

**RAPID**

# VAR, Add, Incr, Decr, Clear;

**VAR** num a:=0; → Crea la variable a con valor 0

Add a,1; → suma a la variable “a” 1

Incr a; → suma a la variable “a” 1

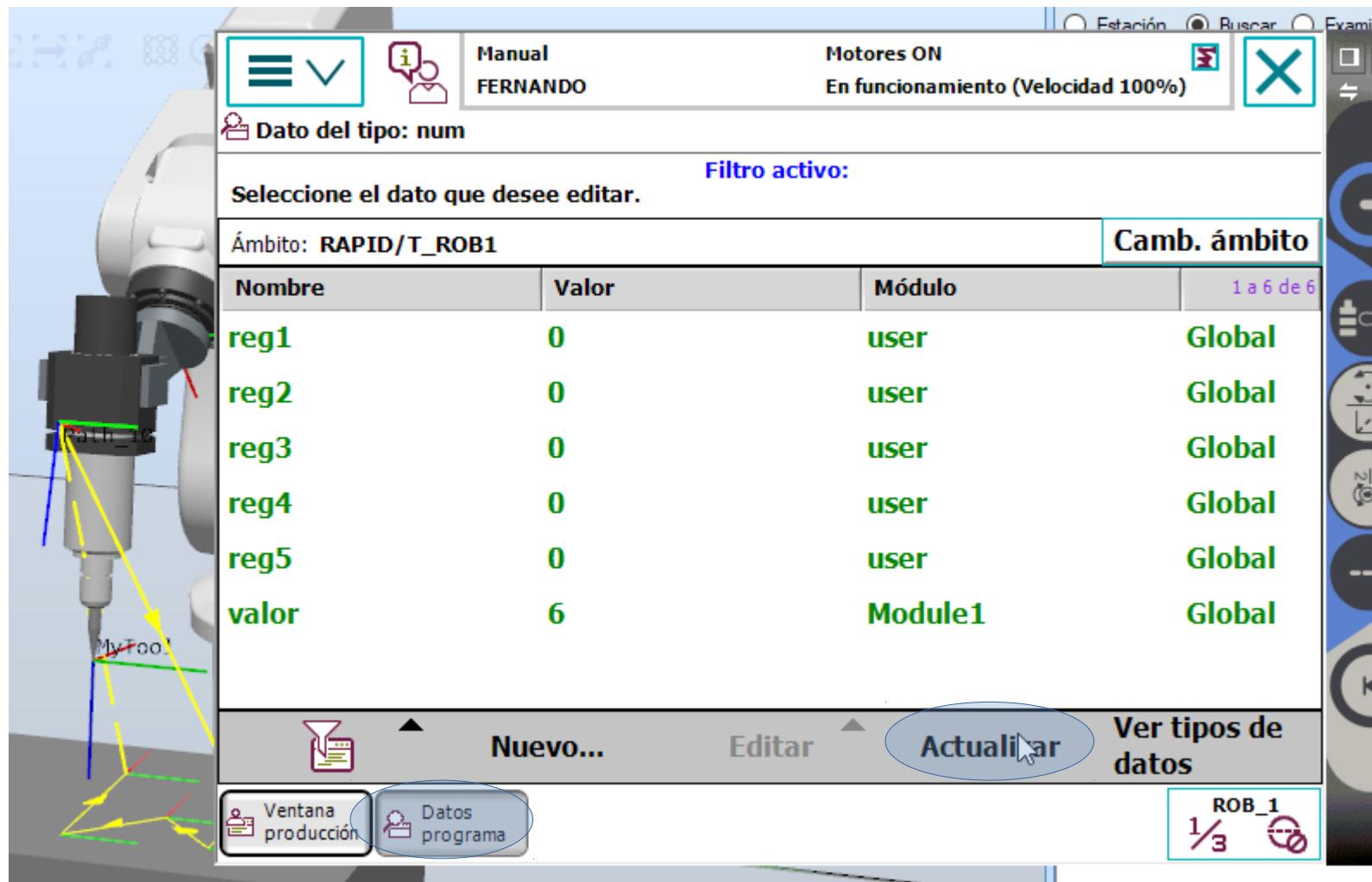
a:=a+1 → suma a la variable “a” 1

Decr a; → resta a la variable “a” 1

Clear a; → pone a cero una variable

A:=0; → pone a cero una variable

# Ver el valor de una variable en flexpendant



# Ver el valor en Robotstudio:RAPID

The screenshot shows the RobotStudio interface with the following components:

- Toolbar:** Archivo, Posición inicial, Modelado, Simulación, Controlador, RAPID (selected), Complementos.
- SolidWorks View:** Shows a 3D model of a robotic arm (IRB120) positioned above a workpiece, with a tool path labeled "Path\_10".
- RAPID Editor:** Displays the RAPID code for the program "T\_ROB1/Module1". The code includes declarations for target points and a main loop that increments a variable "valor".
- Observación de RAPID (RAPID Observation) Table:** A table showing the value of the variable "valor" as 51. The table has columns: Nombre (Name), Valor (Value), Tipo (Type), and Origen (Origin). The "Nombre" column is highlighted with a blue border.

A large arrow points from the text "SE PUEDE INGRESAR EL VALOR DESDE EL TECLADO" at the bottom to the "Valor" column in the table.

Nombre	Valor	Tipo	Origen
valor	51	num	IRB120_3_58/RAPID/T_ROB1/Module1/valor

SE PUEDE INGRESAR EL  
VALOR DESDE EL TECLADO

# WHILE

WHILE a<cajas DO

Incr a;

Acciones;

ENDWHILE

# Ejemplo de instrucciones

ESTACIÓN RAPID.rsstn

```
MODULE Module1
  VAR num a:=0;
  VAR num cajas;
  ****
  PROC main()
    TPErase;
    Clear a;
    TPReadnum cajas,"cuantas cajas quieres soldar";
    TPWrite "vas a soldar "\num:=cajas;
    WHILE a<cajas DO
      Path_10;
      Path_20;
      Incr a;
      TPErase;
      TPWrite "cajas soldadas "\num:=a;
    ENDWHILE
    WaitTime 5;

  ENDPROC
```

CARPETA 19\_PROGRAMAS → PROGRAMA1

# IF - IF \_THEN ELSEIF \_THEN ELSE ENDIF

IF valor=1 GOTO etiqueta;

IF valor=1 procedimientoA;

\*\*\*\*\*

IF valor=1 THEN

acciones;

ELSEIF valor=2 THEN (confirmar)

acciones;

ELSE

acciones;

ENDIF

**FOR i FROM 1 TO valor DO**

FOR i FROM 1 TO valor DO

- Secuencias

ENDFOR

# TEST

TEST valor

CASE 1:

acciones;

CASE 2:

acciones;

CASE 3,4:

acciones;

DEFAULT:

acciones;

ENDTEST

# OFFS

**MoveJ offs (punto, desplazamiento x,y,z),...**

MoveL Offs (Target\_30,0,0,10),v1000,z0,MyTool\WObj:=wobjBASE;

En el siguiente ejemplo, cada vez que realiza el ciclo, aumenta el desplazamiento 10mm

Var num despl=0;

despl:=despl+10;

MoveL Offs (Target\_30,0,0,despl),v1000,z0,MyTool\WObj:=wobjBASE;

## **WAITTIME 2;**

- El programa se para 2 segundos

**TPWrite " ";**

- Pagina 51 manual 04 ABB

**TPWrite " Pulsa marcha";  
TPWrite"" \Num:=a;**

```
PROC main()
    TPWrite "Pulsa marcha";
    WaitTime 5;
    Path_10;
    Path_20;
ENDPROC
```

**TPErase;**

**PROC main()**

**!Añada aquí su código**

TPWrite "Pulsa marcha";

WaitTime 5;

TPErase;

Path\_10;

Path\_20;

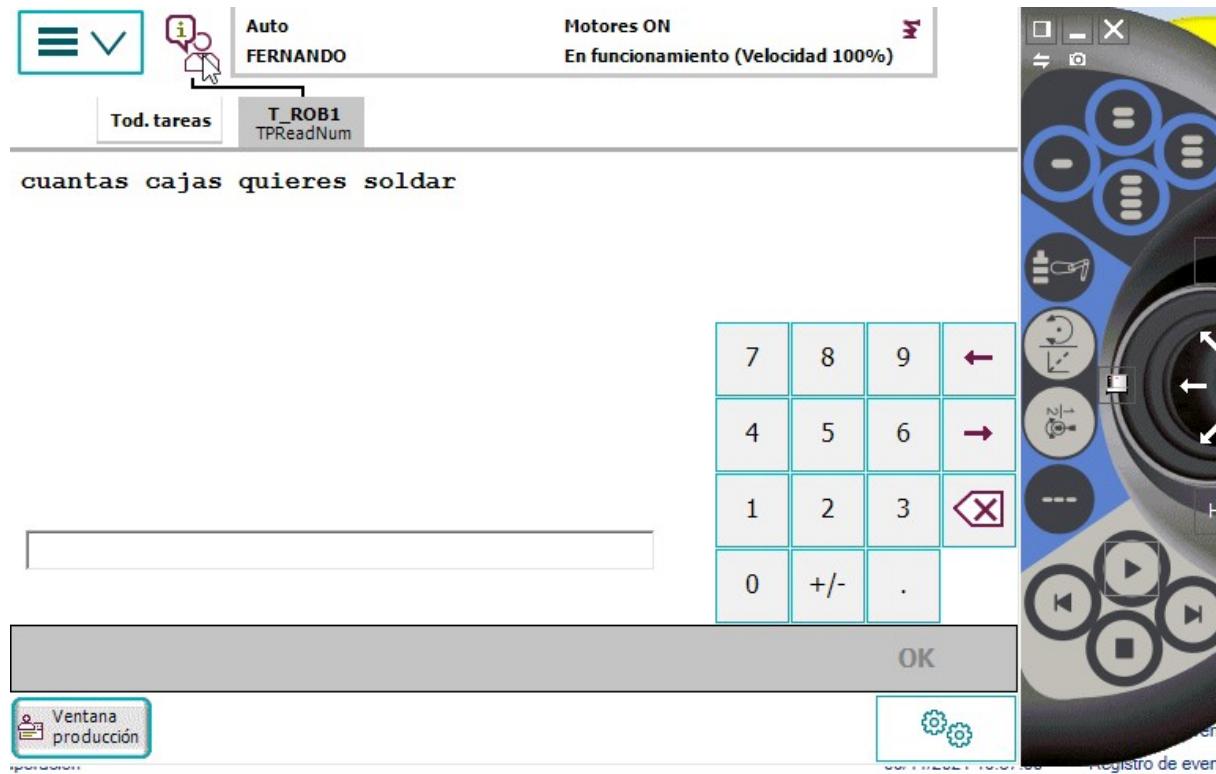
**ENDPROC**

# TPReadnum

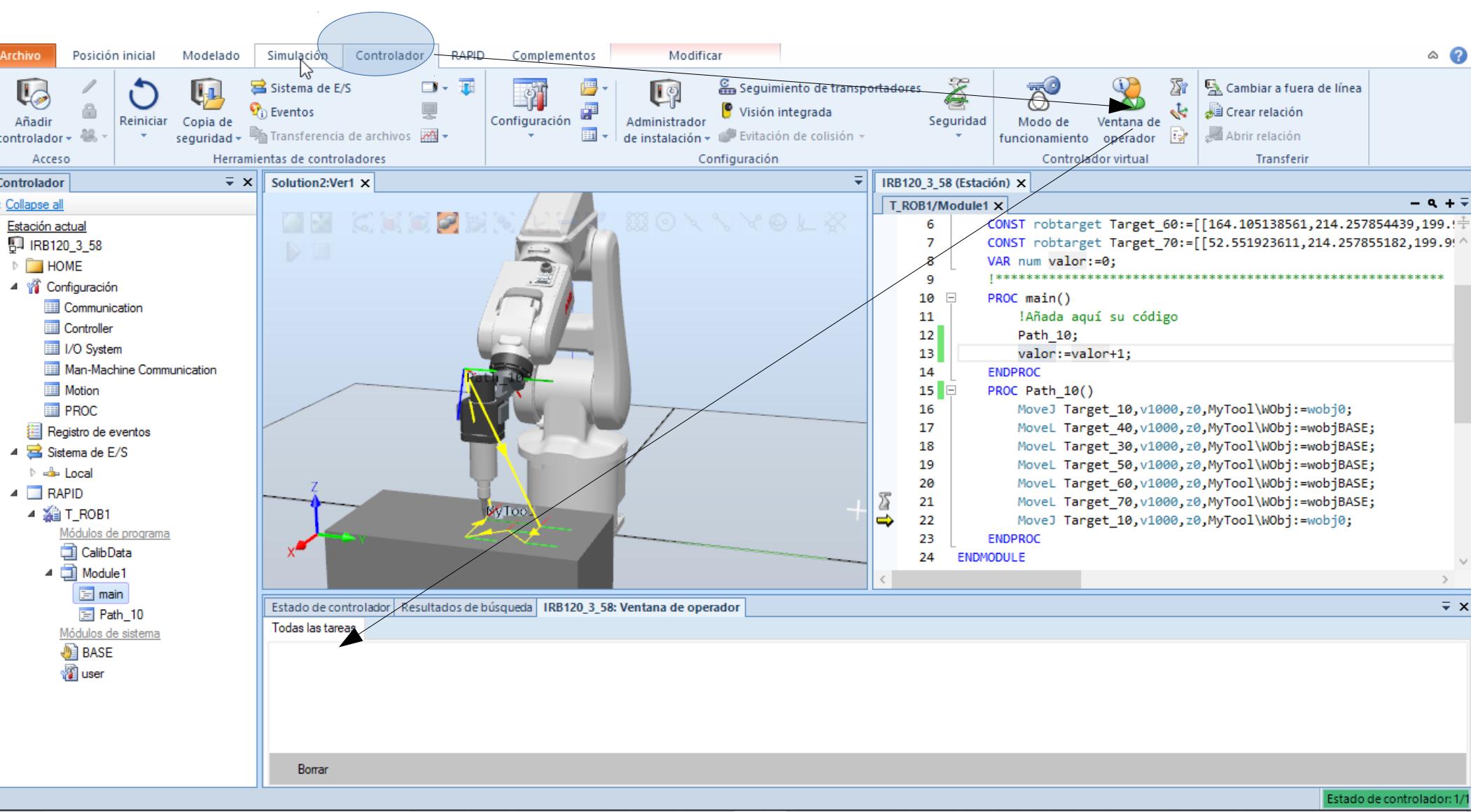
VAR num cajas;

TPReadnum cajas,"cuantas cajas quieres soldar";

→ carga la variable cajas con el numero que introduzcas



# VENTANA DEL OPERADOR NOS EVITA TENER QUE ABRIR LA FLEXPENDANT



# CallByVar

- Para llamar a un procedimiento simplemente ponemos un nombre  
`path_10;`
- Llama a un procedimiento con una raíz común diferenciado por algún carácter  
`CallByVar "NombreComunDelProcedimiento",valor;`  
donde valor puede ser una variable y así llamar a distintos procedimientos con una instrucción