



Autómata programable

Eunea

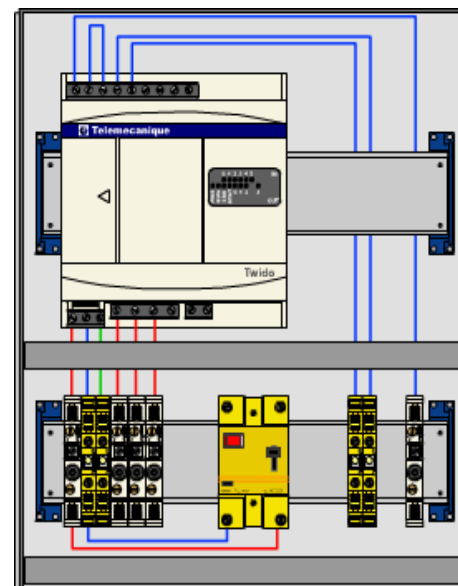
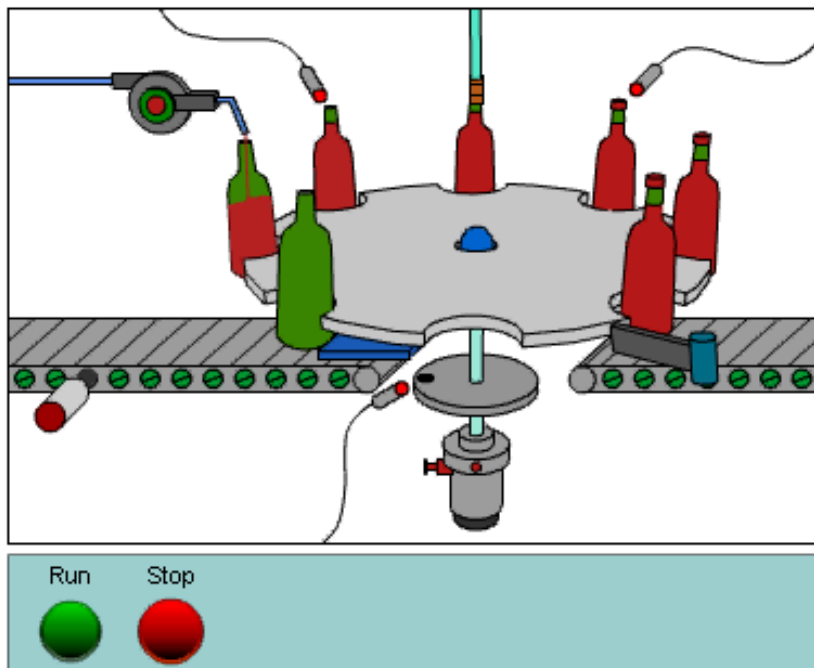
Merlin Gerin

Square D

Telemecanique



Integración de un PLC en un sistema automatizado



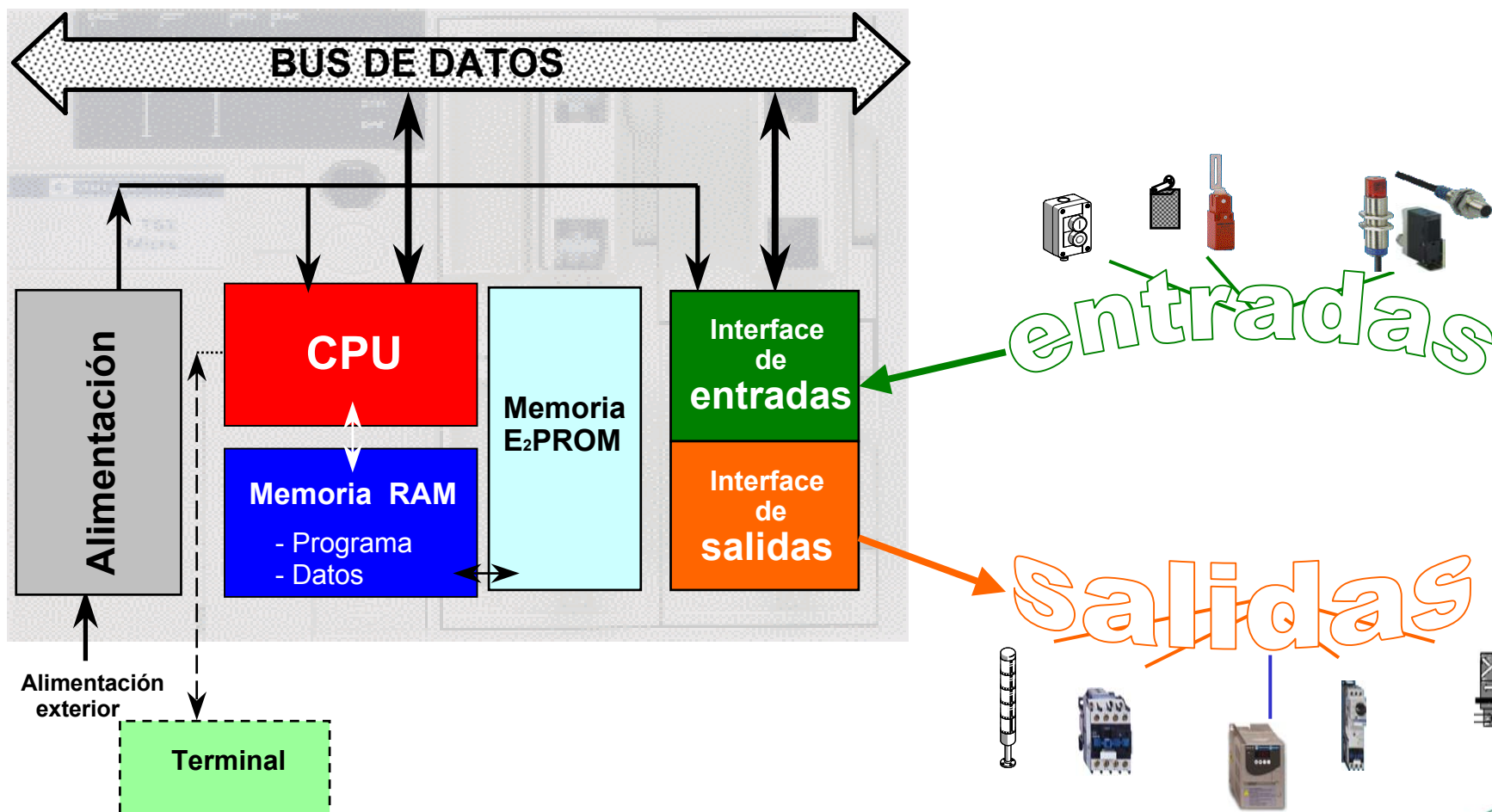
Fusible

Terminal

Disyuntor



Estructura de un AUTOMATA PROGRAMABLE



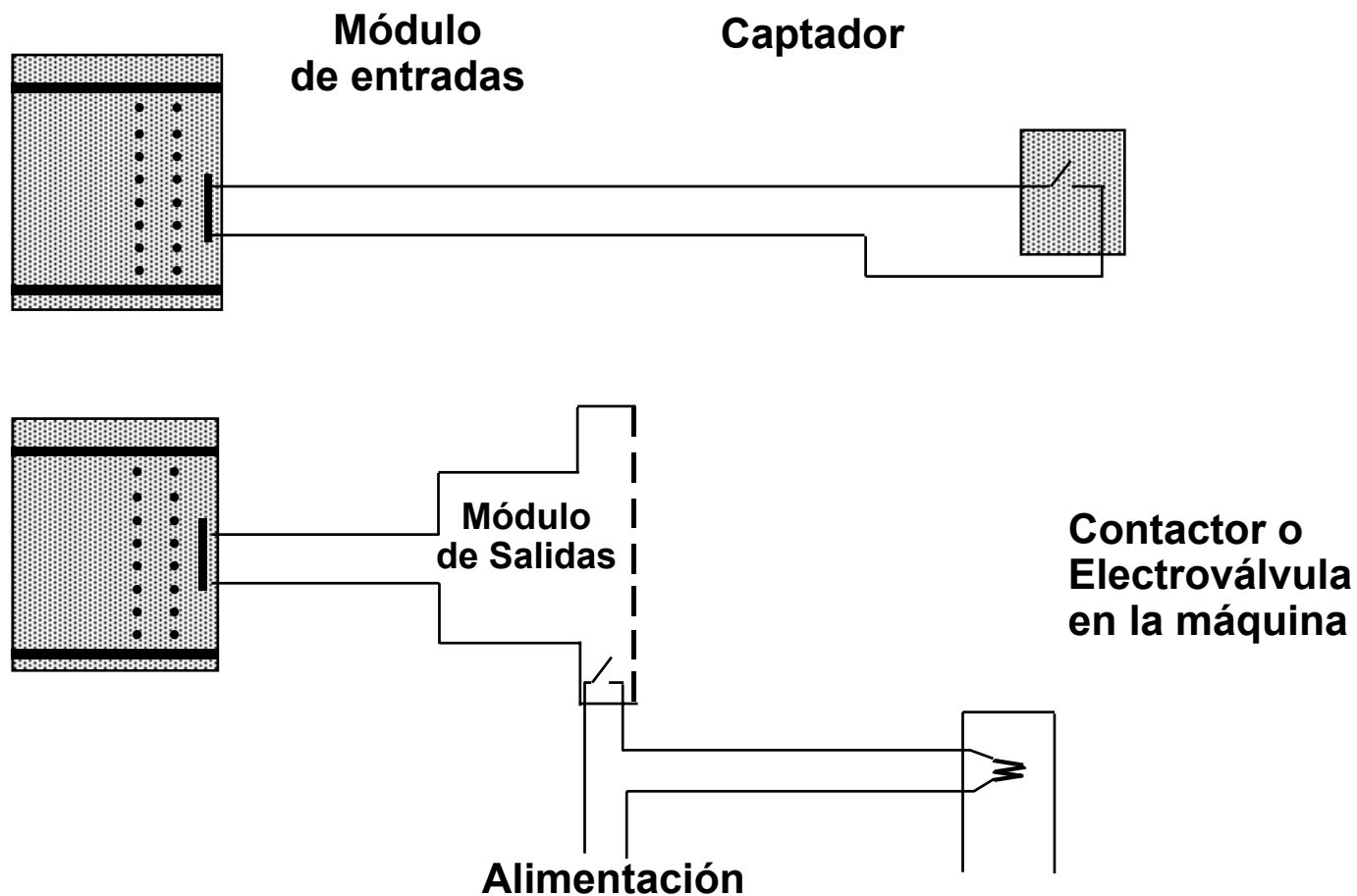


La cadena completa de un automatismo





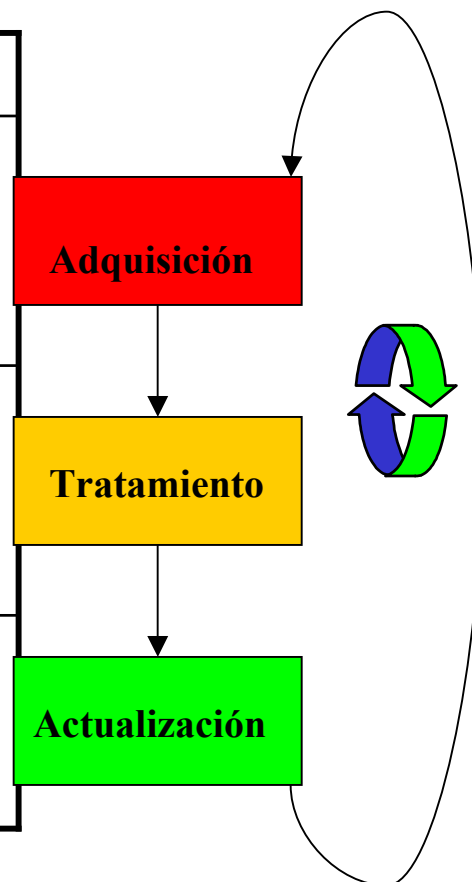
Funcionamiento general y estructura





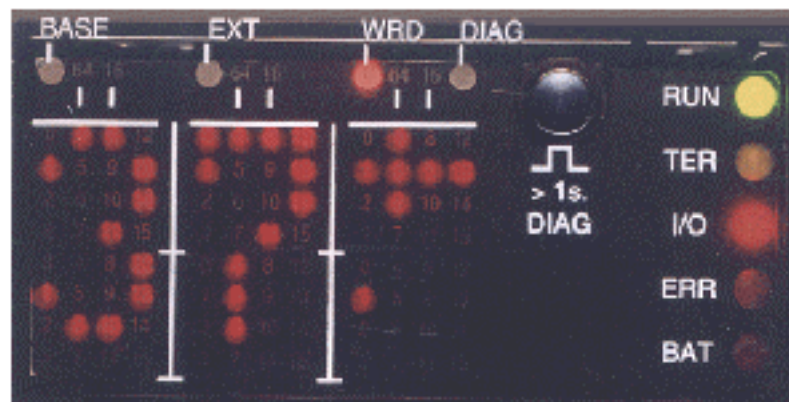
Funcionamiento general y estructura

FASE	ACCION
Primera	Adquisición del estado de las entradas (y memorización de las mismas en la memoria de datos).
Segunda	Tratamiento del programa (y actualización de las imágenes de las salidas en la memoria de datos).
Tercera	Actualización de las salidas (las imágenes de las salidas se transfieren a los interfaces de salida)





Indicadores visuales de la CPU y E/S



140	140	140	140	140	140
CPS 114 00	CPU 113 00	DAI 553 00	DAO 842 10	DDI 353 00	DDO 353 00
AC Power Supply	CPU Module	115 VAC In	100-230 VAC	24 VDC In	24 VDC Out
R Active F	R Active F	R Active F	R Active F	R Active F	R Active F
1	1	1	1	1	1
2	2	2	2	2	2
3	3	3	3	3	3
4	4	4	4	4	4
5	5	5	5	5	5
6	6	6	6	6	6
7	7	7	7	7	7
8	8	8	8	8	8



Direccionamiento de E/S

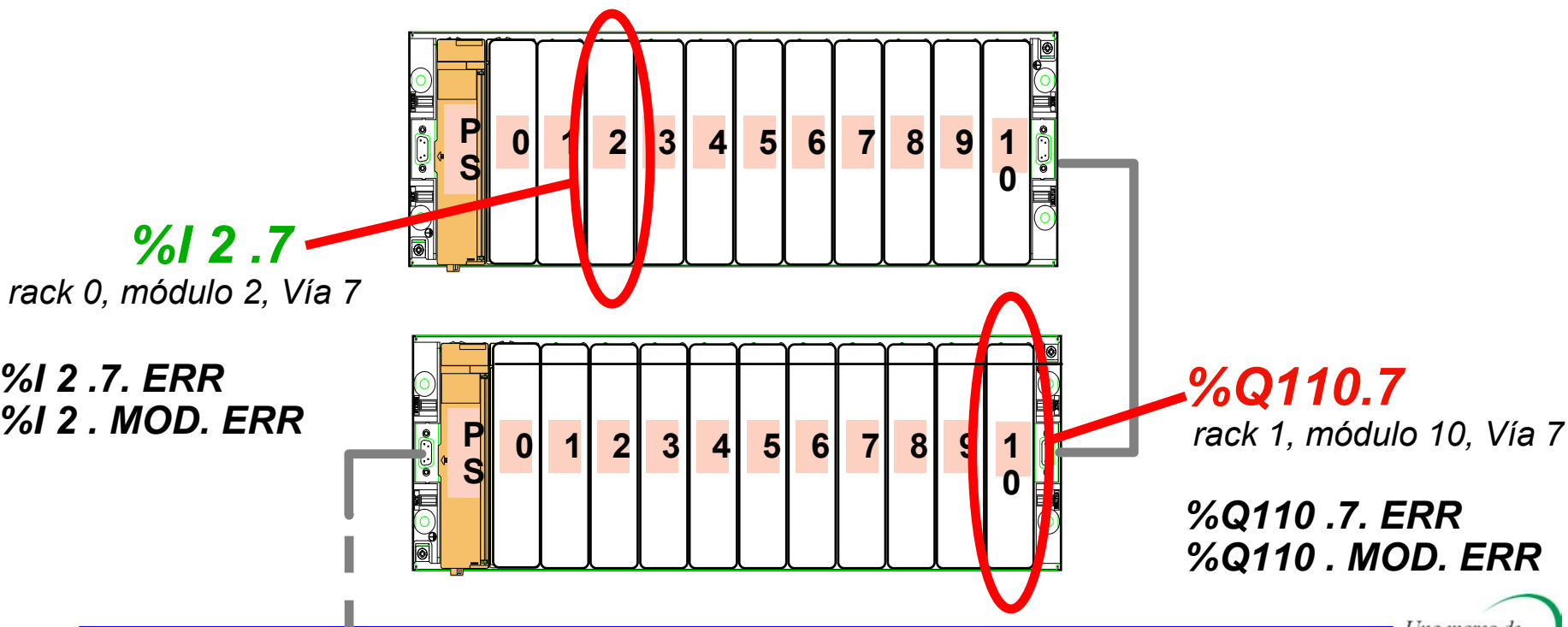
% I x y . i / % Q x y . i

x = Rack : 0 a 7

y = Posición del módulo : 00 a 14

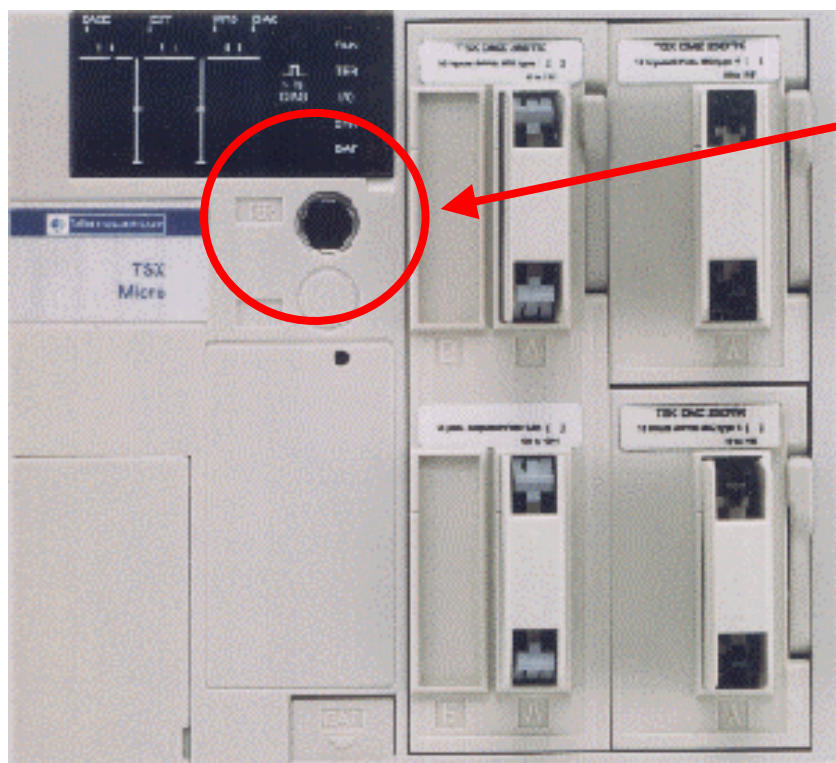
i = Vía 0 a 63

% IW x y . i / % QW x y . i



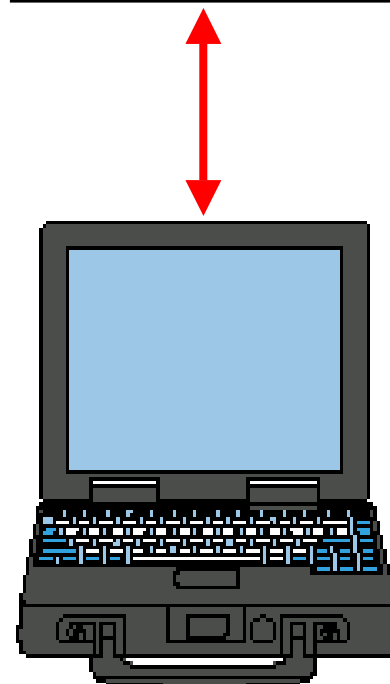


Utiles de programación



**Autómata
Programable**

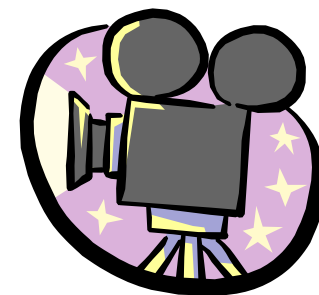
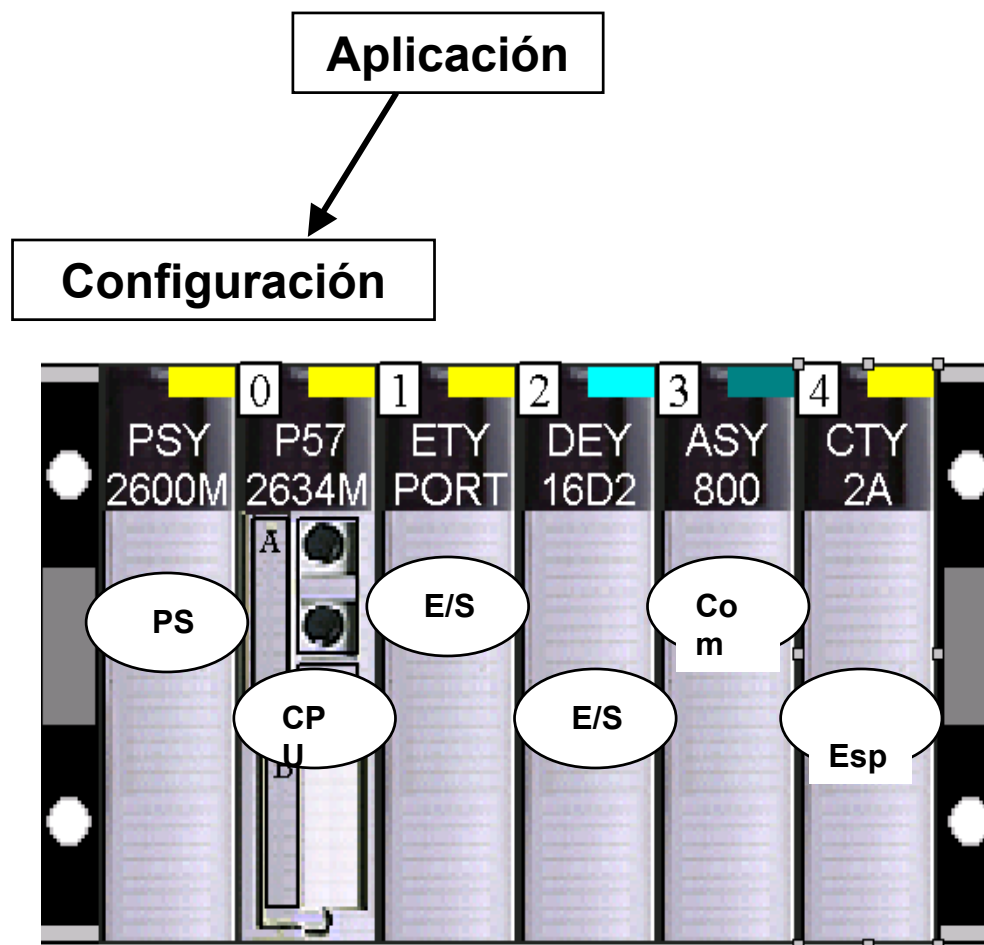
**Puerto de
comunicación**



**Consola de
programación**



Estructura de una Aplicación





Estructura de una Aplicación

Aplicación

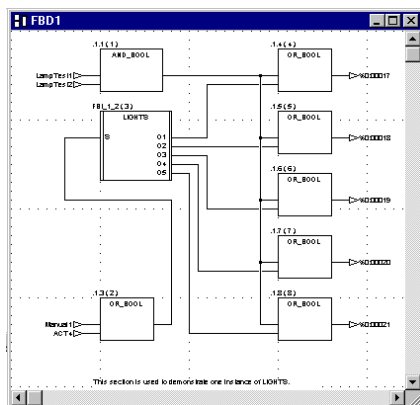
Sección 1

Sección 2

...

Sección n

Programación



Sección 1

```
Structured Text Start
VAR
  TIMER : TON;
END_VAR

TIMER(IN := NOT pulse,
      PT := T#1s); (* Blink timer *)
pulse := TIMER.Q;

(* Count every pulse *)
IF pulse = 1 THEN
  count := count + 1;
END_IF;

(* Animate lights according to counter *)
CASE count OF
  1: out1 := TRUE;
  2: out2 := TRUE;
  3: out3 := TRUE;
  4: out4 := TRUE;
  5: out5 := TRUE;
  6: out6 := TRUE;
  7: out7 := TRUE;
  8: out8 := TRUE;
ELSE (* All lights are on, switch them off again *)
  out1 := FALSE;
  out2 := FALSE;
  out3 := FALSE;
  out4 := FALSE;
  out5 := FALSE;
  out6 := FALSE;
  out7 := FALSE;
  out8 := FALSE;
  count := 0;
END_CASE;
Structured Text End
```

Sección 2

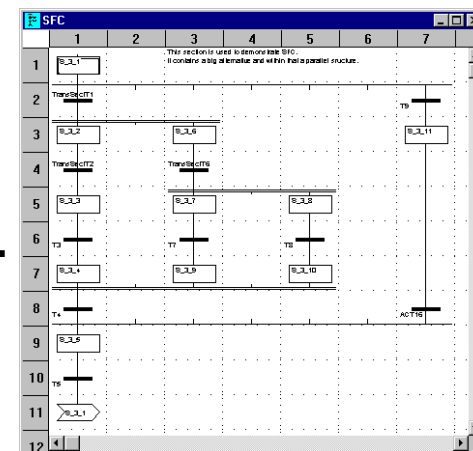
```
Instruction List Start
VAR
  RUN_TIMER : TON; (* Blink timer *)
END_VAR

(* Default for the marker *)
LD run_light1
ST run_light

(* Create a 1.0 Hz. pulse *)
LD run_pulse
STN RUN_TIMER.IN
CAL RUN_TIMER.PT := T#1s
LD RUN_TIMER.QI
LD animateTime
LD RUN_TIMER.Q
ST run_pulse
JMPCH end

(* No pulse yet, nothing to do *)
LD run_light8
ST run_light
LD run_light7
ST run_light
LD run_light6
ST run_light
LD run_light5
ST run_light
LD run_light4
ST run_light
LD run_light3
ST run_light
LD run_light2
ST run_light
LD run_light1
ST run_light
end:LD
ST run_light
```

Sección 3



...

Sección n



El Estándar IEC 61131

- **IEC 1131-1**
Información
General

- **IEC 1131-2**
Requerimiento de los
Equipos y pruebas

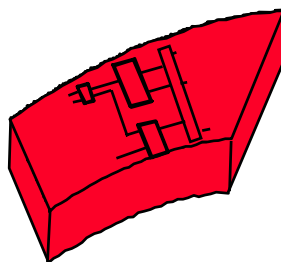
- **IEC 1131-3**
Lenguajes de Programación

- **IEC 1131-4**
Guías de Usuario

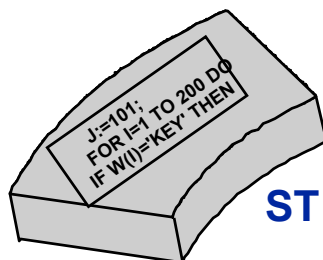
- **IEC 1131-5**
Especificación de los
mensajes de ayuda



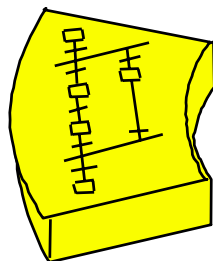
Lenguajes de programación – La norma IEC 61131



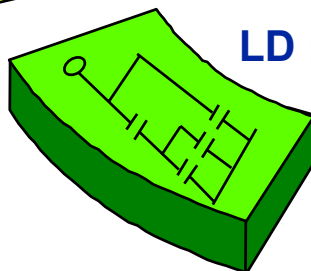
FBD : Bloques Función



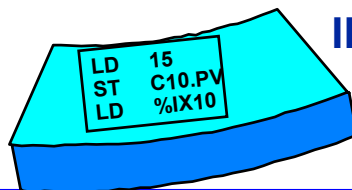
ST : Texto estructurado



SFC : Grafcet

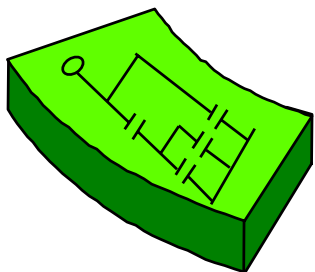


LD : Diagrama contactos



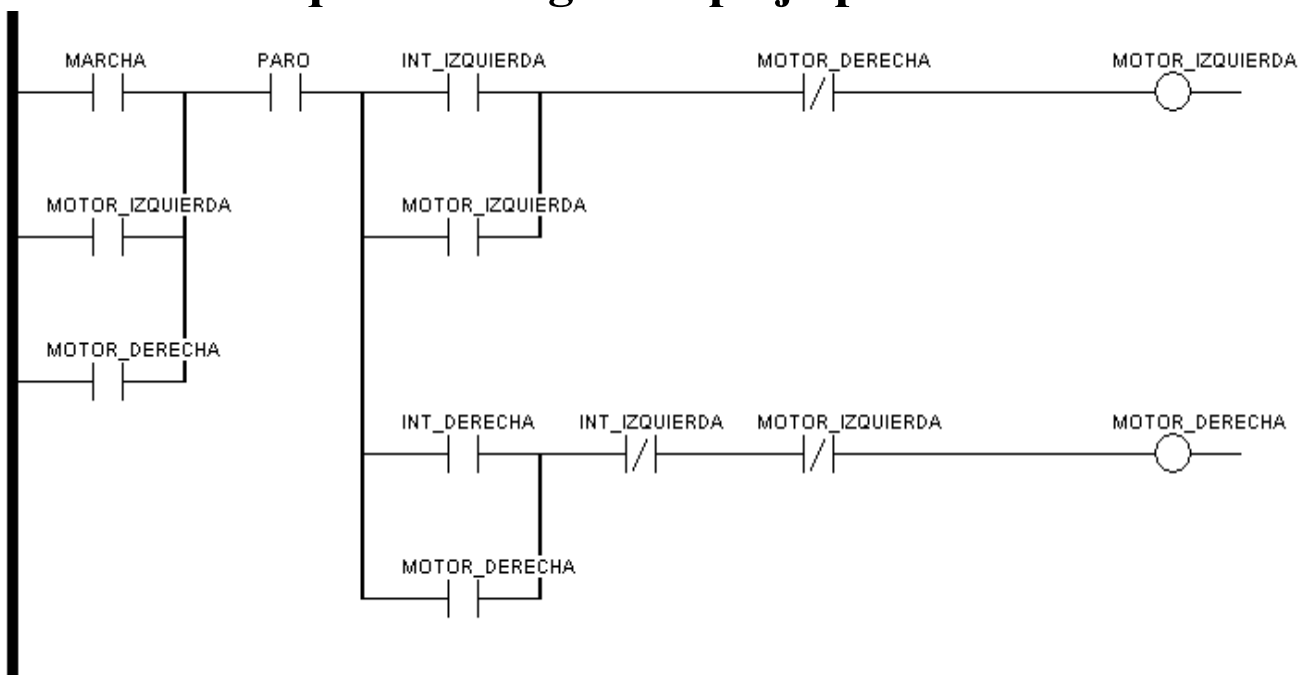
IL : Lista Instrucciones





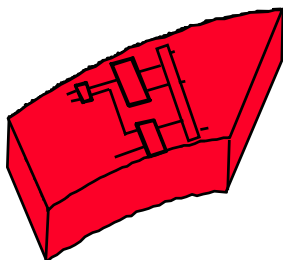
LD: Diagrama de Contactos

- Gráfico / Simbólico
- Orientado al control Discreto
- Familiar → Muy utilizado
- Fácil de mantener
- Requiere código complejo para funciones avanzadas

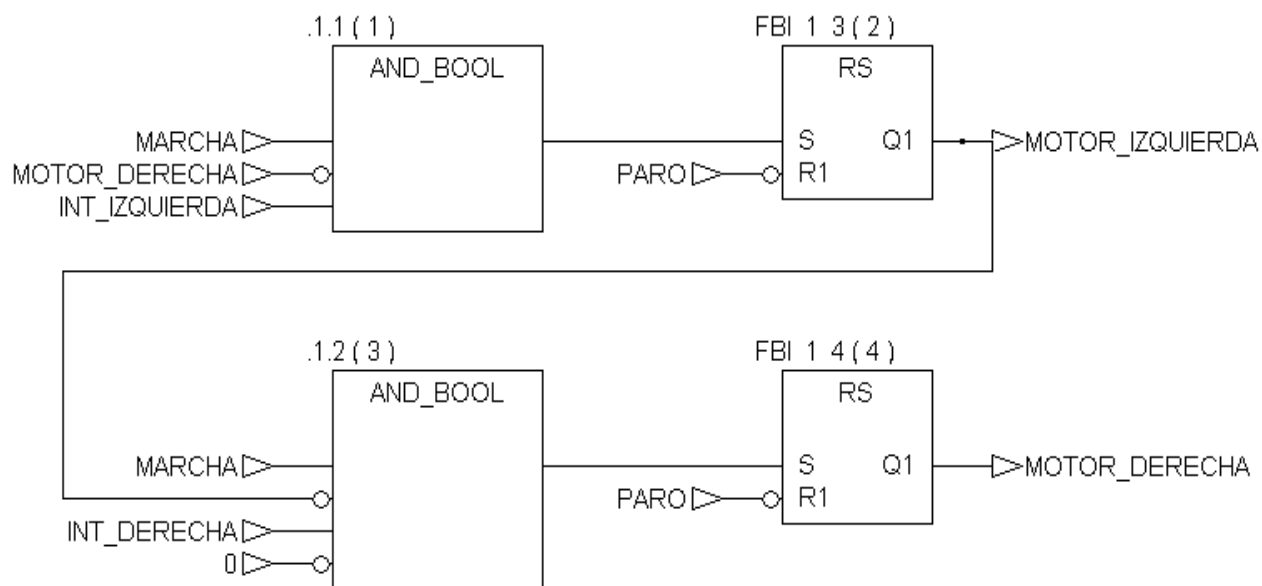


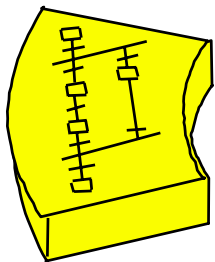


FBD: Diagrama de Bloques de Función



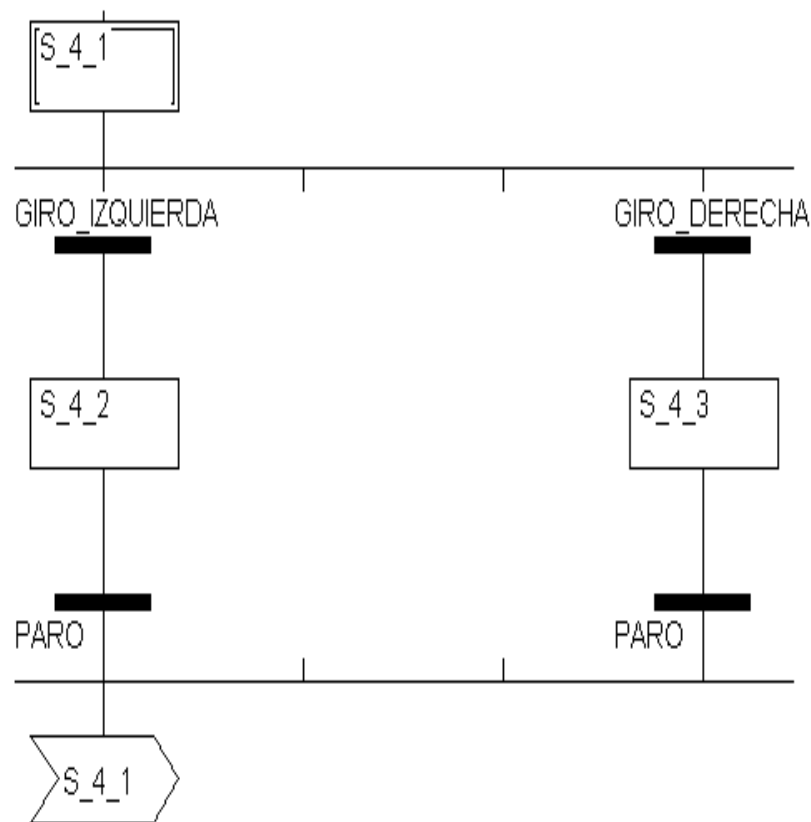
- **Gráfico / Simbólico**
- **Orientado al Proceso**
- **Funcionalidad Jerárquica**
- **Básicos & Derivados**





SFC: Secuencia de Funciones

- **Gáfico**
- **Diagrama de representación del Proceso**
- **“Steps & Transitions” (Pasos y Transiciones)**
- **Las condiciones de “Transición” pueden ser definidas en: LD, FBD, IL & ST**
- **Orientado a la Actividad de Flujo (Lenguaje Secuencial)**
- **Simplifica el mantenimiento**





ST: Texto Estructurado



- **Literal**
- **Lenguaje de Alto Nivel (tipo Pascal)**
- **Fácil de Importar/Exportar**
- **Facilita el manejo Matemático**
- **Simplifica la estructura de los programas en el uso de DFB's**

```
IF NOT PARO THEN
```

```
    IF MARCHA AND INT_IZQUIERDA AND NOT MOTOR_DERECHA THEN  
        MOTOR_IZQUIERDA:=1;  
    END_IF;
```

```
    IF MARCHA AND INT_DERECHA AND NOT INT_IZQUIERDA AND NOT MOTOR_IZQUIERDA THEN  
        MOTOR_DERECHA:=1;  
    END_IF;
```

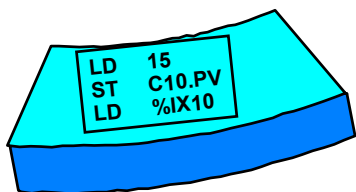
```
ELSE
```

```
    MOTOR_IZQUIERDA:=0;  
    MOTOR_DERECHA:=0;
```

```
END_IF;
```



IL: Lista de instrucciones

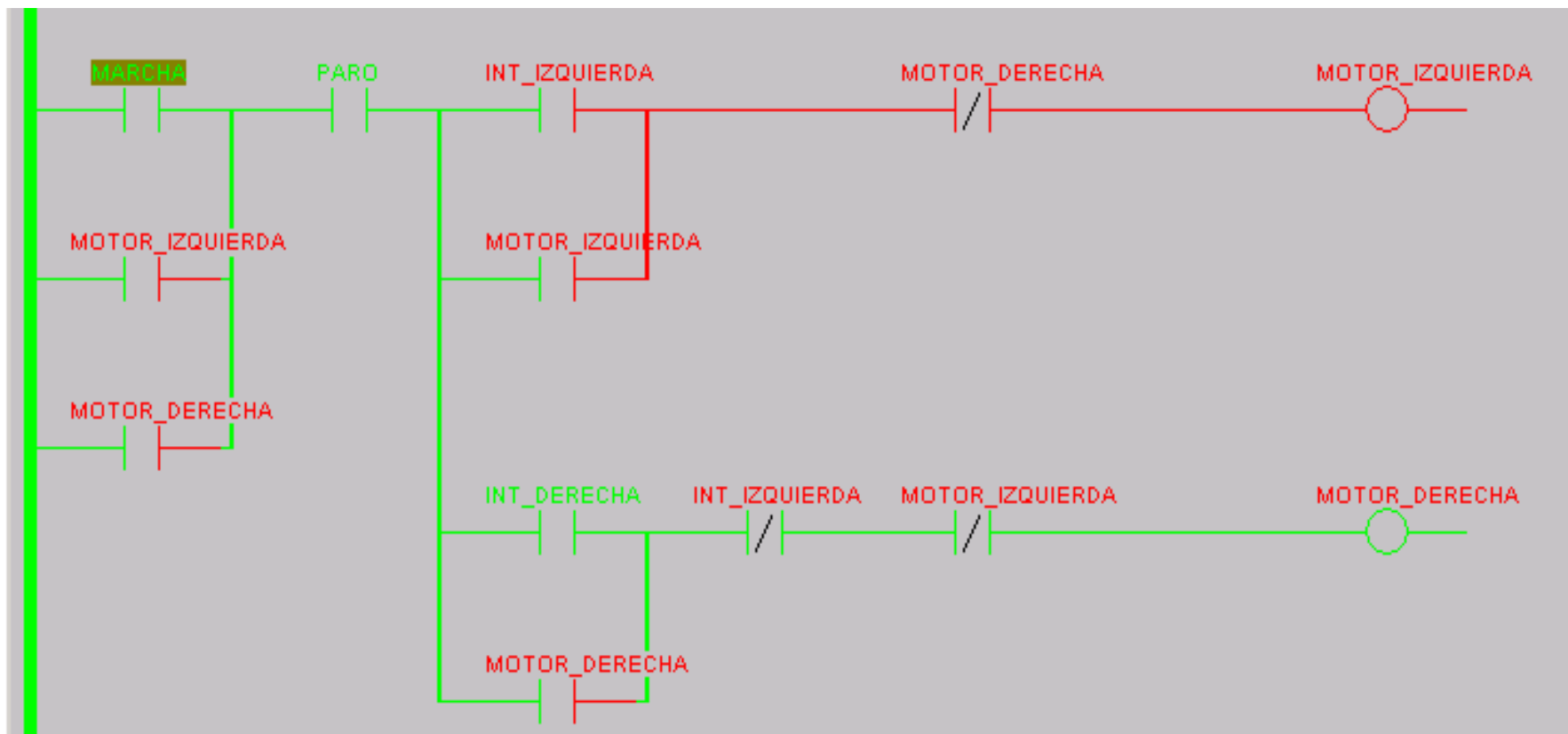


- ✓ Editor Textual (tipo Assembler)
- ✓ Aceptación Limitada
- ✓ No tiene base de Ingeniería
- ✓ Difícil de visualizar en pruebas
- ✓ Fácil de Importar/Exportar

LD	15
ST	C10.PV
LD	%IX10
ST	C10.CU
CAL	C10

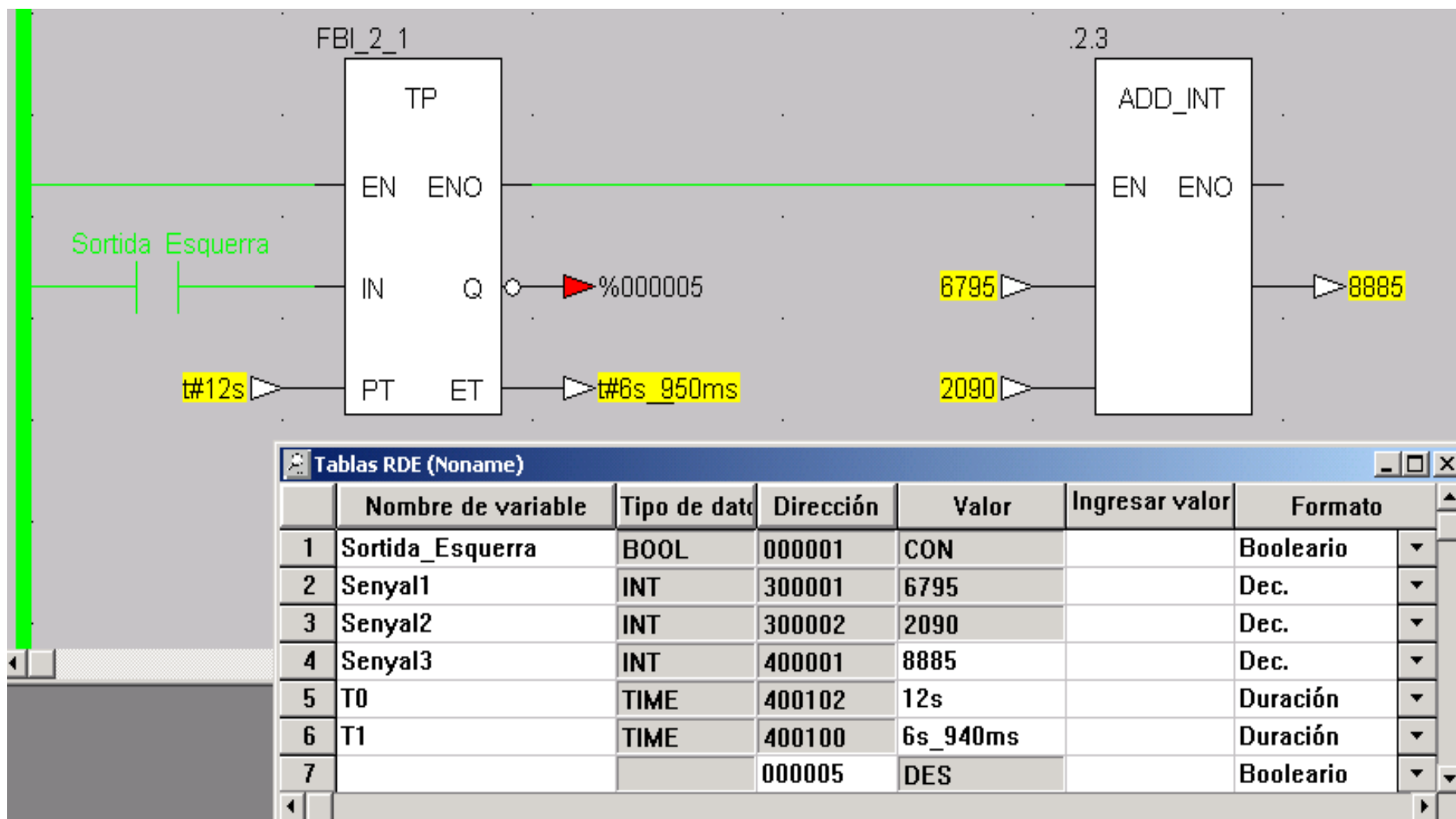


Animación de la programación



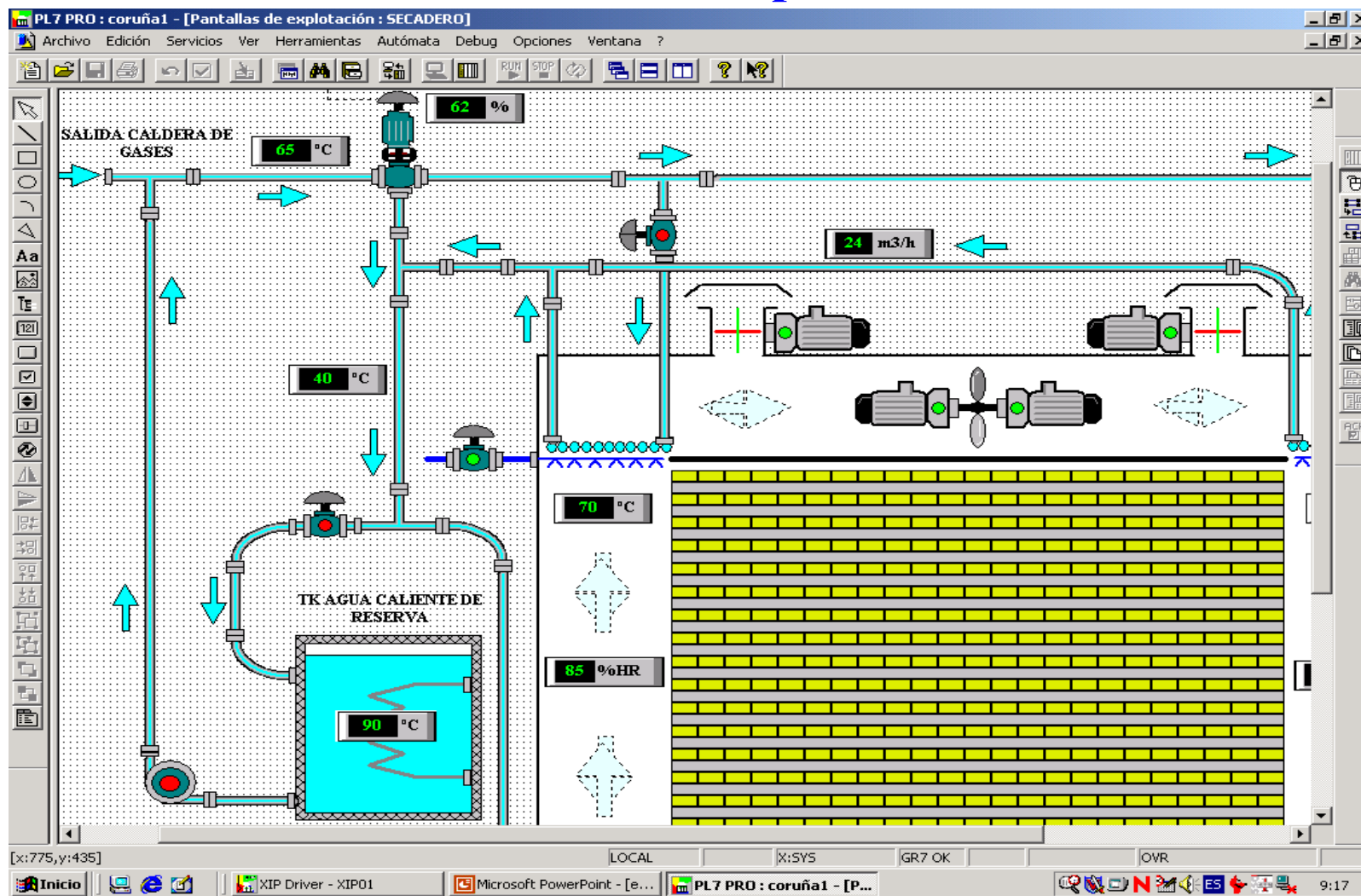


Tablas de animación



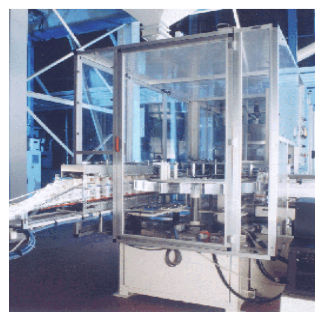


Pantallas de explotación





Gama Autómatas



↑
aplicación
centralizada

aplicación
distribuida

↓
aplicación
autónoma

Compact

Momentum

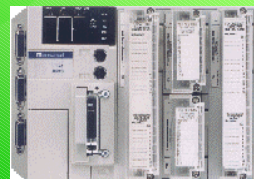


Premium



Quantum

Twido



Micro

248 E/S

468 E/S

2048 E/S

65000 E/S

Maquinaria

Manufacturero

Semi-continuo

Continuo



Factores fundamentales en las aplicaciones de automatización



- ✓ **Número de E/S digitales, E/S analógicas (en rack o distribuidas)**
- ✓ **Funciones especiales: conteo rápido, control ejes, pesaje,**
- ✓ **Distribución geográfica de las E/S: bus de campo. Distancias a cubrir.**
- ✓ **Distribución de inteligencia**
- ✓ **Posibilidad de comunicación con otros PLCs, Pantallas HMI, Web browser,.. Tráfico requerido.**
- ✓ **Capacidad de telediagnóstico y telecontrol**