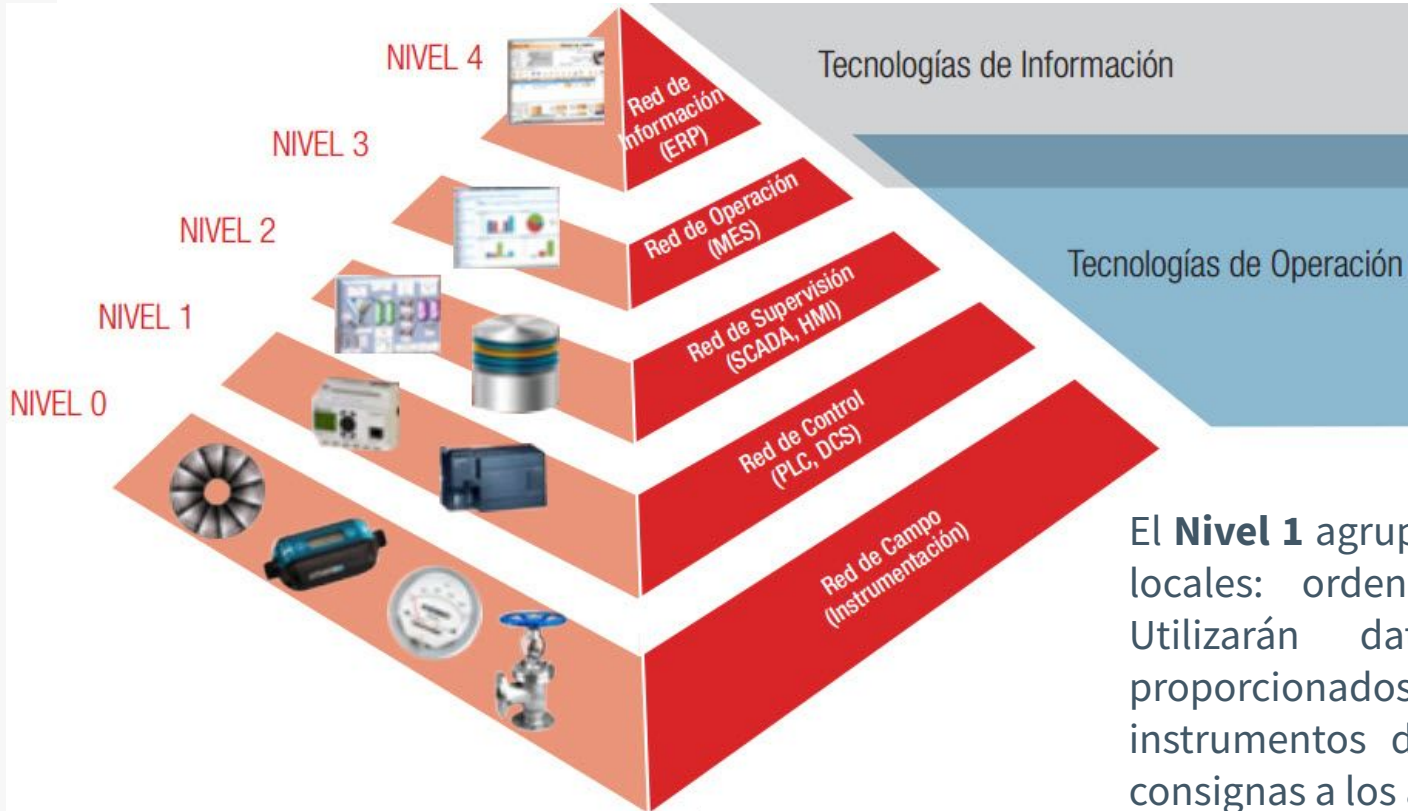


Programación de PLC's



Msc Andrea Enríquez Zúñiga

Pirámide de automatización



El **Nivel 1** agrupa los controladores locales: ordenadores, PLCs, etc. Utilizarán datos del proceso proporcionados por los instrumentos del Nivel 0 y darán consignas a los actuadores.

Dispositivos de entrada

Pulsadores, Switches, Finales de carrera, Sensores (fotoeléctricos, inductivos, capacitivos, temperatura etc.).

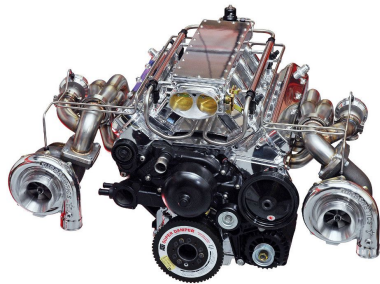


Dispositivos de salida

Electroválvulas, Servoválvulas,
Contactores, pilotos luminosos,
variadores, relevos auxiliares etc..).



Elementos de la automatización industrial



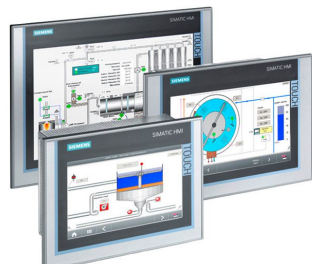
Movimiento

Elementos mecánicos, electrónicos o electromecánicos, hidráulicos, neumáticos



Controlador lógico programable

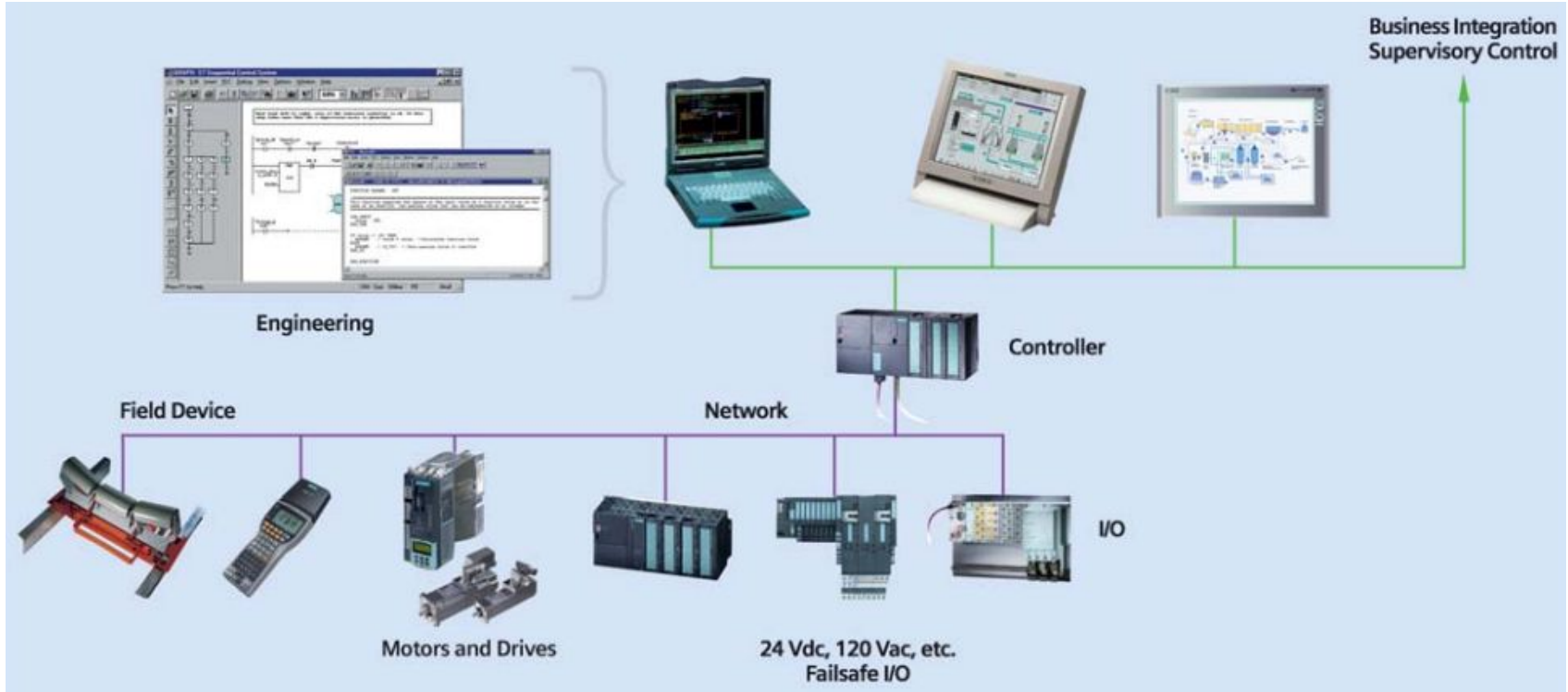
Se encarga de controlar las secuencias, tomar decisiones basadas en una programación preestablecida, se encarga de las repeticiones en el proceso.



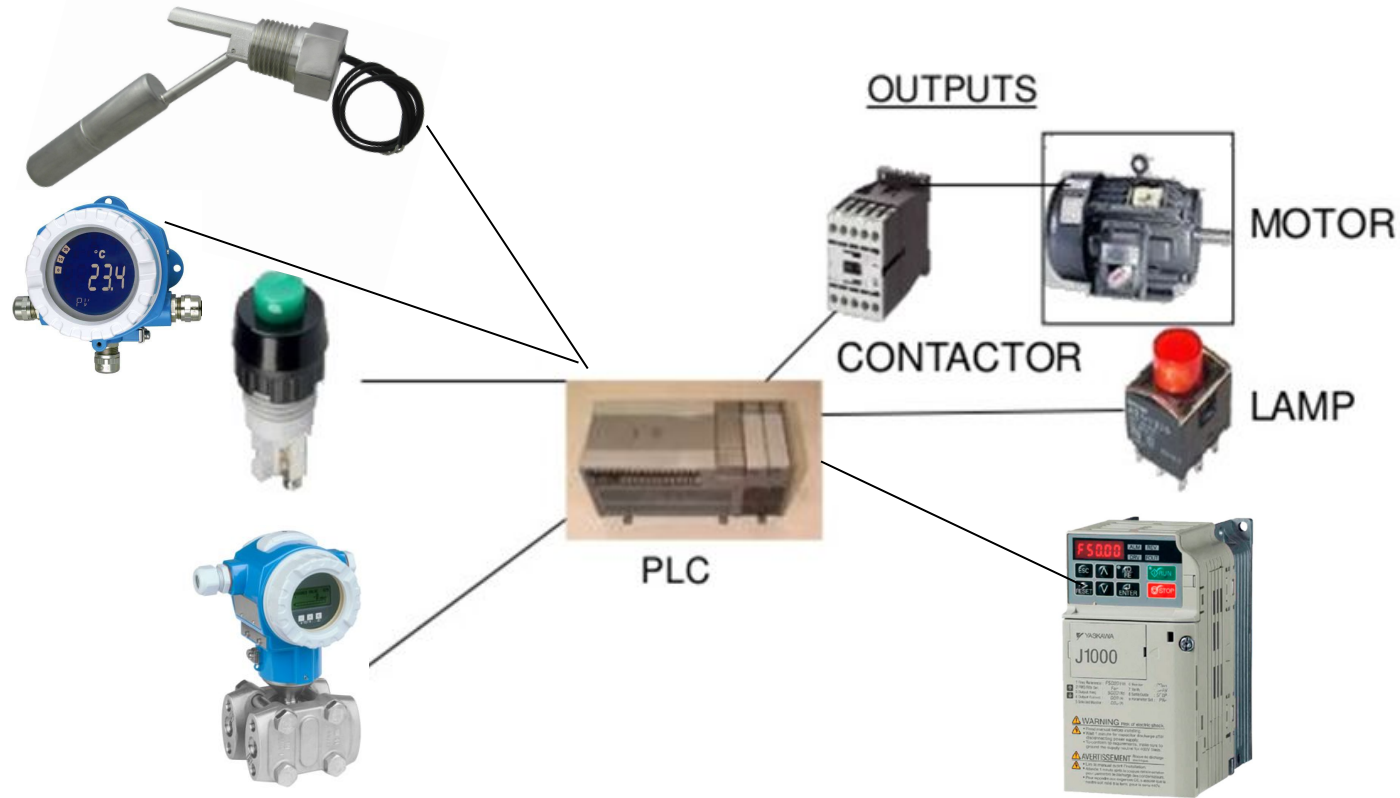
Interfaz Hombre-Máquina

HMI es la principal herramienta con la cual los operadores y los supervisores de la línea coordinan y controlan los procesos industriales y de fabricación en la planta.

Arquitectura típica de un sistema de automatización basado en PLC



Dispositivos de entrada y salida



Controlador Lógico Programable PLC



- **Definición IEC 61131**

Un **autómata programable (AP)** es una máquina electrónica programable diseñada para ser utilizada en un entorno industrial (hostil), que utiliza una **memoria programable** para el almacenamiento interno de instrucciones orientadas al usuario, para implantar soluciones específicas tales como funciones lógicas, secuencias, temporizadores, recuentos y funciones aritméticas, con el fin de **controlar** mediante entradas y salidas, digitales y análogas diversos tipos de **máquinas o procesos**.

AP = PLC

Autómata Programable = Controlador Lógico Programable

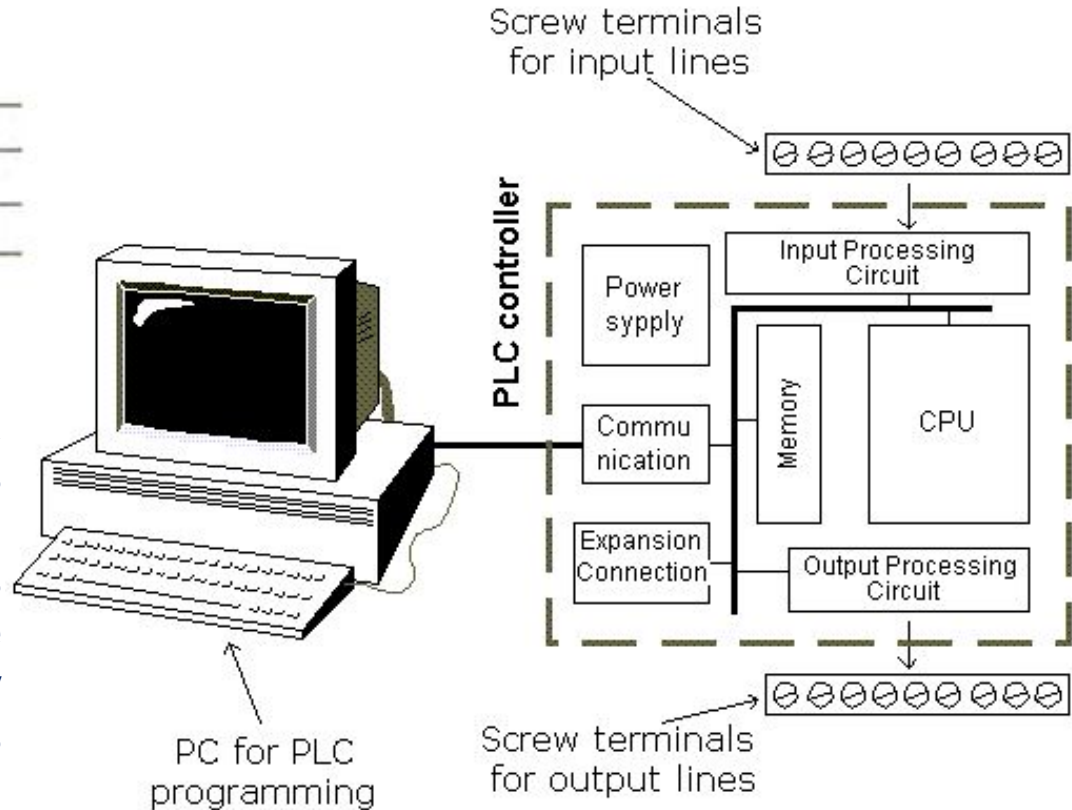




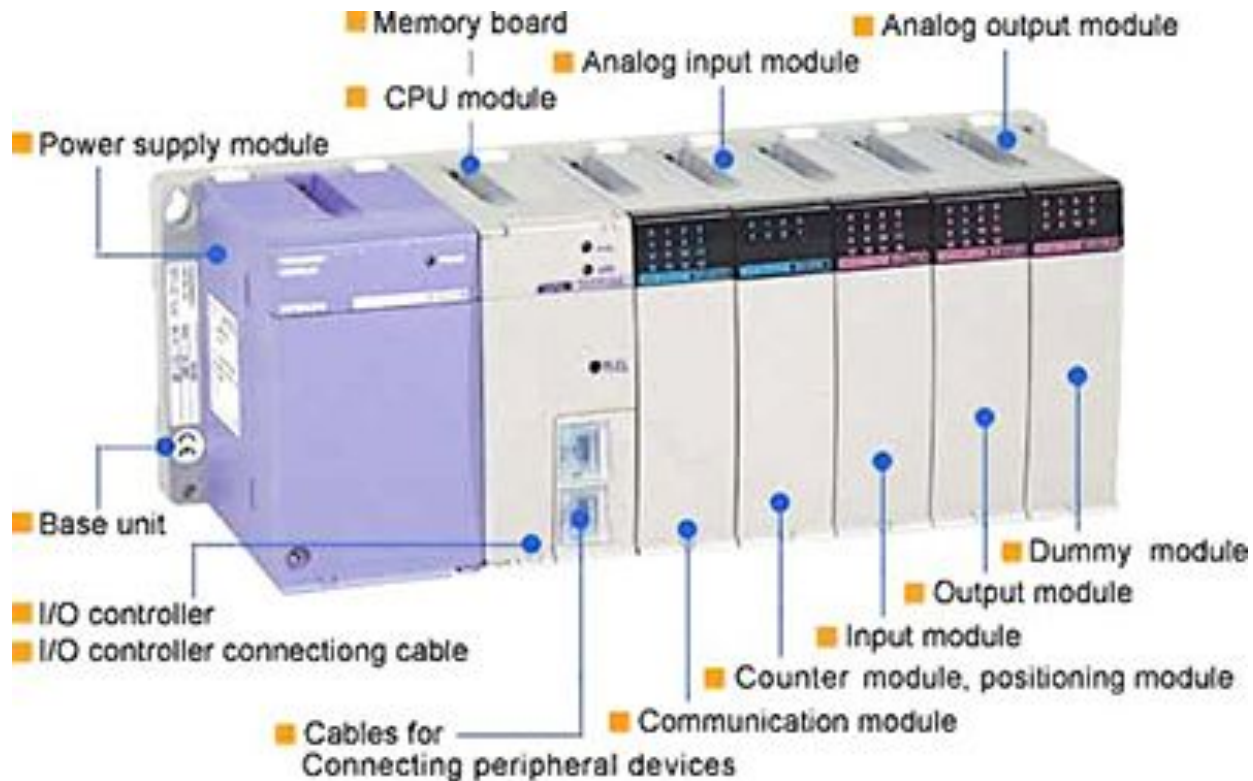
Estructura general de un PLC



Internamente compuesto de dispositivos electrónicos para poder alojar las instrucciones básicas del funcionamiento del PLC, así como las unidades para procesar instrucciones de un programa precargado y realizar las tareas especificadas en él.



Estructura general de un PLC





Clasificación de PLC

- Según número de entradas y salidas

De gama baja:

- con $E/S < 256$

De gama media:

- con $256 \leq E/S \leq 1024$

De gama alta:

- con $E/S > 1024$

Micro

- Típicamente menos de 32 I/O

Pequeño

- Típicamente menos de 128 I/O

Mediano

- Típicamente menos de 1024 I/O

Grande

- Típicamente más de 1024 I/O

Clasificación de PLC

- Según su conformación

Compactos



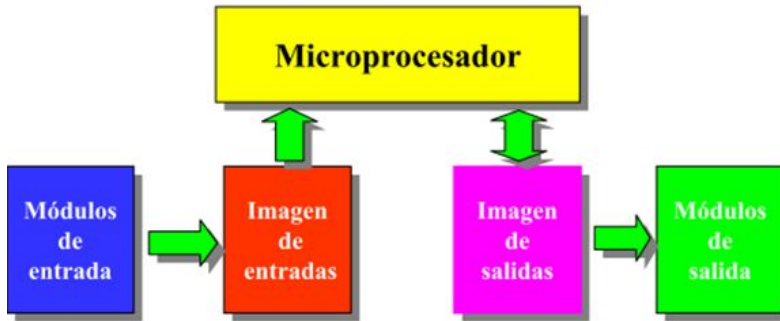
Tienen sus componentes integrados (Fuente, CPU; E/S), no permiten expansión mediante módulos.

Modulares

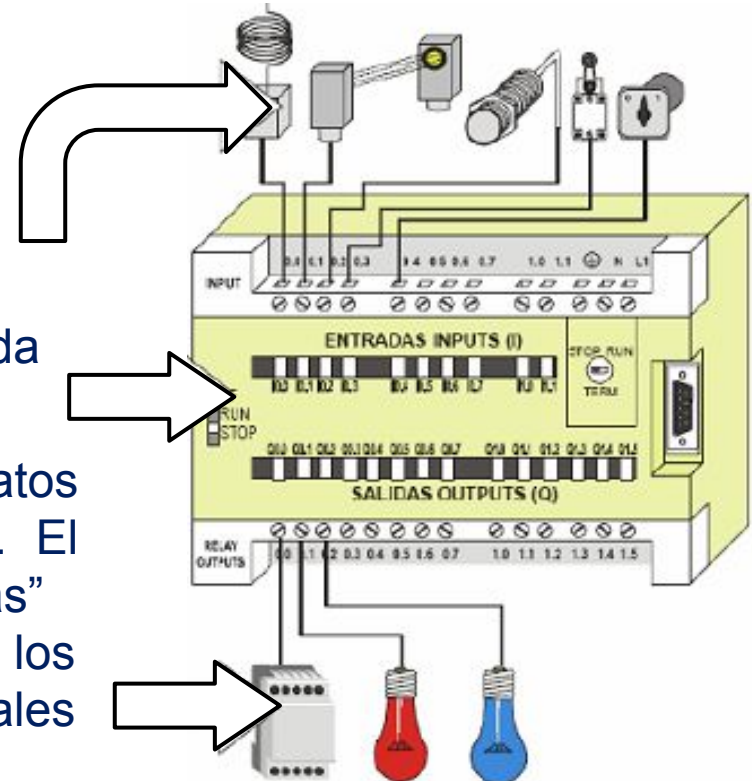


Permiten la expansión del sistema mediante módulos adicionales, ej: Comunicaciones, E/S

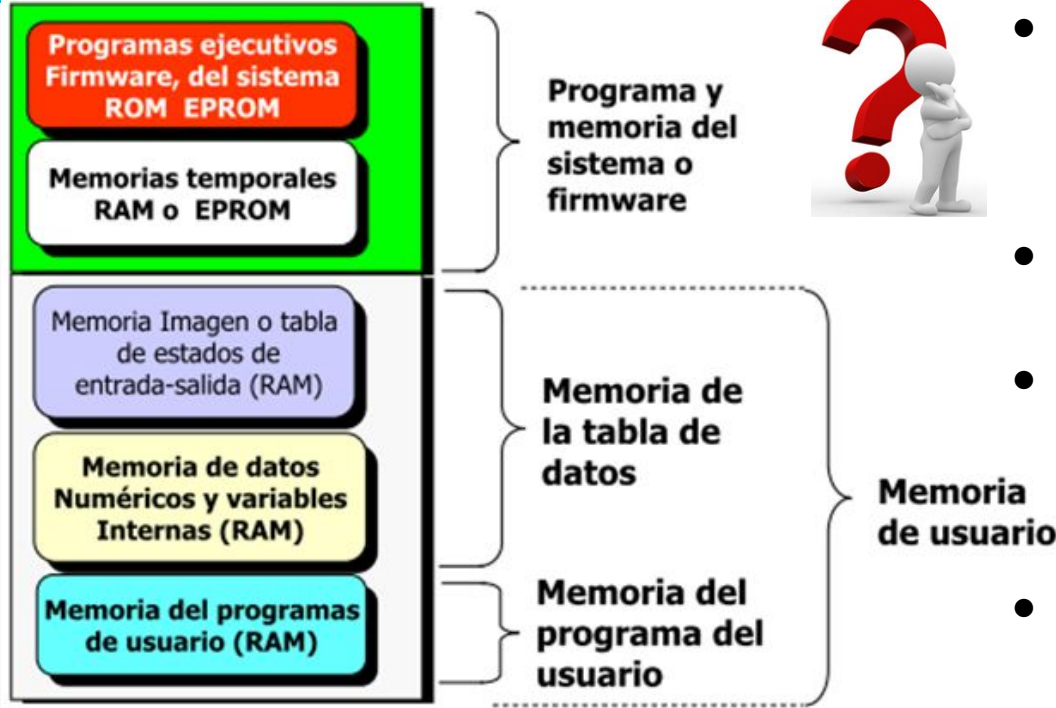
Estructura interna de PLC



1. Consulta estado de entradas y guarda información en “imagen de entrada”
2. Consulta al programa según datos almacenados en “imagen de entrada”. El resultado lo guarda en “imagen de salidas”
3. Según la “imagen de salidas”, los dispositivos conectados a sus terminales serán activados o desactivados



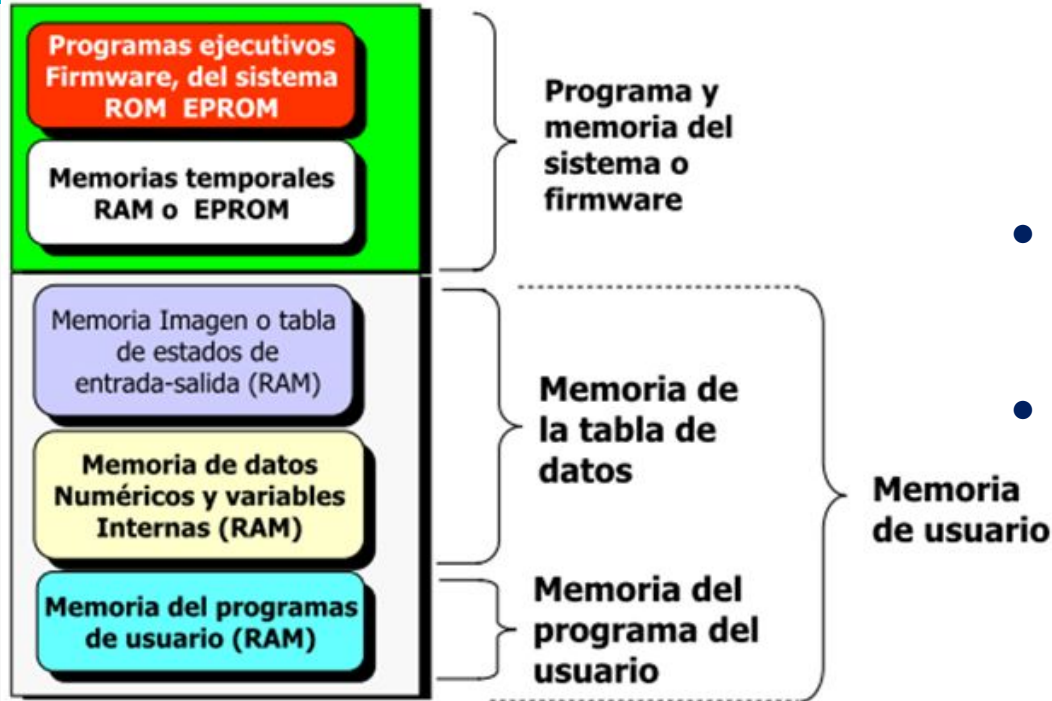
Estructura interna de PLC



Tipos de memoria

- **RAM (Random Access Memory):** Memoria de lectura y escritura. su información desaparece al faltarle la corriente
- **ROM (Read Only Memory):** Memoria de solo lectura, no reprogramable
- **EPROM (Erasable Programmable Read-Only Memory):** Memoria de solo lectura, reprogramables con borrado por ultravioletas.
- **EEPROM (Electrically Erasable Programmable Read-Only Memory):** Memoria de solo lectura, alterables por medios eléctricos.

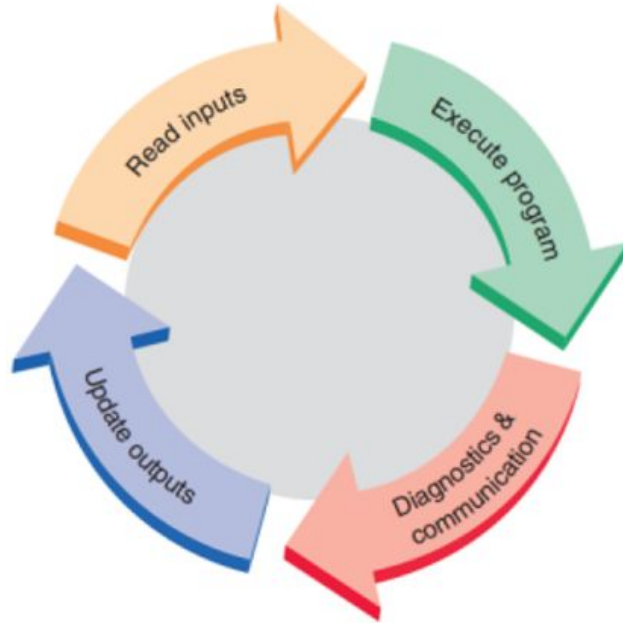
Estructura interna de PLC



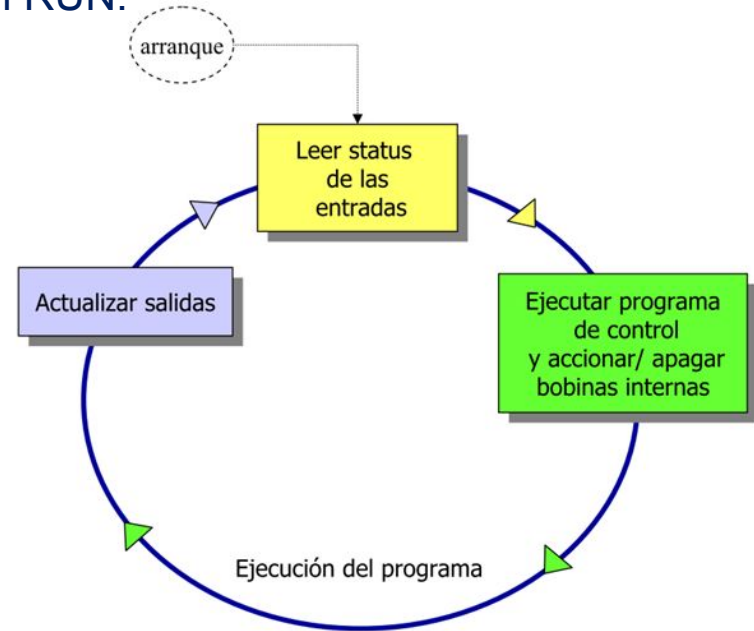
- **Firmware:** programa informático que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo.
- **EPROM** se utilizan para almacenar el programa de usuario, una vez que ha sido convenientemente depurada.
- Más frecuente uso de combinaciones **RAM + EEPROM**, utilizando estas últimas como memorias de seguridad que salvan el contenido de las RAM. Una vez reanudada la alimentación, el contenido de la EEPROM se vuelca sobre la RAM

¿Cómo trabaja un PLC?

CICLO DE SCAN

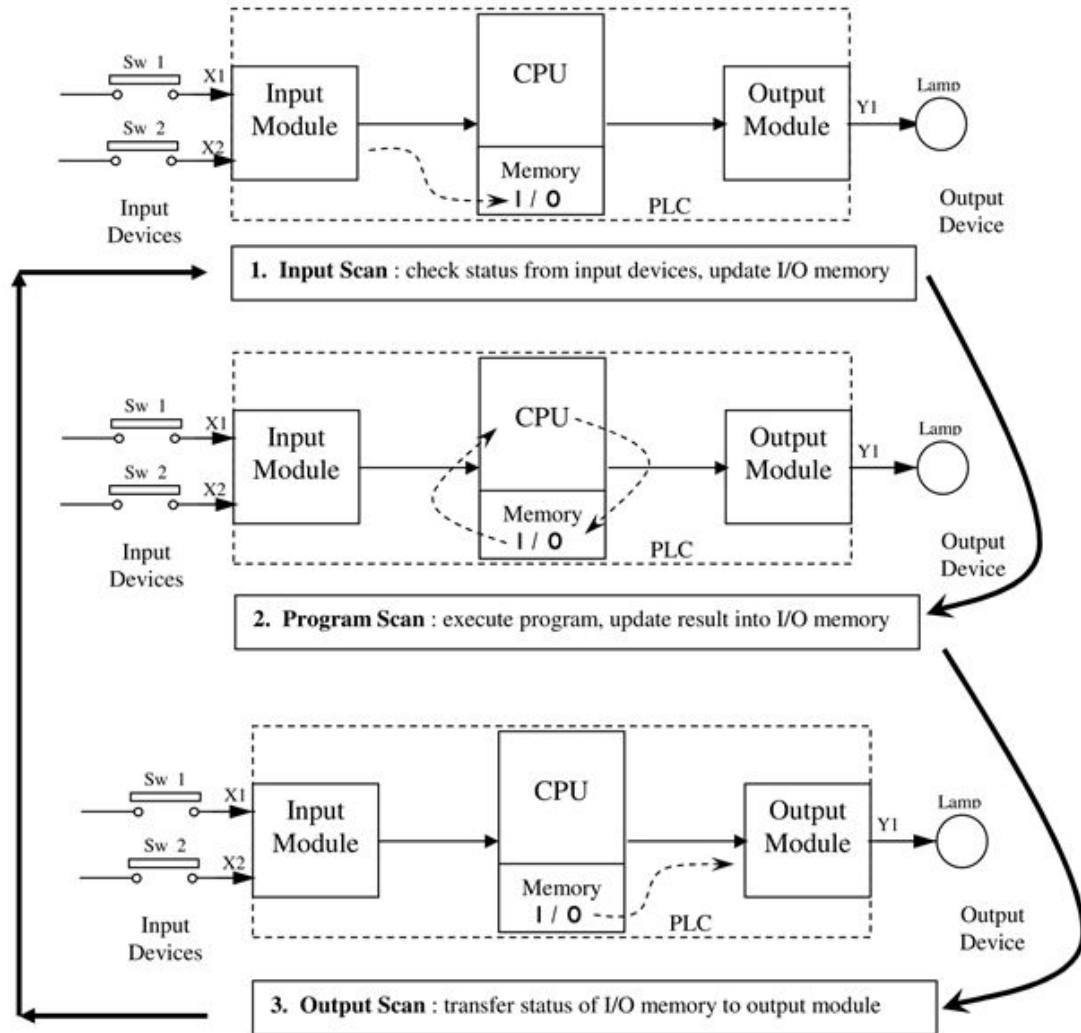


- Secuencia de operaciones que realiza el autómatas de manera repetitiva una vez que entra en RUN.



El tiempo total del ciclo de SCAN se mide en mili-segundos y depende de la capacidad de proceso de la CPU así como de lo grande del programa.

¿Cómo trabaja un PLC?



MÉTODO DE TRANSFERENCIA



¿Automatización Industrial?

1. Hand Terminal
2. Teclado integrado
3. Conexión a PC, RS232, USB
4. Ethernet, etc

MÉTODO DE TRANSFERENCIA



IP:

192.168.1.65

Mascara de subred:

255.255.255.0

Nombre del PC:

IP:

192.168.1.20

Mascara de subred:

255.255.255.0

Nombre:

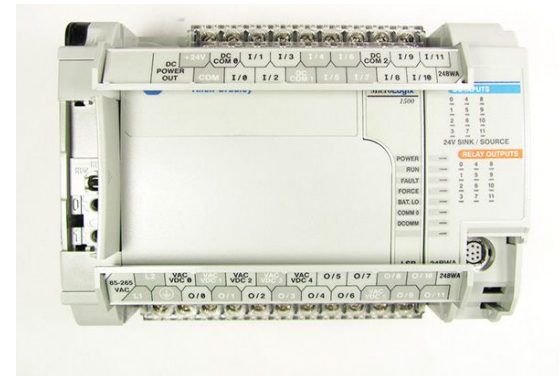
¿Cómo trabaja un PLC?

3 Modos de programación



1. Modo programación:

- Las salidas son desenergizadas
- El código de usuario no se ejecuta
- Permite realizar la programación de un nuevo código
- No se puede cambiar de modo.

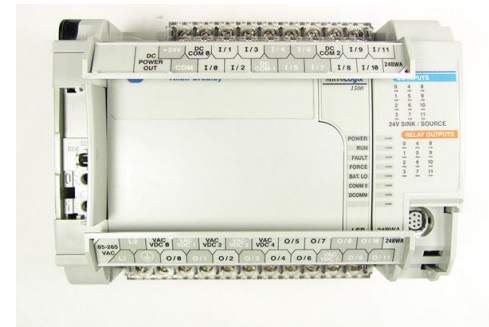


¿Cómo trabaja un PLC?

3 Modos de programación

2. Modo remoto:

- Se utiliza para pruebas.
- Permite cambiar el código o segmentos de código del programa.
- Permite el cambio de modo de operación.

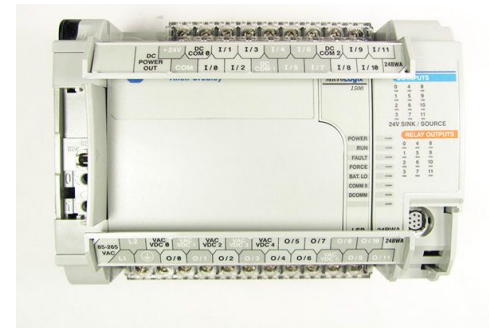


¿Cómo trabaja un PLC?

3 Modos de programación

3. Modo run:

- Ejecución normal del código.
- Deshabilita el uso del programador para edición de código.
- Energiza las salidas según código



LENGUAJES DE PROGRAMACIÓN PLC's

Programa

Conjunto de instrucciones, órdenes y símbolos reconocibles por el PLC, a través de su unidad de programación, que le permiten ejecutar una secuencia de control deseada.

Lenguaje de programación

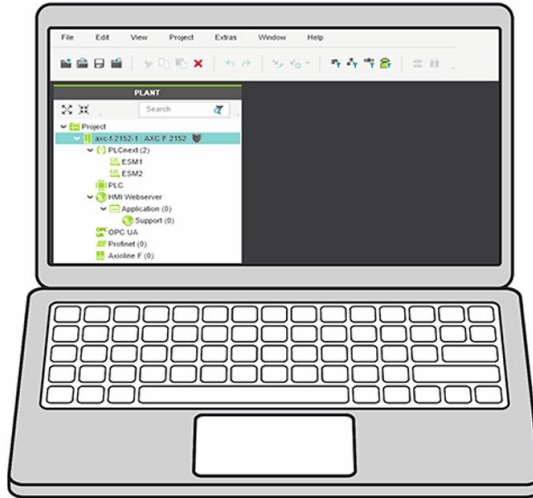
Permite al usuario ingresar un programa de control en la memoria del PLC, usando una sintaxis establecida.



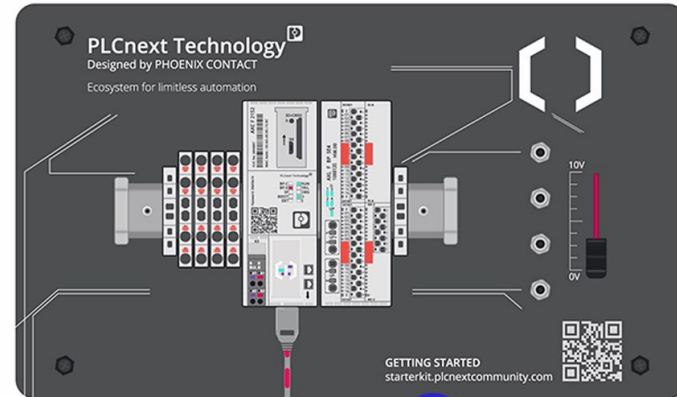
LENGUAJES DE PROGRAMACIÓN PLC's



 PLCnext Engineer



PLCnext Starterkit



15

Ethernet IP address:
192.168.1.1

REALPARS

SOFTWARE DE PROGRAMACIÓN PLC's



PLCnext Engineer

File Extras Help

Start Page

PHENIX CONTACT
INSPIRING INNOVATIONS

PLCnext Engineer 2022.0.1

(Build 5.0.546.0)

Seleccionar referencia PLC

Recent projects

Project templates

Need help?

Welcome to PLCnext Engineer
The Start Page welcomes you introducing PLCnext Engineer. The Start Page is shown every time you start closed when opening or creating a project.

Recent projects
The 'Recent projects' list shows the projects that you have recently opened in PLCnext Engineer. Click the

Project templates
The 'Project templates' list provides a list of predefined project templates to help you get started. Selecting a project with the desired controller type already added.

Further help
If you do not know what to do after a project is opened, you can find further help [here](#).

PLCnext Engineer - C:\Users\AndreitaE\AppData\Roaming\PHOENIX CONTACT\PLCnext Engineer\2022.0\NEWPROJECT1.pcwex

File Edit View Project Extras Window Help

PHOENIX CONTACT

PLANT

- Project
 - axc-f-1152-1 : AXC F 1152
 - PLCnext (1)
 - PLC
 - HMI Web Server
 - OPC UA
 - Profinet (0)
 - Axioline F (0)
 - PLCnext Components (0)

COMPONENTS

- Programming (300)
- PLCnext Components & Programs (0)
- Network (511)
- HMI (34)
- Libraries (1)

0 errors, 0 warnings 100%

IEC 61131-3

Comisión Electrónica Internacional



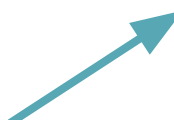

- Parte 1: información general Ed. 2.0 (2003)
- Parte 2: Especificaciones y ensayos de los equipos Ed 3.0 (2007) (hardware)
- **Parte 3: Lenguaje de programación Ed 2.0 (2003).**
- Parte 4: Guías de usuario Ed 2.0 (2004).
- Parte 5: Comunicaciones Ed. 1.0 (2000).
- Parte 6: Seguridad Funcional Ed. 1.0 (2012)
- Parte 7: Programación para control difuso (Fuzzy) Ed. 1.0 (2000)
- Parte 8: Directrices para la aplicación e implementación de lenguajes de programación Ed. 2.0 (2003)

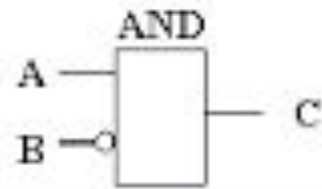
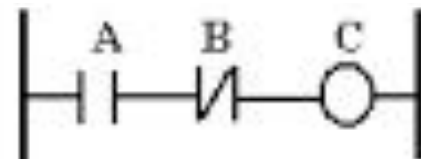


IEC 61131-3


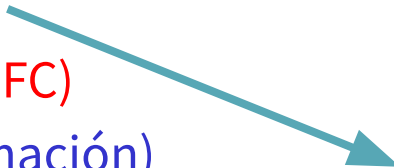
Comisión Electrónica Internacional

1. Lenguajes Gráficos:

- Diagrama de escalera o Ladder (LD) 
- Diagrama de bloques funcionales (FBD) 



2. Lenguajes Escritos:

- Lista de instrucciones (IL) 
- Texto Estructurado (ST) 

```
LD    A
ANDN  B
ST    C
```

- ### 3. Sequential Function Chart (SFC)
- Grafcet (modelado y programación) 

$C = A \text{ AND NOT } B$

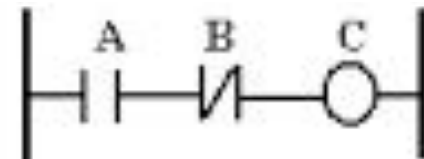
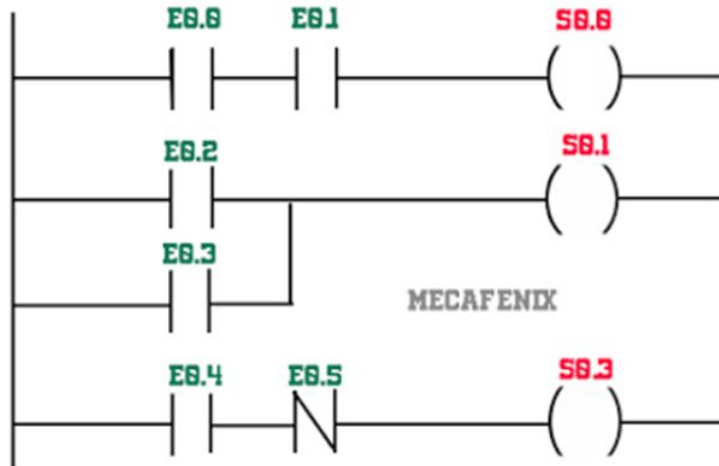
IEC 61131-3

Comisión Electrónica Internacional



Ladder

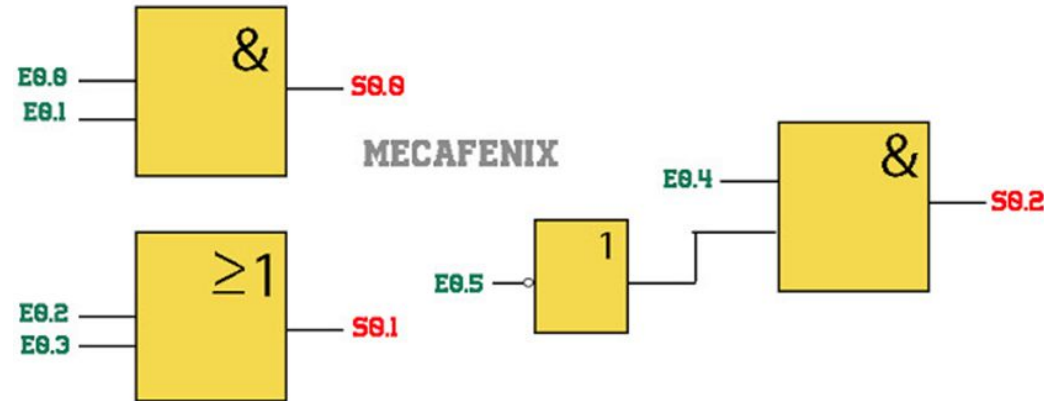
Lenguaje de programación inspirado en el concepto tradicional de la **lógica cableada**, Se basa en barras de alimentación, elementos de enlace, contactos, bobinas y bloques funcionales.



Bloques Funcionales FBD

Lenguaje de programación gráfico basado en **bloques con entradas y salidas** que pueden ser interconectados entre sí para formar una función lógica - aritmética, los tipos de datos dependen del bloque.

Ej: Bloque AND, Suma, Resta.



Lista de Instrucciones IL

Es un lenguaje tipo **ensamblador**. Usa la sintaxis del Álgebra de Boole para ingresar y explicar la lógica de control. Consiste en elaborar una lista de instrucciones o nemónicos, haciendo uso de operadores Booleanos (AND, OR, NOT, etc.) y otras instrucciones nemónicas, para implementar el circuito de control.

```

000 LD    %I0.1  Bp. inicio ciclo
    AND  %I0.0  Dp. presencia vehiculo
    AND  %M3    Bit autorización reloj calendario
    AND  %I0.5  Fc. alto rodillo
    AND  %I0.4  Fc. detrás pórtico
005 S    %M0    Memo inicio ciclo
    LD    %M2
    AND  %I0.5
    OR   %I0.2  Bp. parada ciclo
    R    %M0
010 LD    %M0
    ST   %Q0.0  Piloto ciclo
  
```

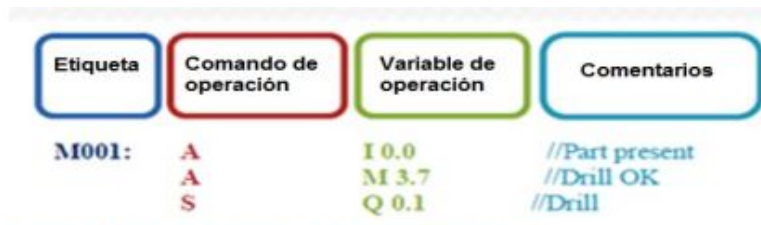


Imagen 7: Estructura de una orden en programación IL/STL.



Texto Estructurado

Es un lenguaje escrito, representa un modo de programación de **alto nivel**, algunos similares a pascal, C, C++.

```
IF ((E0.0 == TRUE) && (E0.1 == TRUE))
{
  S0.0 = TRUE;
}
ELSE S0.0 = FALSE;
```

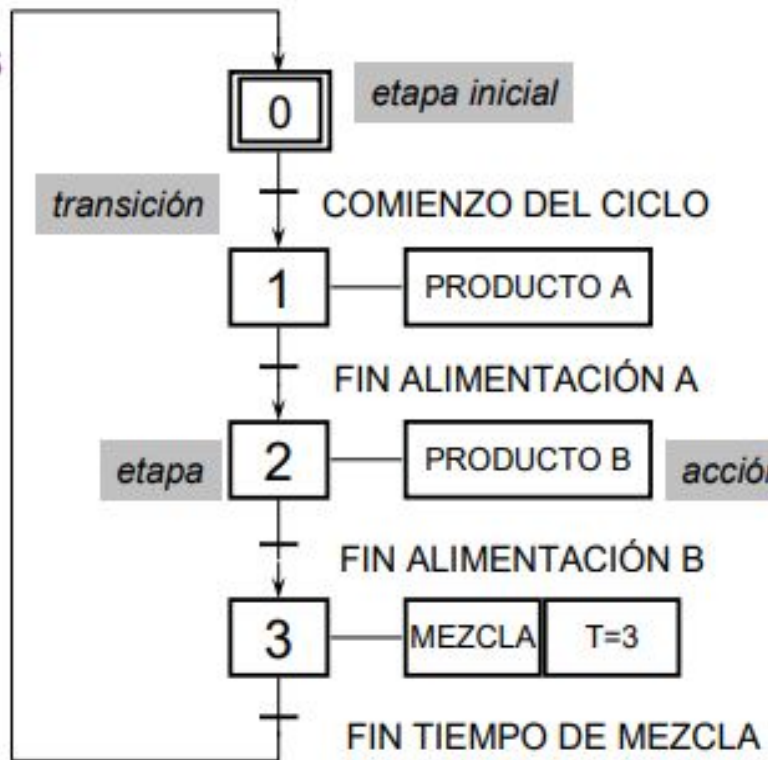
```
IF ((E0.2 == TRUE) || (E0.3 == TRUE))
{
  S0.1 = TRUE;
}
ELSE S0.1 = FALSE;
```

```
IF ((E0.4 == TRUE) && (E0.5 == FALSE))
{
  S0.2 = TRUE;
}
ELSE S0.2 = FALSE;
```

```
IF Dato = "EOF" THEN
  FOR indice:=1 TO 128 DO
    x:= Read_dato(Canal[indice]);
    IF x>2500 THEN Alarma:=TRUE;
    END_IF;
  END_FOR;
END_IF
```

SFC

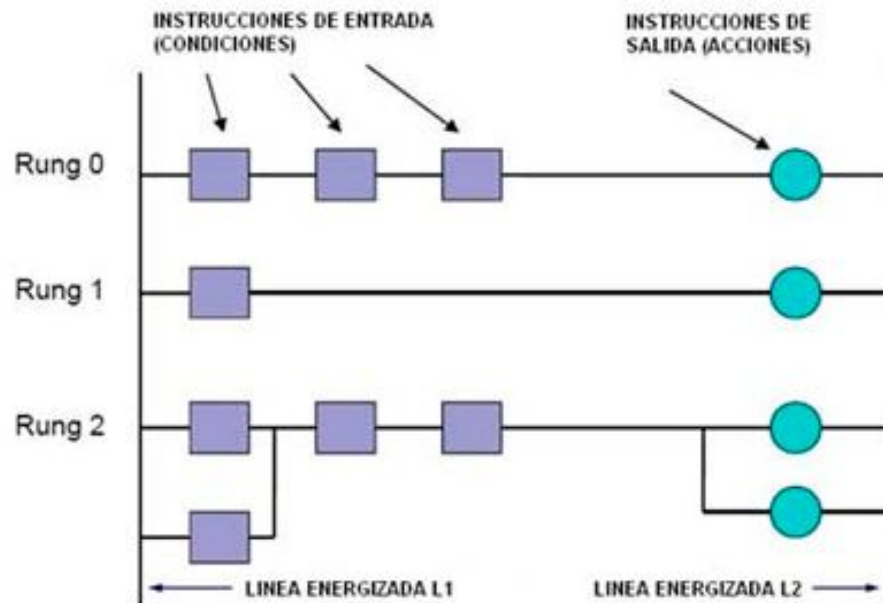
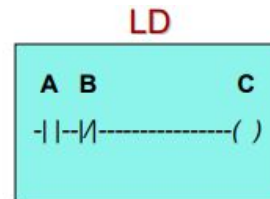
Sequential Function Chart



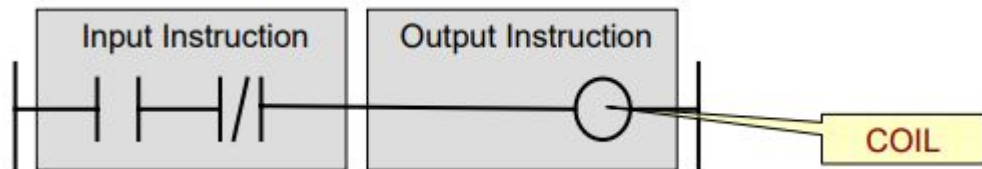
- Las etapas o estados implican acciones asociadas
- Las transiciones gobiernan los cambios de estado
- Las flechas indican la dirección de cambio
- Pueden darse esquemas menos lineales

Ladder o escalera

- Lenguaje más usado para la programación de plcs
- Fue el primero con el que se empezó a programar, de allí su semejanza con diagramas eléctricos de escalera utilizados por los técnicos antes de la aparición del autómat.



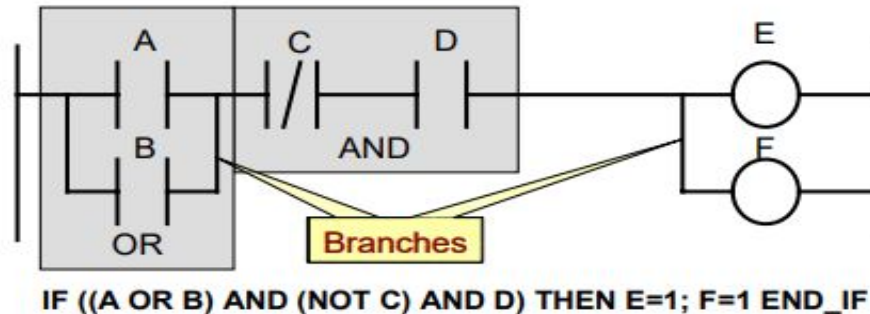
¿Rung?



- Es una línea de programa
- Contiene instrucciones de entrada y salida
 - **Entrada:** permite una comparación o test de las condiciones y se obtiene el resultado de la evaluación
 - **Salida** (coil): examinan el resultado de la evaluación y si es *true* ejecutan alguna operación o función
 - En algunos casos pueden ser el estado del rung

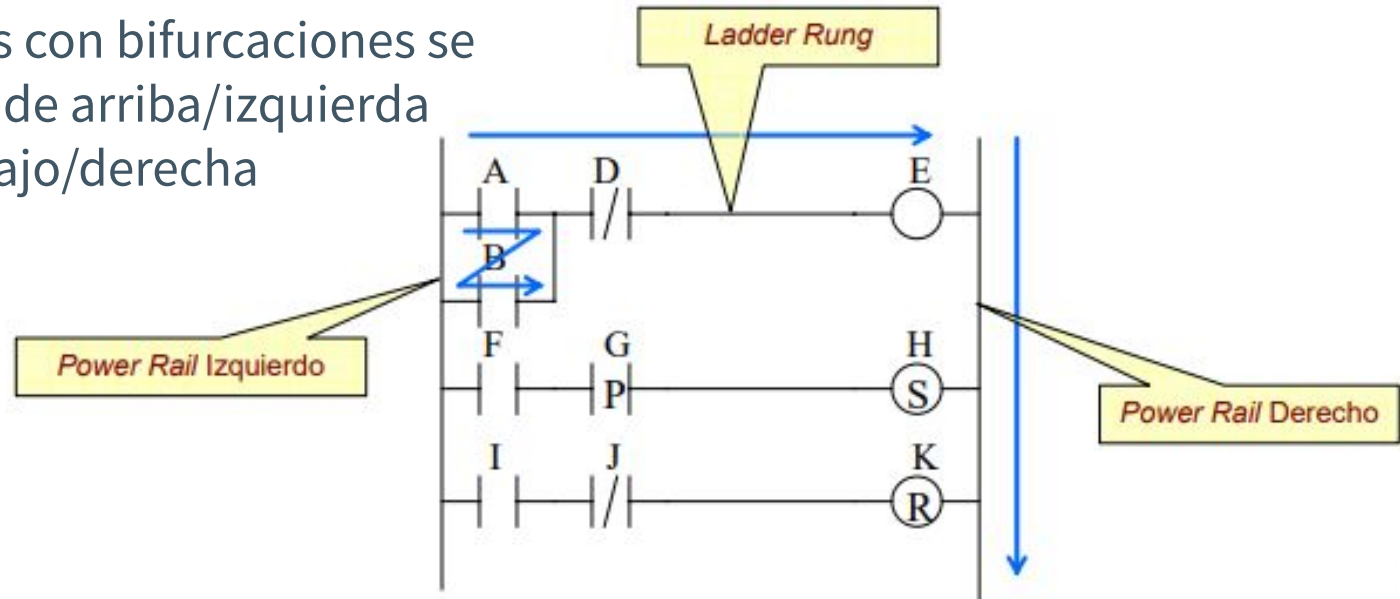
Ladder: Operaciones en Serie y en Paralelo

- Las instrucciones de entrada pueden ejecutarse mediante relaciones lógicas AND y OR en un sencillo formato.
 - Si las relaciones están en **serie** se evalúa un **AND**
 - Si las relaciones están en **paralelo** se evalúa un **OR**
- Salidas en paralelo permiten activar operaciones o funciones con el mismo resultado de la evaluación



Ladder: Ejecución lógica

- Los Rungs se ejecutan de **izquierda a derecha** y de **arriba hacia abajo**
- Los rungs con bifurcaciones se ejecutan de arriba/izquierda
Hacia abajo/derecha





Ladder: Contactos

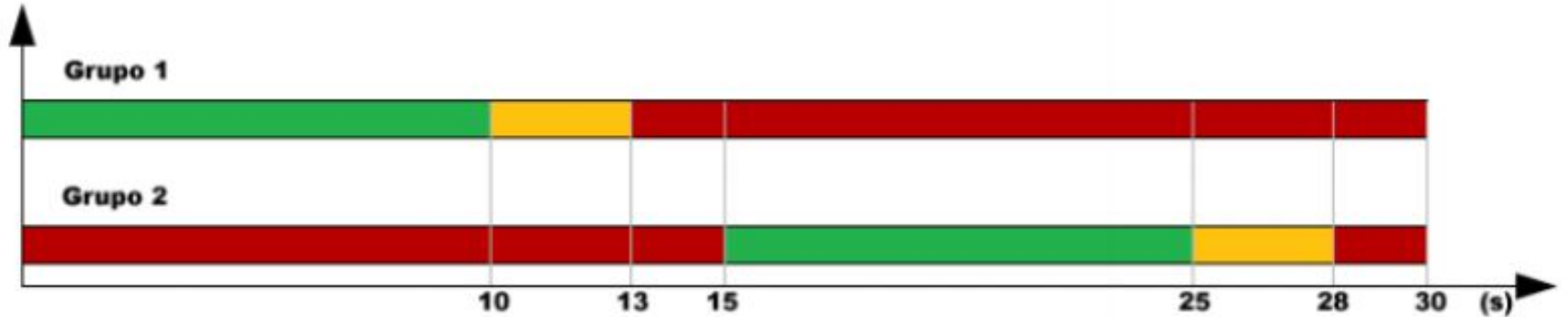
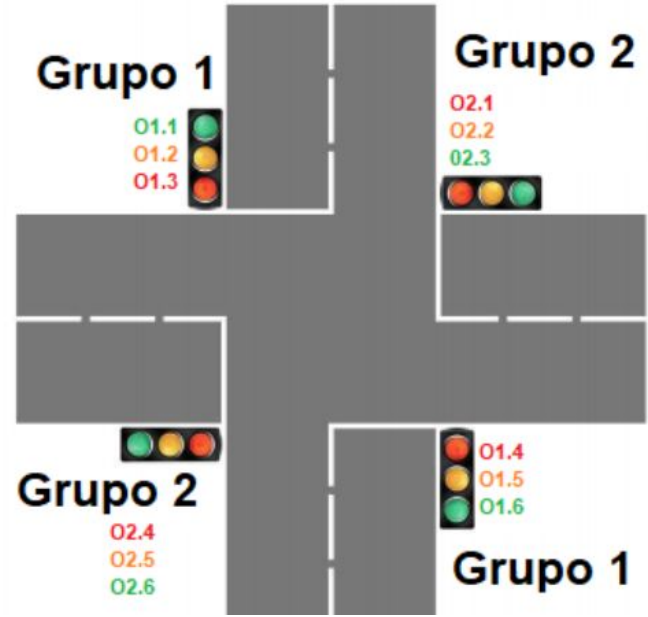
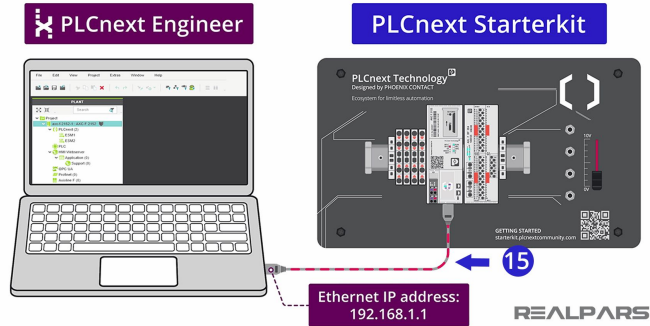
- Normalmente abierto **--| |--**
 - Activa el rung hacia la derecha cuando el contacto se **activa**
- Normalmente cerrado **--|/ |--**
 - Activa el rung hacia la derecha cuando el contacto se **desactiva**
- Transición positiva **--|P |--**
 - Activa el rung hacia la derecha de la instrucción cuando el contacto está **desactivado** en el scan anterior y está **activado** en el scan actual (Allen Bradley utiliza **ONS**)
- Transición negativa **--|N |--**
 - Activa el rung hacia la derecha de la instrucción cuando el contacto está **activado** en el scan anterior y está **desactivado** en el scan actual



Ladder: Acciones (Coils)

- Acción **--()--**
 - Activa un **bit** cuando el rung es *true* y lo *desactiva* cuando es *false*.
- Acción negada **--(/)--**
 - Activa un **bit** cuando el rung es *false* y lo *desactiva* cuando es *true*
- Enclavamiento (Latch) **--(S)--**
 - Activa un **bit** cuando el rung es *true* y no hace nada cuando es *false*
- Desenclavamiento (Unlatch) **--(R)--**
 - Desactiva un **bit** cuando el rung es *true* y no hace nada cuando es *false*

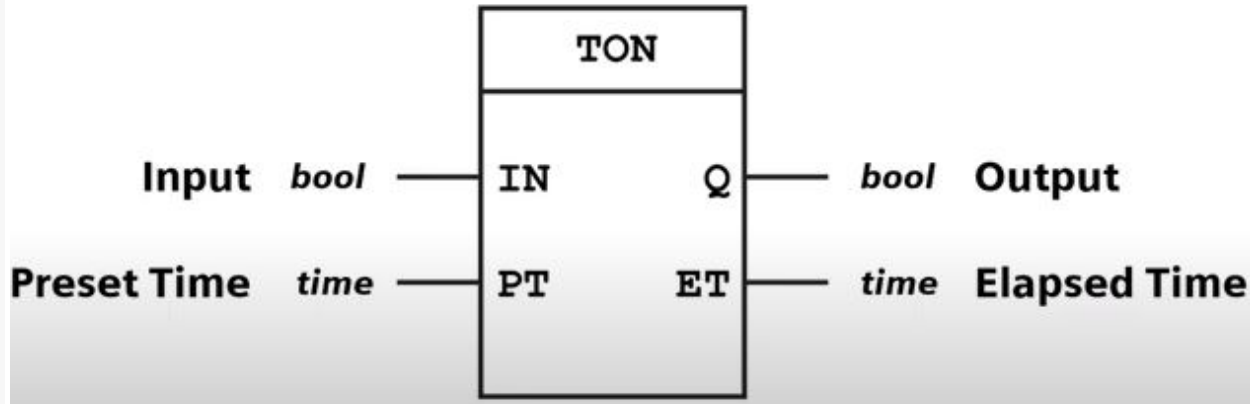
Ejercicio 1





Ladder: Temporizadores

- **TON (Timer ON-Delay) : Temporizador**
 - Acumula tiempo cuando la instrucción está activada.
 - 2 segundos= 2000 en preset



- **PT:** Preset Valor preestablecido de conteo (milisegundos)
- **Q:** Bit que indica que la operación de conteo ha finalizado
- **ET:** Intervalo de tiempo transcurrido



Preguntas?

- ▶ yenriquez@unicauca.edu.co
- ▶ Oficina: 432