



# **INTRODUCCION AL ESTUDIO DE AUTOMATISMOS**

**CURSO – INSTRUMENTACION INDUSTRIAL**

## **FASES DE ESTUDIO EN LA ELABORACIÓN DE UN AUTOMATISMO.**

### **Datos con los que se debe contar:**

- 1- Especificaciones técnicas del sistema o proceso a automatizar y su correcta interpretación.
- 2- La parte económica asignada para no caer en el error de elaborar una buena opción desde el punto de vista técnico, pero inviable económicamente.
- 3- Los materiales, aparatos, etc., existentes en el mercado que se van a utilizar para diseñar el automatismo. En este apartado es importante conocer también:
  - Calidad de la información técnica de los equipos.
  - Disponibilidad y rapidez en cuanto a recambios y asistencia técnica.

### **Fases.**

#### **a- Estudio previo:**

Conocimiento del mayor detalle posible las características, el funcionamiento, las distintas funciones (elaborar la mayor cantidad de planos posibles del proceso), etc.

#### **b- Estudio técnico económico:**

Es la parte técnica de especificaciones del automatismo: relación de materiales, bastidores, su adaptación al sistema y al entorno en el que se haya inscrito, etc.

#### **c- Decisión final:**

En el apartado anterior se han debido estudiar las dos posibilidades u opciones tecnológicas generales posibles: lógica cableada y lógica programada.



Los parámetros que se deben valorar para una decisión correcta pueden ser muchos y variados, p.e.:

- Ventajas e inconvenientes que se le asignan a cada opción en relación a su fiabilidad, vida media y mantenimiento.
- Posibilidad de ampliación y de aprovechamiento de lo existente en cada caso.
- Posibilidades económicas y rentabilidad de la inversión realizada en cada opción.
- Ahorro desde el punto de vista de necesidades para su manejo y mantenimiento

**Una vez realizado éste análisis sólo queda adoptar la solución final elegida.**

**OPCIONES TECNOLÓGICAS.**

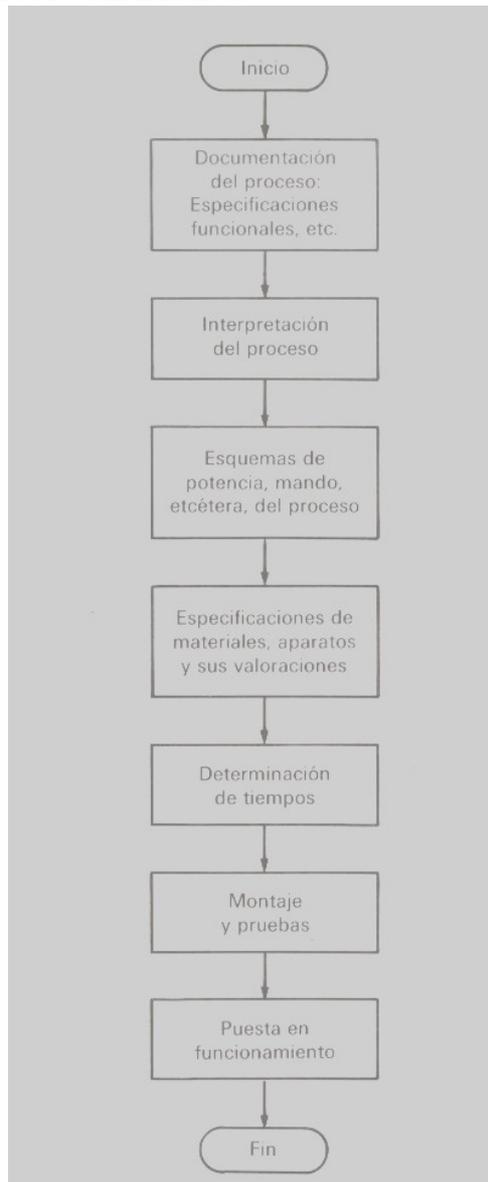
<b>Tipo</b>	<b>Familia Tecnológica</b>	<b>Subfamilias Específicas</b>	
Lógica Cableada	Eléctrica	Relés electromagnéticos	
		Electroneumática	
		Electrohidráulica	
	Electrónica	Electrónica Estática	
Lógica Programada	Electrónica	Sistemas	Microordenadores
		Informáticos	Miniordenadores
		Microsistemas (universales específicos)	
		Autómatas Programables	



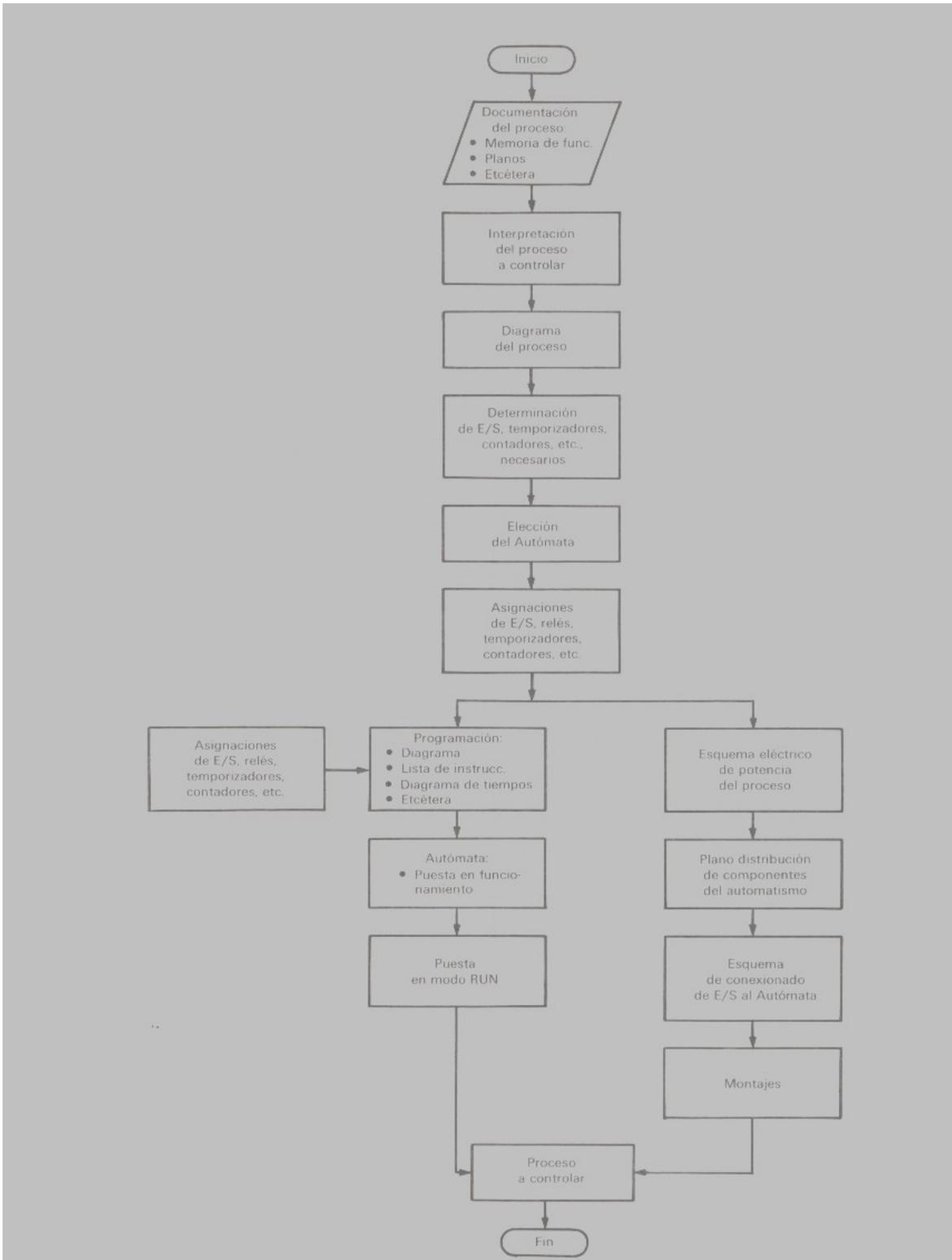
## ORGANIGRAMAS PARA DESARROLLAR EL CONTROL DE UN PROCESO.

Según sea la opción tecnológica elegida:

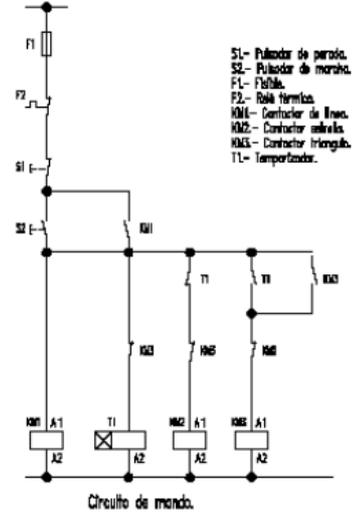
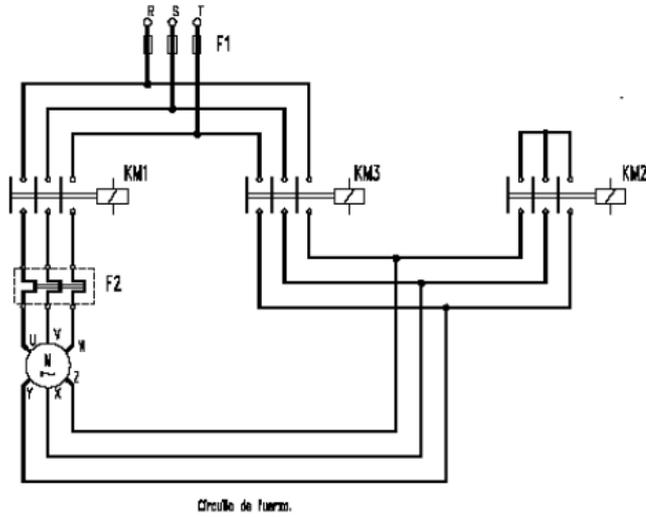
### PROCESO CON LOGICA CABLEADA:



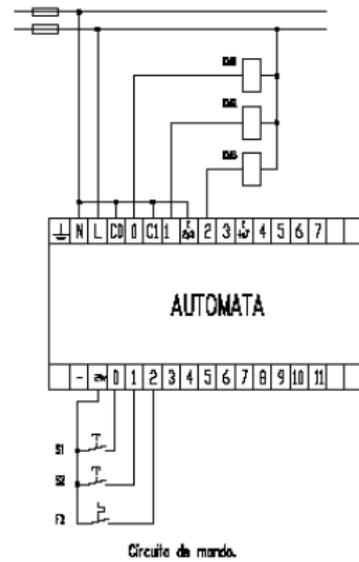
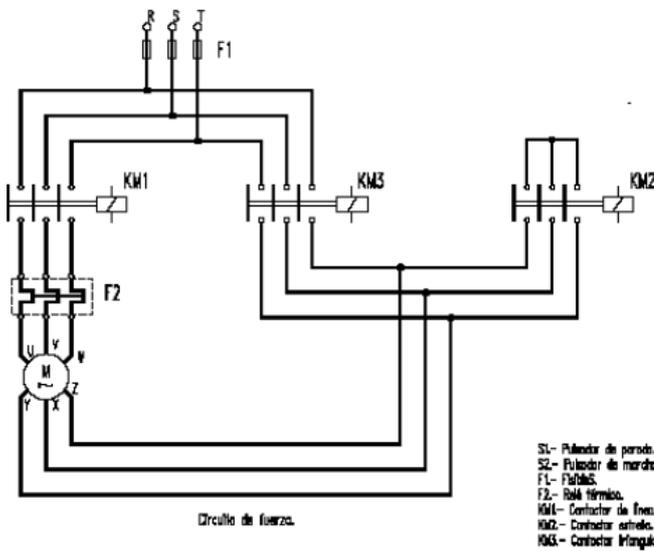
**PROCESO CON LOGICA PROGRAMABLE:**



## EJEMPLOS

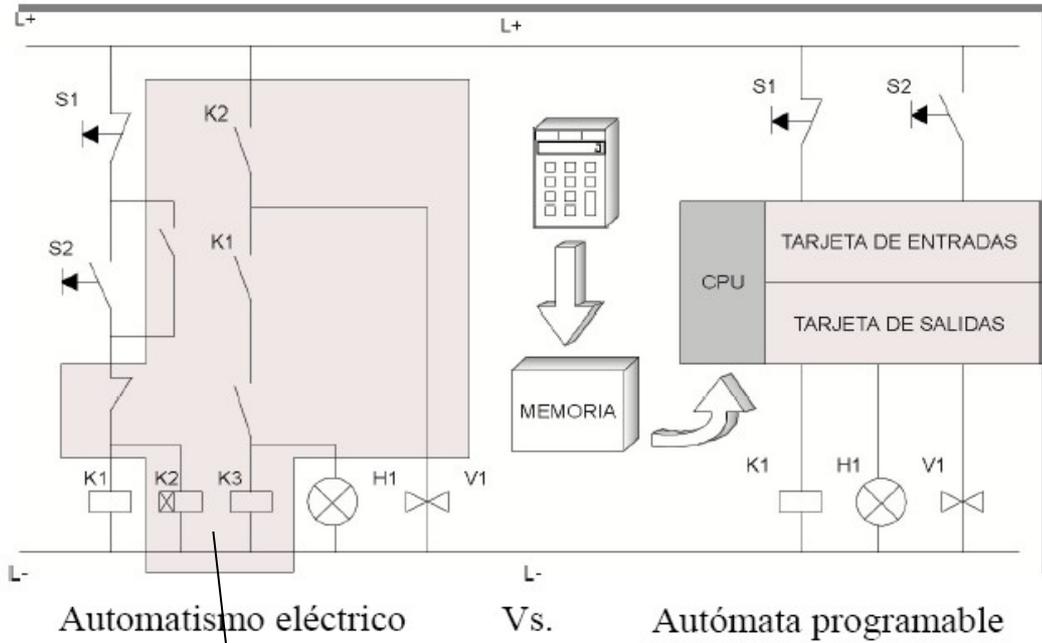


**Arrancador estrella-triángulo con temporizador -TECNICA CABLEADA**



**Arrancador estrella-triángulo con temporizador -TECNICA PROGRAMADA**

## CONCLUSION



Relés	
Temporizadores	
Botoneras	



# **AUTOMATAS**

# **CONTROLADORES LOGICOS PROGRAMABLES**

# **PLC's**

## Definición de Controlador Lógico Programable.

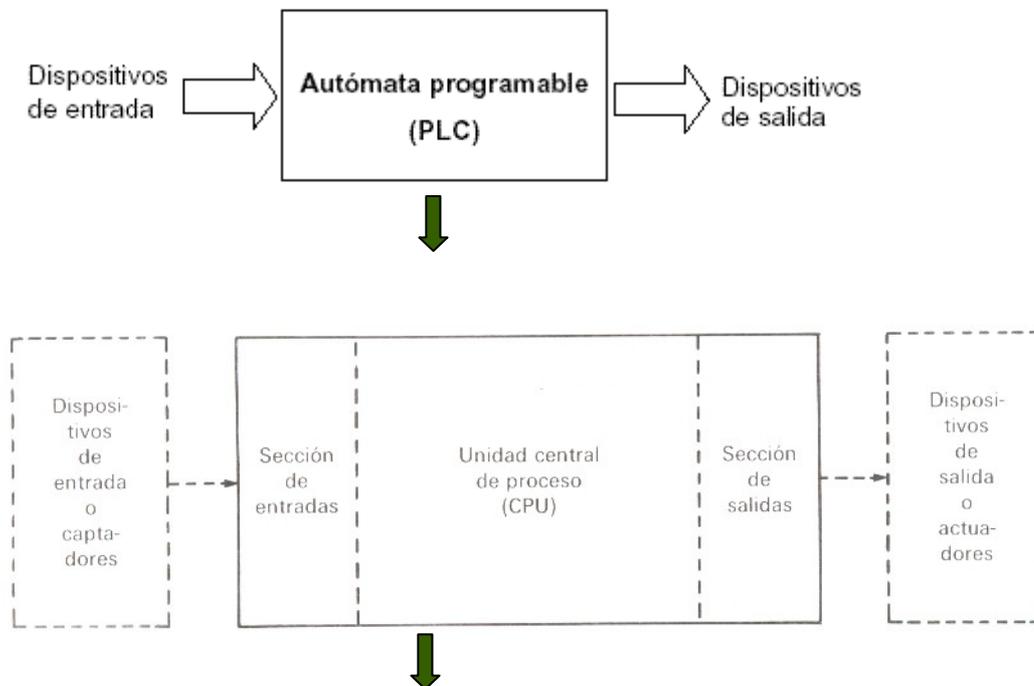
### 1- Controlador Lógico Programable:

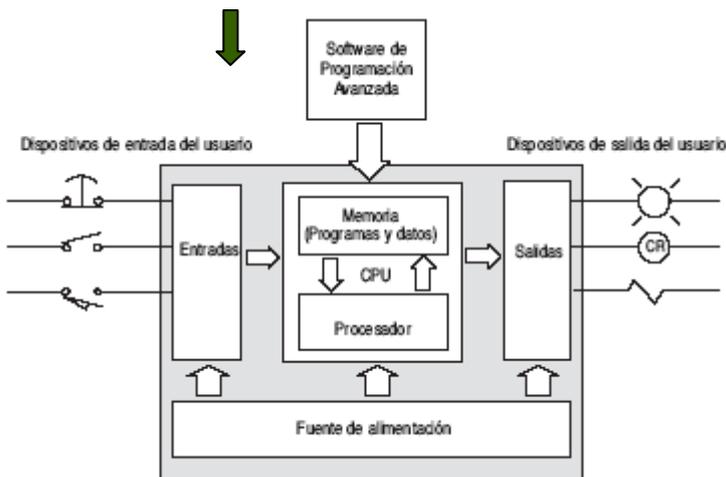
*Se entiende por controlador lógico programable (PLC), o autómatas programables, a toda máquina electrónica diseñada para controlar en tiempo real y en medio industrial procesos secuenciales.*

Esta definición se está quedando un poco desfasada, ya que han aparecido los micro-plc's, destinados a pequeñas necesidades y al alcance de cualquier persona.

### 2- Controlador Lógico Programable.

De acuerdo con la definición de la "NEMA" (National Electrical Manufacturers Association) un controlador programable es: *"Un aparato electrónico operado digitalmente, que usa una memoria programable para el almacenamiento interno de instrucciones para implementar funciones específicas, tales como lógica, secuenciación, registro y control de tiempos, conteo y operaciones aritméticas para controlar, a través de módulos de entrada/salida digitales (ON/OFF) o analógicos (1 5 VDC, 4 20 mA, etc.), varios tipos de máquinas o procesos.*

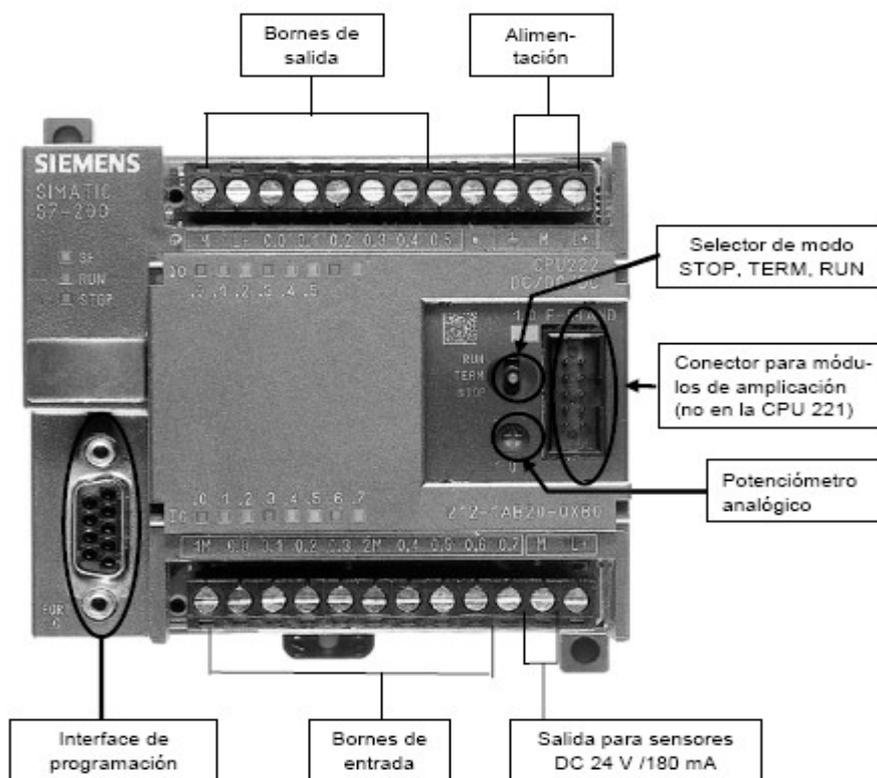




Las entradas pueden recibir señales de tipo digital, por ejemplo: interruptores, o de tipo analógico, como son los sensores de temperatura.

Estas señales son transformadas internamente en señales compatibles con los microprocesadores y demás circuitos integrados de procesamiento interno.

Su funcionamiento depende de un programa que se le ha introducido previamente, con el cual se ejecuta cierta secuencia en forma completamente automática.



## Campos de aplicación.

Un autómata programable suele emplearse en procesos industriales que tengan una o varias de las siguientes necesidades:

- Espacio reducido.
- Procesos de producción periódicamente cambiantes.
- Procesos secuenciales.
- Maquinaria de procesos variables.
- Instalaciones de procesos complejos y amplios.
- Chequeo de programación centralizada de las partes del proceso.

Aplicaciones generales:

- Maniobra de máquinas.
- Maniobra de instalaciones.
- Señalización y control.

Tal y como dijimos anteriormente, esto se refiere a los autómatas programables industriales, dejando de lado los pequeños autómatas para uso más personal (que se pueden emplear, incluso, para automatizar procesos en el hogar, como la puerta de un cochera o las luces de la casa).

## Ventajas e inconvenientes de los PLC's.

Entre la **ventajas** tenemos:

- Menor tiempo de elaboración de proyectos.
- Posibilidad de añadir modificaciones sin costo añadido en otros componentes.
- Mínimo espacio de ocupación.
- Menor costo de mano de obra.
- Mantenimiento económico.
- Posibilidad de gobernar varias máquinas con el mismo autómata.
- Menor tiempo de puesta en funcionamiento.
- Si el autómata queda pequeño para el proceso industrial puede seguir siendo de utilidad en otras máquinas o sistemas de producción.

Y entre los **inconvenientes**:

- Adiestramiento de técnicos.
- Costo \*.

\* Obs.: El punto de equilibrio a partir del cual su precio es comparable, o incluso inferior a los tradicionales de lógica cableada disminuye constantemente.

## Estructura externa.

Todos los autómatas programables, poseen una de las siguientes estructuras:

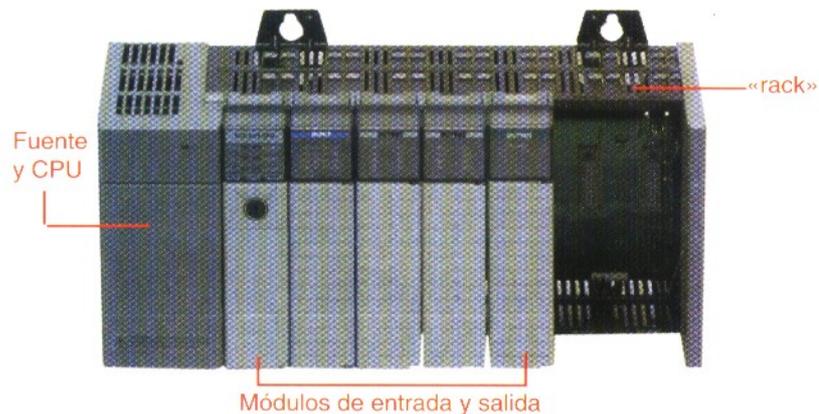
- **Compacta o integrada:** en un solo bloque están todos los elementos.



- **Modular:**

- **Estructura americana:** separa las E/S del resto del autómata.
- **Estructura europea:** cada módulo es una función (fuente de alimentación, CPU, E/S, etc.).

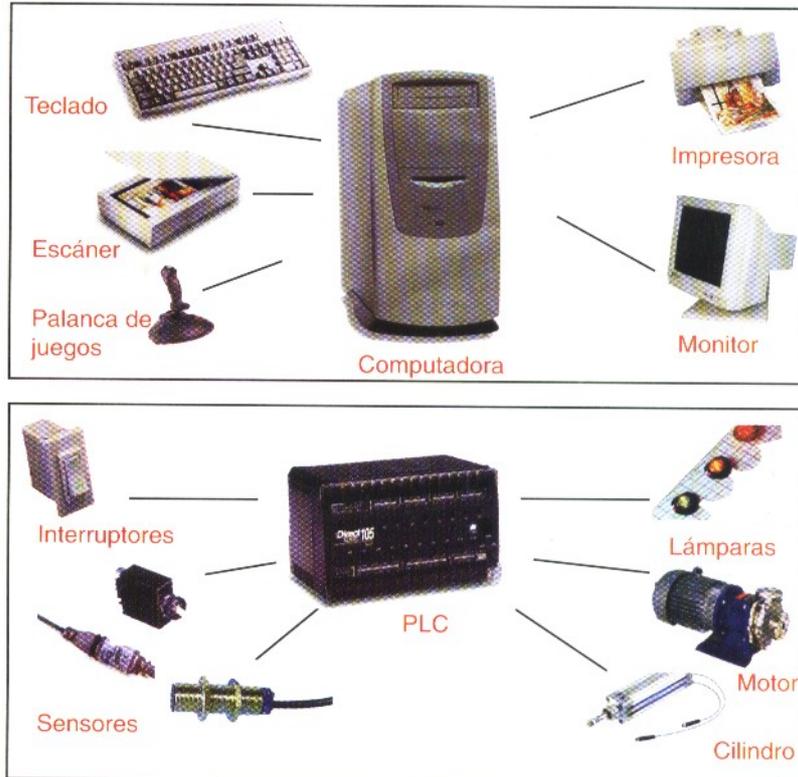
Exteriormente nos encontraremos con cajas que contienen una de estas estructuras, las cuales poseen indicadores y conectores en función del modelo y fabricante.



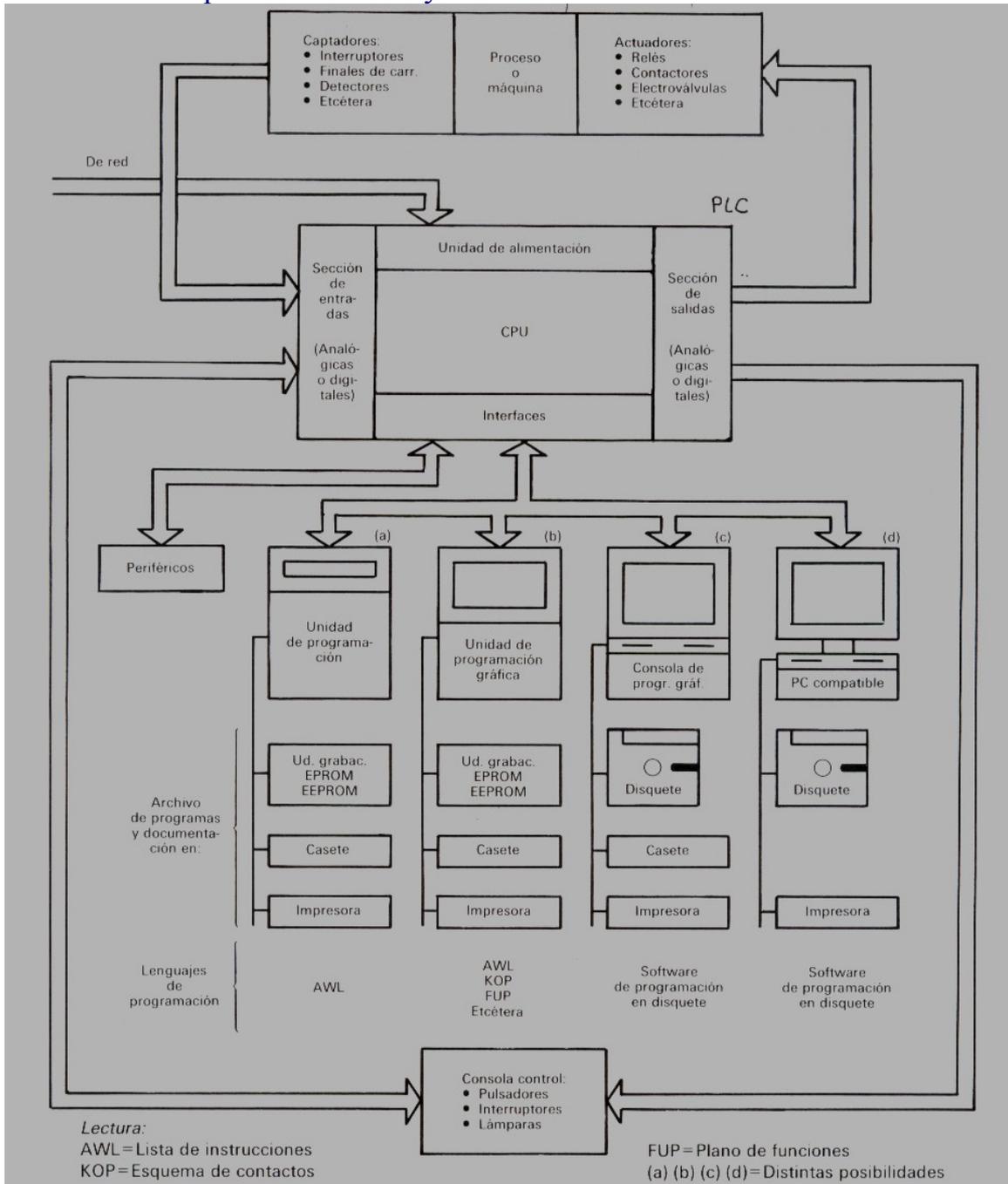
Para el caso de una estructura modular se dispone de la posibilidad de fijar los distintos módulos en railes normalizados, para que el conjunto sea compacto y resistente.

Los micro-autómatas suelen venir sin caja, en formato kit, ya que su empleo no es determinado y se suele incluir dentro de un conjunto más grande de control o dentro de la misma maquinaria que se debe controlar.

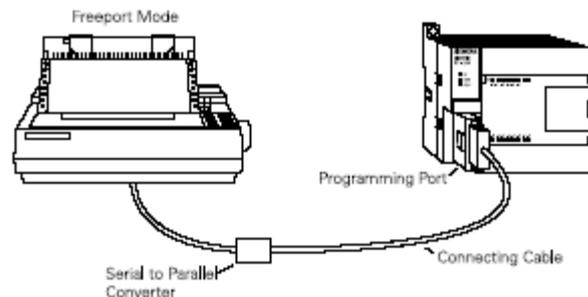
### Analogía entre un PLC y una computadora.



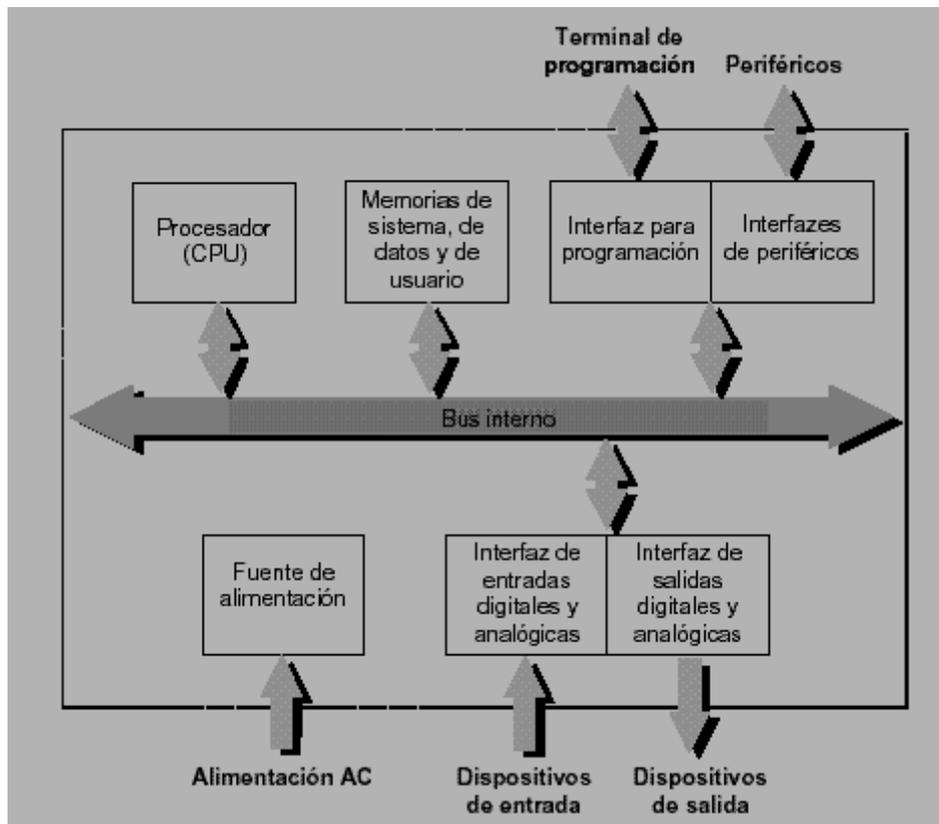
Estructura completa de un PLC y su entorno:



**Obs: un PLC, puede tener como una salida también una impresora.**



## Estructura interna.



Los elementos esenciales, que todo autómatas programable posee como mínimo, son:

**Sección de entradas:** se trata de líneas de entrada, las cuales pueden ser de tipo digital o analógico. En ambos casos tenemos unos rangos de tensión característicos, los cuales se encuentran en las hojas de características del fabricante. A estas líneas conectaremos los sensores.

**Sección de salidas:** son una serie de líneas de salida, que también pueden ser de carácter digital o analógico. A estas líneas conectaremos los actuadores.

**Unidad central de proceso (CPU):** se encarga de procesar el programa de usuario que le introduciremos.

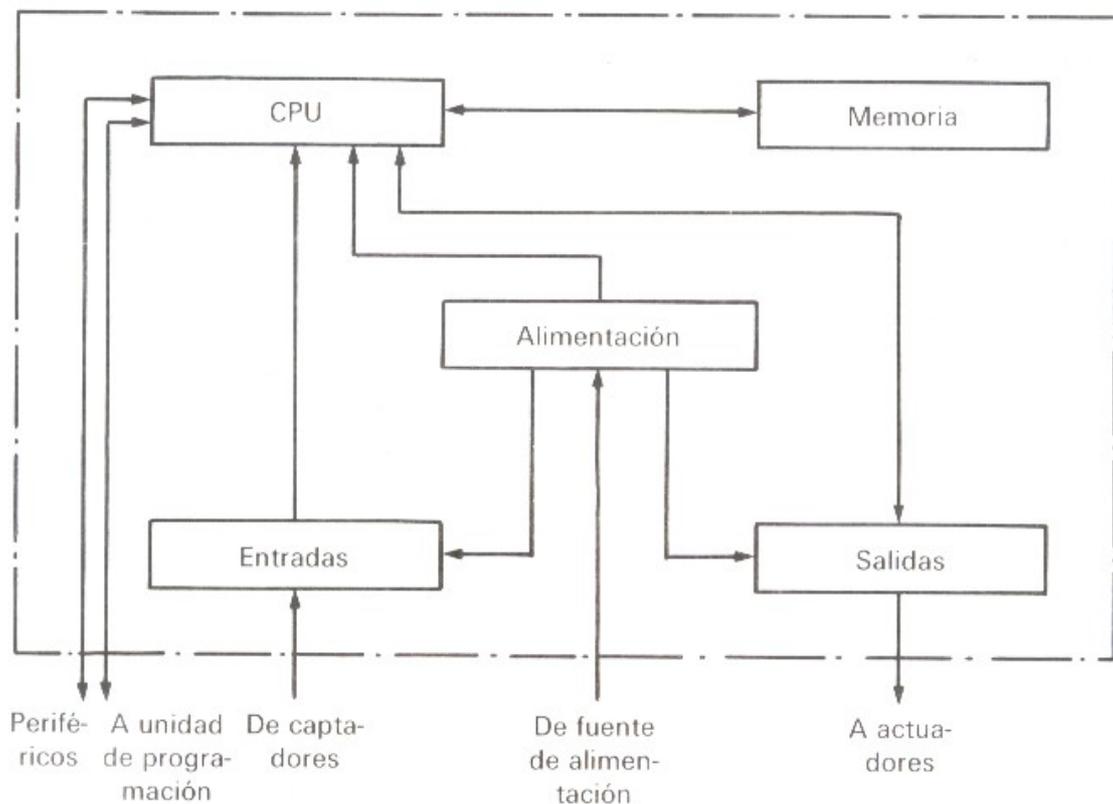
Adicionalmente, en determinados modelos más avanzados, podemos disponer de funciones ya integradas en la CPU; como reguladores PID, control de posición, etc.

**Tanto las entradas como las salidas están aisladas de la CPU según el tipo de autómatas que utilizemos. Normalmente se suelen emplear optoacopladores en las entradas y relés/optoacopladores en las salidas.**

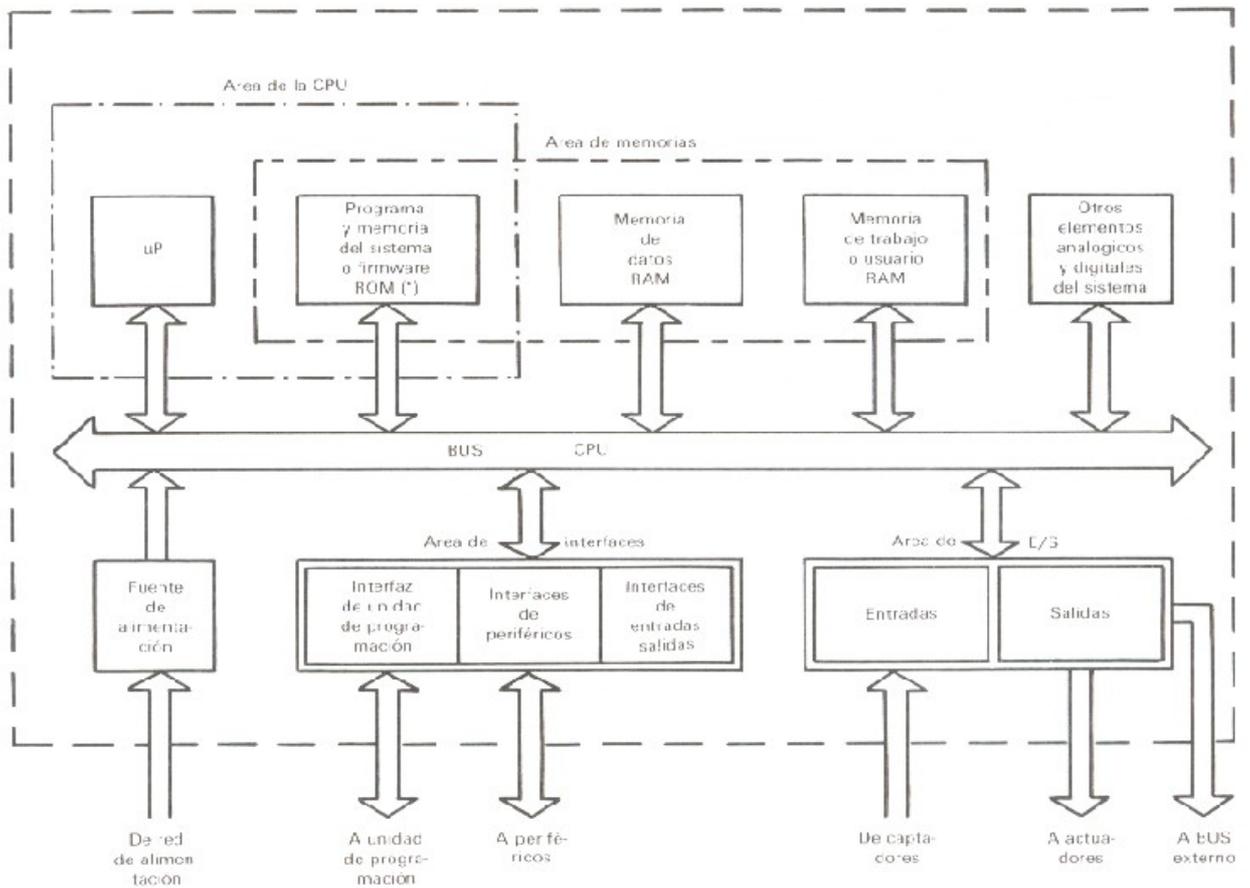
*Aparte de estos elementos podemos disponer de los siguientes:*

- **Unidad de alimentación:** (algunas CPU la llevan incluida).
- **Unidad o consola de programación:** que nos permitirá introducir, modificar y supervisar el programa de usuario.
- **Dispositivos periféricos:** como nuevas unidades de E/S, más memoria, unidades de comunicación en red, etc.
- **Interfaces:** facilitan la comunicación del autómata mediante enlace serie con otros dispositivos (como un PC).

Esquema simplificado de un PLC:



## Arquitectura de un PLC:



## Memorias.

Tipo de memoria	Sistema de Programación	Sistema de borrado	Ante el corte de tensión, la memoria...
RAM o memoria de lectura-escritura	Eléctrica	Eléctrico	Se pierde, es volátil
ROM o memoria de solo lectura	Durante el proceso de fabricación	Es imposible su borrado	Se mantiene
PROM o memoria programable	Eléctrica	Es imposible su borrado	Se mantiene
EPROM o memoria modificable	Eléctrica	Por rayos ultravioletas	Se mantiene
EEPROM o memoria modificable	Eléctrica	Eléctrico	Se mantiene

Dentro de la CPU vamos a disponer de un área de memoria, la cual emplearemos para diversas funciones:

- **Memoria del programa de usuario:** aquí introduciremos el programa que el autómata va a ejecutar cíclicamente (RAM). Respaldo por una **batería tampón** que impide su borrado.

NOTA: En algunos PLC, la memoria RAM se auxilia de una memoria sombra del tipo EEPROM i.e. la combinación de RAM + EEPROM resulta más flexible, pues actúan como memoria de seguridad ante un fallo en el suministro de energía, y una vez reestablecida ésta, el contenido de la EEPROM se vuelca sobre la RAM. Las soluciones de éste tipo están sustituyendo a las clásicas RAM + batería.

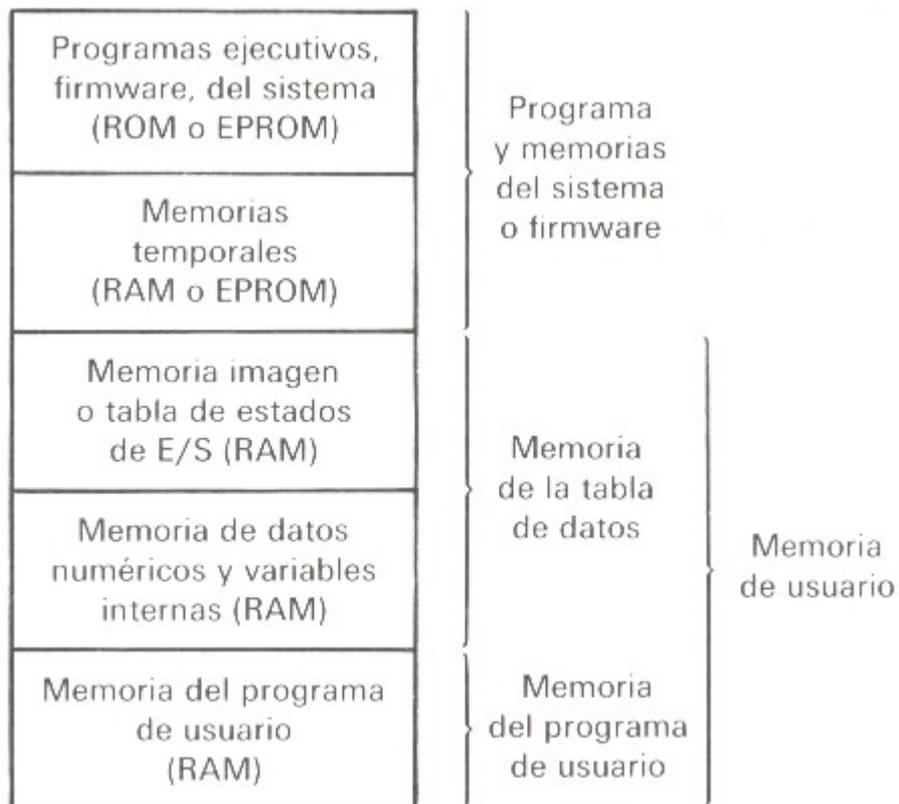
- **Memoria de la tabla de datos:** se suele subdividir en zonas según el tipo de datos (como marcas de memoria, temporizadores, contadores, etc.). Es del tipo RAM.

- **Memoria del sistema:** aquí se encuentra el programa en código máquina que monitoriza el sistema (programa del sistema o firmware). Este programa es ejecutado directamente por el microprocesador/microcontrolador que posee el PLC.

- **Memoria de almacenamiento:** se trata de memoria externa que empleamos para almacenar el programa de usuario, y en ciertos casos parte de la memoria de la tabla de datos. Suele ser de uno de los siguientes tipos: EPROM, EEPROM, o FLASH.

Cada autómata divide su memoria de esta forma genérica, haciendo subdivisiones específicas según el modelo y fabricante.

### Resumen de memorias de un PLC:



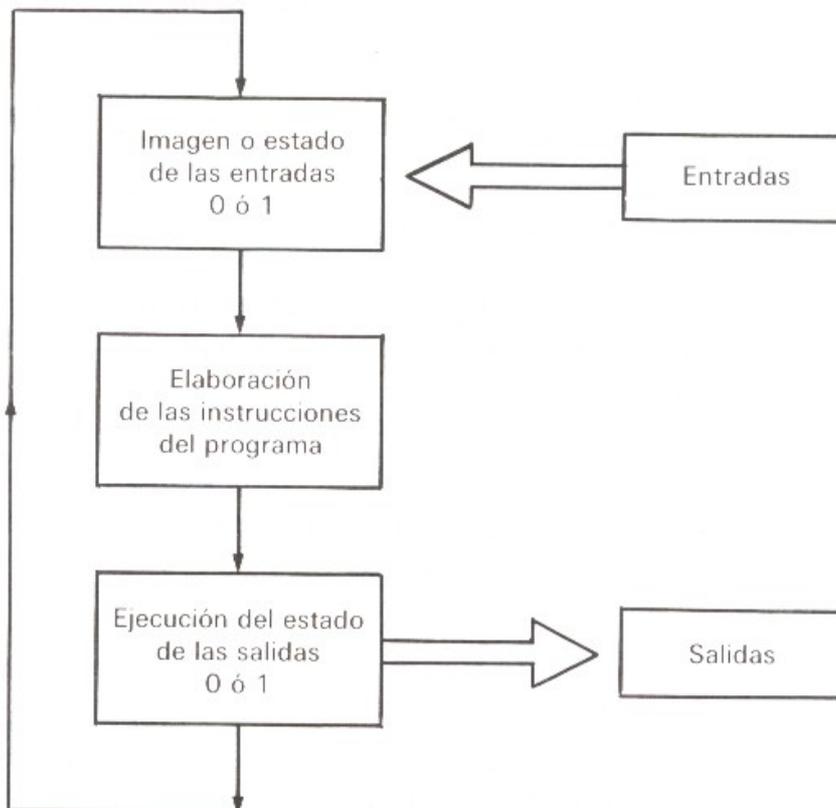
## CPU.

La CPU es el corazón del autómata programable (procesador, memoria y circuitos auxiliares). Es la encargada de ejecutar el programa de usuario mediante el programa del sistema.

Sus funciones son:

- **Vigilar** que el tiempo de ejecución del programa de usuario no excede un determinado tiempo máximo (tiempo de ciclo máximo). A esta función se le suele denominar Watchdog (perro guardián).
- **Ejecutar** el programa de usuario.
- **Crear** una imagen de las entradas, ya que el programa de usuario no debe acceder directamente a dichas entradas.
- **Renovar** el estado de las salidas en función de la imagen de las mismas obtenida al final del ciclo de ejecución del programa de usuario.
- **Chequeo** del sistema.

Para ello el autómata va a poseer un ciclo de trabajo, que ejecutará de forma continua:



## Unidades de E/S.

Son los dispositivos básicos por dónde se toma la información de los captadores, en el caso de las entradas, y por donde se realiza la activación de los actuadores, en las salidas.

Sus funciones principales son adaptar las tensiones e intensidades de trabajo de los captadores a las de trabajo de los circuitos electrónicos del PLC; realizar una separación eléctrica entre los circuitos lógicos de los de potencia, generalmente a través de optoacopladores, y proporcionar el medio de identificación de los captadores y actuadores ante el procesador.



Generalmente vamos a disponer de dos tipos de E/S: **analógicas** y **digitales**.

### → Entradas:

#### - Analógicas.

Su principio de funcionamiento se basa en la conversión de la señal analógica a código binario mediante un convertidor analógico/digital (A/D).

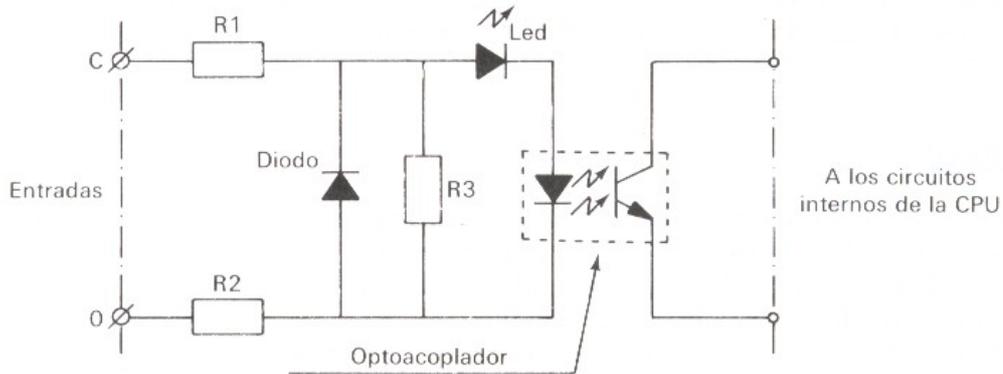
#### Parámetros más significativos de éste tipo de módulos:

Campo o rango de intensidad o tensión	Resolución	Tiempo de conversión	Precisión
0 - 10 V	8 bits	1 ms	+/- (1% + 1 bit) en entradas +/- 1% en salidas
4 - 20 mA	8 bits	1 ms	
0 - 10 V	12 bits	1 ms	
4 -20 mA	12 bits	1 ms	

- Digitales.

Son las más utilizadas y corresponden a una señal de entrada a un nivel de tensión o a la ausencia de la misma.

En el siguiente esquema simplificado, se muestra el circuito de entrada por transistor del tipo NPN, y en el que se destaca como elemento principal, el **optoacoplador**.



→ **Salidas:**

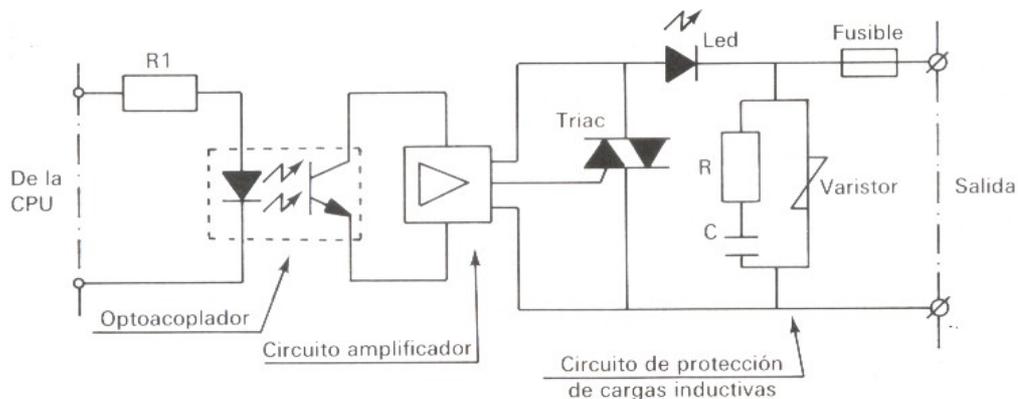
En las salidas donde se conectan o acoplan los dispositivos de salida o actuadores, e incluyen un indicador luminoso LED de activado.

Tres son los tipos de salidas que se pueden dar:

- i. A relé.**
- ii. A triac.**
- iii. A transistor.**

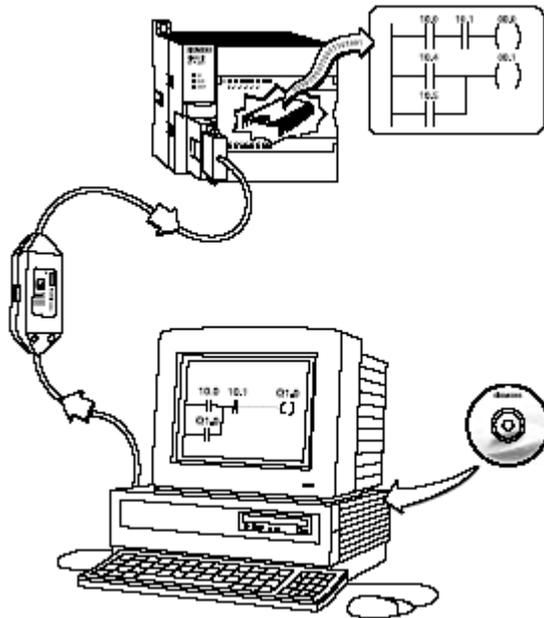
Mientras que la salida a transistor se utiliza cuando los actuadores son a DC, las de relés y triacs suelen utilizarse para actuadores s CA.

Por ejemplo: circuito de salida a triac:



## Interfaces.

Todo autómeta, salvo casos excepcionales, posee la virtud de poder comunicarse con otros dispositivos (como un PC).



Lo normal es que posea una E/S serie del tipo RS-232 / RS-422, y efectúan la comunicación mediante el código ASCII.

Asimismo, permiten la introducción, verificación y depuración del programa mediante la consola de programación, así como la grabación del programa a diskete, en memoria EPROM, comunicación con un monitor, impresora, etc.

## Batería.

Mantiene energizada la memoria RAM que almacena el programa mientras el PLC permanece des-energizado. El PLC, con batería, generalmente puede mantener el programa durante alrededor de tres meses.

La vida útil de la batería, varía de 1 a 3 años. En aquellos PLCs que contienen memoria no volátil (EEPROM o de cualquier otro tipo), la batería no es indispensable.

## Equipos o unidades de programación.

### Funciones principales:

#### 1-Programación

- i. Introducción de programas.
- ii. Búsqueda de instrucciones o posiciones de memoria.
- iii. Modificación del programa:
  - borrado de instrucciones.
  - inserción de instrucciones.
  - modificación de instrucciones.
- iv. Detección de errores de sintaxis o formato.
- v. Visualización del programa de usuario o parte del mismo, contenido en la memoria de usuario.
- vi. Forzamiento del estado de marcas, registros, contadores, temporizadores, etc.

#### 2- Grabación de programas.

- i. En disquete mediante impresora.
- ii. En chip de memoria EPROM o EEPROM.
- iii. En papel mediante impresora.

#### 3- Visualización y verificación dinámica del programa.

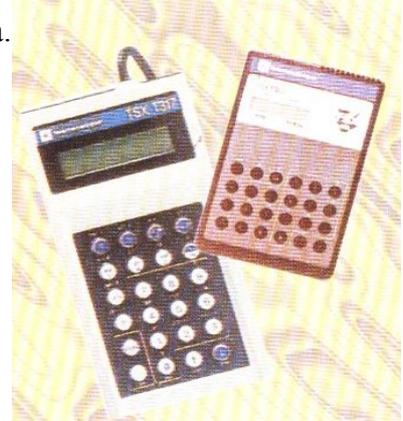
- i. Del programa o parte de él.
- ii. De entradas y salidas.
- iii. De temporizadores, contadores, registros, etc.

#### 4- Modos de servicio.

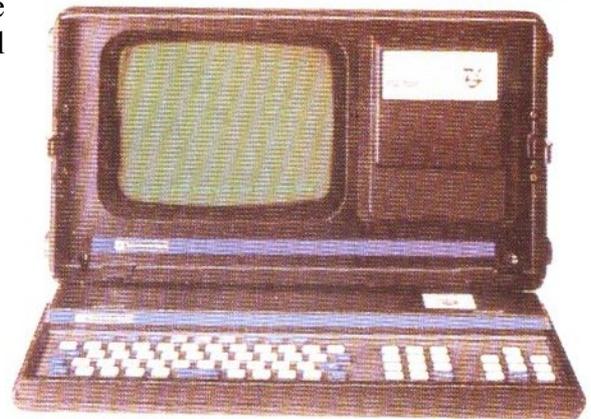
- i. **STOP** (off-line), o salidas en reposo.
- ii. **RUN** (on-line), o ejecutando programa.
- iii. Otros modos intermedios como **Monitorización**, etc.

El autómatas debe disponer de alguna forma de programación, la cual se suele realizar empleando alguno de los siguientes elementos:

**Unidad de programación:** suele ser en forma de calculadora. Usualmente se la conoce como **Hand-held**



**Consola de programación:** es un terminal a modo de ordenador que proporciona una forma más cómoda de realizar el programa de usuario y observar parámetros internos del autómatas. Desfasado actualmente.



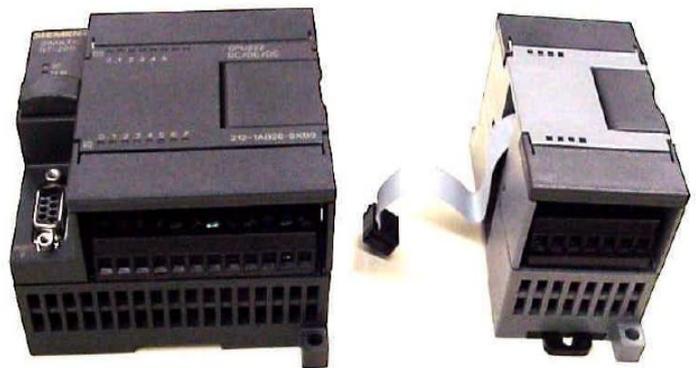
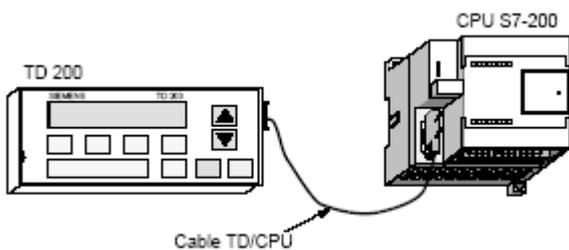
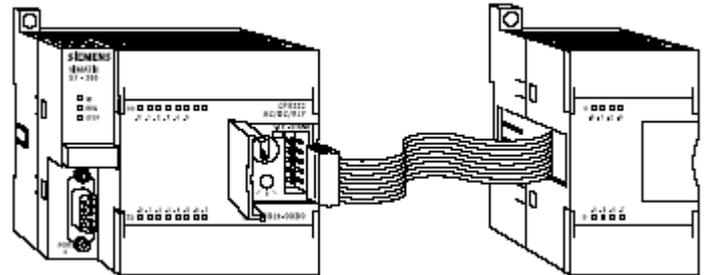
**PC:** es el modo más potente y empleado en la actualidad. Permite programar desde un ordenador personal estándar, con todo lo que ello supone: herramientas más potentes, posibilidad de almacenamiento en soporte magnético, impresión, transferencia de datos, monitorización mediante software SCADA, etc.



## Dispositivos periféricos.

El autómata programable, en la mayoría de los casos, puede ser ampliable. Las ampliaciones abarcan un gran abanico de posibilidades, que van desde las redes internas (LAN, etc.), módulos auxiliares de E/S, memoria adicional... hasta la conexión con otros autómatas del mismo modelo.

Cada fabricante facilita las posibilidades de ampliación de sus modelos, los cuales pueden variar incluso entre modelos de la misma serie.



### NOTA: Configuración del sistema.

El software de interfaz entre la computadora personal y el PLC puede ser configurado fácilmente por el usuario para seleccionar la velocidad de comunicación, el puerto serial a utilizar, el puerto paralelo a utilizar por la impresora, los colores de la pantalla, y opciones de video rápido, entre una gran diversidad de opciones que difieren entre fabricantes de PLC que son los desarrolladores del software. Actualmente, se presentan en ambiente Windows, y con opciones de tipo común a todos los programas, permiten un ambiente de trabajo amigable.

## TAMAÑO DE LOS CONTROLADORES LOGICO PROGRAMABLES.

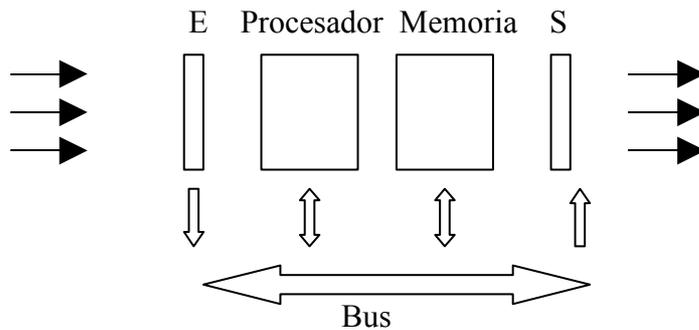
La clasificación de los PLCs se realiza en función del número de sus entradas y salidas; son admitidos los tres grupos siguientes:

- **GAMA BAJA**: Hasta un máximo de 128 entradas/salidas.  
La memoria de usuario de que disponen suele alcanzar un valor máximo de 4K de instrucciones.
- **GAMA MEDIA**: De 128 a 512 entradas/salidas.  
La memoria de usuario de que disponen suele alcanzar un valor máximo de hasta 16K de instrucciones.
- **GAMA ALTA**: Más de 512 entradas/salidas.  
Su memoria de usuario supera en algunos de ellos los 100 K de instrucciones.

## PROGRAMACION DEL CONTROLADOR – PLC.

### El ciclo:

- Lectura de las entradas.
- Tratamiento del programa.
- Escritura de las salidas.



El PLC se descompone en 4 subconjuntos principales:

- ✓ 1- Interfaces de entrada.
- ✓ 2- Procesador (lee las entradas y en función de ellas y de las instrucciones del programa, *escribe* las salidas).
- ✓ 3- Memoria.
- ✓ 4- Interfaces de salida.

Los intercambios entre la unidad central y las interfaces de E/S se realizan de manera cíclica (algunas decenas de mseg. por ciclo).

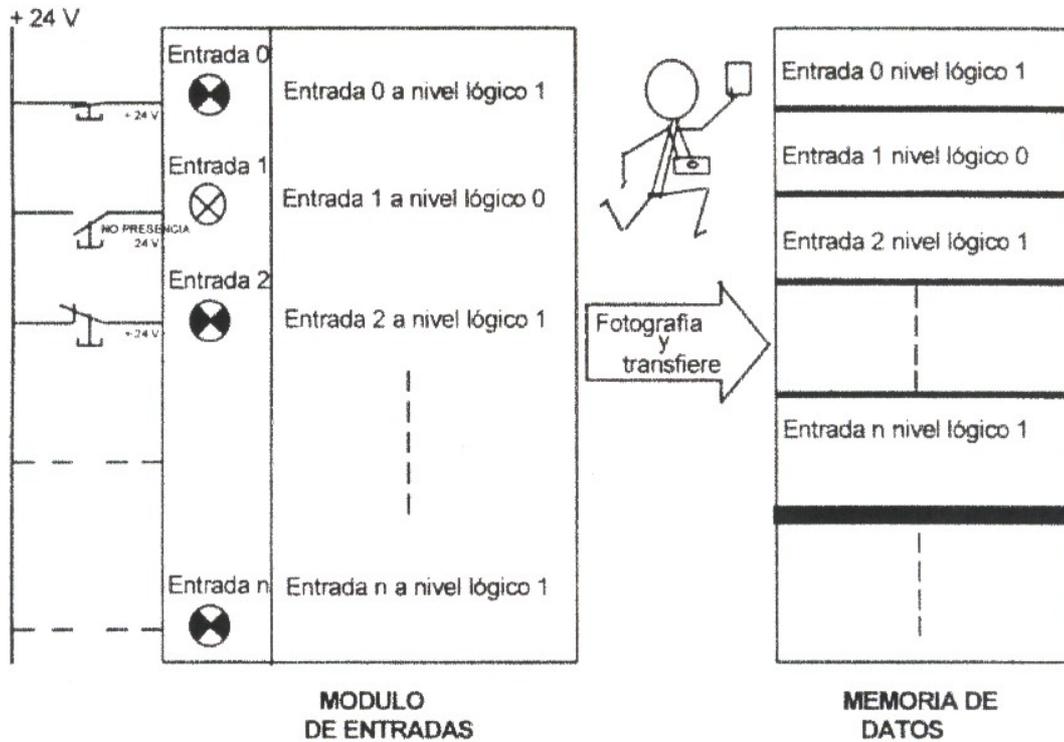
El ciclo de funcionamiento comprende 3 fases sucesivas.

**FASE 1:** Adquisición del estado de las entradas (y memorización de las mismas en la memoria de datos).

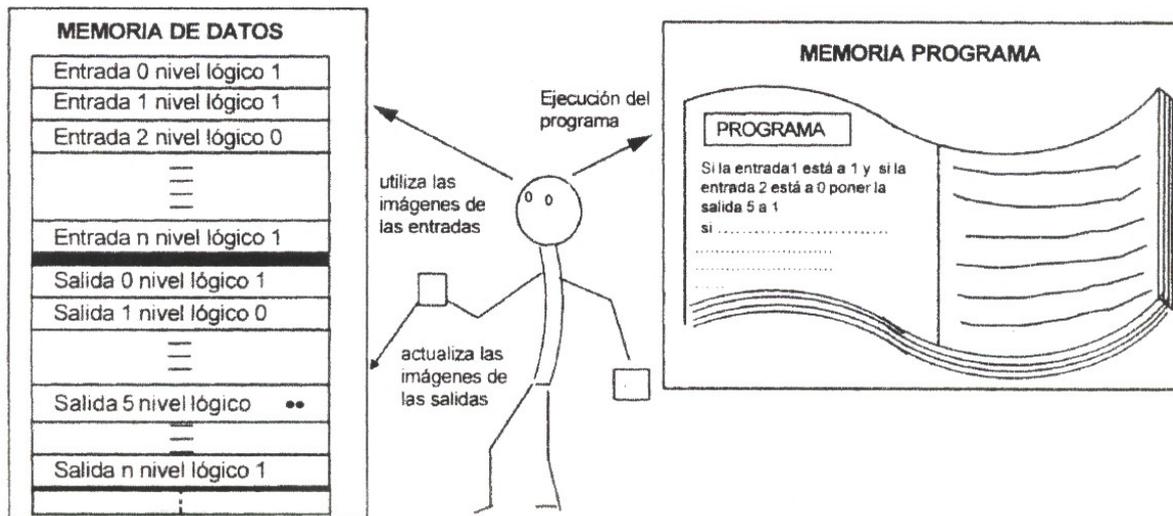
**FASE 2:** Tratamiento del programa (y actualización de las imágenes de las salidas en la memoria de datos).

**FASE 3:** Actualización de las salidas (las imágenes de las salidas se transfieren a las interfaces de salida).

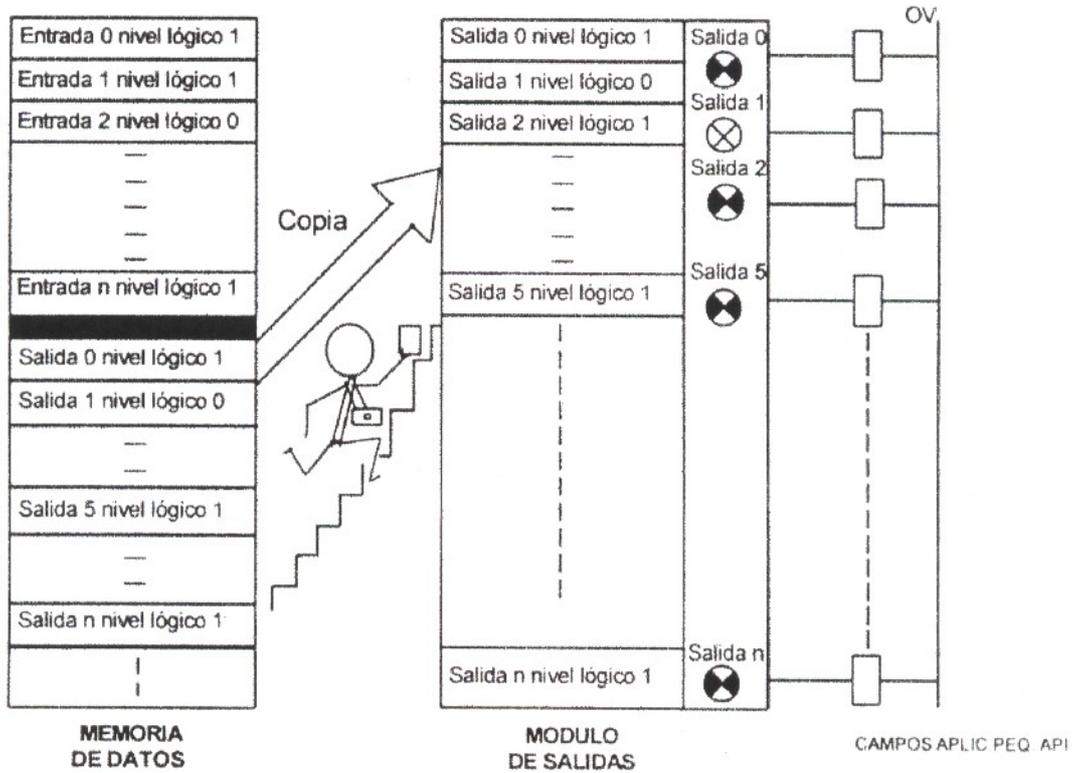
**FASE 1:** el procesador “fotografía”, el estado lógico de las entradas y después transfiere la imagen obtenida en la memoria de datos.



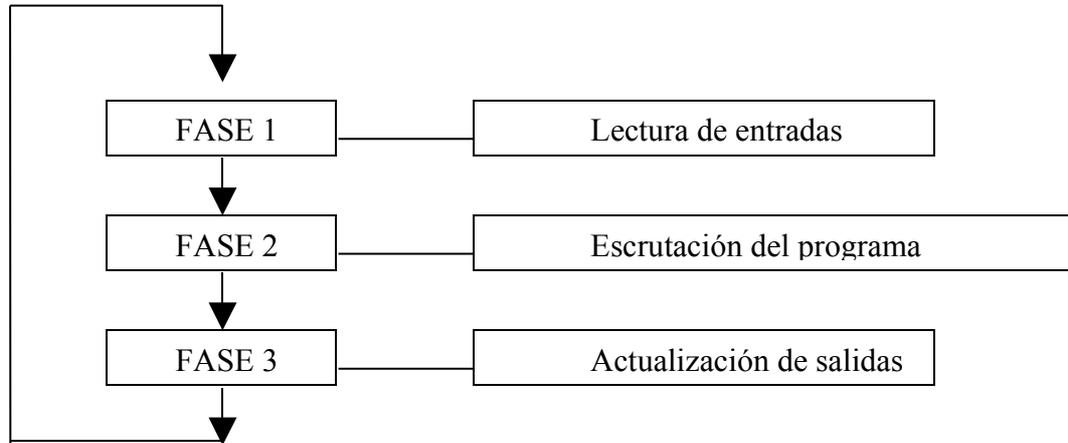
**FASE 2:** Ejecución de operaciones lógicas contenidas en la memoria de programa, una tras otra hasta la última. Para ello utiliza la imagen del estado de las entradas contenida en la memoria de datos, y actualiza el resultado de cada operación lógica en la memoria de datos (imágenes de las salidas).



**FASE 3:** copia sobre los módulos de salida, el conjunto de las imágenes (estados lógicos de las salidas) contenidos en la memoria de datos.

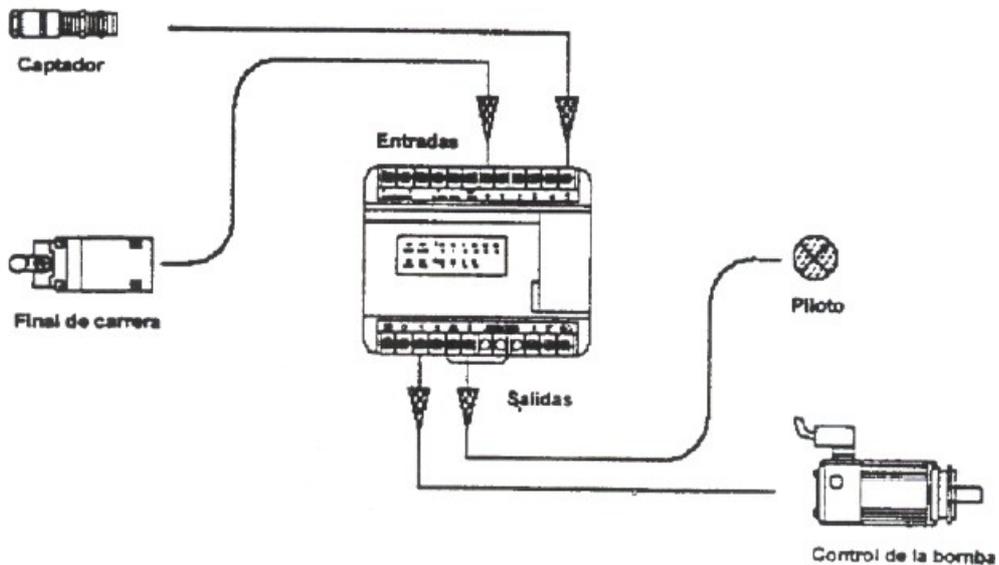


## TIEMPO DE CICLO DEL PLC.



El tiempo de ciclo (o de escrutación) es el tiempo transcurrido entre 2 escruciones de una entrada física (eléctrica).

La máxima duración del programa de usuario está controlada por el autómata (watchdog del programa) y no debe rebasar los 150 ms.



- ⇒ **EL CICLO**
- 1 Lectura de las Entradas
  - 2 Tratamiento del programa
  - 3 Escritura de las Salidas

## El programa.

**Nota:** El tema que a continuación se analizará, es muy específico hacia el PLC particular que se quiera programar. Es por eso que fue de nuestro mayor esfuerzo, generalizar lo mayor posible la información que sigue.

**Definición:** Un programa es una sucesión o lista en un determinado orden de distintas órdenes de trabajo también llamadas instrucciones y capaz de hacer ejecutar al PLC la secuencia de trabajo pretendido.

### Instrucción u orden de trabajo.

Es la parte más pequeña de un programa y consta de dos partes:

- operación.
- operando.

Instrucción	
Operación ¿qué?	Operando ¿dónde?
	Símbolo      Parámetro

La **operación**, es el código de la instrucción. Puede venir como código numérico (08) o código nemónico (AND).

El **operando** es el complemento al código u operación. Mediante el operando indicamos la dirección del elemento de que se trate (contadores, temporizadores, E/S, etc.), así como las contadas, temporizaciones, bits de registro de desplazamiento, etc.

En los PLC más sencillos, el símbolo no aparece, ya que al ser fijas las entradas/salidas y tener asignados números distintos unas y otras, al indicarle a la CPU el número, ya queda determinada, en el caso de las entradas la distinción con las marcas internas y las salidas con el código de la operación.

La operación le indica a la CPU  $\longrightarrow$  **qué tiene que hacer** (clase de instrucción que ha de ejecutar)

Por ej.:

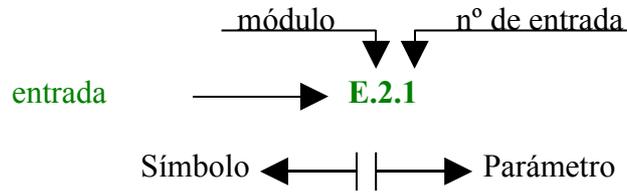
AND (Y): formar una concatenación serie.

OR (O): formar una concatenación paralelo.

OUT(=): asignar una salida a lo precedente.

El operando le indica a la CPU  $\longrightarrow$  **dónde debe de hacerlo** (dónde debe realizarse la instrucción).

Por ej.:



Cuando se programa, cada instrucción del programa se aloja en una celda o plaza de memoria que están numeradas desde la dirección 0000 hasta el último número, en función de la capacidad de memoria; en el caso de una memoria de usuario de 1 K palabras, las direcciones disponibles serían de la 0000 a la 1023.

*Nota:* se ha supuesto que cada instrucción ocupa una palabra, que en general, es de 16 bits o 2 bytes, si la instrucción ocupa más de 2 bytes como ocurre en algunos casos, el número de direcciones disponibles se reduce.

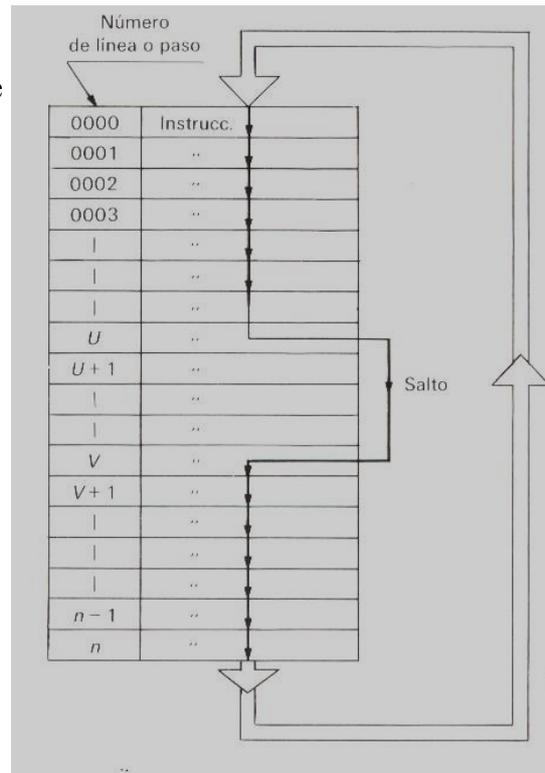
Otro concepto a tener en cuenta es la ***línea o línea de programa***. Una línea contiene dirección o paso, operación y operando, por tanto, se puede decir que una línea de programa consta de una instrucción, salvo algunos casos en el que son necesarias dos líneas para alojar una sola instrucción.

Para poder elaborar un programa no es suficiente con las instrucciones de mando o de programa, son necesarias otro tipo de instrucciones que reciben el nombre de ***instrucciones de servicio u órdenes de manejo*** y por medio de las cuales se consigue la elaboración, análisis y puesta a punto del programa, así como otras posibilidades.



### Salto condicional.

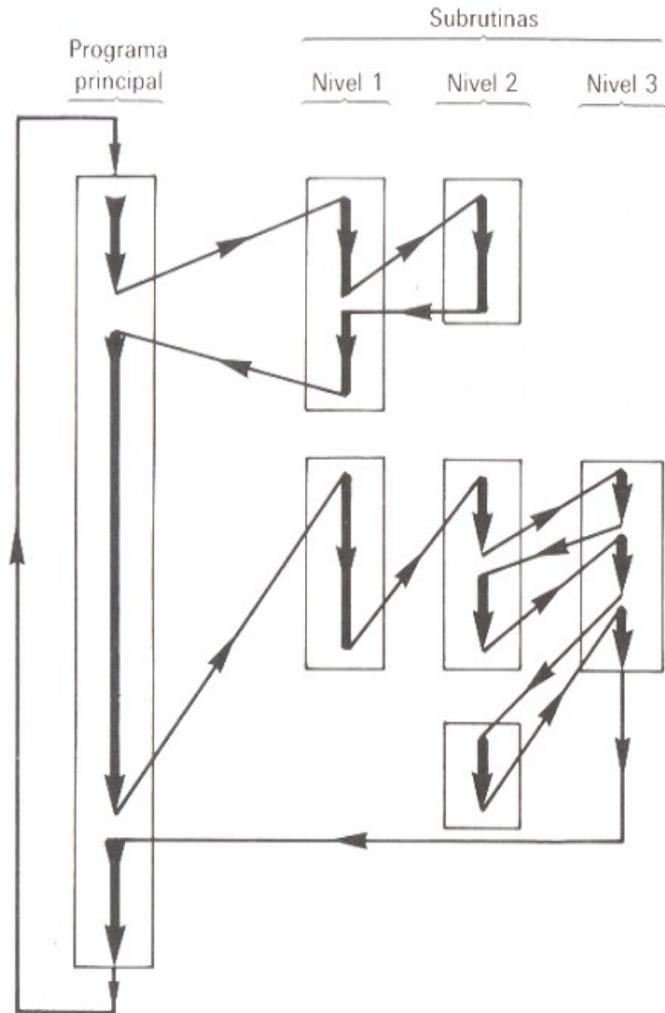
Cuando el ciclo de ejecución de un programa tiene la posibilidad, previa establecida, de alterar la secuencia línea a línea del mismo y dar un salto a otras línea de programa, dejando X líneas sin ejecutar.



### Salto o subrutinas.

En algunas ocasiones ocurre que en un programa hay uno o más grupos de secuencias de instrucciones idénticas que se repiten y que habrá que reescribir tantas veces como éstas se repitan en dicho programa principal.

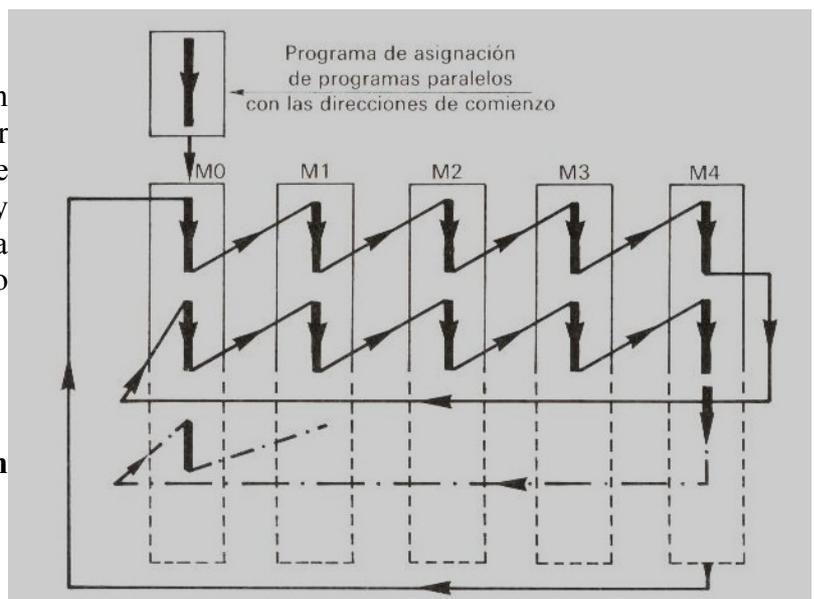
En estos caso, es muy útil escribir una sola vez ésta secuencia o subrutina, e ir a ella cuando se requiera (por lo gral. Los PLC de la gama bajo, no contienen ésta posibilidad).



### Programas paralelos.

En aquellos casos en que con un único PLC querramos controlar varios procesos totalmente independientes, éste sistema es muy útil, aunque también se utiliza controlando funciones de un proceso único.

**Esta posibilidad no la contienen los PLC de la gama baja.**



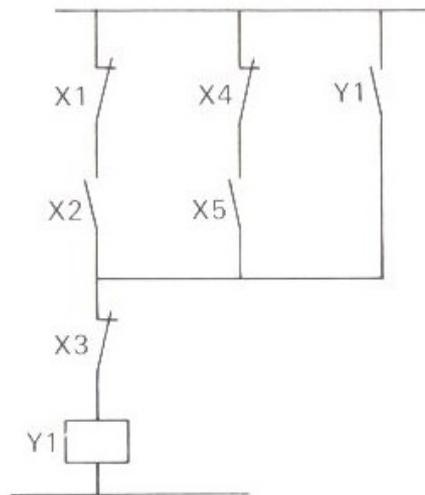
## LENGUAJES DE PROGRAMACIÓN.

En gral. se podría decir que los lenguajes de programación más usados son aquellos que transfieren directamente el esquema de contactos y las ecuaciones lógicas o los logigramas, pero éstos no son los únicos.

Los lenguajes más usados:

- ✓ **Nemónico**, conocido como lista de instrucciones, booleano. → **AWL**
- ✓ **Diagrama de contactos**, (ladder diagram) → **KOP**
- ✓ **Plano de funciones**, o bloques funcionales → **FUP**
- ✓ **Grafcet**, o diagrama funcional, diagrama de etapas o fases.
- ✓ **Texto estructurado** (informáticos).

Dado el siguiente circuito con lógica de relés, veremos como es su programación en los diversos, lenguajes nombrados:



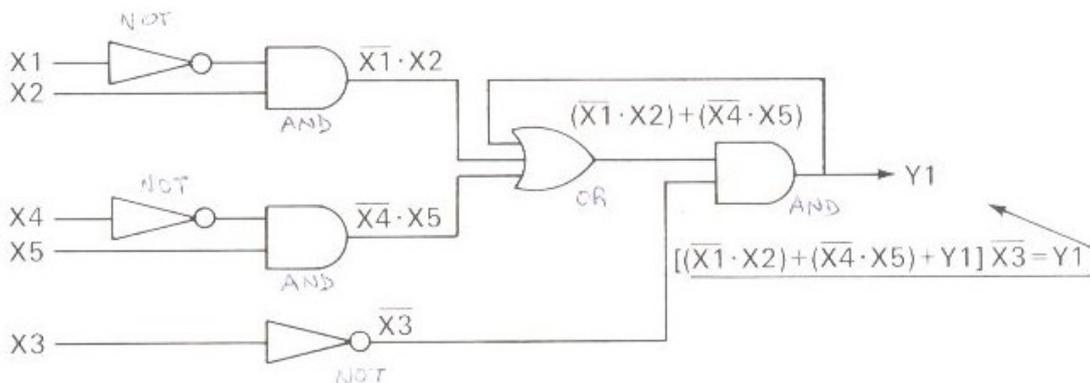
Ecuación lógica:

$$[(\bar{X1} \cdot X2) + (\bar{X4} \cdot X5) + Y1] \bar{X3} = Y1$$

Ecuación:

$$[(\bar{X1} \cdot X2) + (\bar{X4} \cdot X5) + Y1] \bar{X3} = Y1$$

Esquema de relés o lógica cableada de la ecuación propuesta.



Implementación de la ecuación lógica propuesta.

## 📁 Nemónicos o booleanos - AWL.

Alguno de los nemónicos son:

- STR: operación inicio contacto abierto.
- STR NOT: operación inicio contacto cerrado.
- AND (Y): contacto serie abierto.
- OR (O): contacto paralelo abierto.
- AND NOT: contacto serie cerrado.
- OR NOT: contacto paralelo cerrado.
- OUT: bobina de relé de salida.
- TMR: temporizador.
- CNT: contador.

- MCS: conexión de una función a un grupo de salidas.
- SFR: registro de desplazamiento.
- etc.

El ejemplo quedaría:

STR NOT	X1
AND	X2
STR NOT	X4
AND	X5
OR STR	
OR	Y1
AND NOT	X3
OUT	Y1

*Observaciones:*

- Permite crear programas de control introduciendo la nemotécnica de las operaciones.
- El editor AWL también permite crear ciertos programas que, de otra forma, no se podrían programar con los editores KOP ni FUP.
- Ello se debe a que AWL es el lenguaje nativo de la CPU, a diferencia de los editores gráficos en los que son aplicables ciertas restricciones para poder dibujar los diagramas correctamente.

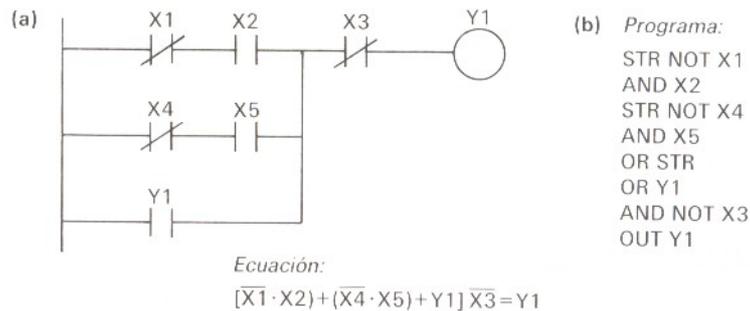
*Ventajas:*

- Más apropiado para los programadores expertos.
- En algunos casos AWL permite solucionar problemas que no podrían solucionarse con KOP o FUP
- En tanto que el editor AWL se puede utilizar siempre para ver o editar un programa creado con los editores KOP o FUP, lo contrario no es posible en todos los casos.

## Diagrama ladder - KOP.

Es una sucesión de redes de contactos que transfieren la información lógica de las entradas a las salidas. El resultado depende de las funciones programadas.

Consiste en asignar los contactos a las entradas del controlador y las bobinas a los relés , a las salidas o a los bits internos del PLC.



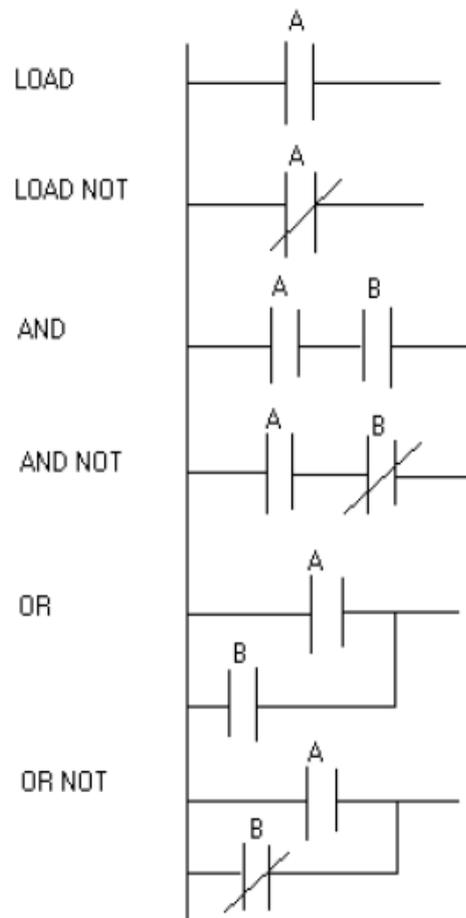
### Observaciones:

- La lógica se divide en unidades pequeñas y de fácil comprensión llamadas llamadas “segmentos” o “networks”
- El programa se ejecuta segmento por segmento, de izquierda a derecha.  
y luego de arriba a abajo.
- Tras alcanzar la CPU el final del programa, comienza nuevamente en la en la primera operación del mismo.

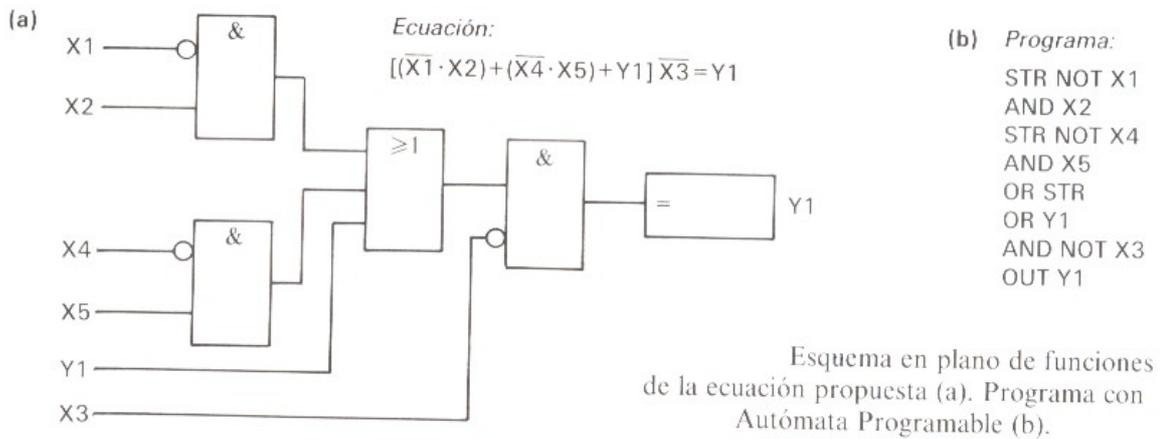
### Ventajas del lenguaje ladder:

- El lenguaje KOP les facilita el trabajo a los programadores principiantes.
- La representación gráfica es a menudo fácil de comprender, siendo popular en el mundo entero.
- El editor AWL siempre se puede utilizar para visualizar un programa creado en KOP.

Funciones básicas:



Plano de funciones - FUP.



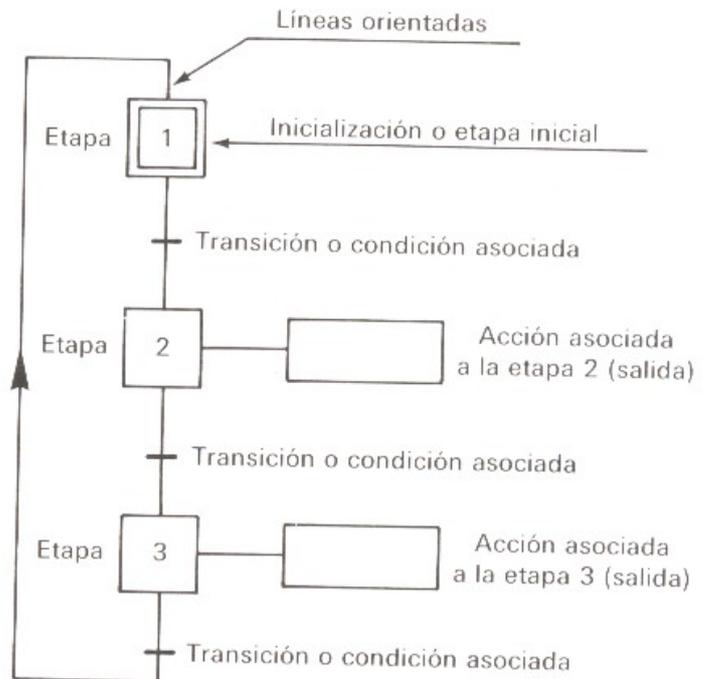
Observar su similitud con los símbolos lógicos o puertas lógicas.

Grafcet (Graphe de Comande Etape Transition).

Es un método gráfico de análisis.

Consiste en descomponer todo automatismo secuencial en una sucesión de etapas, a las que están asociadas acciones, transiciones y receptividades.

El paso de una etapa a otra está condicionada por una transición.



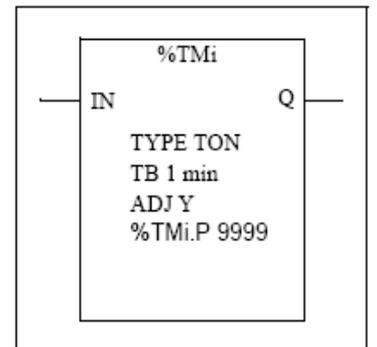
## FUNCIONALIDADES DE BASE.

### i. LOS TEMPORIZADORES %TM

- Son posiciones de memoria, que se actualizan automáticamente por el SO a ritmo de reloj.
- Permiten sincronizar operaciones de entradas/salidas, medir tiempos, etc..
- Los temporizadores tienen asociados un bit lógico, para indicar su estado.

Los parámetros del temporizador:

Parámetro	Etiqueta	Valor
Número de temporizador	%Tmi	0 a 63: TWDLCAA10DRF y TWDLCAA16DRF. 0 a 127 para los demás controladores.
Tipo	TON	• retraso durante el ajuste (predeterminado)
	TOF	• retraso durante el restablecimiento
	TP	• pulso (monoestable)
Base de tiempo	TB	1 min (predeterminado), 1 s, 100 ms, 10 ms, 1 ms
Valor actual	%Tmi.V	Palabra que aumenta de 0 a %Tmi.P cuando el temporizador está en funcionamiento. Se puede leer y comprobar, pero no se puede escribir desde el programa. %Tmi.V se puede modificar utilizando el editor de tablas de animación.
Valor preestablecido	%Tmi.P	0 - 9999. Palabra que se puede leer, comprobar y escribir desde el programa. El valor predeterminado es 9999. El período o retardo generado es igual a %Tmi.P x TB.
Editor de tablas de animación	Y/N	Y: Sí, el valor preestablecido %Tmi.P se puede modificar mediante el editor de tablas de animación. Número: No, el valor preestablecido %Tmi.P no se puede modificar.
Entrada de validación (o de la instrucción)	IN	Inicia el temporizador en flanco ascendente (tipos TON o TP) o en flanco descendente (tipo TOF).
Salida del temporizador	Q	El bit asociado %Tmi.Q se ajusta a 1 dependiendo de la función realizada: TON, TOF, o TP.



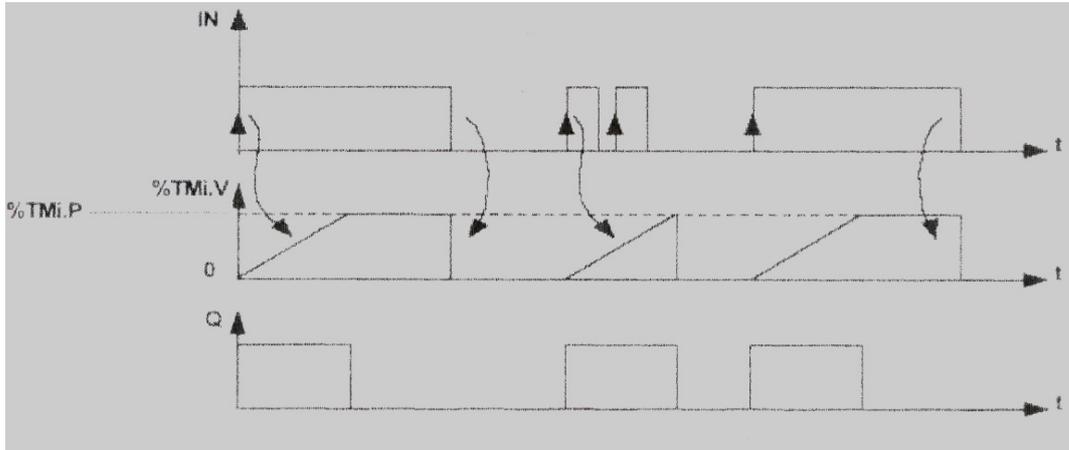
Bloque de función del temporizador



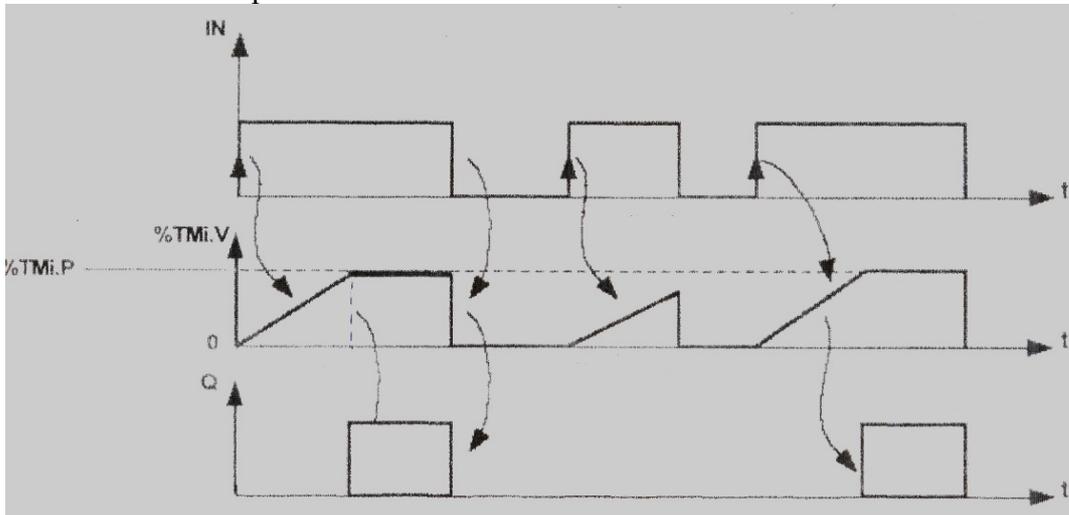
Hay **3 tipos** (básicos) o modos de funcionamiento:

- ✓ **TP:** monoestable, permite generar un impulso de duración precisa (duración programable).
- ✓ **TON:** Permite controlar el retardo en la conexión (retardo programable).
- ✓ **TOF:** Permite controlar el retardo en la desconexión (retardo programable).

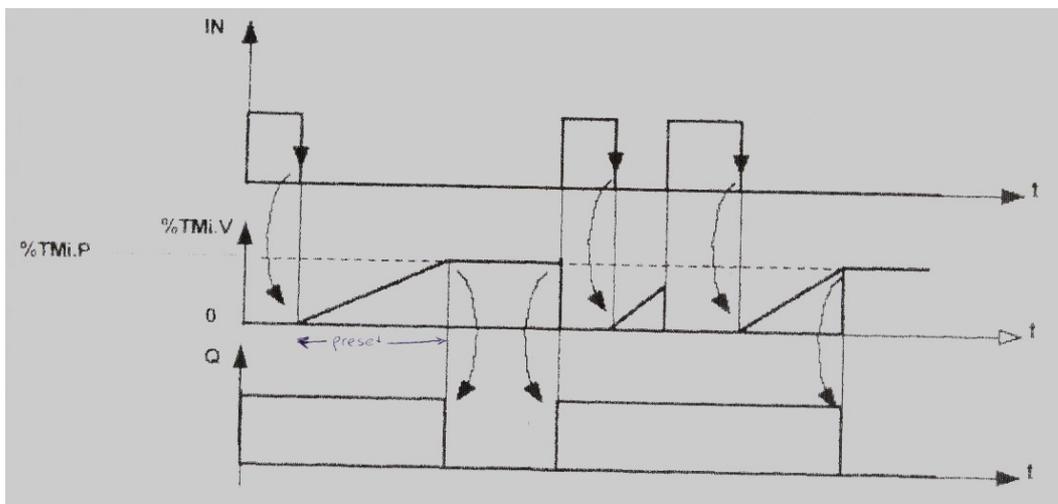
Utilización en monoestable TP:



Utilización del temporizador con retardo en la conexión TON:



Utilización de la temporización con retardo en la desconexión TOF:



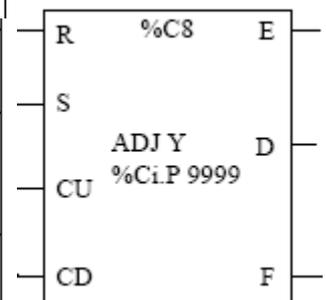


## ii. LOS CONTADORES/ DESCONTADORES %Ci

- ✓ contar.
  - ✓ descontar.
  - ✓ contar/descontar.
- 
- Un contador es un posición de memoria que se actualiza según unas determinadas instrucciones.
  - Trabajan con flanco.

Los parámetros del contador:

Parámetro	Etiqueta	Valor
Número de contador	%Ci	0 à 127
Valor actual	%Ci.V	La palabra se incrementa o reduce de acuerdo con las entradas (o instrucciones) CU y CD. Se puede leer y comprobar, pero no se puede escribir desde el programa. Si desea modificar %Ci.V, utilice el editor de datos.
Valor preestablecido	%Ci.P	$0 \leq \%Ci.P \leq 9999$ . La palabra se puede leer, comprobar y escribir (valor predeterminado: 9999).
Edición con el editor de tablas de animación	ADJ	<ul style="list-style-type: none"> <li>● Y: Sí, el valor preestablecido se puede modificar mediante el editor de tablas de animación.</li> <li>● Número: No, el valor preestablecido no se puede modificar mediante el editor de tablas de animación.</li> </ul>
Restablecer entrada (o instrucción)	R	En estado 1: %Ci.V = 0.
Restablecer entrada (o instrucción)	S	En estado 1: %Ci.V = %Ci.P.
Conteo progresivo de la entrada (o instrucción)	CU	Incrementos %Ci.V en un flanco ascendente.
Conteo regresivo de la entrada (o instrucción)	CD	Disminuciones %Ci.V en un flanco ascendente.
Conteo regresivo de la salida de desborde	E (vacío)	El bit asociado %Ci.E = 1, cuando el contador regresivo %Ci.V cambia de 0 a 9999 (establecido en 1 cuando %Ci.V alcanza 9999 y en 0 si el contador continúa con el conteo regresivo).
Salida predeterminada alcanzada	D (hecho)	El bit asociado %Ci.D = 1, cuando %Ci.V = %Ci.P.
Conteo progresivo de la salida de desborde	F (llena)	El bit asociado %Ci.F = 1, cuando %Ci.V cambia de 9999 a 0 (ajustado a 1 cuando %Ci.V alcanza 0 y a 0 si el contador continúa con el conteo progresivo).



### iii. INSTRUCCIONES DE COMPARACION

- Las instrucciones de comparacion permite cortejar un valor contenido en una tabla de datos con respecto a un valor fijo.
- Estas instrucciones se consideran como instrucciones de entrada.
- Entre otras:

Instrucción	Función
>	Comprueba si el operando 1 es mayor que el operando 2.
>=	Comprueba si el operando 1 es mayor o igual que el operando 2.
<	Comprueba si el operando 1 es menor que el operando 2.
<=	Comprueba si el operando 1 es menor o igual que el operando 2.
=	Comprueba si el operando 1 es igual que el operando 2.
<>	Comprueba si el operando 1 es distinto que el operando 2.

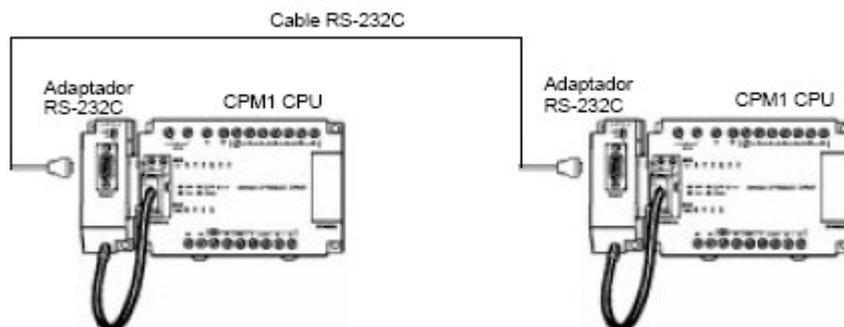
### OTRAS FUNCIONALIDADES DE BASE.

**OBJETOS PALABRAS.**

**FUNCION REGULACION ANALOGICA**

**EXTENSION DE ENTRADAS/SALIDAS**

**COMUNICACIONES ENTRE PLCs.**



**Ejemplos de nomenclatura según distintos fabricantes:**

<i>Fabricante</i>	<i>Nomenclatura utilizada en los módulos de entrada</i>	<i>Nomenclatura utilizada en los módulos de salida</i>
<b>Allen Bradley</b> (Modelos: Micrologix , SLC 500, PLC5)	<b>I : A . B / C</b> Donde: <b>I</b> = “Input” o Entrada <b>A</b> = Número de ranura, módulo o “slot” <b>B</b> = “Word” o Palabra del módulo <b>A</b> . <b>C</b> = Número de “bit” de la palabra <b>B</b> = Número del punto de conexión.	<b>O : A . B / C</b> Donde: <b>O</b> = “Output” o Salida <b>A</b> = Número de módulo o “slot”. <b>B</b> = Número de Palabra o “Word” del módulo <b>A</b> . <b>C</b> = Número de “bit” en la palabra <b>B</b> = Número del punto de conexión.
<b>Mitsubishi</b> (PLC modelo: FX1s , ...)	<b>X A</b> Donde: <b>X</b> = Input <b>A</b> = Número de conexión en el módulo de entradas (0 a n  n =>0 y n =7)	<b>Y B</b> Donde: <b>Y</b> = Output <b>B</b> = Número de conexión en el módulo de salidas (0 a n  n =>0 y n =5)
<b>Omron</b> (PLC modelo: PCM1, ... )	<b>XXXXX</b> Donde: <b>00000 a 00915</b> son entradas	<b>XXXXX</b> Donde: <b>01000 a 01915</b> son salidas.
<b>Schneider</b> (PLC modelo: TSX, ...)	<b>% I 2 0 5 .3</b> Donde: <b>I</b> = Input <b>2</b> = Rack 2 <b>05</b> = Módulo 5 <b>3</b> = Punto de conexión	<b>% Q 2 0 5 .3</b> Donde: <b>Q</b> = Ouotput <b>2</b> = Rack 2 <b>05</b> = Módulo 5 <b>3</b> = Punto de conexión
<b>Siemens</b> (PLC series: 505 )	<b>XA.B</b> Donde: <b>I</b> = Input <b>A</b> = Módulo <b>B</b> = Punto de conexión	<b>YA.B</b> Donde: <b>Q</b> = Salida <b>A</b> = Módulo <b>B</b> = Punto de conexión

## CRITERIOS DE SELECCION

### CRITERIOS CUANTITATIVOS:

Dentro de estos criterios se pueden englobar todas aquellas características que definen a este tipo de equipos y que pueden ser medidas y por tanto comparadas, a saber:

**CICLO DE EJECUCION.** Mide el tiempo que el automata tarda en ejecutar una instrucción o un kbyte de instrucciones. Depende directamente de la velocidad de la CPU del equipo e influirá directamente en el ciclo de SCAN.

**CAPACIDAD DE ENTRADAS-SALIDAS.** Define el número de E/S que se pueden conectar a un equipo.

**CARACTERISTICAS DE LAS ENTRADAS-SALIDAS.** Determina el tipo de E/S que se pueden conectar al automata. Determina por tanto la forma en que el equipo se relaciona con el proceso.

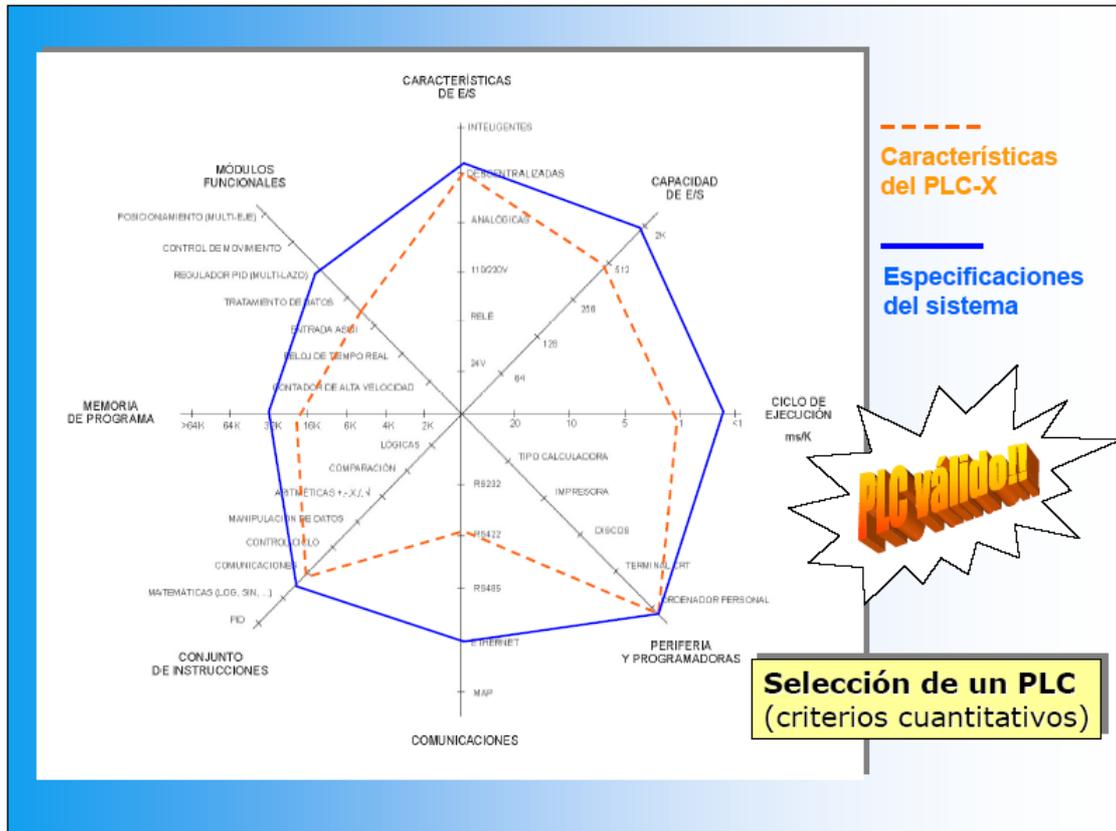
**MODULOS FUNCIONALES.** Algunos tipos de acciones que se pueden llevar a cabo sobre el proceso como por ejemplo el posicionamiento de ejes, el control de procesos continuos, la noción del tiempo, etc.,

**MEMORIA DE PROGRAMA.** Define el tamaño de la memoria del autómata programable y al igual que en caso de los PC's, esta se mide en bytes o múltiplos de bytes.

**CONJUNTO DE INSTRUCCIONES.** No todos los autómatas son capaces de ejecutar los mismos tipos de instrucciones. Por ejemplo, no todos los autómatas son capaces de realizar operaciones en coma flotante. Esta característica determinará la potencia del equipo para afrontar el control de ciertos tipos de procesos.

**COMUNICACIONES.** Otra de las características más importantes a la hora de poder establecer comparaciones entre autómatas es la capacidad que tengan para intercambiar información con otros autómatas u otros módulos de E-S.

**PERIFERIA Y PROGRAMADORAS.** Los autómatas programables pueden ser complementados con dispositivos auxiliares que sin ser esenciales para llevar a cabo su principal función (controlar un proceso), si facilitan ciertas tareas secundarias.



De forma gráfica se puede confeccionar un diagrama que permite comparar de manera visual varios autómatas programables para determinar cual es el que mejor se adapta a nuestras necesidades.

*El polígono que mayor área tenga representará el autómata más potente.*

**OBS.:** En la mayoría de los casos para llevar a cabo la elección final del autómata es necesario tener en cuenta otros criterios que son difícilmente medibles y por tanto comparables. Son los llamados criterios cualitativos.

### CRITERIOS CUALITATIVOS:

**AYUDAS AL DESARROLLO DEL PROGRAMA.** Hace referencia a la cantidad de información y ayuda que presta la empresa distribuidora del equipo a nivel local.

**FIABILIDAD DEL PRODUCTO.** Por lo general la marca o nombre del fabricante del equipo es un aval suficiente para conocer el grado de fiabilidad del mismo.

**SERVICIOS DEL SUMINISTRADOR.** Es muy importante evaluar que servicios extra aporta el distribuidor del equipo a nivel local, tales como: pequeños cursos de formación, su precio, el lugar de impartición y el número de fechas disponibles, capacidad para aportar recursos técnicos, etc.

**NORMALIZACIÓN EN PLANTA.** Esta característica hace referencia a la capacidad del equipo para ser conectado e intercambiar información de manera correcta y efectiva con el resto de los dispositivos y equipos que ya estén instalados en planta. Vendrá determinada por la capacidad del autómatas de soportar estándares internacionales de comunicación, de programación, de conexión, de arquitectura, etc.

Tendencia en el pasado reciente era hacia la “fidelización” del cliente mediante el empleo de sistemas de comunicación y arquitecturas totalmente incompatibles con el resto de los fabricantes de autómatas.

**COMPATIBILIDAD CON EQUIPOS DE OTRAS GAMAS.** Ahondando en el punto anterior, incluso puede darse el caso en que dos equipos del mismo fabricante pero de gamas distintas no sean compatibles entre sí, desde el punto de vista de la conexión y la capacidad de comunicación.

**COSTO.** Aunque el precio de un producto es un criterio cuantitativo que puede ser medido y comparado, su costo o valor apreciado no lo es. Este hace referencia a la apreciación de cuánto caro o barato le parece a una persona un producto.

### **NOTA:**

*Estas características cualitativas a la larga acaban convirtiéndose en las más importantes a la hora de seleccionar no tanto qué equipo adquirir sino de qué fabricante en cuestión. Esto es debido fundamentalmente a que hoy en día todos los fabricantes proporcionan equipos con características cuantitativas muy similares, y con una alta gama de formatos. Esto hace que el punto de vista cuantitativo no sea muy determinante a la hora de seleccionar el autómatas programable más adecuado.*