

# PLCs DE SCHNEIDER

Ignacio Bravo  
Departamento de Electrónica. Universidad de Alcalá  
Escuela Politécnica Superior. Campus Universitario.  
Ctra. Madrid-Barcelona Km. 33.600  
28871 – Alcalá de Henares (Madrid)  
<http://www.depeca.uah.es> [ibravo@depeca.uah.es](mailto:ibravo@depeca.uah.es)

# ÍNDICE DEL TEMA

---

- 1.- Introducción Schneider Electric**
- 2.- PLCs de Schneider**
- 3.- PLC M340**
- 4.- UNITY PRO**
- 5.- Configuración del Hardware**
- 6.- Descarga de programas**



# 1.- Introducción Schneider

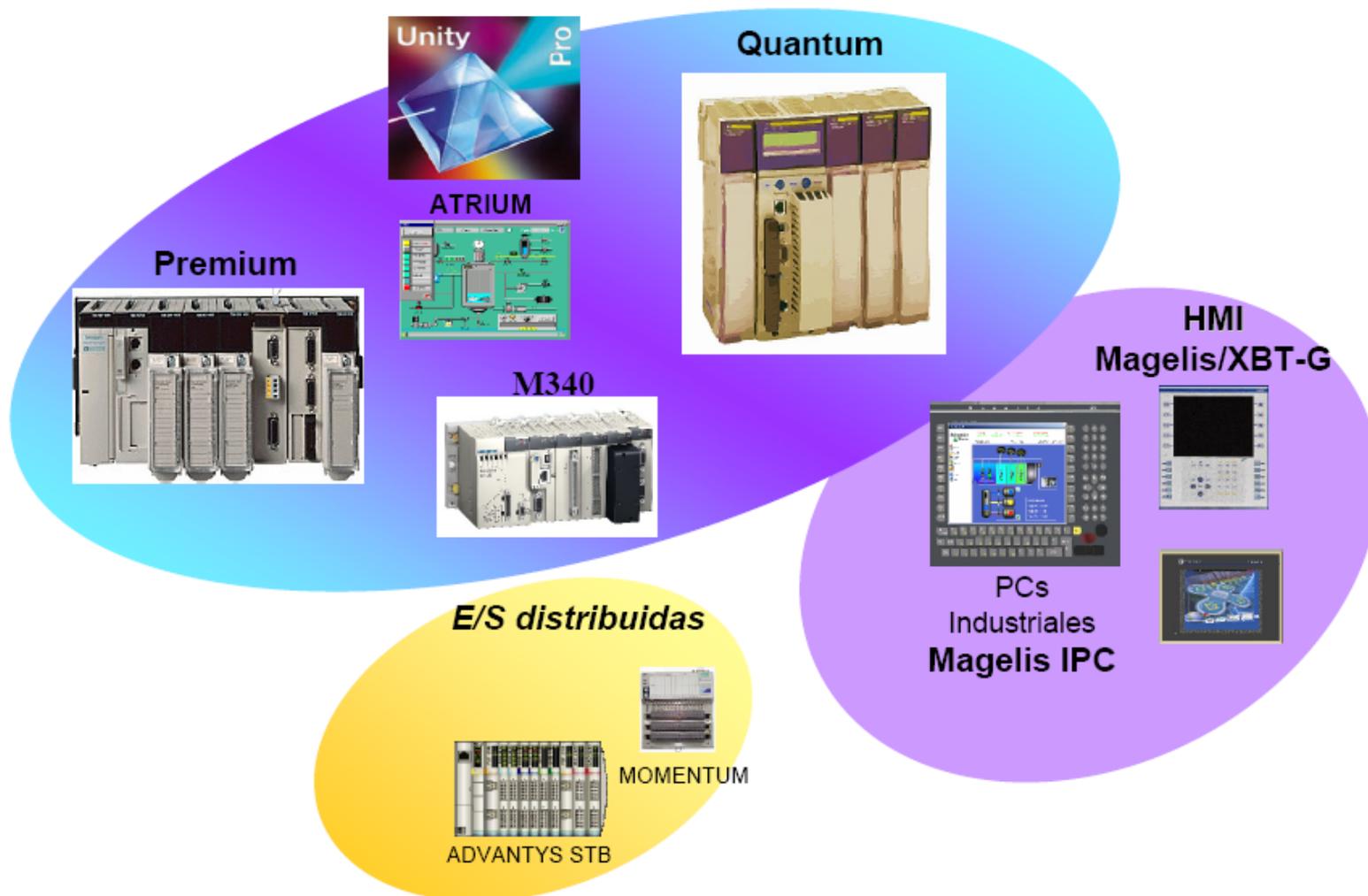
- Schneider es un gran grupo que ofrece soluciones completas dentro del campo de la gestión y control de la energía.
- Surge como evolución/fusión de varias empresas desde el año 1836.
- Schneider (1999) → Schneider Electric 
- Modicon (1996) , Square D (1991).  
Telemecanique (1988), Merlin Gerin (1975) →  
Pasa a ser una marca del grupo Schneider

# 1.- Introducción Schneider

---

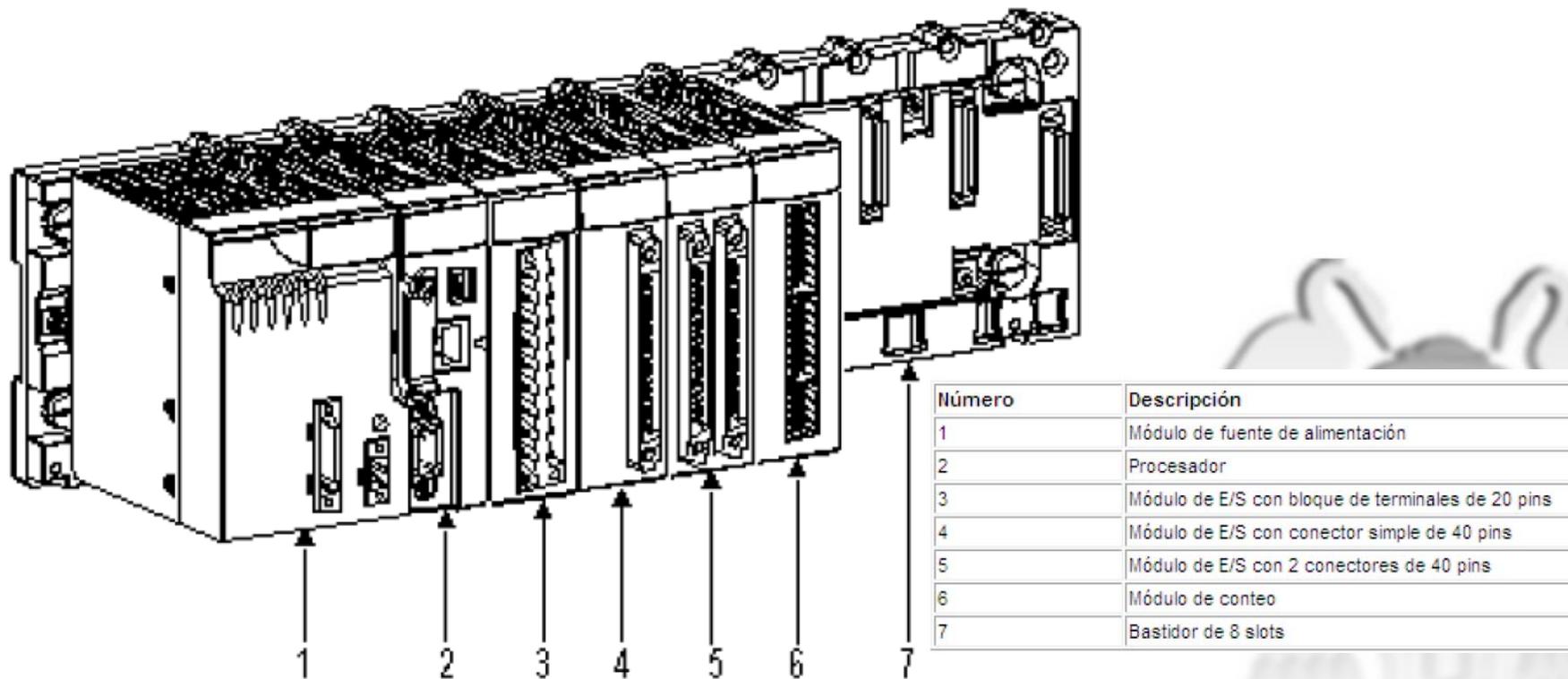
- Líneas de productos de Schneider:
  - Automatización y Control Industrial (PLCs, Software, todo tipo de accionamientos y sensores)
  - Distribución Eléctrica (alumbrado de emergencia, canalización, mecanismos, medidores, relés, etc.)
  - Sistemas de Instalación y Control (domótica, redes, etc.)
  - Automatización y seguridad de edificios (video vigilancia, identificación, etc.)
  - Servicios de Energía y Refrigeración crítica (SAI, MGE, Software de climatización, etc.)
  - Soluciones de Energía Renovable (Medidores de energía, conexiones a red eléctrica, interruptores automáticos, etc.)

## 2.- PLCs de Schneider



## 2.- PLCs de Schneider

- Estructura de un rack MODICON: todos los módulos se pueden quitar y poner en “caliente” con la CPU en RUN (excepto PS y la CPU)



## 3.- PLC MODICON M340

- Especificaciones del sistema:
- Restricciones mecánicas:
  - Choques: 50g
  - Vibraciones: 5g
  - Temperatura: 0-60°C
- Trabaja normalmente hasta altitudes de 4000m



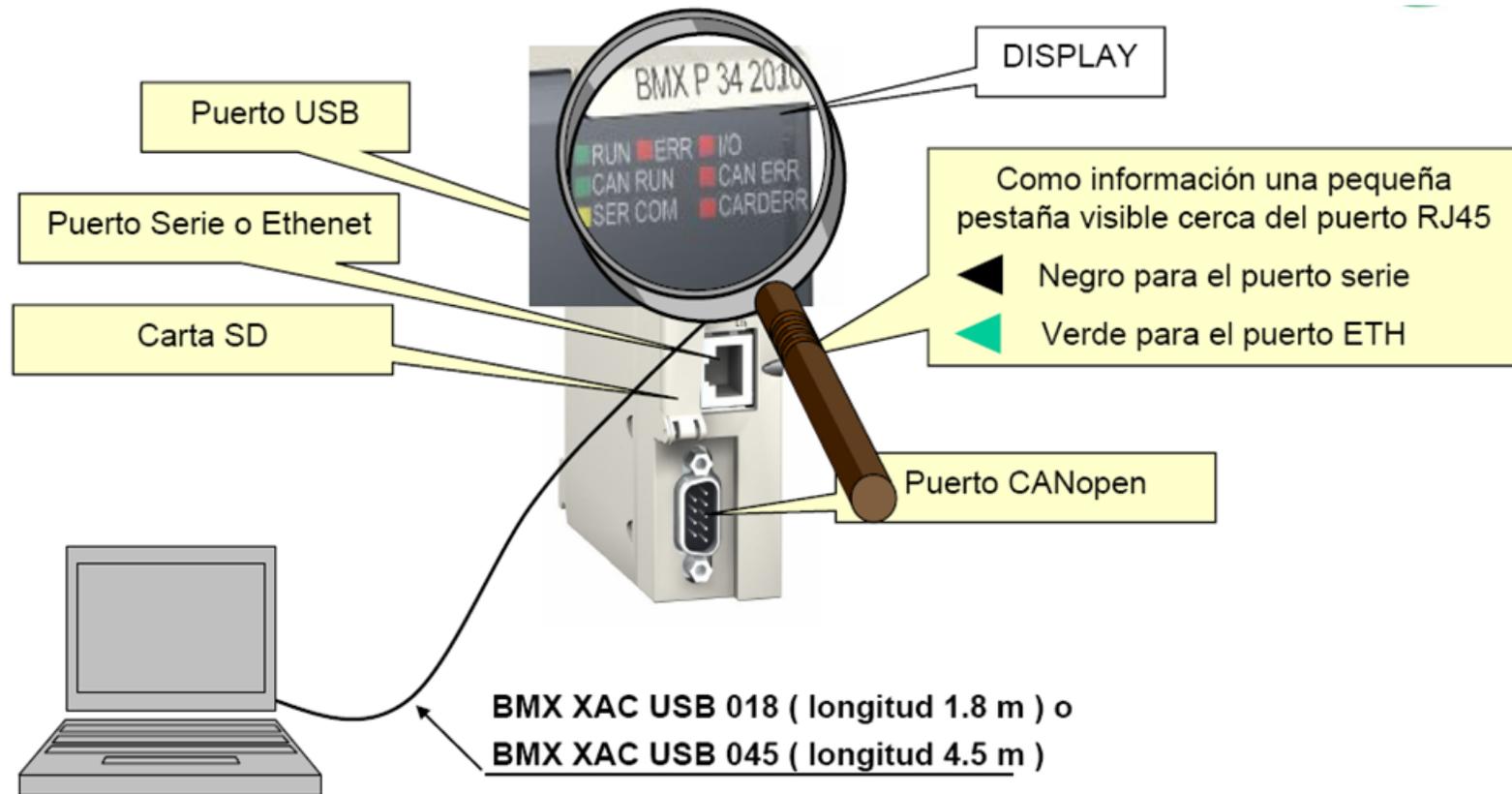
## 3.- PLC MODICON M340

---

- La CPU que se va a utilizar para todas las aplicaciones es la BMX P34 2010.
- Sus características principales son:
  - Tamaño pequeño
  - Alta densidad (hasta 1024 E/S digitales, hasta 256 E/S analógicas)
  - Puertos de comunicación integrados (Ethernet, CANOPEN, Modbus Serie).
- Una CPU está siempre instalado sobre un rack en la dirección 0 y posición 00



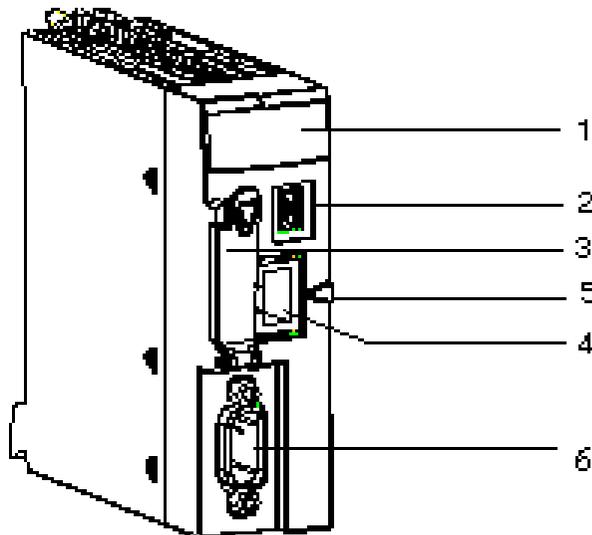
### 3.- PLC MODICON M340



## 3.- PLC MODICON M340

### Características CPU BMX P34 2010:

- Bus CANOpen y Bus MODBUS integrado
- Tarjeta de memoria



Procesador BMX P34 2010

Número	Función
1	Panel de visualización
2	Puerto USB
3	Puerto de protección de la tarjeta de memoria
4	Puerto serie
5	Anillo de identificación del puerto serie (negro)
6	Puerto CANopen

## 3.- PLC MODICON M340

### Tarjeta de memoria para BMX P34 2010

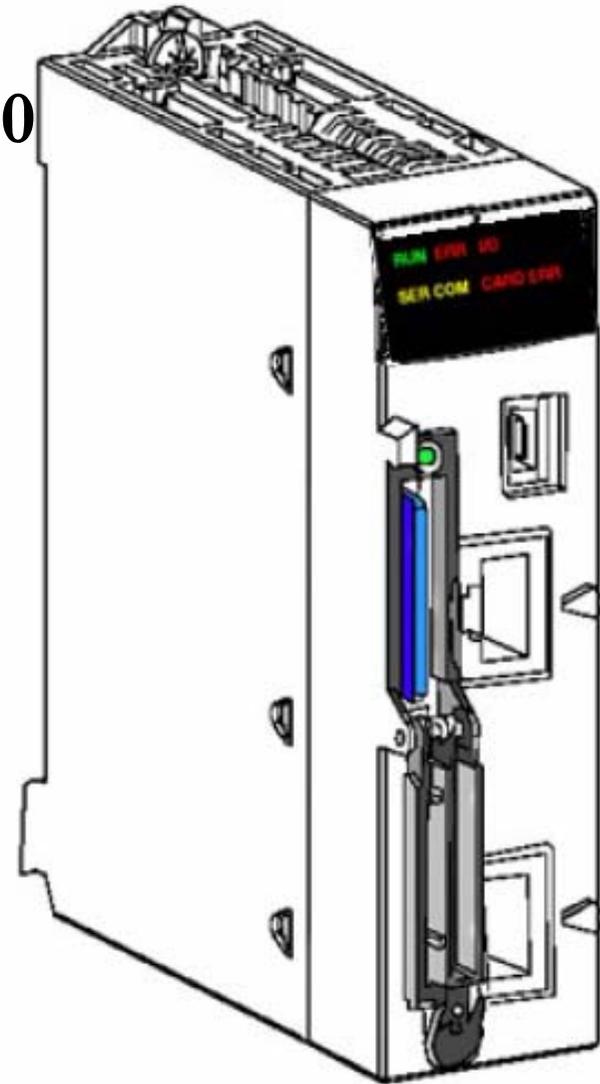


- Las CPU BMX P34 xxx de MODICON M340 usan una tarjeta de memoria BMX RMS 008MPx para guardar aplicaciones, datos y páginas web.
- Las tarjetas de memoria Schneider son las únicas compatibles con los procesadores BMX P34xxx.

## 3.- PLC MODICON M340

### Tarjeta de memoria para BMX P34 2010

- La aplicación que corre en la CPU es la que está en la tarjeta de memoria.
- Cuando se enciende la alimentación se descarga automáticamente.
- También se puede forzar manualmente la transferencia CPU a la tarjeta mediante un bit del sistema.



## 3.- PLC MODICON M340

### Diagnóstico de la CPU

- Existen varios indicadores LED que permiten un diagnóstico rápido del estado del PLC:
  - Funcionamiento del PLC
  - La tarjeta de memoria
  - La comunicación con los módulos
  - La comunicación serie
  - La comunicación en la red CANopen y Ethernet

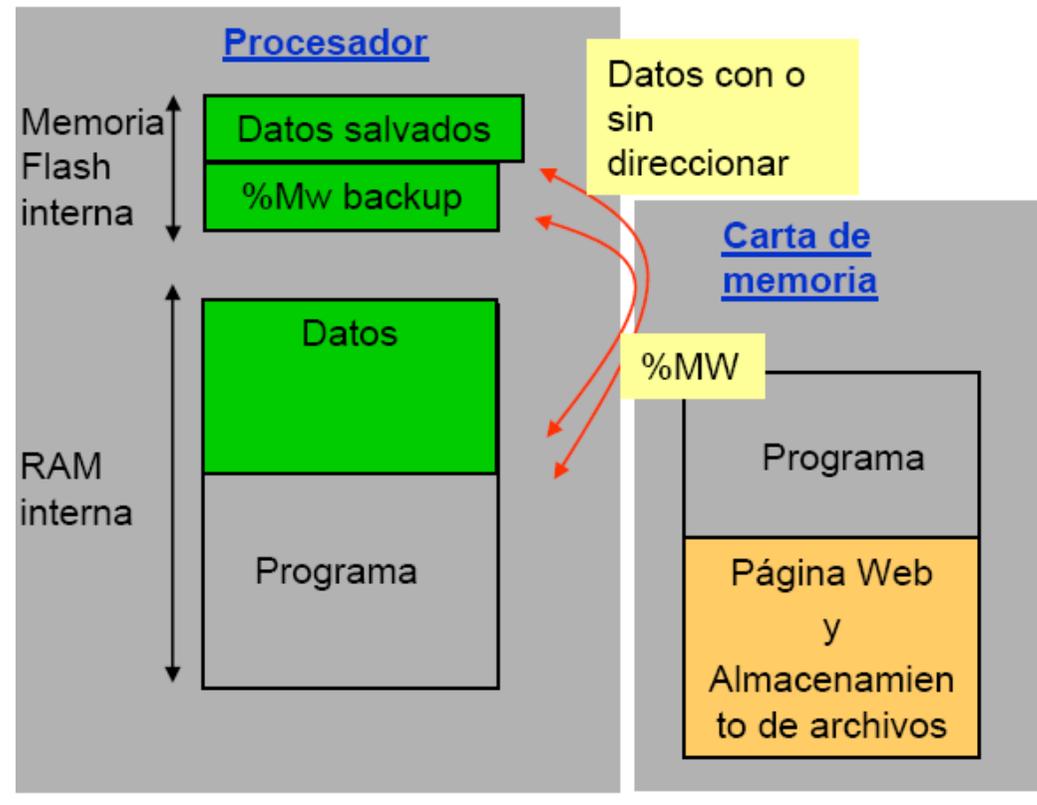
Etiqueta	Modelo	Significado
RUN (verde): estado operativo	Encendido	PLC en funcionamiento normal, ejecución del programa.
	Intermitente	PLC en modalidad STOP o bloqueado por un error de software
	Apagado	PLC no configurado (aplicación ausente, no válida o incompatible)
ERR (rojo): error	Encendido	Fallo del procesador o del sistema
	Intermitente	<ul style="list-style-type: none"> <li>PLC no configurado (aplicación ausente, no válida o incompatible).</li> <li>PLC bloqueado por un error de software.</li> </ul>
	Apagado	Estado normal, sin errores internos.
E/S (rojo): estado de entrada/salida	Encendido	<ul style="list-style-type: none"> <li>Error de entrada/salida procedente de un módulo o de un canal.</li> <li>error de configuración.</li> </ul>
	Apagado	Estado normal, sin errores internos
SER COM (amarillo): serie estado de los dato	Intermitente	Intercambio de datos en curso en la conexión serie (recibiendo o enviando)
	Apagado	No se están intercambiando datos en la conexión serie.
CAN RUN (verde): operaciones CANopen	Encendido	Red CANopen operativa
	Intermitencia rápida (activa durante 50 ms, inactiva durante 50 ms, repitiéndose)	Detección automática del flujo de datos o de los servicios LSS en curso (se alterna con CAN ERR)
	Intermitencia lenta (encendido durante 200 ms, apagado durante 200 ms, repitiéndose)	Red CANopen preoperativa
	1 intermitencia	Red CANopen detenida
CAN ERR (rojo): errores de CANopen	3 intermitencias	Descarga del firmware CANopen
	Encendido	Bus CANopen detenido
	Intermitencia rápida (encendido durante 50 ms, apagado durante 50 ms, repitiéndose)	Detección automática del flujo de datos o de los servicios LSS en curso (se alterna con CAN RUN)
	Intermitencia lenta (encendido durante 200 ms, apagado durante 200 ms, repitiéndose)	La configuración de CANopen no es válida.
	1 intermitencia	Al menos uno de los contadores de errores ha alcanzado o superado el nivel de alerta.
	2 intermitencias	Se ha producido un evento de guard (esclavo o maestro NMT) o de heartbeat.
	3 intermitencias	No se recibió el mensaje SYNC antes de que finalizara el periodo del ciclo de comunicación.
Apagado	No hay error de CANopen.	
CARDERR (rojo): error de tarjeta de memoria	Encendido	<ul style="list-style-type: none"> <li>Falta la tarjeta de memoria.</li> <li>Tarjeta de memoria no utilizable (formato incorrecto, tipo no reconocido)</li> <li>Contenido de la tarjeta de memoria incoherente con la aplicación RAM interna.</li> </ul>
	Apagado	<ul style="list-style-type: none"> <li>Tarjeta de memoria correcta</li> <li>Aplicación de la tarjeta coherente con la aplicación RAM interna.</li> </ul>
CARDAC (verde): acceso a la tarjeta de memoria	Encendido	Acceso a la tarjeta habilitado
	Intermitente	Actividad en la tarjeta; en cada acceso, el indicador LED de la tarjeta se establece en DES y luego vuelve a CON.
	Apagado	Acceso a la tarjeta deshabilitado. Puede extraerse la tarjeta después de haber bloqueado el acceso a la tarjeta generando un flanco ascendente en el bit %S65

RUN ERR IO  
CAN RUN CAN ERR  
SER COM CARD ERR

## 3.- PLC MODICON M340

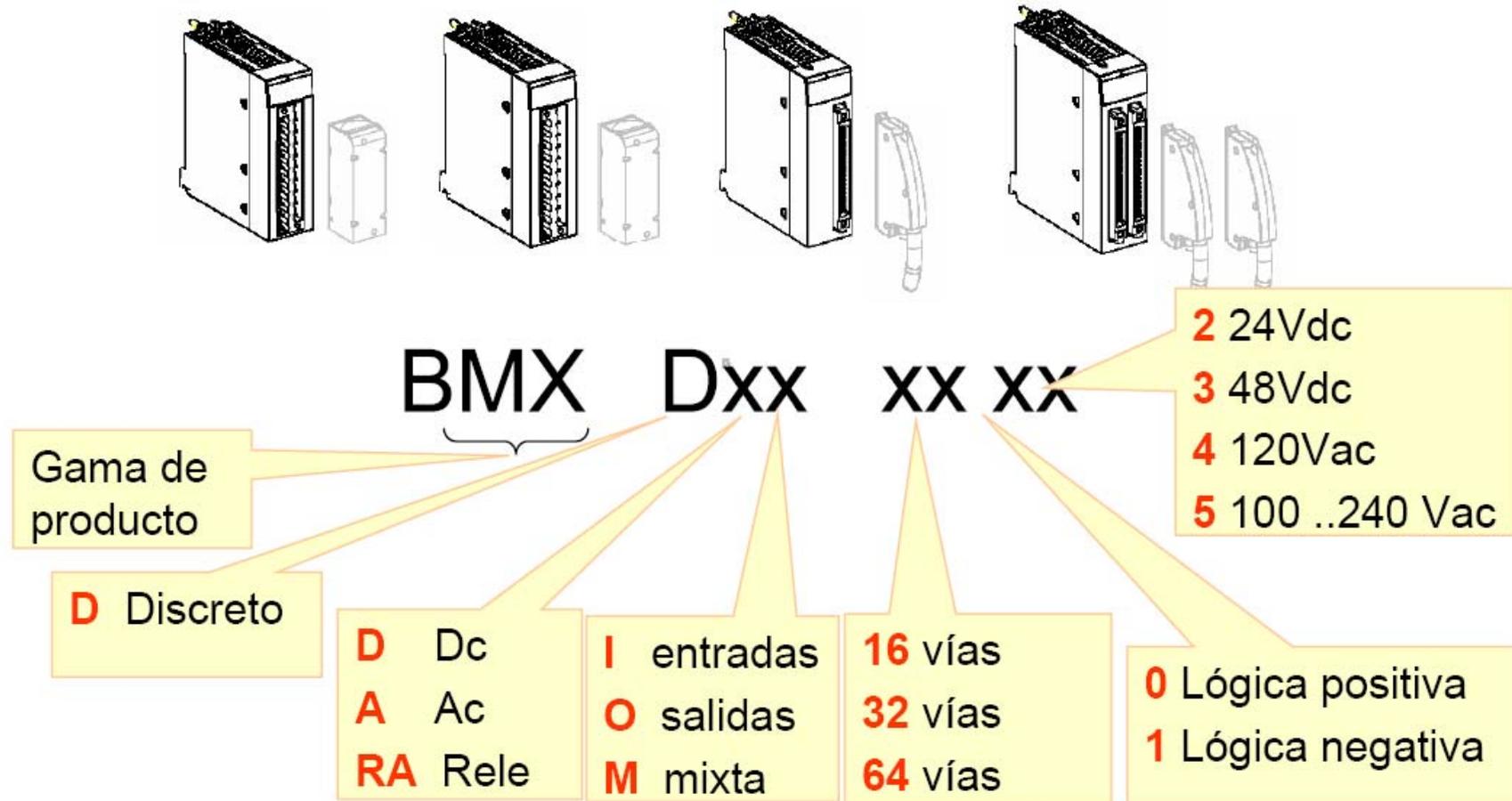
### Memoria Física de la CPU

- Hay que distinguir entre la memoria RAM, Flash y Carta memoria (tarjeta SD).
- El programa se salva automáticamente en la SD después de una descarga o una modificación online



### 3.- PLC MODICON M340

- Tarjetas Digitales



## 4.- UNITY PRO

## Definición

- Unity Pro es el software único de programación, puesta a punto y explotación de los autómatas Modicon, M340, Premium, Quantum y coprocesadores Atrium.
- Unity Pro recoge todas las ventajas de los software PL7 y Concept y ofrece un conjunto completo de nuevas funciones para obtener más productividad y apertura hacia otros softwares.

# 4.- UNITY PRO

# Metodología (1/2)



## 4.- UNITY PRO

## Metodología (2/2)

1. Definir los ajustes
  - Ajustes estación de trabajo accesible vía Herramientas / Opciones
  - Ajustes del proyecto accesible vía Herramientas / Ajustes del proyecto
2. Configurar el hardware y las redes (racks, módulos,...)
3. Definir y editar las variables (variables elementales, arrays, estructuras, instancias de bloques función,...)
4. Estructurar la aplicación (tareas, secciones, módulos funcionales)
5. Editar las secciones de programa (Ladder, FBD, ...)
6. Generar el código (construcción)
7. Seleccionar el destino (PLC o simulador) y transferir la aplicación (descargar)
8. Depurar la aplicación (punto de parada, paso a paso, ...)



## 4.- UNITY PRO

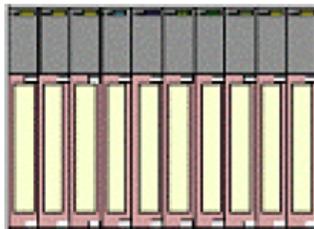
## Generalidades (1/3)

- **Unity Pro de aplicaciones simples a complejas**
  - **Conversores integrados y 5 lenguajes IEC 61131-3**
  - **Datos estructurados y bloques de función**
  - **Módulos funcionales**
  - **Simulador de PLC en el PC**
  - **Diagnóstico del sistema y de la aplicación**
  - **Integración de pantallas de operador**

## 4.- UNITY PRO

## Generalidades (2/3)

- Hardware soportado con UNITY

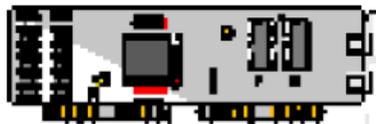
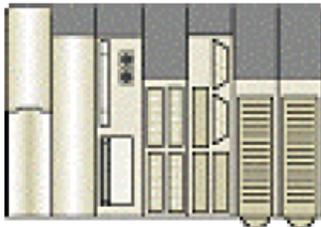


- Quantum: procesadores

- Premium: procesadores

- Atrium: coprocesador ( PCI standard )

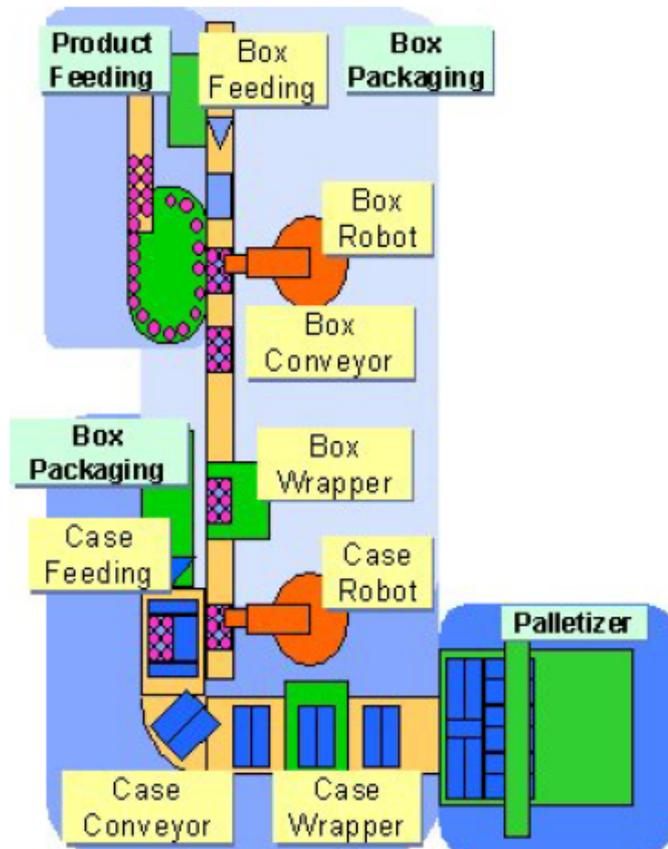
- Momentum: como entradas / salidas distribuidas



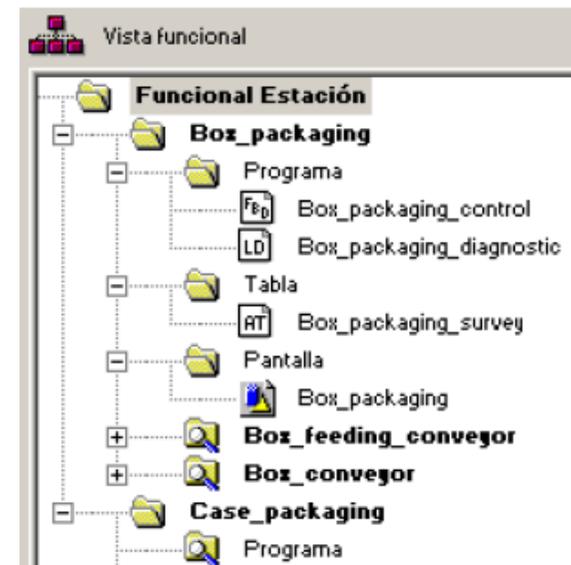
## 4.- UNITY PRO

## Generalidades (3/3)

### ● Vista funcional



Para acceder directamente a la parte de un programa usando la estructura de la máquina



## 4.- UNITY PRO

## Funcionalidad (1/2)

### ● Funcionalidad

#### ● Estructura de aplicación

- Entorno de proyecto
- Explorador de proyecto con vista funcional / estructural
- Biblioteca de funciones ( EF / EFB / DFB )
- Editor de variables ( estructuradas / alocadas / no alocadas )
- monotarea o multitarea
- Aplicación de archivo único ( STU )

#### ● Programación flexible

- Independientemente de la configuración del hardware
- Conjunto de instrucciones genéricas ( compatible con ambas plataformas Premium y Quantum )
- 5 lenguajes IEC

## 4.- UNITY PRO

## Funcionalidad (2/2)

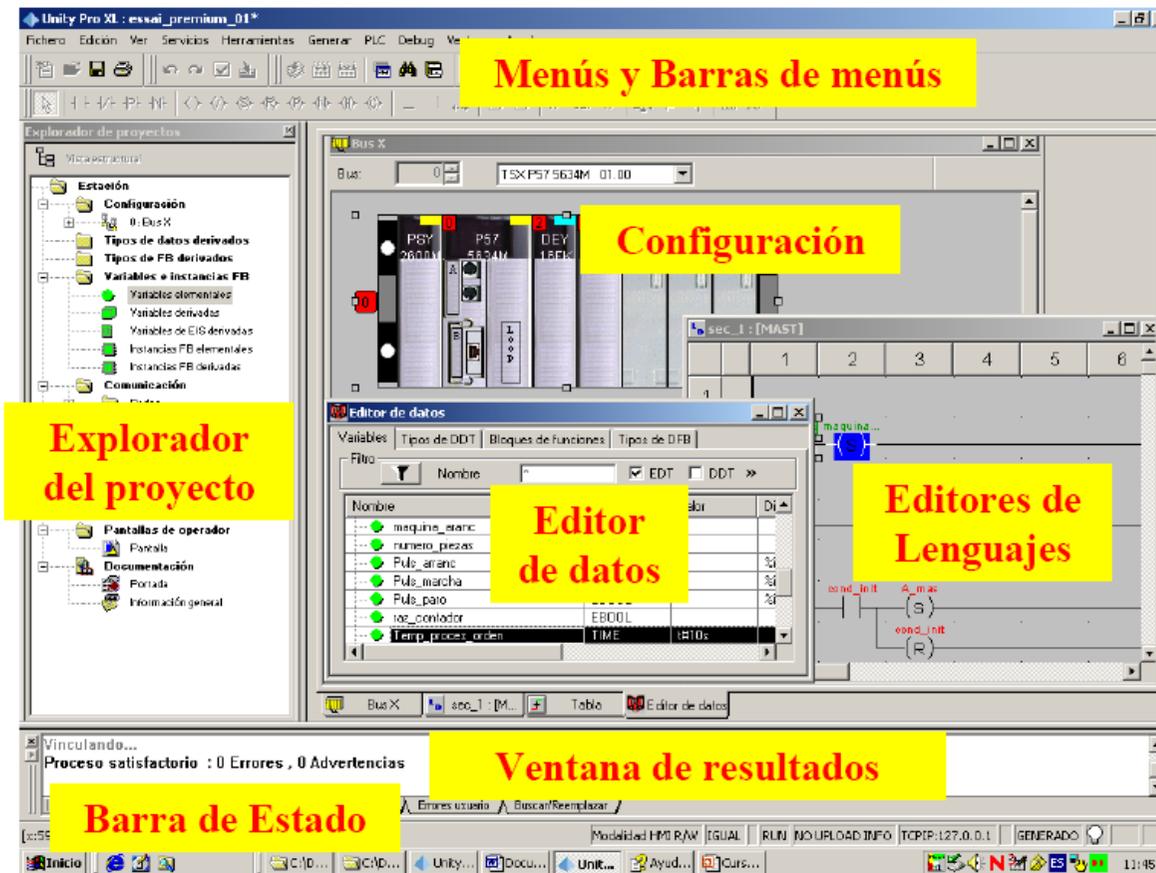
### ● Funcionalidad ( continuación )

- Configuración gráfica de módulos de entrada / salida, módulos expertos
- Comunicación
  - Arquitectura de red dependiente del hardware
  - Módulos específicos de comunicación
- Depuración y reajuste
  - Pantallas de Explotación y de diagnostico
  - Simulador del PLC
  - Punto de observación / Punto de parada y modificaciones en línea
  - Consumo de potencia y utilización de memoria
- Apertura ( hipervínculo, importación / exportación en formato XML, conversores )
- Seguridad ( perfil de usuario)

# 4.- UNITY PRO

# Entorno (1/5)

- Interface de usuario



## 4.- UNITY PRO

## Entorno (2/5)

- Barras de herramientas comunes



- **Gestión de archivos** ( Nuevo, Abrir, Guardar, Imprimir )

- **Edición de archivos** ( deshacer, rehacer, confirmar, ir a )

- **Gestión de aplicaciones** ( analizar, generar, regenerar todo, explorador de proyectos, buscar, gestor de librería de datos)

- **Gestión del PLC** ( descargar, cargar, conectar, desconectar, ejecutar, detener, animación, modalidad estándar, modalidad de simulación)

- **Gestión de depuración** ( establecer punto de parada, detener punto de parada, continuar, paso a paso por función, paso a paso por instrucciones , paso a paso para salir, mostrar paso actual, stack de llamadas)

- **Gestión de Ventanas** ( cascada, organización horizontal, organización vertical )

- **Gestión de la ayuda** ( ayuda, descripción )

## 4.- UNITY PRO

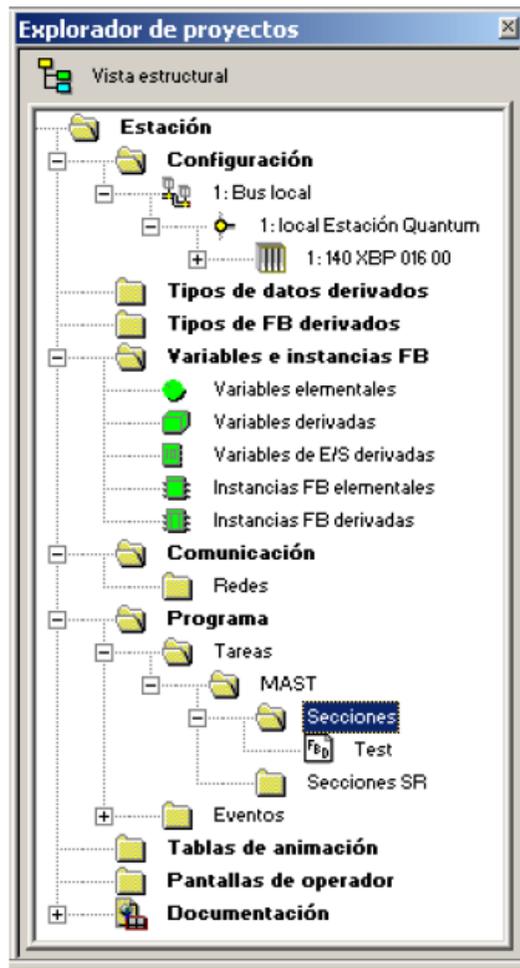
## Entorno (3/5)

- Biblioteca de funciones: existe la posibilidad de crear funciones propias (DFB) (librerías) para posteriormente ser reutilizadas en otros procesos.
- Posee al igual que Siemens un Administrador Hardware para configurar el rack.
- Posee herramientas de depuración. Concretamente se dispone de un simulador con las mismas características que el PLC real. También se dispone de forzado de variables, watch-point y pantallas gráficas

## 4.- UNITY PRO

## Entorno (4/5)

### ● Explorador del proyecto



● Presenta el contenido del proyecto en un formato de estructura en árbol y permite moverse alrededor de los distintos elementos: configuración, datos, programa, etc.

● Visualización de 2 maneras diferentes :

● Vista estructural

● Vista funcional

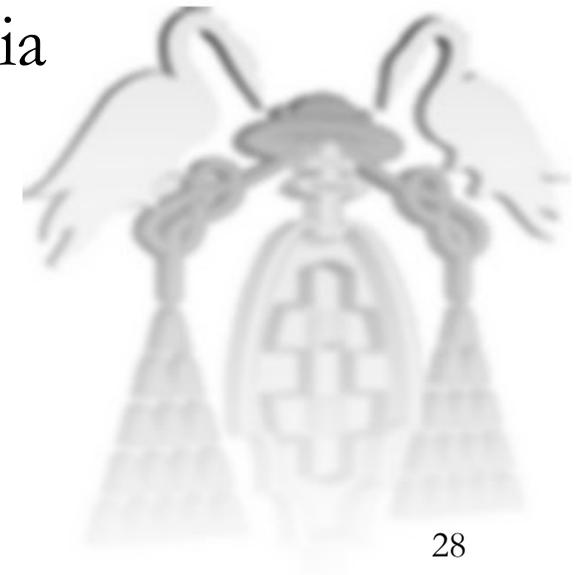
● Trae los componentes de la aplicación ( Secciones ) hacia delante antes que las herramientas se usen

● Permite al usuario estructurar las aplicaciones en conjuntos funcionales

## 4.- UNITY PRO

## Entorno (5/5)

- Lenguajes soportados por UNITY:
  - LD: Lenguaje de contactos
  - FDB: Lenguaje de bloques funcionales
  - IL: Lista de instrucciones
  - ST: Literal estructurado
  - SFC: Diagrama funcional en secuencia



## 4.- UNITY PRO

## Variables (1/6)

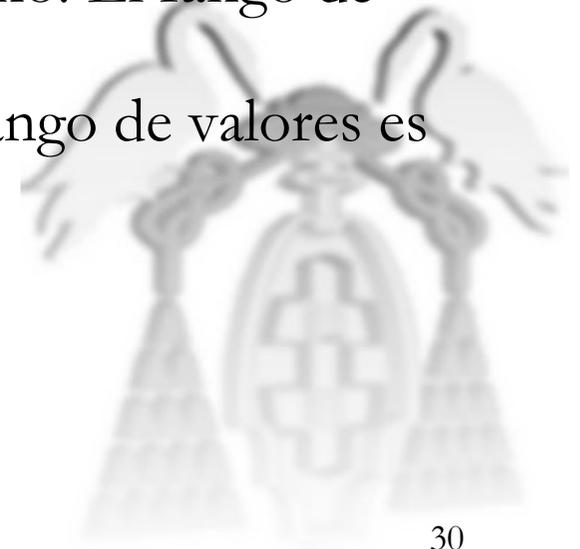
- Una variable es una zona de Memoria de diferentes tipos cuyos contenidos pueden ser modificados por el programa durante su ejecución. Pose un nombre simbólico (32 caracteres) y siempre un tipo.
  - Variable alcatada (direccionada): variable asociada a una dirección de memoria física determinada
  - Variable no alcatada (no direccionada): variable a la que el sistema da una dirección libre. No sabemos su dirección

Nombre	Tipo	Dirección	Valor
base_500	EBOOL		
coderior	UDINT	%mw2	
dfb_lang	EBOOL	%i0.2.4	
diag1	diag		
fdb_lang	EBOOL	%i0	
hecho	EBOOL		
hecho_ld	EBOOL		
hecho_st	EBOOL		
inc	INT		
ld_lang	EBOOL	%i0.2.1	
mult1	INT		
mult2	INT		
out_mux	WORD		
...	EBOOL		

## 4.- UNITY PRO

## Variables (2/6)

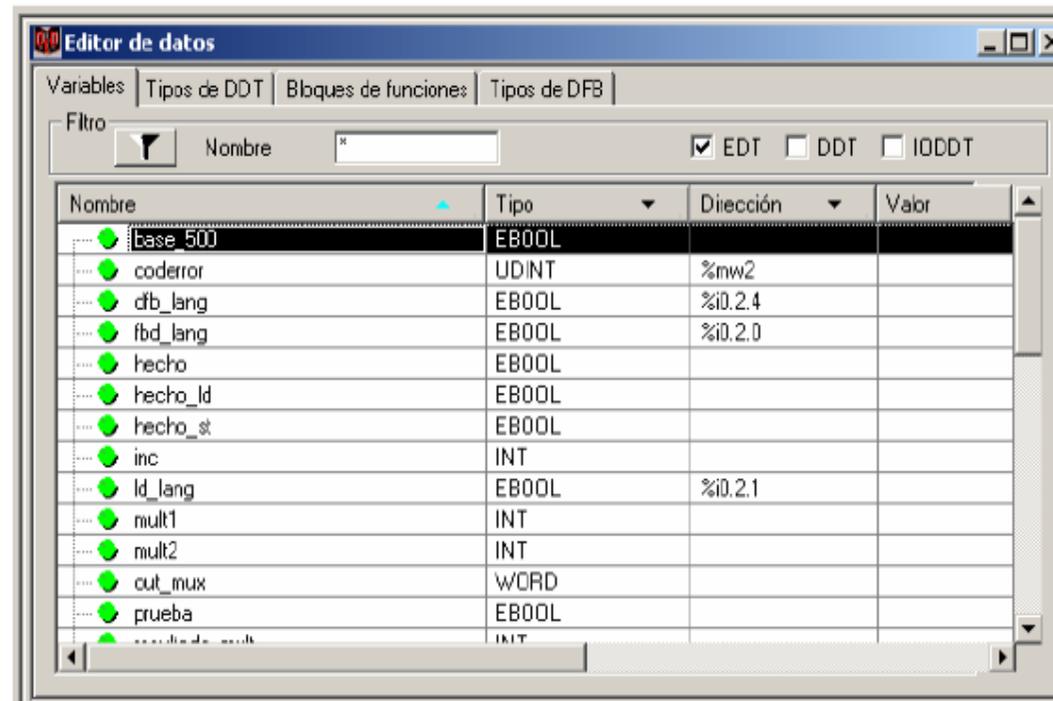
- Existen diferentes tipos de variables:
  - **BOOL/EBOOL** : Variable Booleana será FALSA (0) o VERDADERA (1). EBOOL gestiona también la detección de flancos
  - **WORD** : Representa una cadena de 16 bits , significando una longitud de datos de 16 bits
  - **INT** : representa un valor entero. El rango de valores es de -32768 hasta 32767
  - **UINT** : Representa un valor entero sin signo. El rango de valores es de 0 hasta 65535
  - **REAL** : Representa un valor decimal. El rango de valores es de  $-3.40e+38$  a  $3.40e+38$



## 4.- UNITY PRO

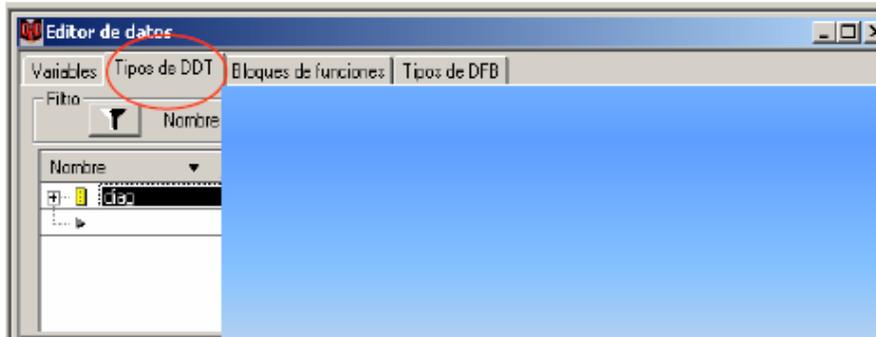
## Variables (3/6)

- Una variable se puede crear bien en la edición de un programa o dentro del módulo de edición de variables.

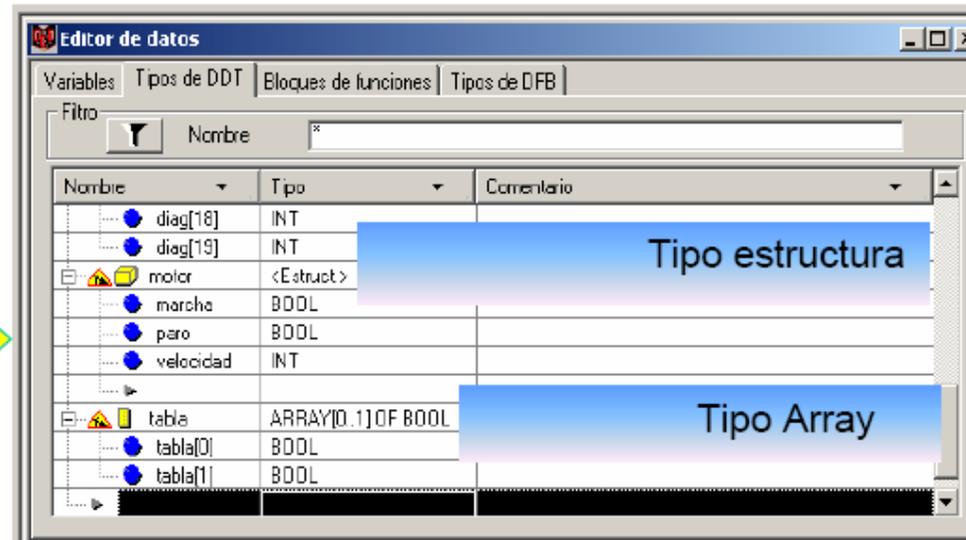
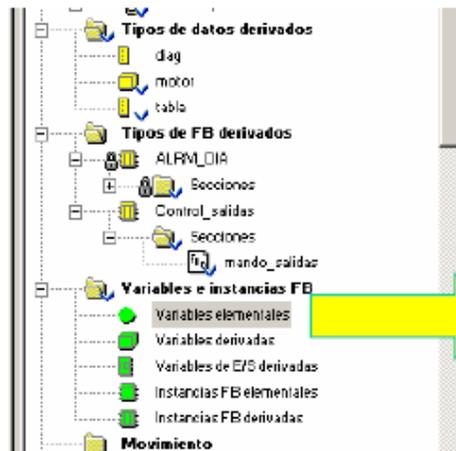


# 4.- UNITY PRO

# Variables (4/6)



**Tipos de DDT:**  
 Permite crear:  
 -una estructura (tipo de dato diferente) o  
 -un Array (tipo de dato igual)



## 4.- UNITY PRO

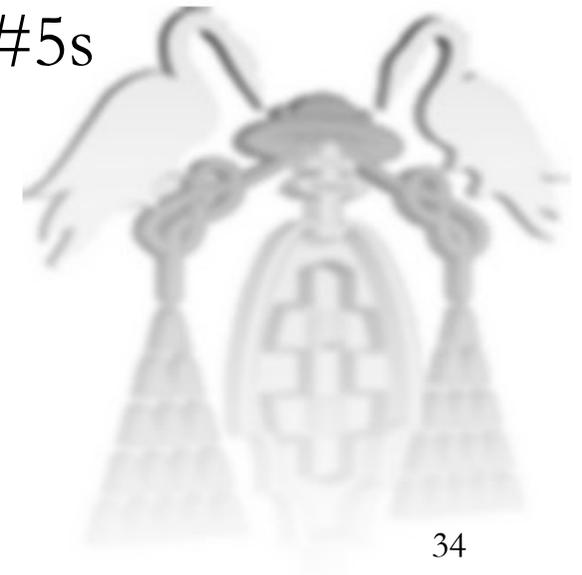
## Variables (5/6)

- ¿Cómo se llaman a las variables?
  - %<Tipo\_Mem><bastidor>.<slot\_modulo>.<canal>
  - Entradas Digitales: %Ir.m.c
  - Salidas Digitales: %Qr.m.c
  - Entradas Analógicas: %IW.r.m.c
  - Salidas Analógicas: %QW.r.m.c
- Las **constantes** son unas variables del tipo INT, DINT o REAL alcatadas en el campo constante ( %K ), o variables usadas en direcciones directas ( %KW, %KD, o % KF). Sus contenidos no pueden modificarse por el programa durante la ejecución.

## 4.- UNITY PRO

## Variables (6/6)

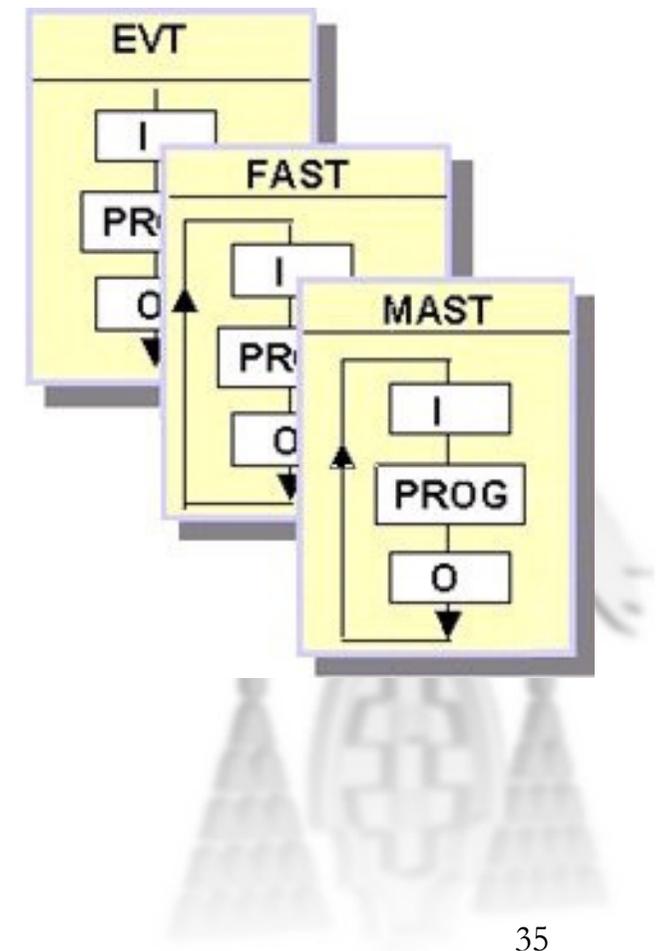
- **Se pueden entrar valores literales como**
  - Base 2 (binario) `2#111111111111111111`
  - base 8 (octal) `8#177777`
  - base 10 (decimal) `65535` (`10#` no se necesita)
  - base 16 (hex) `16#FFFF`
  - Tiempo de 5 segundos : `t#5s` o `time#5s`



## 4.- UNITY PRO

## Estructura Aplicación (1/7)

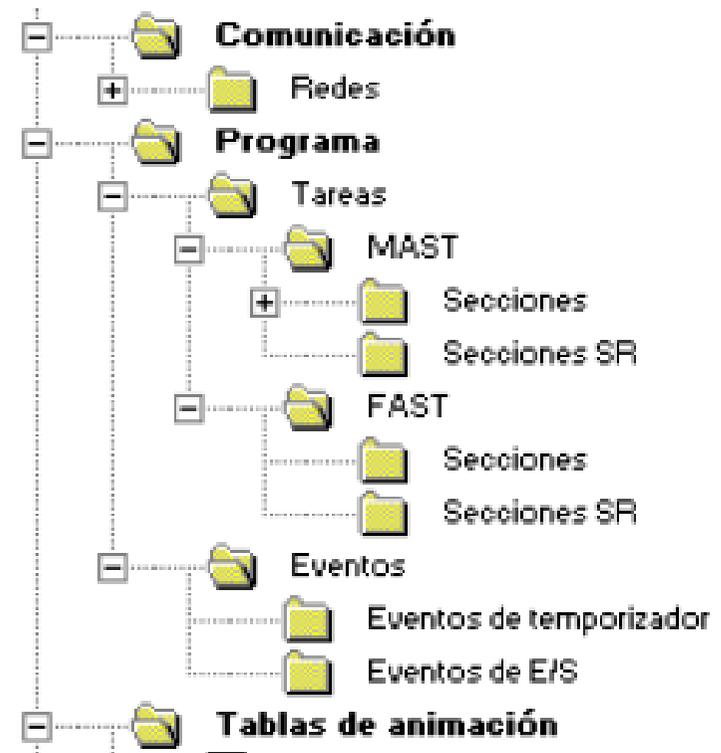
- **Diferentes tareas**
  - Tarea única (solamente MAST) o multitarea (MAST y FAST)
  - Tarea de eventos : Evento de E/S (EVT) y evento de temporizador
- **Modo de ejecución**
  - Ejecución cíclica (solamente para tarea MAST)
  - Ejecución periódica
- **Secciones**
  - Las tareas están estructuradas en secciones de programa
- **Subrutinas (SR)**
  - Las tareas MAST y FAST pueden contener subrutinas.
  - Una subrutina se llama desde una sección o desde otra subrutina



## 4.- UNITY PRO

## Estructura Aplicación (2/7)

- **MAST:** Tarea principal
  - con secciones y subrutinas
  - Programación en LD, FBD, IL, ST o SFC
  - Ejecución cíclica o periódica ( 0..255 ms, 0 = operación cíclica )
  - Controlado por el watch dog, bits y palabras sistema
- **FAST:** Tarea rápida
  - Con secciones y subrutinas
  - Programación en LD, FBD, IL, ST
  - Ejecución periódica ( 1..255 ms )
  - Controlado por el watch dog, bits y palabras sistema



## 4.- UNITY PRO

## Estructura Aplicación (3/7)

---

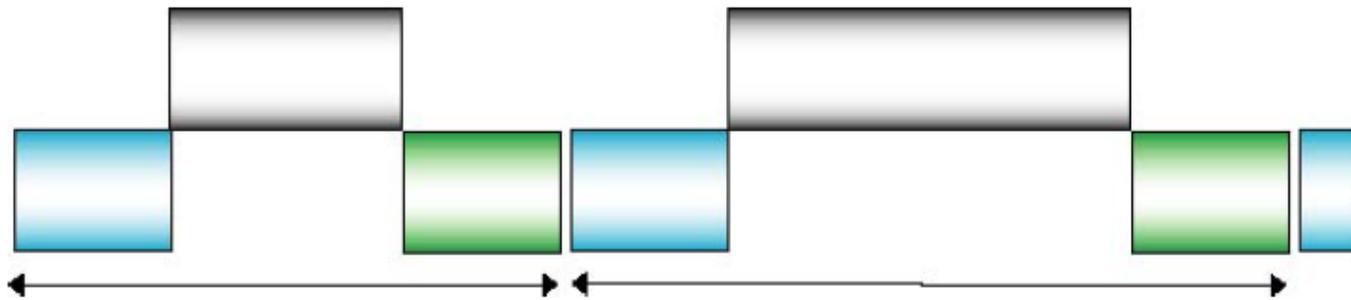
- **EVT o TIMER:** Tareas de eventos
  - Permiten reducir el tiempo de respuesta del programa de aplicación a los eventos desde módulos de entrada / salida, sucesos temporizados
  - Programación en secciones simples en LD, FBD, IL, ST
  - EVTi: eventos procedentes de módulos de entrada / salida
  - TIMERi: eventos procedentes de eventos temporizados ( función ITCNTRL )

# 4.- UNITY PRO

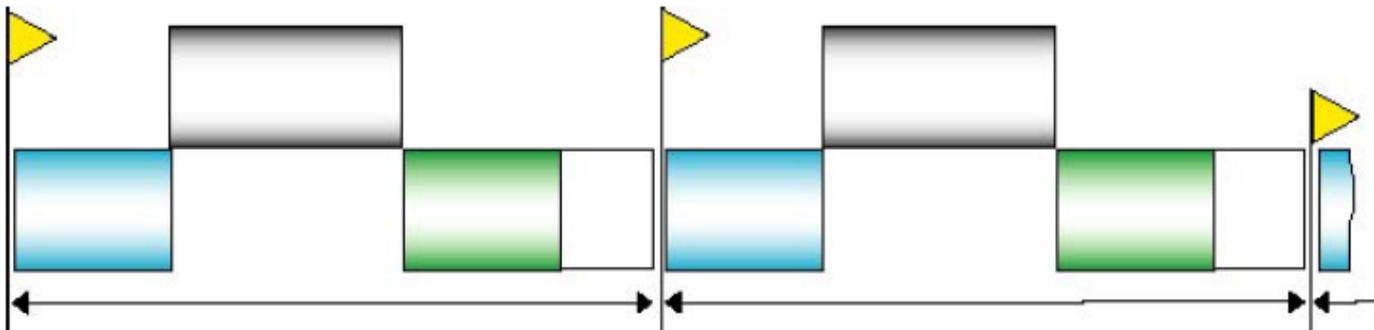
# Estructura Aplicación (4/7)

- Modos de ejecución

**● Cíclico**



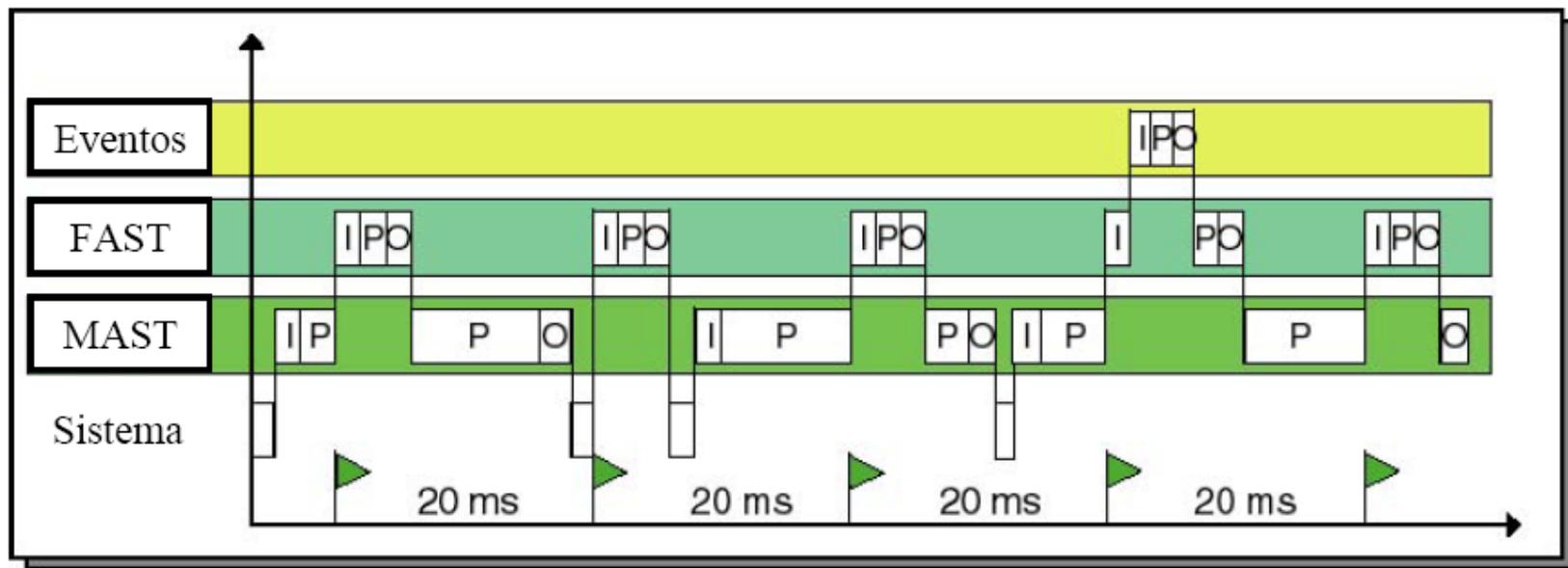
**● Periódico**



## 4.- UNITY PRO

## Estructura Aplicación (5/7)

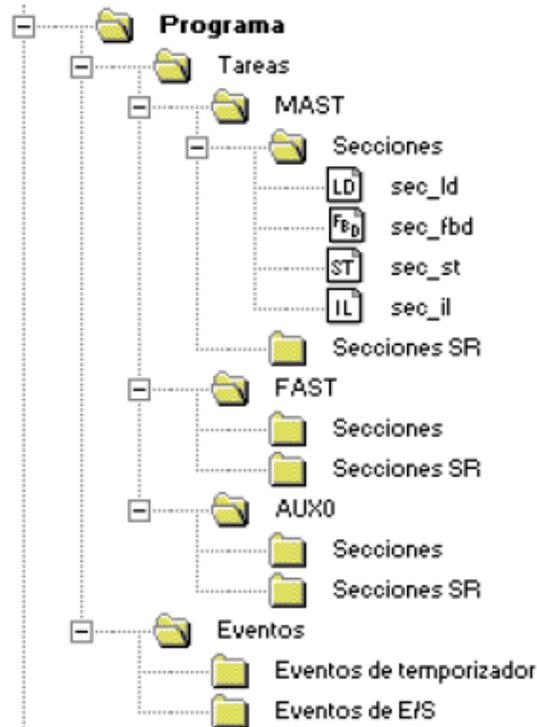
- Ejemplo de tarea principal (MAST) y tarea rápida (periodo=20ms)



## 4.- UNITY PRO

## Estructura Aplicación (6/7)

### SECCIONES



Cada tarea puede estar compuesta de un número de secciones

Atributos asociados con una Sección

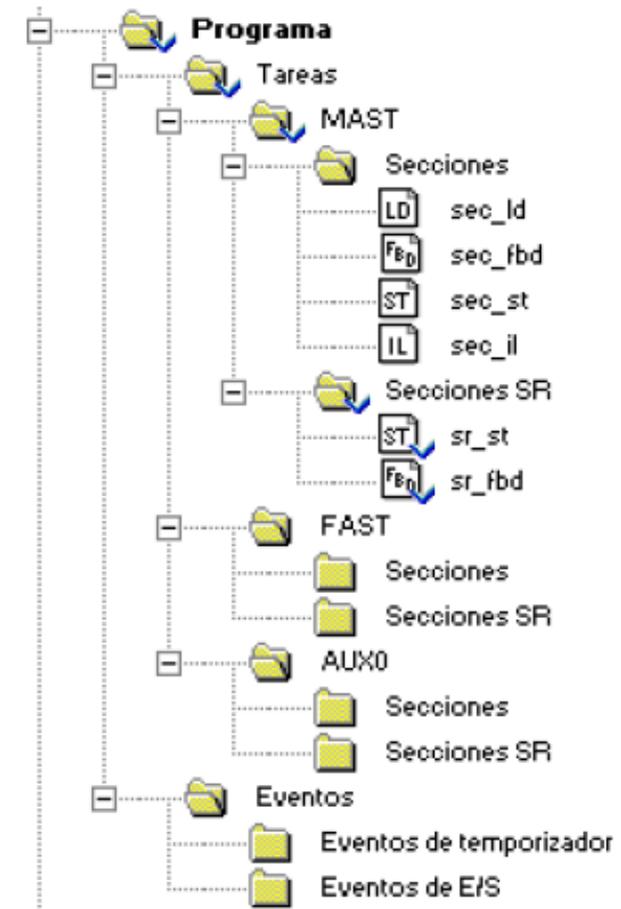
- Nombre: 32 caracteres máximo ( los acentos son posibles, pero los espacios no)
  - Lenguaje: LD, FBD, IL, ST o SFC
  - Tareas asociadas: LD, FBD, IL, ST o SFC
  - Condición ( opcional ) : validación bit para ejecutar, configuración directa
  - Comentario: 256 caracteres como máximo
  - Protección: protección escritura, protección de lectura / escritura
- Orden de ejecución de las secciones
    - Orden en el que aparecerán en el explorador

## 4.- UNITY PRO

## Estructura Aplicación (7/7)

### SUBROUTINAS

- Programado como entidades separadas, cualquiera en LD, IL, ST, FBD
- Las llamadas a las subrutinas se cargan fuera en las secciones o desde otras subrutinas
- El número de *anidamientos* está limitado a 8. Una subrutina no puede llamarse a sí misma ( no recursivo )
- Conectado a una tarea, la misma subrutina no puede llamarse desde varias tareas diferentes

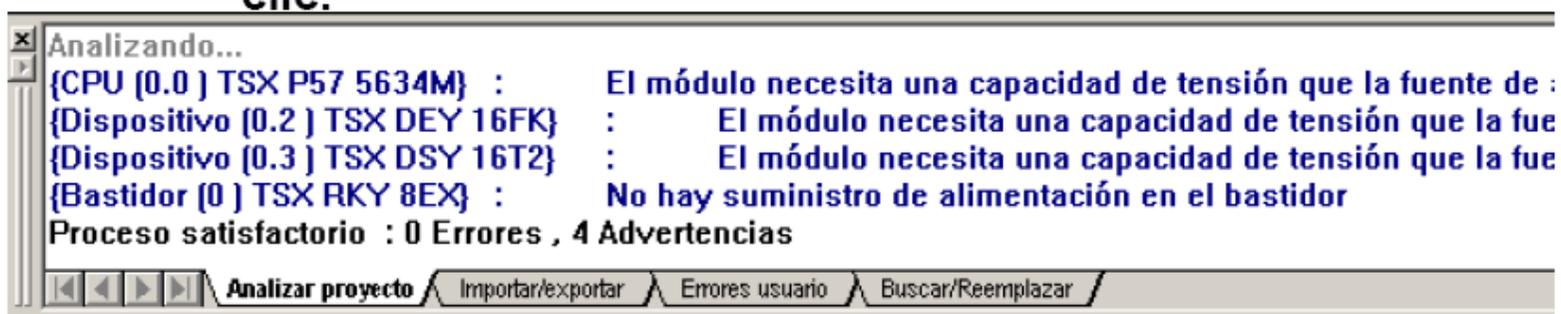


## 4.- UNITY PRO

## Generación Aplicación (1/2)

### Análisis de la aplicación

- Permite la detección de errores generados para cada modificación de la aplicación
- Todos los tipos de errores detectados durante la fase de análisis se muestran automáticamente en la pantalla en la ventana resultados 
- Este mensaje, que aparece en comas invertidas en la ventana de resultados te permite acceder directamente a la parte del programa para ser corregido haciendo un doble clic.

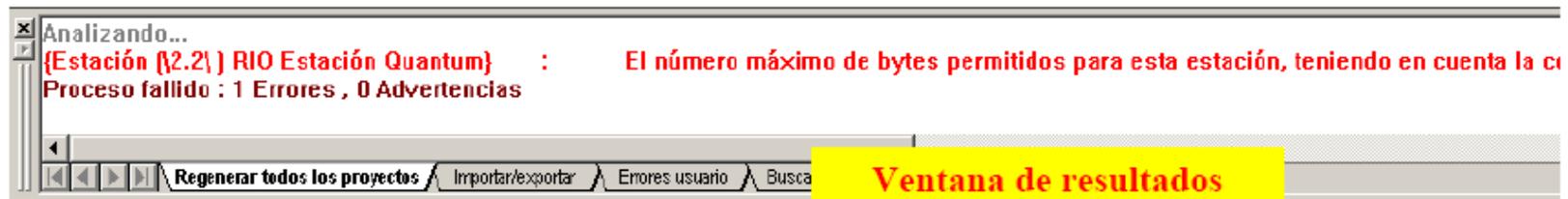
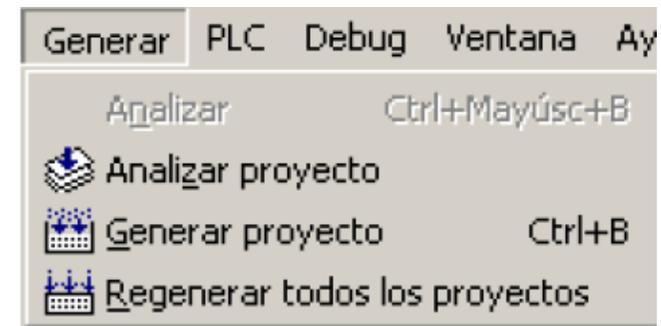


## 4.- UNITY PRO

## Generación Aplicación (2/2)

### Construcción

- El menú “Generar” permite acceder a las herramientas del software usadas para generar un proyecto
- Regenerar todos los proyectos
  - Crear un archivo que pueda telecargarse en el PLC o en el Simulador PLC
- Generar proyecto
  - Solo tiene en cuenta los cambios hechos en el proyecto existente



## 4.- UNITY PRO

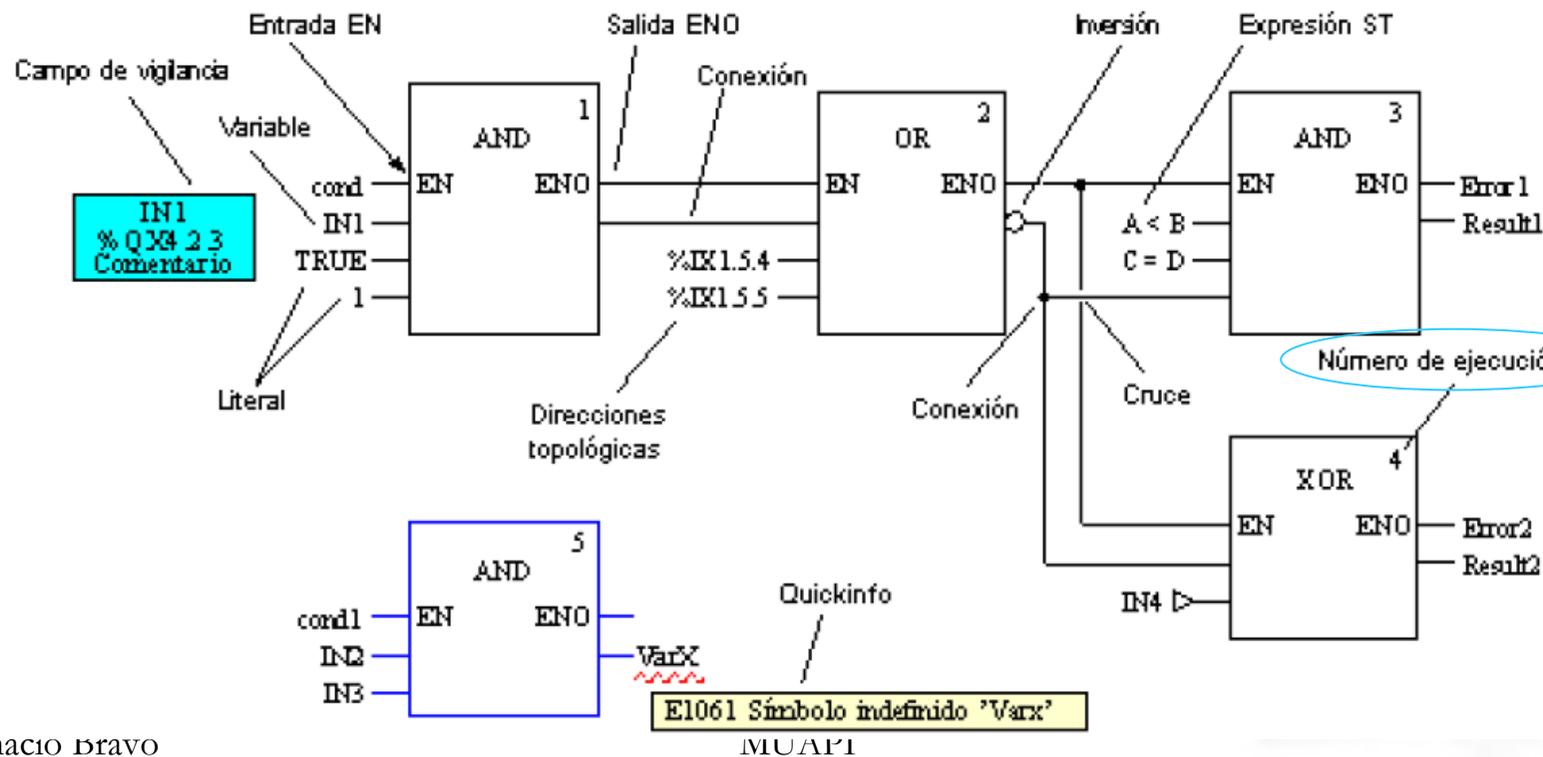
## Editores(1/1)

- Durante la creación de una sección, se ha de establecer con qué lenguaje de programación se desea trabajar. Para la confección de secciones en los distintos lenguajes de programación, se encuentran disponibles editores especiales:
  - **Editor FBD** (Function Block Diagram): Lenguaje de bloques de función;
  - **Editor LD** (Ladder Diagram) :Lenguaje de contactos;
  - **Editor SFC** (Sequential Function Chart):Lenguaje de desarrollo secuencial o Grafcet;
  - **Editor IL** (Instruction List) : Lista de instrucciones;
  - **Editor ST** (Structured Text): Lenguaje literal estructurado.

## 4.- UNITY PRO

## FBD (1/16)

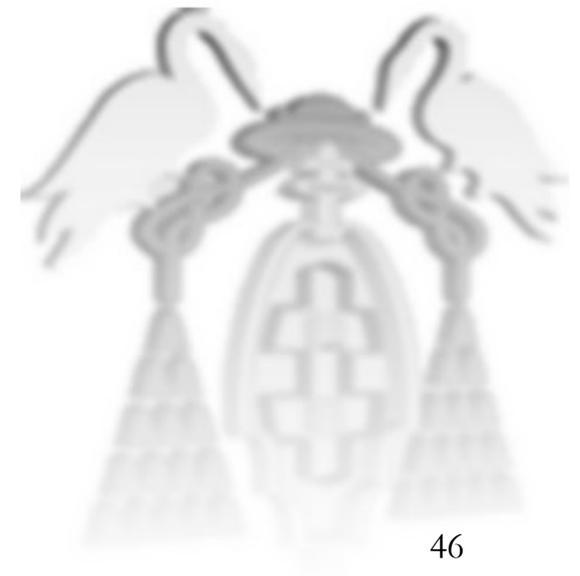
- El lenguaje **FBD** (Function Block Diagram) permite la programación gráfica de los bloques de función según CEI 61131-3.



## 4.- UNITY PRO

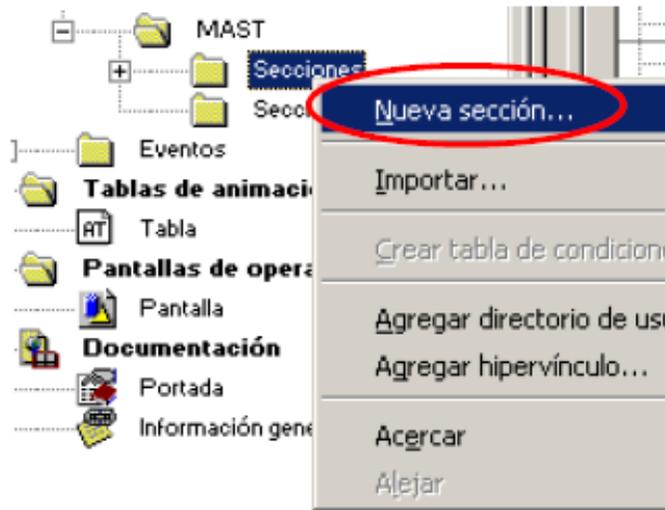
## FBD (2/16)

- Los Diagramas de Bloques de función consisten en un Editor gráfico orientado al dibujo de bloques. El lenguaje consiste en los Bloques de Funciones reusables elementales y derivados.



# 4.- UNITY PRO

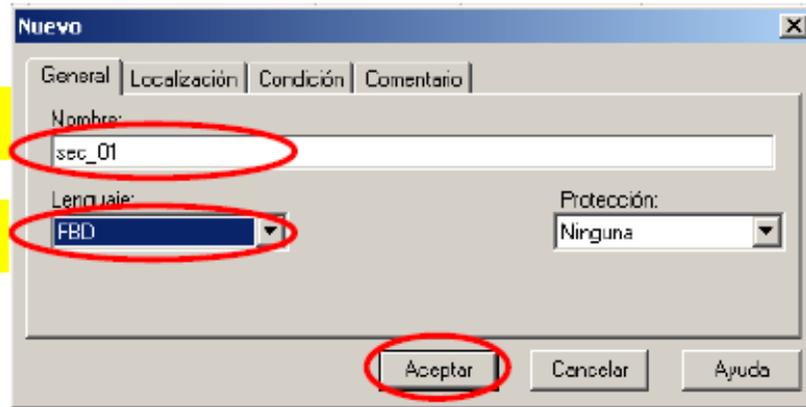
# FBD (3/16)



1- Hacer clic con el botón derecho del ratón y seleccionar "Nueva sección..."

2- Escribir un nombre

3- Seleccionar FBD

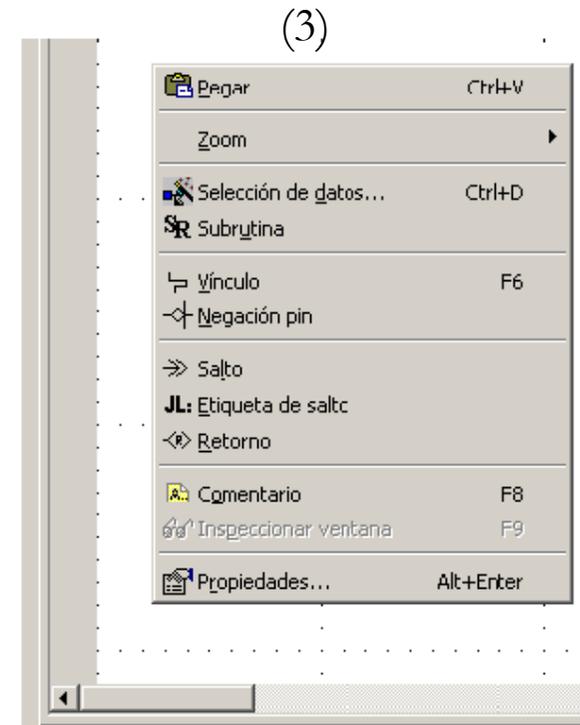
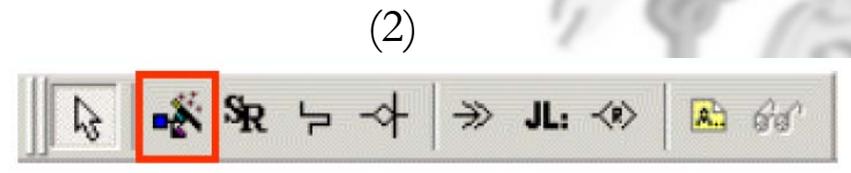


4- Aceptar

## 4.- UNITY PRO

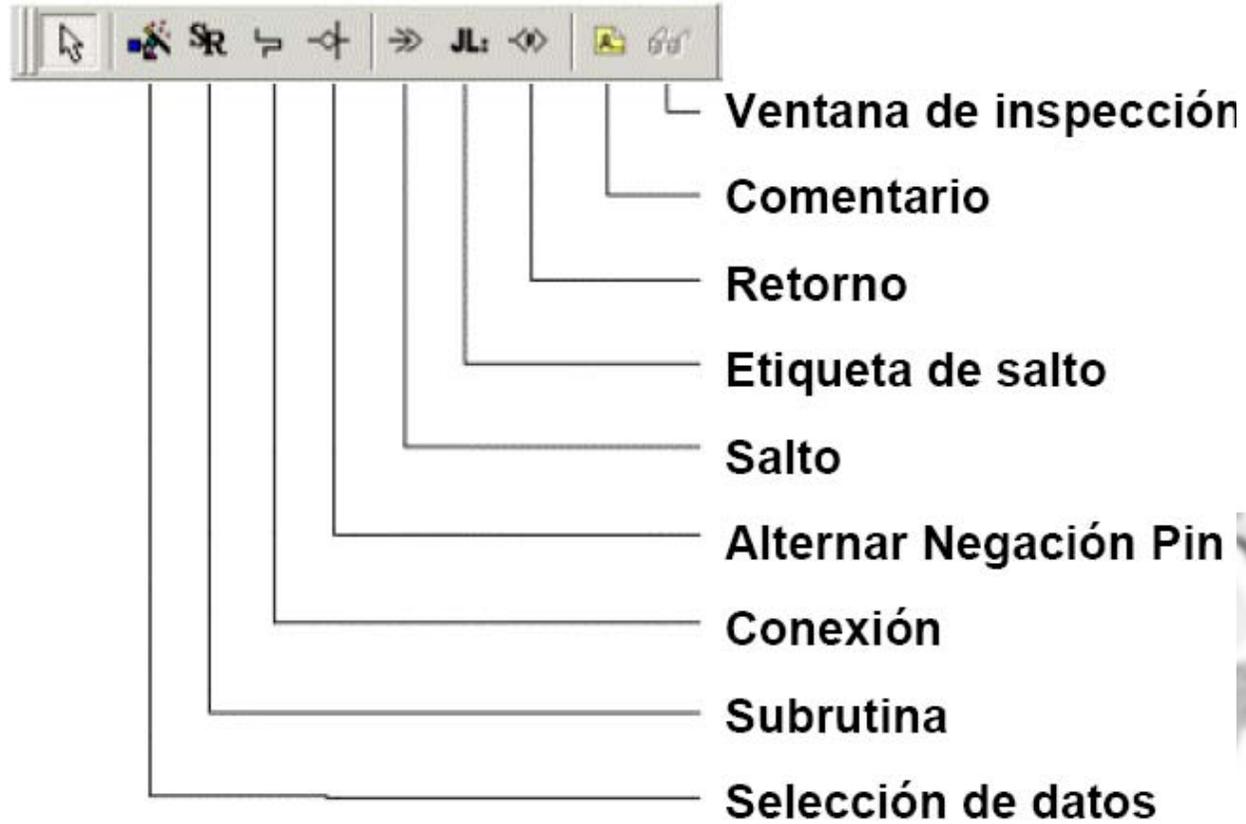
## FDB (4/16)

- Hay 3 métodos de acceso a los objetos de programación FBD:
  1. Hacer clic “ Edición ”, entonces “ nuevo ”...
  2. Usar la barra de herramientas de objetos FBD
  3. Colocar el cursor en el editor lógico de Windows entonces hacer clic con el botón derecho del ratón para seleccionar el grupo de objetos



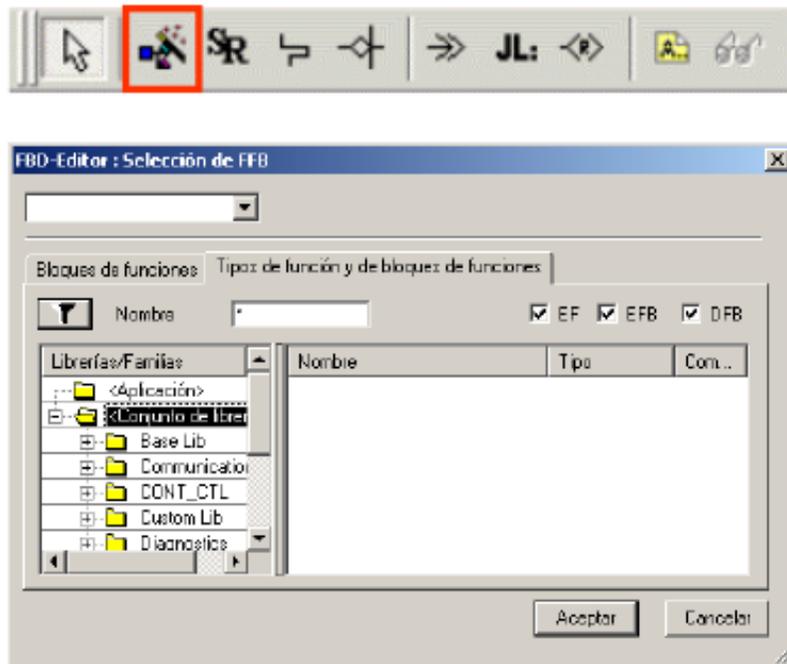
# 4.- UNITY PRO

# FDB (5/16)



## 4.- UNITY PRO

## FDB (6/16)



- Funciones elementales ( EF )
- Bloque de función ( FB )
- Bloque de función elemental ( EFB )
- Bloque de función derivado ( DFB )
- Abreviación general = FFB ( función / bloque de función )
- Macros ( FFBs, DFBs )
- FFBs están en librerías y familias
- Interconexión de FFBs con:
  - Links gráficos o
  - Nombre de variable: “ MOT\_LE ”

## 4.- UNITY PRO

## FDB (7/16)



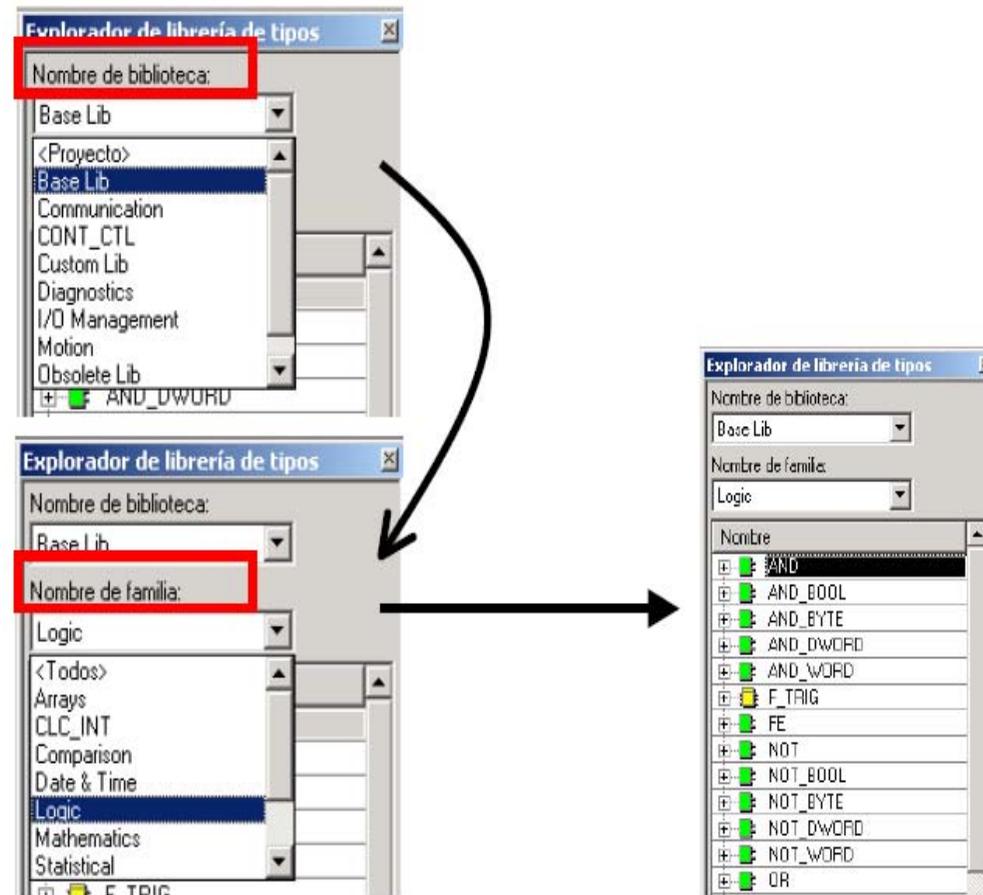
- Introducir un bloque de función en una sección FBD
1. Crear una nueva sección FBD
  2. Hacer clic en el icono FFB de la barra de herramientas
  3. Seleccionar la tabla “ funciones y bloques de funciones ”
  4. Bajo Conjunto de librerías-baselib-lógica, selecciona AND\_BOOL y hacer clic en OK
  5. Ubicar el bloque AND\_BOOL donde se quiera
  6. Asignar las variables a los pins ( entradas / salidas )

*Los Gráficos continúan en la siguiente página*

## 4.- UNITY PRO

## FDB (8/16)

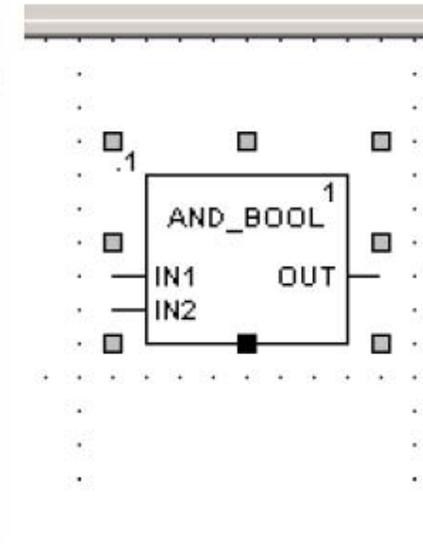
- La librería Base\_Lib contiene las funciones más habituales:
  - Lógicas
  - Tiempo
  - Matemáticas
  - Comparación



## 4.- UNITY PRO

## FDB (9/16)

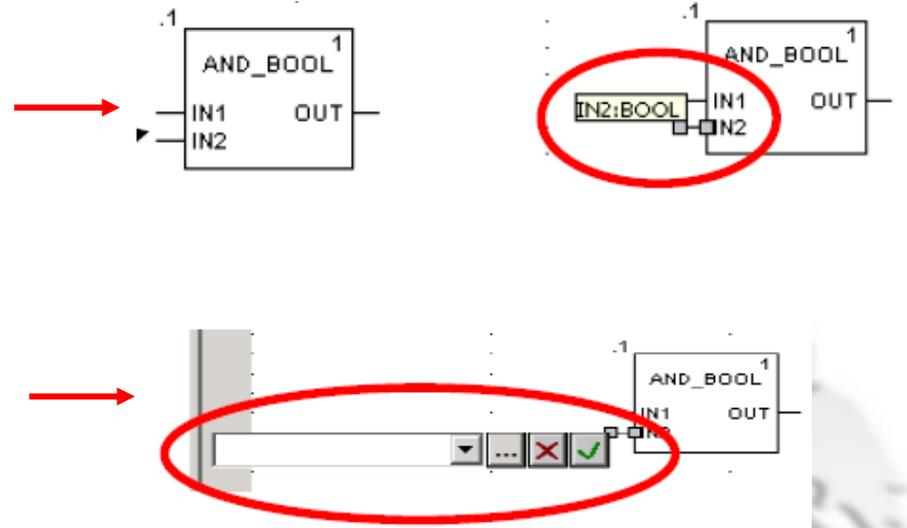
- Una vez elegido el bloque , para poner un bloque en el editor, se ha de hacer un clic izquierdo sobre él, dejar el ratón activado hasta que ha elegido el lugar donde quiere poner el bloque. Por ejemplo: elegimos el bloque “AND\_BOOL”.



## 4.- UNITY PRO

## FDB (10/16)

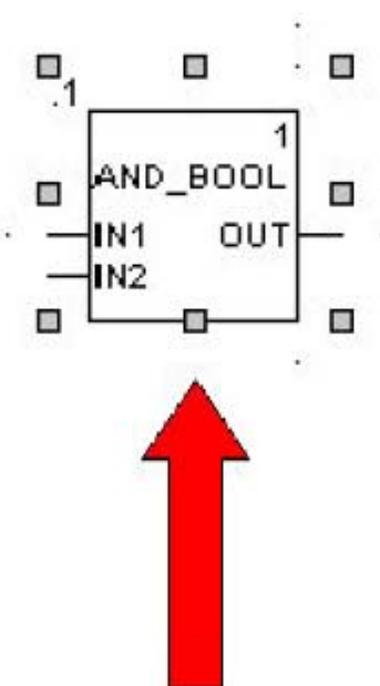
- ¿Cómo asignar una variable, una señal o un literal (constante) a un bloque?:
  1. Una vez puesto el componente en la hoja, clic sobre el terminal de entrada/salida al que asignar variable
  2. Doble clic sobre el terminal de entrada/salida al que asignar variable para que se despliegue el menú de variables



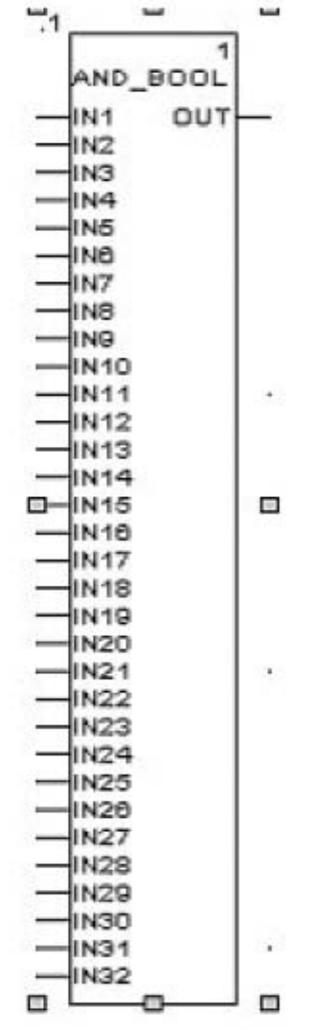
(1)

# 4.- UNITY PRO

# FDB (11/16)



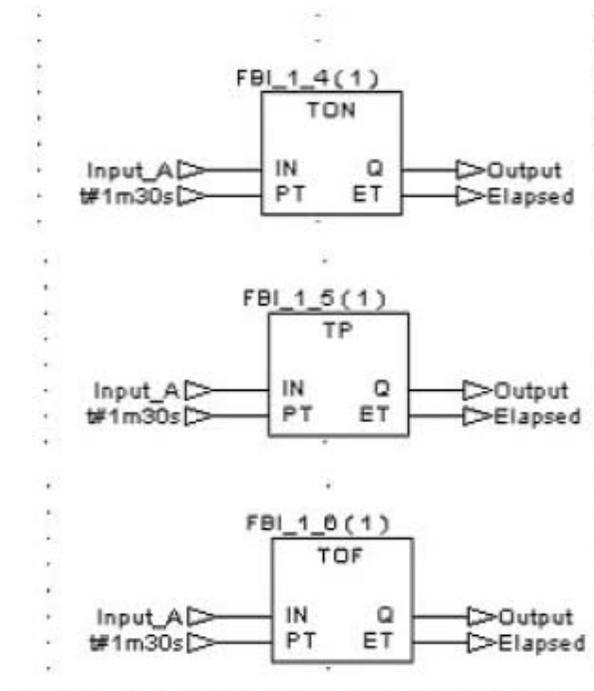
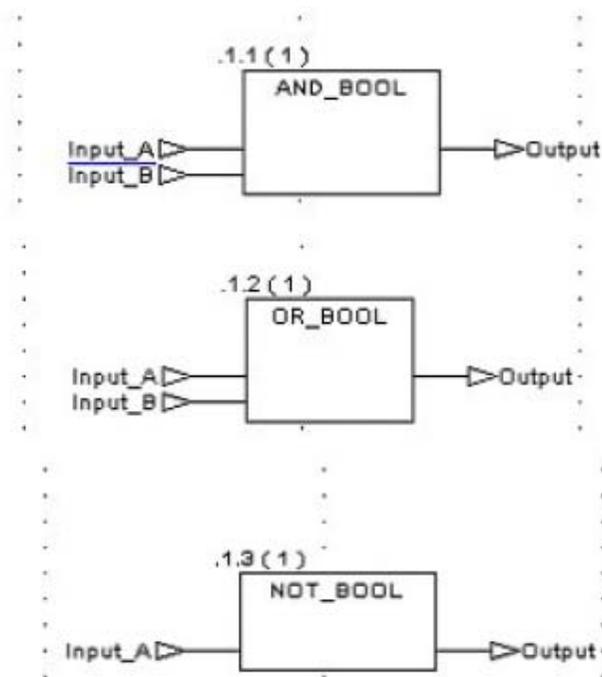
¿ Cuantos pins necesitas ?  
Extiende el bloque de funciones  
¡ Hasta 32 pins !



## 4.- UNITY PRO

## FDB (12/16)

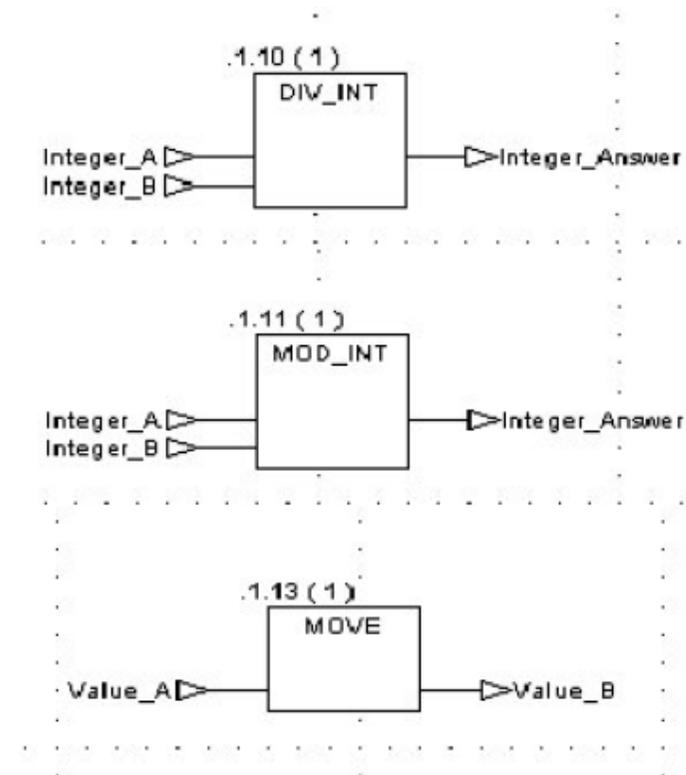
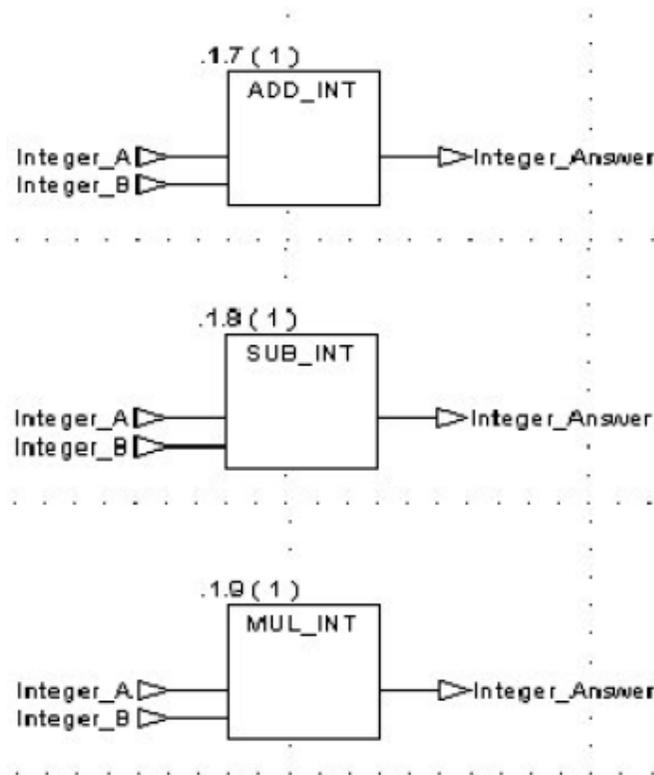
- Lógica básica y funciones de tiempo



## 4.- UNITY PRO

## FDB (13/16)

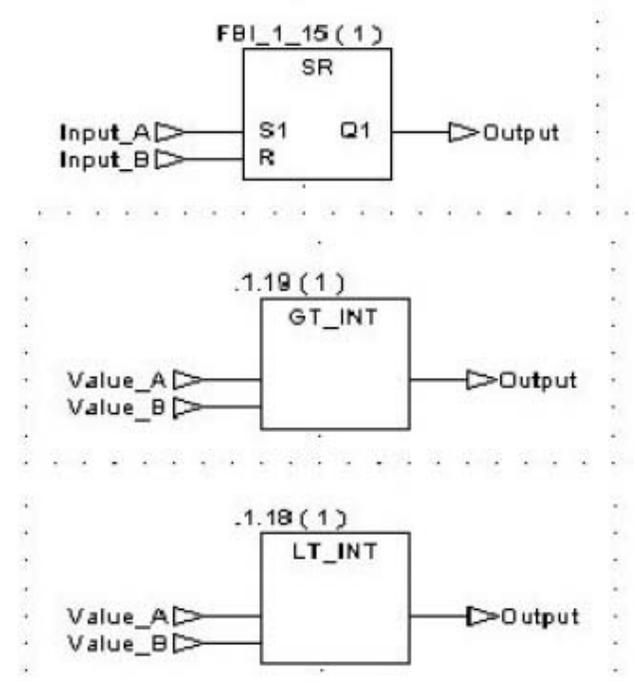
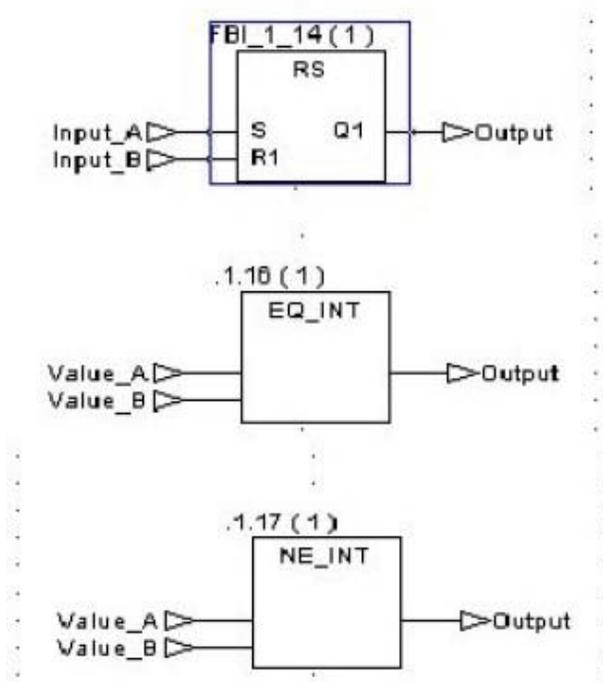
- Funciones básicas matemáticas



## 4.- UNITY PRO

## FDB (14/16)

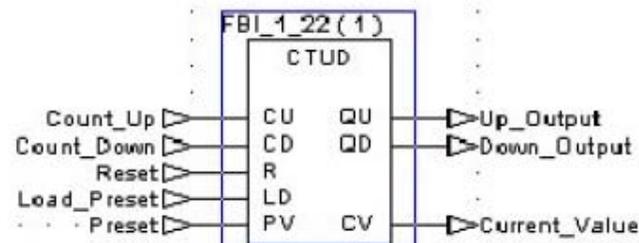
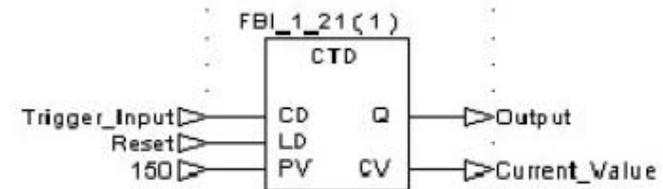
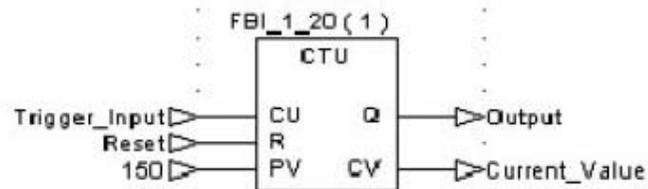
- Latches y comparadores



## 4.- UNITY PRO

## FDB (15/16)

- Contadores

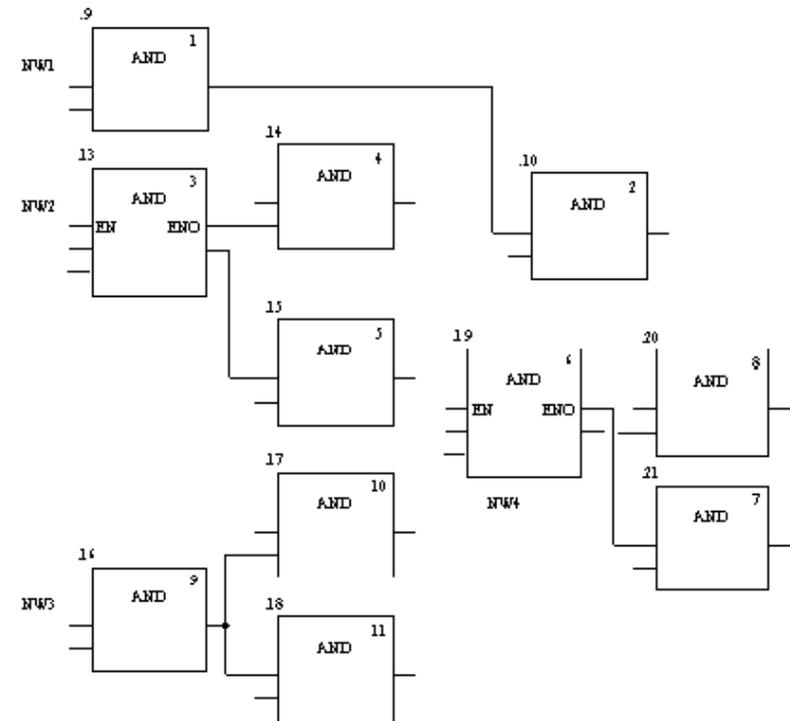


## 4.- UNITY PRO

## FDB (16/16)

### Secuencia de ejecución

- A la derecha hay un ejemplo de la secuencia de objetos en una sección FBD.
- La secuencia de ejecución se determina por la posición de las FFBs dentro de la sección ( ejecutada de izquierda a derecha y desde arriba abajo ). Si los FFBs están conectados gráficamente, la secuencia de ejecución esta determinada por la señal de flujo
- La secuencia de ejecución está indicada por el número de ejecución (número en la esquina superior derecha del FFB)



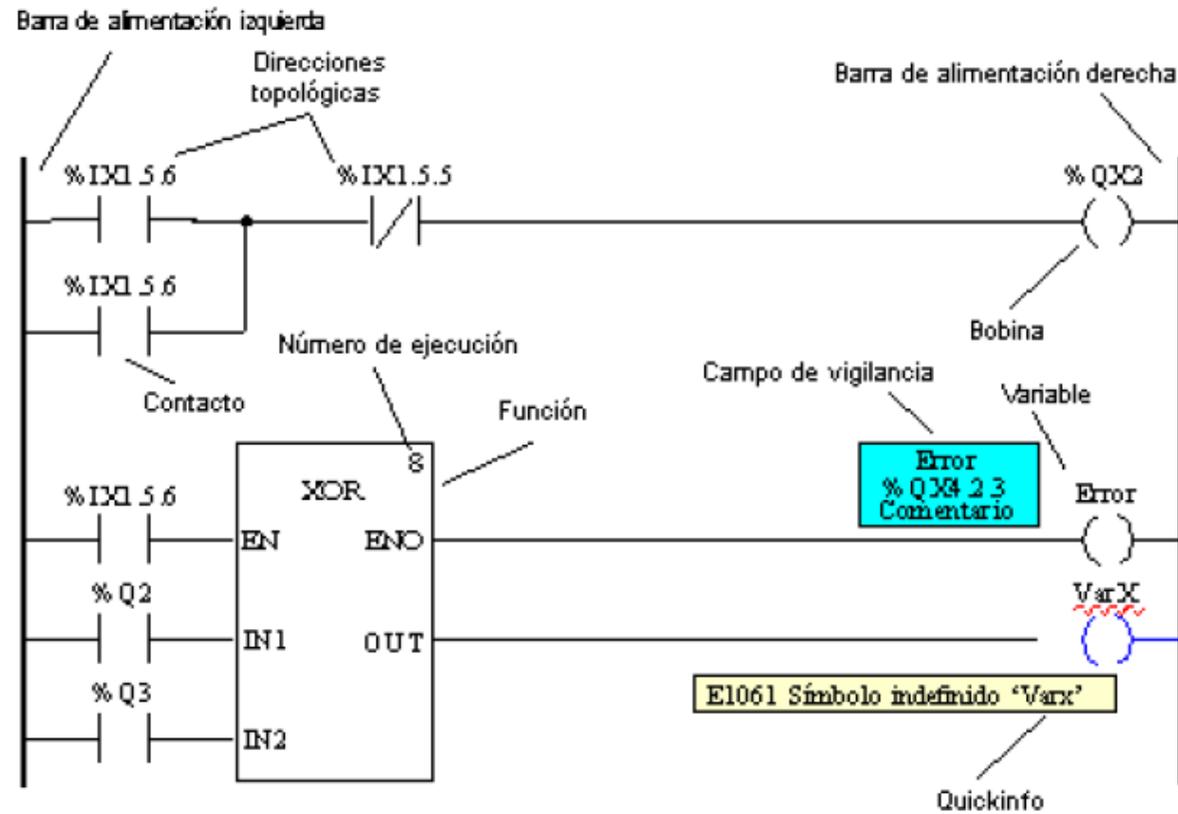
## 4.- UNITY PRO

## Ladder (1/6)

- El lenguaje Ladder (LD) es el de representación de un programa en diagrama de contactos/relés.
- Un grupo de objetos enlazados juntos sin enlaces a otros objetos (excluyendo la línea de potencia) se llama red.
- La secuencia de proceso se determina por el flujo de datos dentro de la sección. Las redes conectadas a la izquierda de la línea de potencia se procesan desde principio a final.
- En el lado izquierdo del **editor LD** se encuentra la denominada "barra de alimentación izquierda". Esta barra de alimentación izquierda corresponde a la fase (conductor L) de un circuito de corriente. Al igual que en un circuito de corriente, en la **programación LD** sólo se "editan" los **objetos LD** que estén conectados a una fuente de alimentación, es decir, que estén conectados a la barra de alimentación izquierda. La barra de alimentación derecha equivale al conductor neutro.

# 4.- UNITY PRO

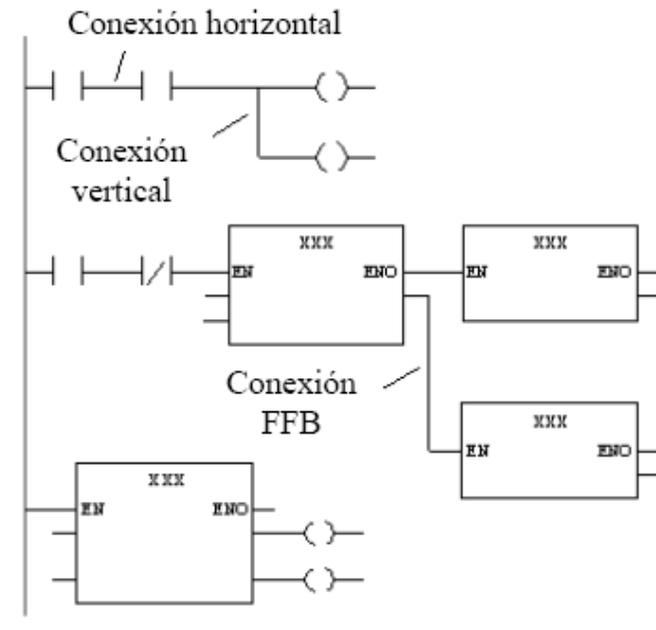
# Ladder (2/6)



## 4.- UNITY PRO

## Ladder (3/6)

- Las comunicaciones son conexiones entre los objetos LD ( contactos, bobinas y FFBs etc.)
- Hay 2 tipos diferentes de comunicaciones:
  - Las comunicaciones booleanas constan de uno o más segmentos de comunicación de objetos Booleanos (contactos, bobinas ) con otros.
  - También hay 2 tipos de estas comunicaciones: Comunicaciones booleanas Horizontales, permiten contactos secuenciales y bobinas intercambiables
  - Comunicaciones booleanas Verticales, permiten contactos paralelos y bobinas intercambiables.
  - Los links FFB – conexiones FFB son combinaciones de los segmentos horizontales y verticales que conectan entradas / salidas FFB con otros objetos.



# 4.- UNITY PRO

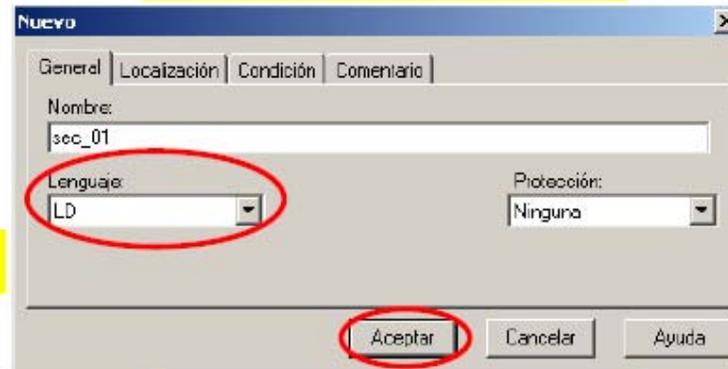
# Ladder (4/6)



1- Hacer clic con el botón derecho del ratón y seleccionar

“Nueva sección...”

2 – Dar un nombre



3- Seleccionar LD

3- Aceptar



## 4.- UNITY PRO

## Ladder (6/6)

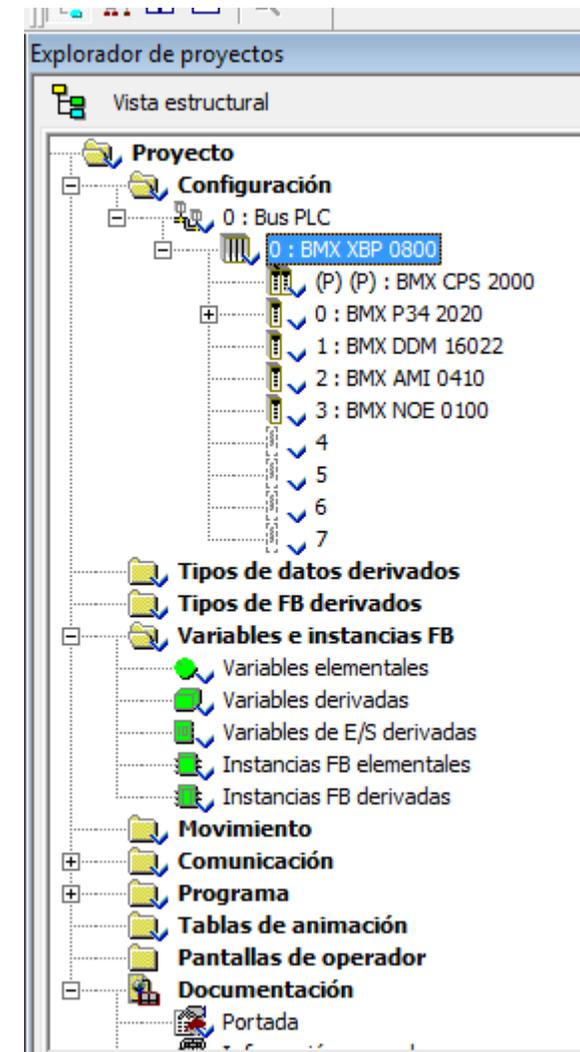
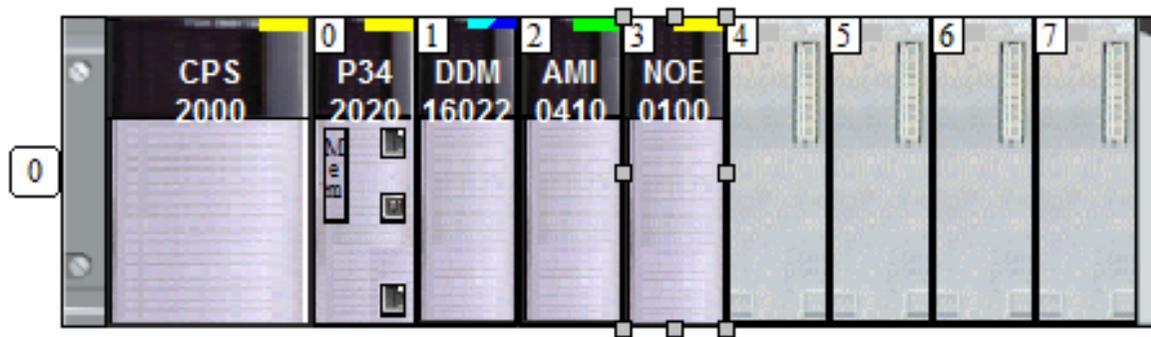
### Conexiones de Objetos:

- Todos los contactos pueden estar conectados en serie ( AND - BOOL )
- Todos los contactos pueden estar conectados en paralelo ( OR - BOOL )
- Todos los contactos pueden estar conectados juntos, en paralelo o en serie
- La red debe estar conectada al rail de potencia
- El rail de potencia se refiere a los datos de tipo BOOL
- FFB's pueden insertarse en cualquier posición de la red
- Las bobinas tienen que estar conectadas a los contactos o a las salidas FFB
- ¡ El lado derecho e izquierdo de las bobinas siempre tienen el mismo valor !

## 5.- Configuración del Hardware (1/3)

### Fases de un diseño con UNITY:

- Crear nuevo proyecto
- Configurar Hardware:
  - CPU BMX P34 2020 1.0 (firmware)
  - Tarjeta E/S Digitales (DDM16022)
  - Tarjeta E Analógicas (AMI0410)
  - Tarjeta Ethernet (NOE0100)



## 5.- Configuración del Hardware (2/3)

---

### ¿Qué rango de direcciones maneja mi CPU?

- Esta CPU posee marcas libres para propósito general:
  - %M0-511 de tipo binario
  - %MW0-1023 de tipo 16 bits
  - %KW0-255
- También maneja direcciones de E/S:
  - Canal, configuración, sistema, estado, comando, implícitos.



## 5.- Configuración del Hardware (3/3)

- Para ver las direcciones que soporta la CPU, dentro del menú configuración doble click en la CPU, pestaña Objetos de E/S

0.0 : BMX P34 2020  
CPU 340-20 Modbus Ethernet

Vista general Configuración Animación **Objetos de E/S**

Objetos de la CPU

Sistema:  %S  %SW

Memoria:  %M  %MW  %MD  %MF  %KW  %KD  %KF

Objetos de E/S

Canal:  %CH

Configuración:  %KW  %KD  %KF

Sistema:  %MW

Estado:  %MW

Parámetro:  %MW  %MD  %MF

Comando:  %MW  %MD  %MF

Implícitos:  %I  %IW  %ID  %IF  %IERR  %Q  %QW  %QD  %QF

Actualizar

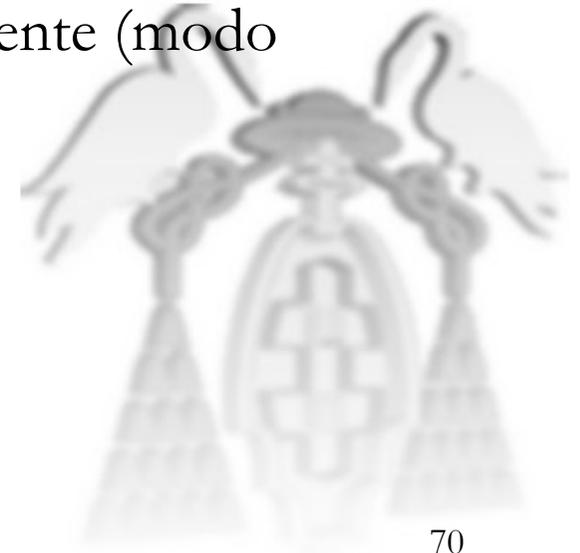
Actualizar cuadrícula con  Direcciones  Nombres, tipos y comentarios

	Dirección	Nombre	Tipo	Comentario
1	%CH0.0.0			
2	%CH0.0.MOD			
3	%CH0.1.MOD			
4	%CH0.1.0			
5	%CH0.1.1			
6	%CH0.1.2			
7	%CH0.1.3			
8	%CH0.1.4			
9	%CH0.1.5			
10	%CH0.1.6			
11	%CH0.1.7			
12	%CH0.1.16			
13	%CH0.1.17			
14	%CH0.1.18			
15	%CH0.1.19			
16	%CH0.1.20			
17	%CH0.1.21			
18	%CH0.1.22			
19	%CH0.1.23			
20	%CH0.2.MOD			
21	%CH0.2.0			
22	%CH0.2.1			

## 6.- Descarga de programas (1/6)

---

- Una vez editado un programa, los siguientes pasos son:
  - Seleccionar modo de funcionamiento: PLC/Simulador
  - Analizar: Detecta posibles errores de sintaxis
  - Generar cambios: Modifica el fichero transferible al PLC con los nuevos cambios
  - Regenerar el proyecto: Preparar el fichero transferible al PLC
  - Conectar al PLC: Conecta física o virtualmente (modo simulador) el PC con el PLC
  - Transferir programa al PLC.
  - RUN: Lanza el programa



## 6.- Descarga de programas (2/6)

- Proyecto no generado

The screenshot displays the Unity Pro XL software interface. The main window shows a ladder logic diagram with three rungs. Rung 1 contains an EQ\_BOOL (Equality Boolean) block with inputs ED1 and ED2, and output OUT. Rung 2 contains an AND\_BOOL (AND Boolean) block with inputs ED1, ED2, and ED3, and output OUT. Rung 3 contains a TON (Timer On Delay) block with input IN, preset time PT set to #5s, and output Q labeled SALIDA3. The left sidebar shows the project structure, including sections for data types, variables, movement, communication, and programs. The status bar at the bottom indicates "Proceso satisfactorio : 0 Errores , 0 Advertencias" and "Regenerar todo el proyecto". A red circle highlights the "NO GENERADO" status indicator in the bottom right corner of the interface.

## 6.- Descarga de programas (3/6)

- Proyecto generado

The screenshot displays the Unity Pro XL software interface. On the left, the 'Explorador de proyectos' (Project Explorer) shows a hierarchical structure of the project, including 'Bus PLC', 'Tipos de datos derivados', 'Tipos de FB derivados', 'Variables e instancias FB', 'Movimiento', 'Comunicación', 'Programa', 'Tareas', 'Secciones', 'Eventos', 'Tablas de animación', 'Pantallas de operador', and 'Documentación'. The main workspace shows a ladder logic diagram with three blocks: 'EQ\_BOOL' (Block 1), 'TON' (Block 2), and 'AND\_BOOL' (Block 3). The 'EQ\_BOOL' block has inputs ED1 and ED2, and output OUT. The 'TON' block has input IN, preset time PT (set to #5s), and output Q (SALIDA3). The 'AND\_BOOL' block has inputs ED1, ED2, and ED3, and output OUT (SALIDA2). The bottom status bar shows 'Proceso satisfactorio : 0 Errores , 0 Advertencias' and a 'Regenerar todo el proyecto' button. A red circle highlights the 'GENERADO' button in the bottom right corner of the status bar.

## 6.- Descarga de programas (4/6)

- PLC Conectado

The screenshot displays the Unity Pro XL software interface for a PLC program. The main workspace shows a ladder logic diagram with three rungs:

- Rung 10:** Contains an **EQ\_BOOL** block with inputs ED1 (IN1) and ED2 (IN2), and output OUT. It is connected to the IN input of a **TON** block.
- Rung 20:** Contains an **AND\_BOOL** block with inputs ED1 (IN1), ED2 (IN2), and ED3 (IN3), and output OUT (SALIDA2).
- Rung 30:** Contains a **TON** block with inputs IN and PT (w5s), and outputs Q (SALIDA3) and ET.

The interface also shows a project explorer on the left with a tree view of the project structure, including folders for 'Tipos de datos derivados', 'Tipos de FB derivados', 'Variables e instancias FB', 'Movimiento', 'Comunicación', 'Programa', and 'Tablas de animación'. The status bar at the bottom indicates 'Proceso satisfactorio : 0 Errores , 0 Advertencias'.

## 6.- Descarga de programas (5/6)

- Proyecto transferido pero no arrancado

The screenshot displays the Unity Pro XL software interface. The main workspace shows a ladder logic diagram with three rungs. Rung 1 contains an EQ\_BOOL (Equality Boolean) block with inputs ED1 and ED2, and output IN2. Rung 2 contains an AND\_BOOL (AND Boolean) block with inputs ED1, ED2, and ED3, and output IN3. Rung 3 contains a TON (Timer On Delay) block with input IN, preset time PT of 5s, and output ET. The output ET is connected to SALIDA3. The project explorer on the left shows a hierarchical structure of the project, including folders for 'Tipos de datos derivados', 'Tipos de FB derivados', 'Variables e instancias FB', 'Movimiento', 'Comunicación', 'Programa', 'Tareas', 'Secciones', 'Eventos', 'Tablas de animación', 'Pantallas de operador', and 'Documentación'. The status bar at the bottom indicates 'Proceso satisfactorio : 0 Errores , 0 Advertencias' and 'Regenerar todo el proyecto'.

## 6.- Descarga de programas (6/6)

- Proyecto transferido y arrancado

The screenshot displays the Unity Pro XL software interface for a PLC project. The main workspace shows a ladder logic diagram with three rungs:

- Rung 10:** Contains an **EQ\_BOOL** block (1) with inputs ED1 and ED2, and output OUT. It is connected to a **TON** block (2) with input IN and output Q. The TON block has a preset time (PT) of #5s and an enable time (ET) of #1s\_295ms. The output Q is labeled **SALIDA3**.
- Rung 20:** Contains an **AND\_BOOL** block (3) with inputs ED1, ED2, and ED3, and output OUT. The output is labeled **SALIDA2**.
- Rung 30:** Is currently empty.

The project explorer on the left shows the following structure:

- 0 : Bus PLC
  - 0 : BMX XBP 0800
    - Tipos de datos derivados
    - Tipos de FB derivados
    - Variables e instancias FB
      - Variables elementales
      - Variables derivadas
      - Variables de E/S derivadas
      - Instancias FB elementales
      - Instancias FB derivadas
    - Movimiento
    - Comunicación
    - Programa
      - Tareas
        - MAST
          - Secciones
            - rejem
            - ld1
            - FBD1
          - Secciones SR
        - Eventos
          - Eventos de temporización
          - Eventos de E/S
        - Tablas de animación
          - Tabla[Editor LD - ld1 : [M...
          - Tabla[Bits forzados]
        - Pantallas de operador
        - Documentación

The status bar at the bottom indicates: **Proceso satisfactorio : 0 Errores , 0 Advertencias**. The bottom right corner shows the status: **Modalidad HMI R/W IGUAL EJE. UPLOAD INFO OK USB:SYS GENERADO**.