

# AUTOMATIZACION

## GUIA DE TRABAJO 5

DOCENTE: VICTOR HUGO BERNAL

### UNIDAD No. 3

#### OBJETIVO GENERAL

Realizar una introducción a los controladores lógicos programables

#### OBJETIVOS ESPECIFICOS:

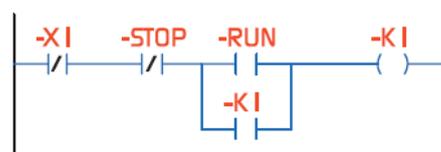
- Reconocer los diferentes lenguajes de programación para un PLC
- Reconocer la programación y la utilización de temporizadores y contadores.

#### Lenguajes de Programación

Los lenguajes de programación son necesarios para la comunicación entre el usuario (sea programador u operario de la máquina o proceso donde se encuentre el PLC) y el PLC. La interacción que tiene el usuario con el PLC la puede realizar por medio de la utilización de un cargador de programa (ladder Program) también reconocida como consola de programación o por medio de un PC (computador Personal). Tenga en cuenta que: En procesos grandes o en ambientes industriales el PLC recibe el nombre también de API (Autómata Programable Industrial) y utiliza como interface para el usuario pantallas de plasma, pantallas de contacto (touch screen) o sistemas SCADA (sistemas para la adquisición de datos, supervisión, monitoreo y control de los procesos), cuyo contenido no serán presentados ni tenidos en cuenta en este curso.

#### Clasificación de los Lenguajes de Programación:

Los lenguajes de programación para PLC son de dos tipos, visuales y escritos. Los visuales admiten estructurar el programa por medio de símbolos gráficos, similares a los que se han venido utilizando para describir los sistemas de automatización, planos esquemáticos y diagramas de bloques. Los escritos son listados de sentencias que describen las funciones a ejecutar.



Lenguaje Visual

AN	-X I
AN	-STOP
(	
A	-RUN
O	-K I
)	
=	-K I

Lenguaje Escrito

Los programadores de PLC poseen formación en múltiples disciplinas y esto determina que exista diversidad de lenguajes. Los programadores de aplicaciones familiarizados con el área

industrial prefieren lenguajes visuales, por su parte quienes tienen formación en electrónica e informática optan, inicialmente por los lenguajes escritos.

### **Niveles de los Lenguajes**

Los lenguajes de programación de sistemas basados en microprocesadores, como es el caso de los PLC, se clasifican en niveles; al microprocesador le corresponde el nivel más bajo, y al usuario el más alto.

#### **Lenguajes de Bajo Nivel:**

##### **Lenguaje de Máquina:**

Código binario encargado de la ejecución del programa directamente en el microprocesador.

##### **Lenguaje Ensamblador:**

Lenguaje sintético de sentencias que representan cada una de las instrucciones que puede ejecutar el microprocesador. Una vez diseñado un programa en lenguaje ensamblador es necesario, para cargarlo en el sistema, convertirlo o compilarlo a lenguaje de máquina. Los programadores de lenguajes de bajo nivel deben estar especializados en microprocesadores y demás circuitos que conforman el sistema.

#### **Lenguajes de Alto Nivel:**

Se basan en la construcción de sentencias orientadas a la estructura lógica de lo deseado; una sentencia de lenguaje de alto nivel representa varias de bajo; cabe la posibilidad que las sentencias de un lenguaje de alto nivel no cubran todas las instrucciones del lenguaje de bajo nivel, lo que limita el control sobre la máquina. Para que un lenguaje de alto nivel sea legible por el sistema, debe traducirse a lenguaje ensamblador y posteriormente a lenguaje de máquina.

Tipos	Descripción	Nivel	Características	
			Acceso a los Recursos	Preferencias de Uso
Visuales	Utilizan los símbolos de planos esquemáticos y diagramas de bloques.	Alto	Restringido a los símbolos que proporciona el lenguaje.	Profesionales en áreas de automatización industrial, mecánica y afines.
Escritos	Utilizan sentencias similares a las de programación de computadores.	Bajo	Total a los recursos de programación.	Profesionales en áreas de electrónica e informática.

#### **Lenguajes de Programación para PLC:**

Los fabricantes de PLC han desarrollado una cantidad de lenguajes de programación en mayoría de los casos siguiendo normas internacionales, con el fin de suplir las necesidades y expectativas de los programadores. En la siguiente tabla se presentan lenguajes de uso común.

Lenguaje	Características	Ejemplos*	Tipo	Nivel
Listas	Lista de Instrucciones	IL AWL STL IL/ST	Escrito	Bajo
Plano	Diagrama Eléctrico	LADDER LD KOP	Visual	Alto
Diagrama de Bloques Funcionales	Diagrama Lógico	FBD FBS FUD		
Organigrama de Bloques Secuenciales	Diagrama Algorítmico	AS SFC PETRI GRAFSET		
Otros	Lenguajes Usados en Otras Áreas de la computación	BASIC C	Escrito	

\* Los nombres fueron asignados por el fabricante

## Niveles de los Lenguajes Específicos para PLC

### 1. Bajo Nivel:

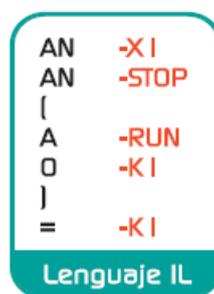
En el ámbito de programación de PLC no se utiliza directamente el lenguaje de máquina o del ensamblador. Se emplea el lenguaje de lista de instrucciones, similar al lenguaje ensamblador, con una sintaxis y vocabulario acordes con la terminología usada en PLC.

### 2. Listas:

Lenguaje que describe lo que debe hacer el PLC instrucción por instrucción.

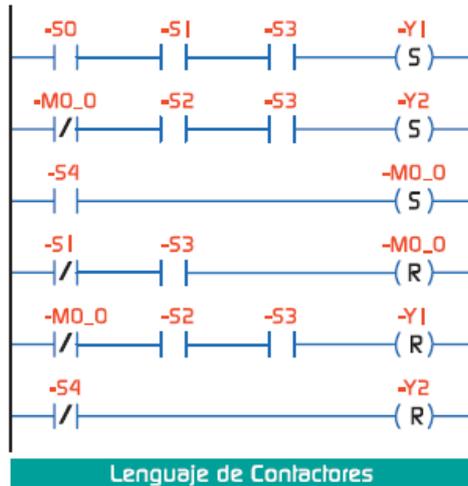
### 3. Alto Nivel:

Se caracterizan principalmente por ser visuales, aunque existen también lenguajes escritos de alto nivel.



### 4. Diagrama de Contactos:

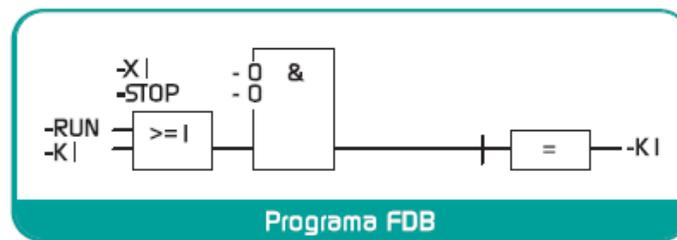
Representa el funcionamiento deseado, como en un circuito de contactores



y relés, fácil de entender y utilizar para usuarios con experiencia en lógica alambrada. En general, nos referimos a este lenguaje como LADDER (escalera), ya que la forma de construcción de su esquema se asemeja a una escalera.

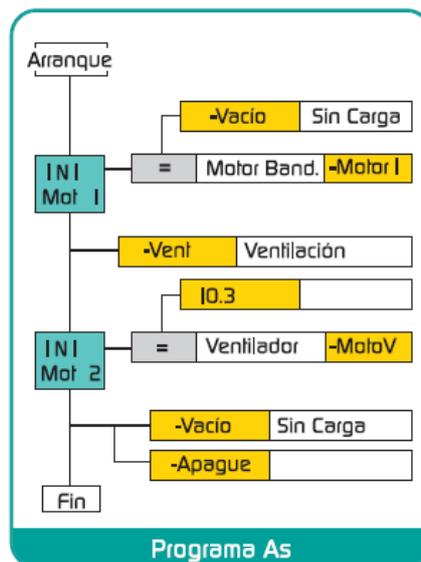
### 5. Diagrama de Bloques Funcionales:

Utiliza los diagramas lógicos de la electrónica digital.



### 6. Organigrama de Bloques Secuenciales:

Explota la concepción algorítmica que todo proceso cumple con una secuencia. Estos lenguajes son los más utilizados por programadores de PLC con mayor trayectoria.



## Reglas del Lenguaje de Plano de Contactos

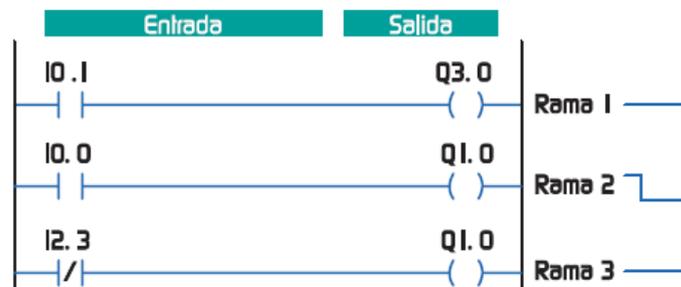
El esquema se realiza entre dos líneas o barras de alimentación dispuestas verticalmente a ambos lados del diagrama, entre ellas se dibujan los elementos del lenguaje.



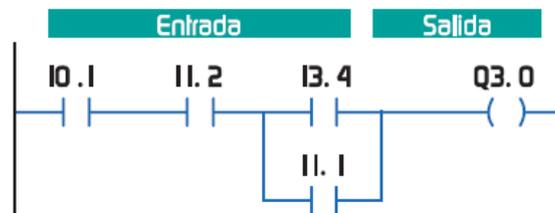
A la derecha del esquema se ubican los elementos de salida y a la izquierda los de entrada



El diagrama puede tener varias ramas o escalones.



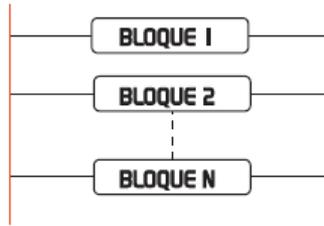
Cada rama permite ubicar varios elementos de entrada pero sólo uno de salida.



La programación en cada bloque de contactos se realiza en el orden de izquierda a derecha.

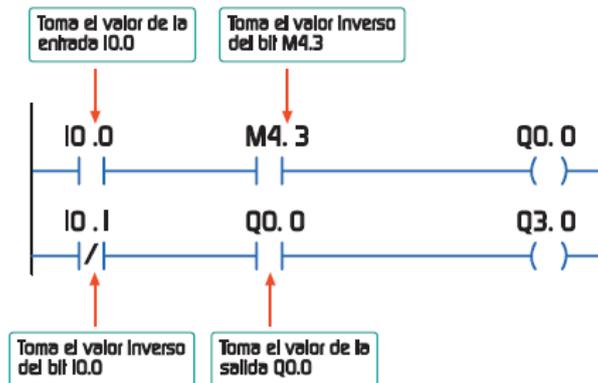


El sentido de programación de los bloques de contactos de un programa de ejecuta en el sentido de arriba abajo.



### Reglas del Lenguaje

El número de contactos que se pueden colocar en un bloque, desde el comienzo de la línea principal hasta la salida, es ilimitado. **Limitación práctica:** Anchura del papel cuando queramos sacar el programa por impresora o anchura en el ambiente de programación.



**No** se puede conectar una salida directamente a la línea principal, en estos casos se intercala un contacto cerrado de una marca o bit o relé interno cualquiera.



Con relación a los contactos, tenga presente lo siguiente:

Contactos de entrada	El número de contactos abiertos o cerrados que se pueden utilizar en un programa, por cada una de las entradas, es ilimitado, es decir que, se puede repetir el mismo número de contacto cuantas veces se quiera.
Contactos de salida	El número de salidas o bobinas de salida o relés de salida OUT es fijo, por lo que no se puede repetir un mismo número de salida. Sin embargo, el número de contactos asociados a cada una de ellas es ilimitado.

### Elementos del Lenguaje

Se clasifican en elementos de entrada y salida. Su estado es evaluado por el PLC para determinar un valor lógico, que recibe distintas denominaciones dependiendo del contexto de trabajo. A continuación, se presenta una tabla donde se relacionan las denominaciones de los contextos con las usadas en este curso (activo e inactivo).

Valores Lógicos		
Contexto	Activo	Inactivo
Informática	True	False
	Verdadero	Falso
Algebra de Boole	V	F
	1	0
Electrónica Digital	High	Low
	H	L

### Elementos de Entrada:

Los contactos, únicos elementos que se colocan a las entradas, son de tipo normal abierto y normal cerrado.

Encima del contacto se escribe la variable a la cual hace referencia. El valor lógico del contacto depende directamente del valor lógico de su variable. Para los contactos normalmente abiertos, si la variable es V el contacto también será V y, si la variable es F el contacto será F.

Los contactos normal cerrado toman el valor inverso de su variable, si la variable es V el contacto será evaluado como F y viceversa.

Contacto	Variable	
Normal Abierto	V	V
	F	F
Normal Cerrado	F	V
	V	F

Las variables a las cuales pueden referirse los contactos son:

Variable	Contacto	Ejemplo
Entradas Digitales	Entrada Digital	<b>I1.3</b>
Salidas Digitales	Valor Salida Digital	<b>Q0.0</b>
Bits en Memoria	Bit localizado en la memoria con posibilidad de ser definido por el usuario (también se conocen como relés internos, bits de estado, control de temporizadores y contadores)	

### Elementos de Salida:

A los elementos de salida, al igual que para los de entrada, se les escribe encima la variable a la cual están referidos. El valor lógico del elemento de salida es determinado por el PLC a partir de los elementos de entrada.

El elemento de salida principal se denomina Asignación o Bobina.

Las Bobinas son de tres tipos: Asignación Simple, Puesta a uno (SET) y Puesta a Cero (RESET) .

**Bobina de Asignación Simple:** Su valor lógico es igual al resultado de la combinación de los contactos en la rama. Si el resultado de la evaluación de los contactos es V entonces la bobina será V; si el resultado es F, la bobina toma el valor F.

**Bobina de Puesta a Uno (SET):** Cuando llega el valor V a esta bobina, su variable asociada se pone y mantiene indefinidamente en estado V sin importar que a la bobina llegue posteriormente un valor F. Una vez retenida la variable en el valor V, para pasarla a F será necesario el uso de una bobina de puesta a 0 (cero).

**Bobina de Puesta a Cero (RESET):** Cuando llega un valor V a esta bobina, su variable asociada se pone y mantiene indefinidamente en estado F sin importar que a la bobina llegue posteriormente un valor F. La única manera de cambiar el estado de la variable es usando una Bobina de Puesta a Uno.

## Temporizadores y Contadores

### Temporizadores

En las tablas, a continuación, se listan los diversos tipos de temporizadores disponibles en lenguaje de plano de contactos especificando su simbología y diagrama de tiempos.

Función	Símbolo		Diagrama de Tiempos
<b>Temporizador de Impulso</b>			
Disparo		Disparo	
Reposición		Reposición	
Detención		Detención	
<b>Temporizador de Impulso Prolongado «Monoestable»</b>			
Disparo		Disparo	
Reposición		Reposición	
Detención		Detención	
<b>Temporizador de Impulso de Retardo de Conexión</b>			
Disparo		Disparo	
Reposición		Reposición	
Detención		Detención	
<b>Temporizador de Retorno de Conexión Memorizado</b>			
Disparo		Disparo	
Reposición		Reposición	
Detención		Detención	
<b>Temporizador de Retardo de Conexión</b>			
Disparo		Disparo	
Reposición		Reposición	
Detención		Detención	

### Definición del Tiempo de Retardo:

El Tiempo de Retardo (**T#xx**) se establece:

En la parte superior del símbolo de disparo del temporizador, en segundos o en milisegundos. Mediante el formato **T#multiplicador.escala**, como producto entre la base de tiempo estipulada por la escala y multiplicador.

Así que

$$\text{Retardo} = \text{Base de tiempo} * \text{multiplicador}$$

Observe en la tabla los posibles valores de base de tiempo.

Valores Base de Tiempo		
Valor de Escala	Base de Tiempo	Ejemplo
0	0.01 S	T#20.0 Retardo= 0.2 S
1	0.1 S	T#15.1 Retardo= 1.5 S
2	1 S	T#30.1 Retardo= 30 S
3	10 S	T#60.3 Retardo= 600 S

## Contadores

Las opciones de programación de los contadores son:

**Asignación** 

Con éste elemento se define el nombre del contador a ser utilizado y el valor inicial de la cuenta.

**Cuenta Ascendente:** 

Un flanco de subida en la entrada del elemento hace que el valor de la cuenta se incremente en uno. El flanco de subida se define como el cambio de una señal de F a V.

**Cuenta Descendente:** 

Con un flanco de subida se hace que el valor de la cuenta descienda en uno.

**Reposición:** 

Obliga a que el contador se reinicie con su valor inicial.

La salida de un contador es un contacto cuya variable de referencia sea el nombre del contador, la variable es F mientras el valor de la cuenta sea 0 y es V si la cuenta es diferente de 0.