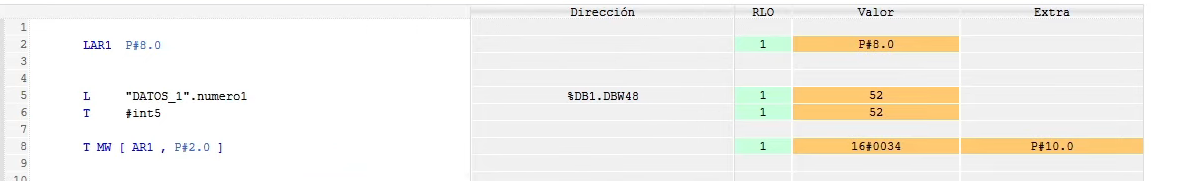


En el Puntero P#1.2 (dirección de memoria) estará almacenado el valor de 10:

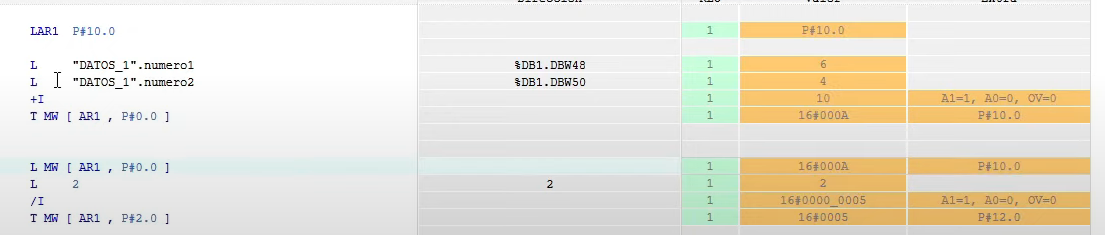








Ejemplo:



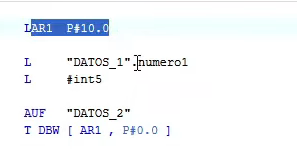
Cargamos el puntero P10.0

Sumamos 2 numero y lo almacenamos en la marca

Cargamos la marca sin offset y dividimos su valor entre dos

Transferimos el resultado a otra marca con offset 2.0 respecto a la otra

Con DB:

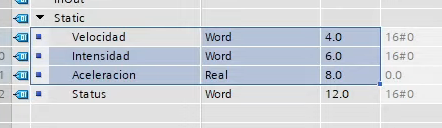


## Punteros ANY

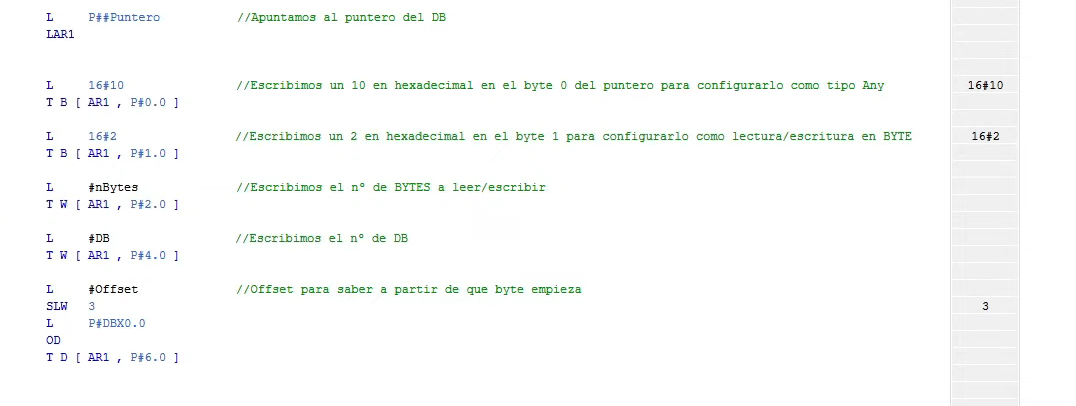
Un puntero ANY es una variable tipo ANY que almacena un área de memoria.

nBytes = 8 Bytes (del 4.0 al 11.7)

Offset = 4 (Empieza en 4.0)



FC\_Puntero



Hola Abel,

Encontré tu canal de Youtube la semana pasada porque estaba buscando una manera de entender AWL. Trabajo con Tia Portal en Alemania desde hace 1 año y estoy con un proyecto donde necesitaba AWL. Me he visto todos tus videos acerca de este lenguaje de instrucciones y me ha ayudado bastente, muchísimas gracias!

Puesto que programo también aplicaciones web con python, me gustaría saber si te interesaría en colaborar conmigo para hacerte un sitio web; el mismo que tienes pero con una pasalera de pago y con cursos de pago, un blog, etc. Un ejemplo sería mi página: [Ingeniería y Programación (ramiboutas.com)](https://ramiboutas.com/).

No te cobraría nada, sería una colaboración en que la acordaríamos un términos y esas cosas. Yo podría escribir artículos en el blog también. No sería hacer la web y yo desentenederme, sino trabajar los 2 conjuntamente en el sitio.

He visto que no dispones de una manera de monetizar o por lo menos no la he encontrado. Creo que se podría hacer algo.

Ya me dices qué te parece.

Gracias por tu atención.

Saludos,

Rami