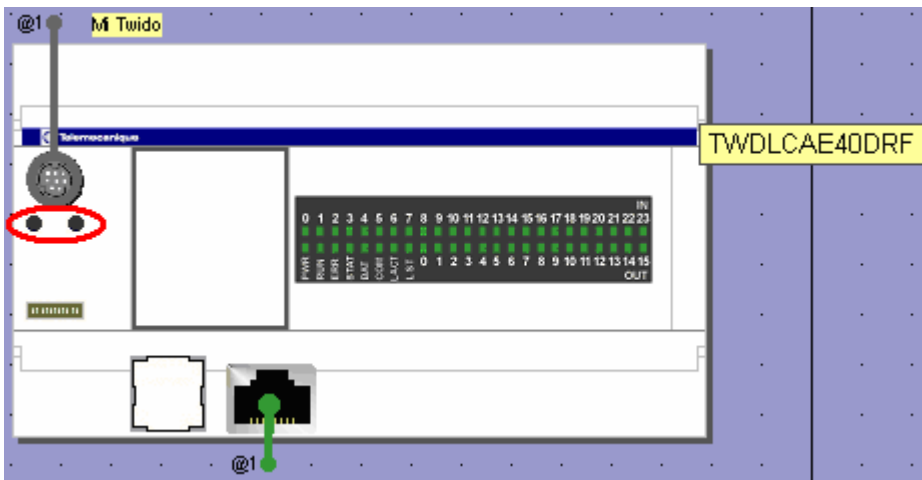




Escalado de 0 -10 de una Entada analógica en Twido

Equipos Utilizados:

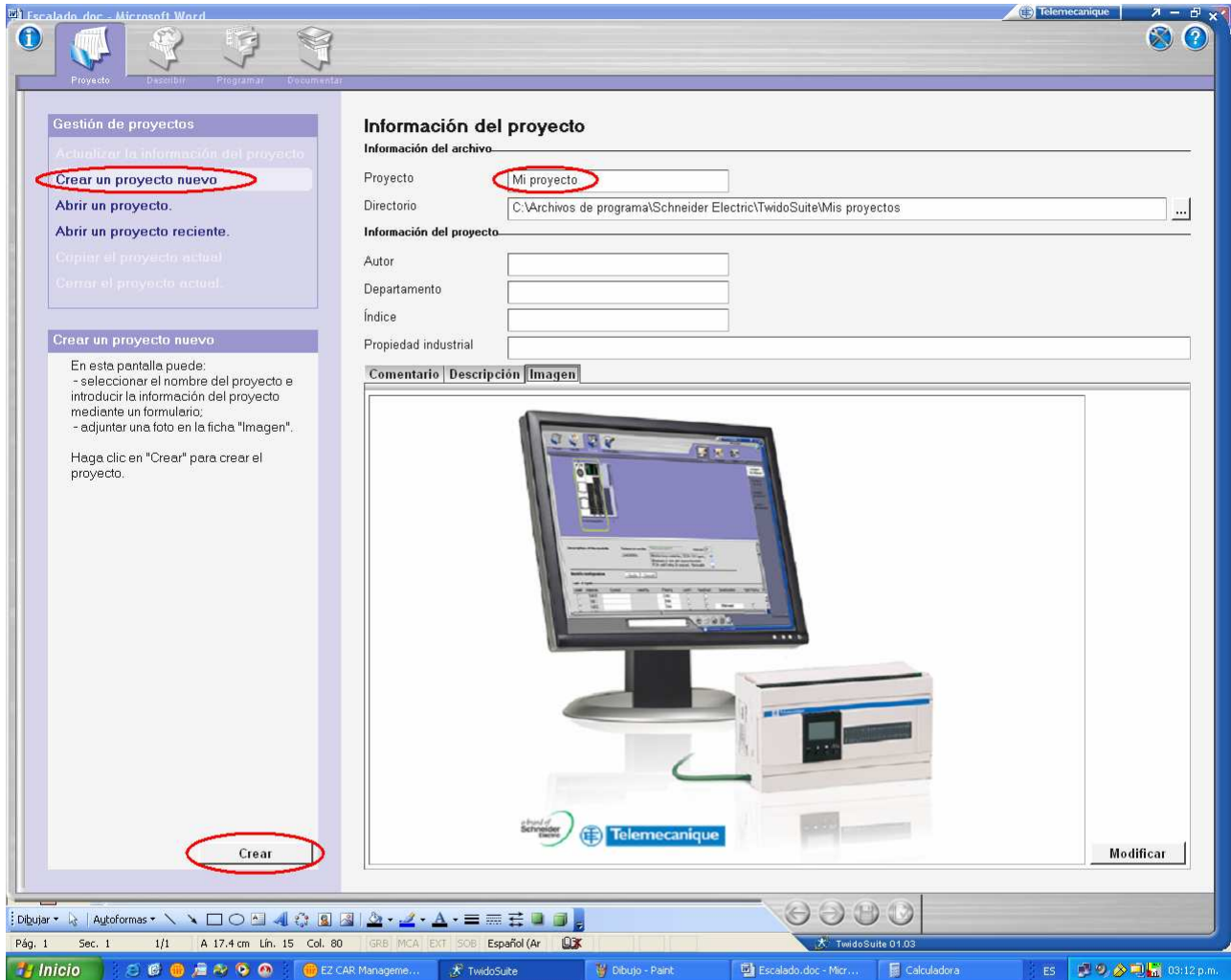
- TWDLCAE40DRF (40 I/O con puerto Ethernet)
- TSXPCX1031 (Cable de programación RS232 – RS485)
- Twido Suite V1.03 (Software de programación Twido)



En este caso la entrada analógica en cuestión será uno de los dos potenciómetros incorporados en la base de este PLC. Estos potenciómetros están mapeados en: %IW0.0 de 10 bits de resolución (0-1023) con un rango de 1024 pasos y %IW0.1 de 9 bits de resolución (0 – 511) con un rango de 512 pasos.

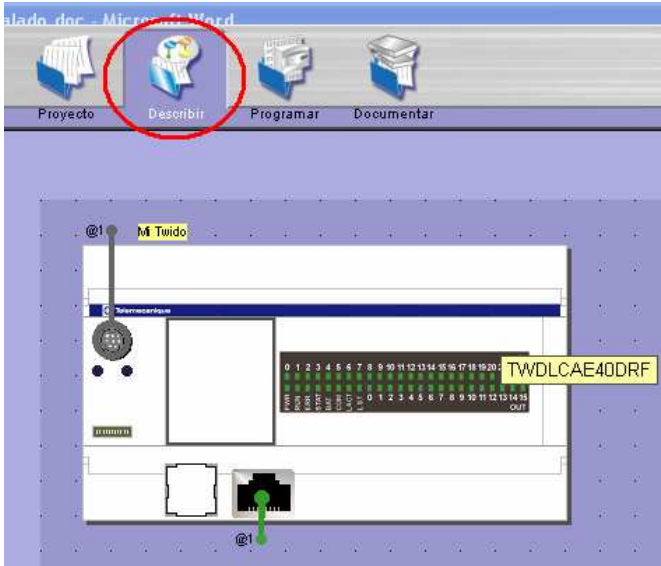


Hay que **Crear un proyecto nuevo**.

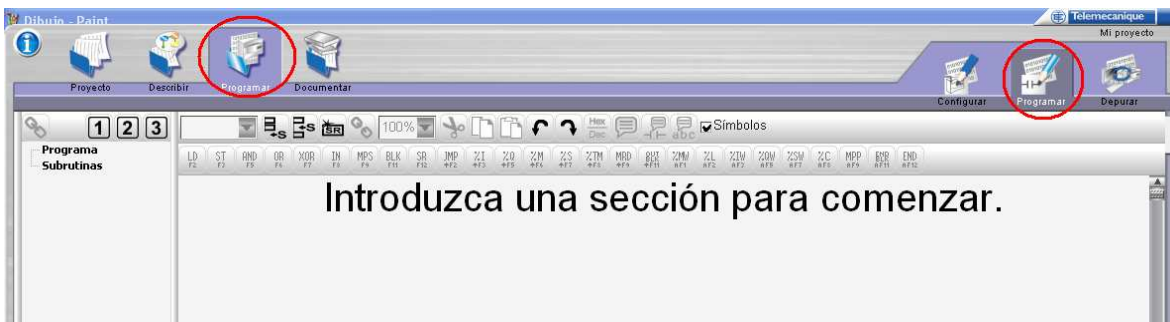




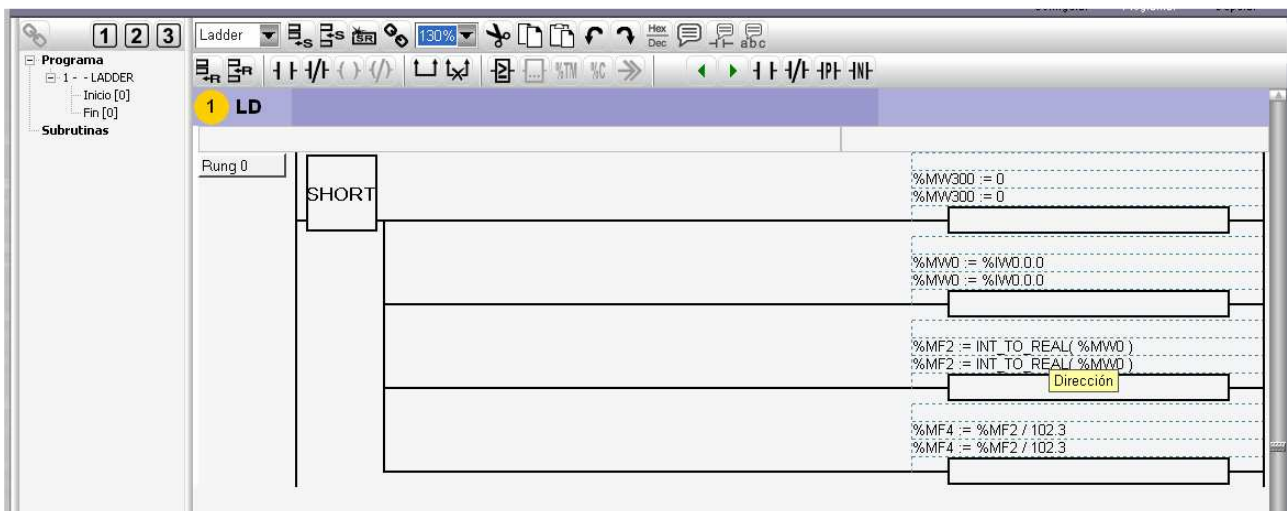
Luego desde la pestaña **Describir**, se arma la configuración.



El próximo paso es desde la pestaña **Programar**, para armar la aplicación.



Aquí se presenta la aplicación hecha en Twido Suite.





Como primer línea de programa se hace una reserva de memoria de 300 posiciones para la aplicación, mediante la asignación %MW300:=0.

```

%MW300 := 0
%MW300 := 0

```

Para no perder precisión en la lectura de la medición de la entrada analógica %IW0.0 hay que trabajar con números flotantes, para esto es necesario convertir el dato en formato entero de la medición, en dato en formato flotante, pero previo a esta conversión hay que asignar a una %MWi el valor de la entrada analógica mediante %MW0:=%IW0.0.

```

%MW0 := %IW0.0
%MW0 := %IW0.0

```

El pasaje de entero a real se hace con la función INT_TO_REAL (Entero a Real).

A continuación se detallan 4 funciones de conversión entre datos enteros y flotantes.

Instrucciones de conversión de entero <-> Flotante

[Temas relacionados](#)



General

Se proponen cuatro instrucciones de conversión.

Lista de las instrucciones de conversión de entero<-> flotante:

| | |
|----------------------|---|
| INT_TO_REAL | conversión de una palabra entera --> flotante |
| DIINT_TO_REAL | conversión de una palabra doble (entera) --> flotante |
| REAL_TO_INT | conversión de una palabra flotante --> entera (el resultado es el valor algebraico más cercano) |
| REAL_TO_DIINT | conversión de una palabra flotante --> entera doble (el resultado es el valor algebraico más cercano) |

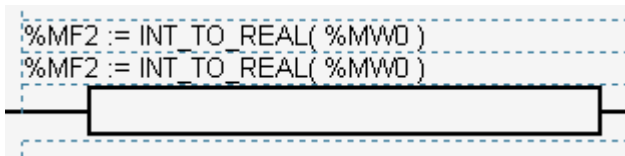
Sintaxis

Operadores y sintaxis (conversión de una palabra entera --> flotante):

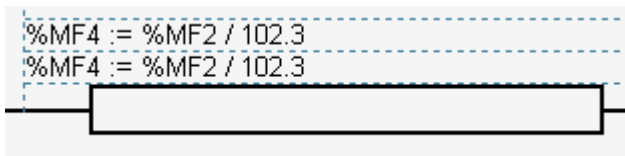
| Operadores | Sintaxis |
|--------------------|-------------------------------|
| INT_TO_REAL | Op1= INT_TO_REAL (Op2) |



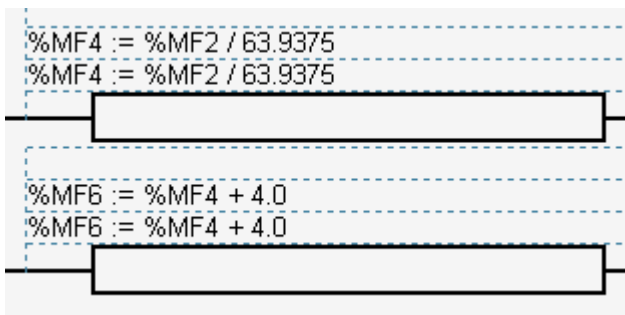
En nuestro ejemplo $\%MF2 := INT_TO_REAL(\%MW0)$, que convierte el valor entero en $\%MW0$ a real en $\%MF2$. Se utiliza $\%MFi$ en lugar de $\%MWi$, porque un valor real ocupa 32 bits y debe ser asignado a un $\%MFi$ (Memory Float) porque es una palabra doble.



Por ultimo es necesario hacer una división para acomodar el rango de variación analógica de 10 bits (0 – 1023), a nuestro rango deseado de 0 a 10, por eso se divide $\%MF2/102.3$ que sale del máximo valor posible en la entrada analógica, dividido por nuestro propio rango de variación.



Para escalar entre 4 y 20, la cuenta seria $\%MF4 := \%MF2 / 63.9375$, y para correr las 4 unidades de la ordenada al origen, se suman 4 a $\%MF4$; entonces $\%MF6 := \%MF4 + 4.0$.



Hay que poner especial atención debido al solapamiento, que existe entre las palabras dobles, por eso el uso de las $\%MFi$ y las $\%MDi$ es de a pares.

Possibilidad de solapamiento entre objetos

Las palabras simples, de doble longitud y flotantes se guardan en el espacio de datos en una misma zona de memoria. Así, la palabra flotante $\%MFi$ y la palabra doble $\%MDi$ se corresponden con las palabras de longitud simple $\%MWi$ y $\%MWi+1$ (la palabra $\%MWi$ contiene los bits menos significativos y la palabra $\%MWi+1$ los bits más significativos de la palabra $\%MFi$).

En la tabla siguiente se muestra el solapamiento de palabras flotantes y palabras dobles internas:

| Flotante y doble | Dirección impar | Palabras internas |
|---------------------|-----------------|-------------------|
| $\%MF0 / \%MD0$ | $\%MF1 / \%MD1$ | $\%MW0$ |
| $\%MF2 / \%MD2$ | $\%MF3 / \%MD3$ | $\%MW1$ |
| $\%MF4 / \%MD4$ | ... | $\%MW2$ |
| ... | $\%MFi / \%MDi$ | $\%MW3$ |
| $\%MFi+1 / \%MDi+1$ | | $\%MW4$ |
| | | $\%MW5$ |
| | | ... |
| | | $\%MWi$ |
| | | $\%MWi+1$ |