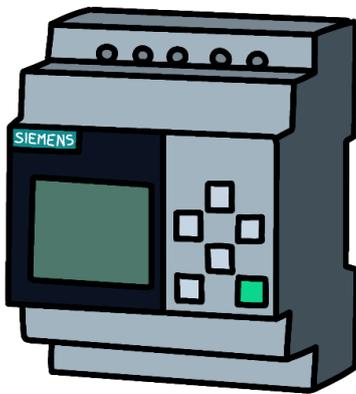


VÍDEO 3:

CONCEPTOS BÁSICOS DE TIAPORTAL:
INTERFAZ E INTRODUCCIÓN AL CÓDIGO LADDER



SIEMENS
TIA Portal

RESUMEN:

Con el tercer vídeo de esta guía se procede a estudiar la interfaz y las herramientas de TIA Portal. Resulta de gran interés ubicar los instrumentos de trabajo dentro del programa y conocer sus funciones o características antes de comenzar con los ejemplos prácticos.

TIA Portal es un software de renombre en el sector de la automatización, y a lo largo de este trabajo se va emplear para la simulación de los PLCs y HMIs virtuales.

Escanea o haz clic sobre
el código QR para ver el vídeo



Vídeo 3: Conceptos básicos de TiaPortal: Interfaz e introducción al código Ladder

by Gonzalo Carrasco Velilla is licensed under a

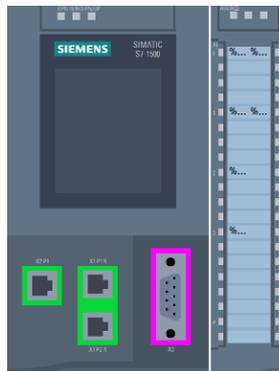
[Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional License.](https://creativecommons.org/licenses/by-nc-nd/4.0/)

TIA PORTAL

Intro

Se presenta a continuación el tercer vídeo, dedicado al diseño de la lógica Ladder y de la interfaz hombre-máquina. Para ello, se utiliza el programa de TIA Portal, un software profesional muy relevante en el sector de la automatización, cuyo uso permite explorar aspectos de la Industria 4.0 por medio de la simulación y configuración de un gran abanico de PLCs y HMIs.

La selección del controlador lógico programable se ha realizado siguiendo las “limitaciones” del programa PLCSIM Advanced, con el cual solo se pueden configurar PLCs virtuales de la gama SIMATIC S7-1500. Por ello, para este trabajo se ha escogido el PLC denominado **CPU 1516-3 PN/PD V1.8**, un dispositivo que incorpora dos puertos Profinet IO, diseñados para el intercambio de datos del controlador lógico programable, un puerto Profinet y un puerto Profibus. Los puertos Profinet IO son los que se utilizarán para conectar el PLC simulado en TIA Portal con el HMI y con el PLC virtual generado desde PLCSIM Advanced. Asimismo, este dispositivo es compatible con módulos de entradas y salidas, tanto analógicas como digitales, permitiendo adaptar el dispositivo a las características de la simulación.



PLC CPU 1516-3 PN/PD V1.8.

Por otro lado, el HMI utilizado en esta guía es el modelo **KTP 900 Basic PN**, con el que el usuario podrá supervisar las operaciones y controlar las acciones del modelo digital.

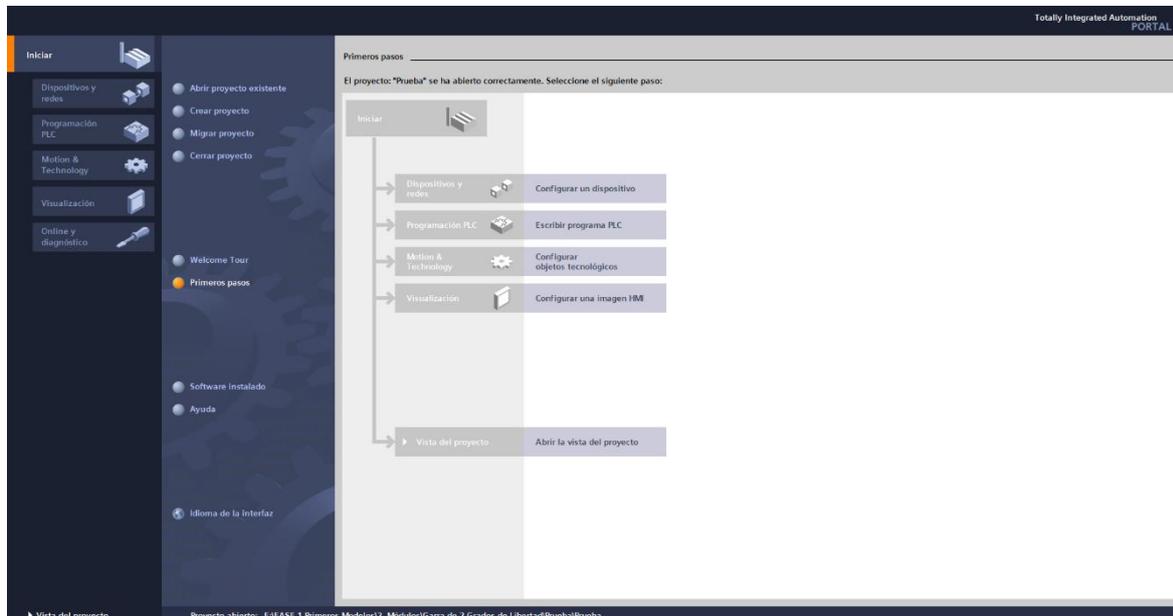


HMI KTP 900 Basic PN.

INTERFAZ DE TIA PORTAL

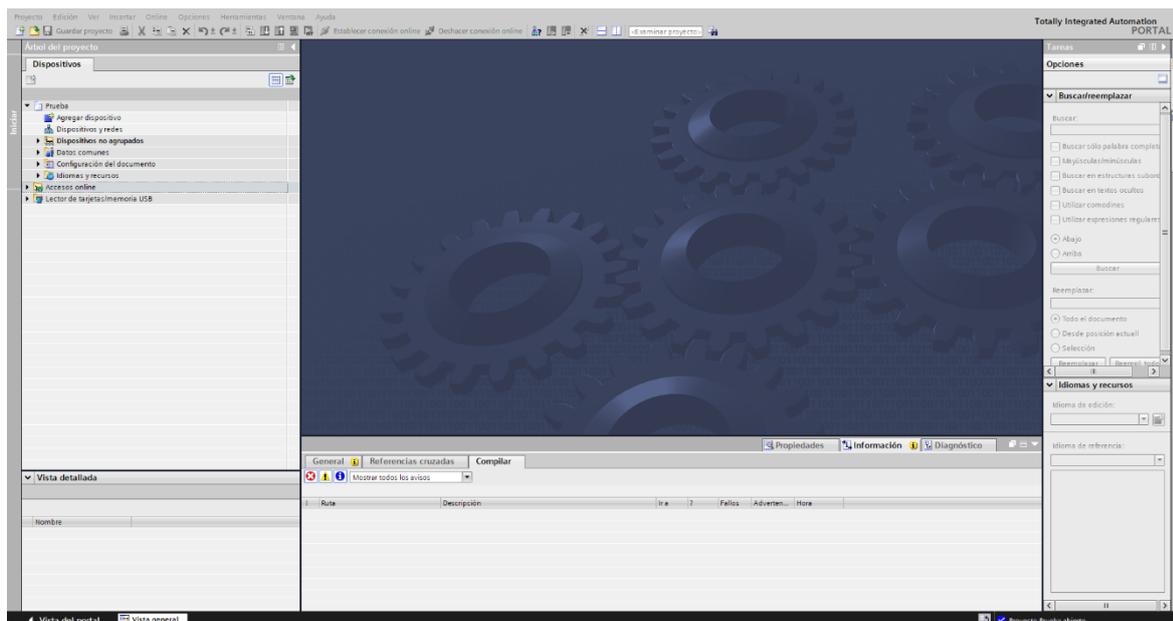
0' 41''

Cuando se abre un nuevo proyecto de TIA Portal aparece la interfaz conocida como *Vista del Portal*, que tiene el siguiente aspecto. Esta ventana cuenta con una interfaz muy bien estructurada, lo que permite navegar fácilmente entre los distintos elementos del proyecto, aunque su elevado tiempo de carga la convierte en una opción poco empleada.



Vista del portal.

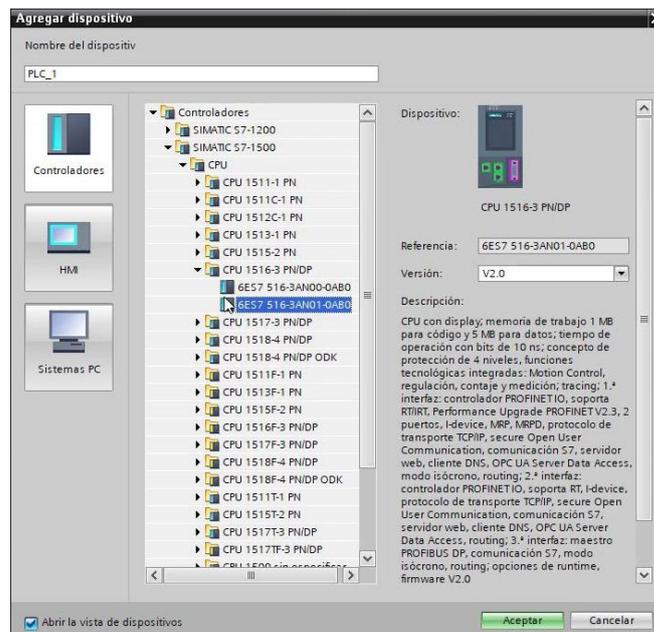
Por ello, a lo largo de esta guía se trabajará desde la *Vista del Proyecto*, a la que se puede acceder con el botón que aparece en la esquina inferior izquierda.



Vista de proyecto.

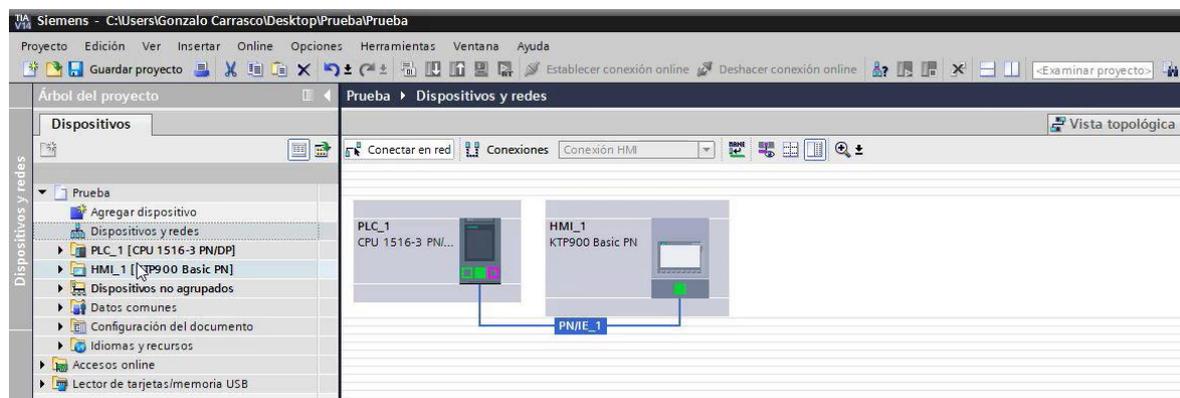
Resulta recomendable configurar el programa para que inicie los proyectos directamente desde la *Vista del Proyecto*. Esto se hace desde la ventana de “Opciones”, donde se encuentra la “Configuración” del programa. Desde aquí, en la pestaña de “General”, se procede a seleccionar la *Vista del Proyecto* como la interfaz de trabajo predeterminada.

Con la configuración del software lista, se procede a agregar los dispositivos con los que se trabajará en el proyecto. Por lo tanto, se incorpora el PLC y el HMI virtual desde la herramienta de *Agregar Dispositivos*.



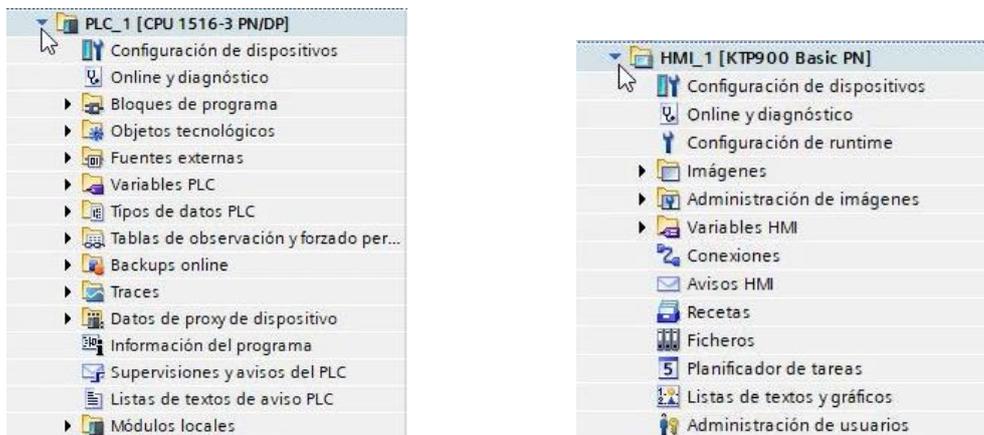
Ventana para agregar nuevos dispositivos.

Una vez los dispositivos están agregados al programa, desde la pestaña *dispositivos y redes*, se procede a establecer la conexión entre el PLC y el HMI. Esto se realiza por medio de los puertos PN/IE (*Profinet/Industrial Ethernet* o Profinet IO) de los dispositivos, que vienen representados por los recuadros verdes de cada aparato.



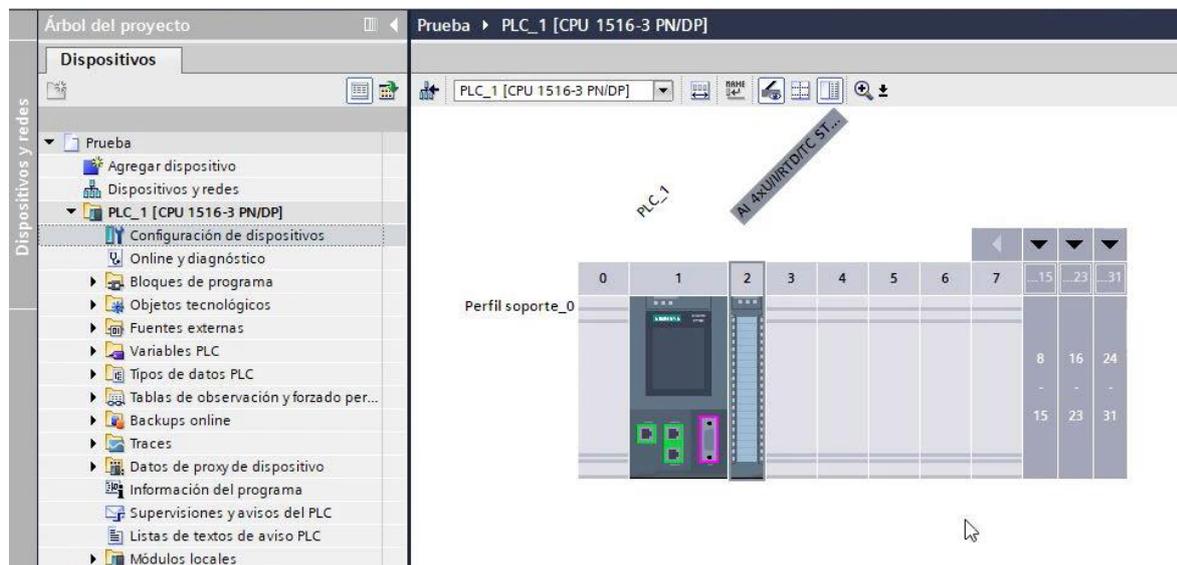
Conexión entre el PLC y el HMI.

Posteriormente, se comprueba que al añadir los dispositivos al proyecto, se crean dos carpetas correspondientes a cada uno de ellos.



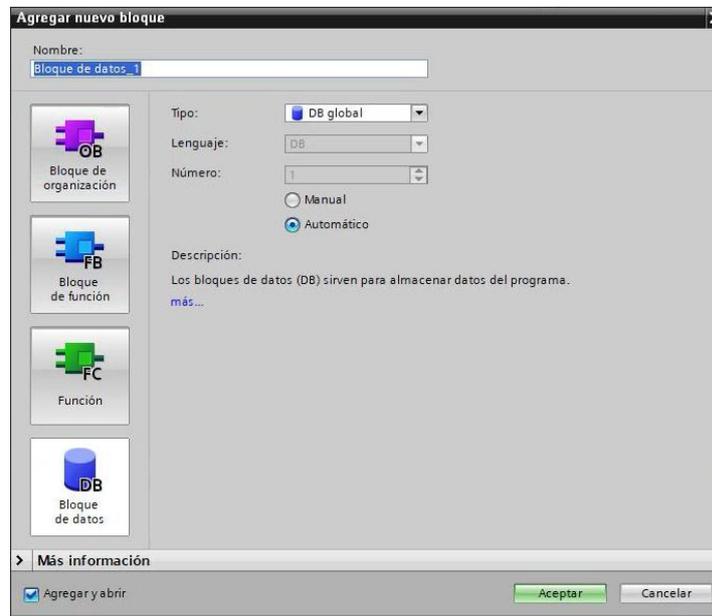
Carpetas correspondientes al PLC y al HMI.

En primer lugar, se analiza la carpeta del PLC, donde se encuentra la pestaña *Configuración de Dispositivos*. Esta ventana se utiliza para visualizar los dispositivos con los que se está trabajando, pudiendo modificarlos por medio de la incorporación de módulos de entradas digitales, módulos de entradas analógicas y módulos tecnológicos entre muchos otros. De esta forma, si se fuera a trabajar con un dispositivo real, este se podría replicar de forma virtual con todos sus componentes integrados.



Configuración del PLC.

Toda la programación se realiza desde la pestaña *bloques de programa*, donde se accede a los bloques en los que se escribe la lógica Ladder del PLC. Cuando se crea un nuevo proyecto, en esta pestaña aparecen dos elementos. Se tiene el bloque *Main*, es decir, el bloque principal de la programación, y la herramienta de *agregar nuevo bloque*, desde la cual se pueden agregar los distintos tipos de bloques disponibles. Estos son: los bloques de organización, las funciones, los bloques función y los bloques de datos.



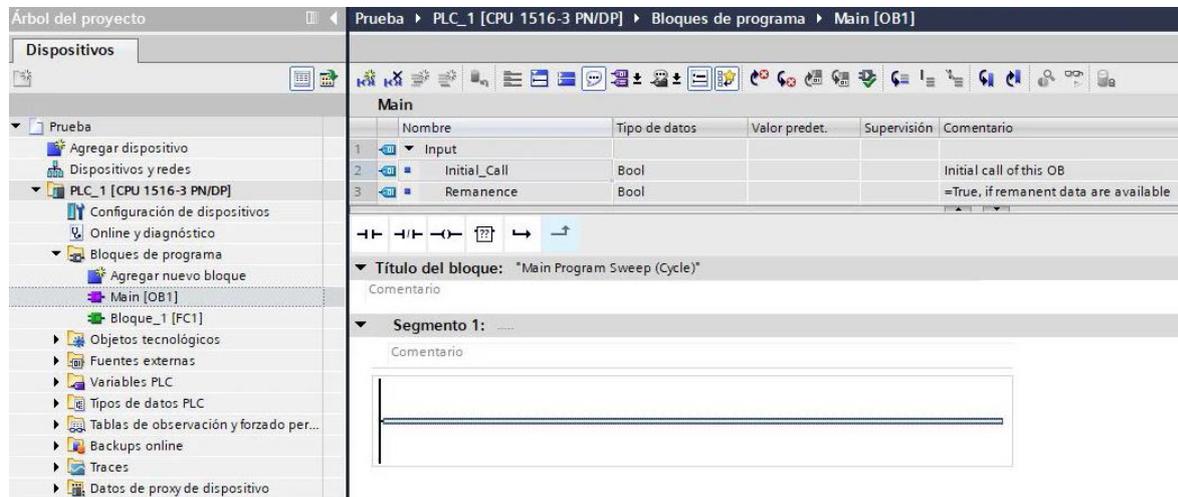
Tipos de bloques.

En primer lugar, se estudian los *bloques de organización*. Este tipo de bloques constituyen la interfaz entre el sistema operativo y el programa del usuario, por lo que son necesarios para poder cargar el código al sistema operativo del PLC.

Por tanto, esto significa que en todo proyecto se debe incorporar un bloque de organización, que en este caso es el bloque *Main*, dentro del cual se introducirían el resto de los bloques del programa. Entre estos bloques restantes, se encuentran las *funciones*, que son muy útiles para crear plantillas y subrutinas dentro del bloque de organización. La diferencia entre las *funciones* y los *bloques de función*, es que estos últimos cuentan con un bloque de datos de instancia, es decir con memoria. Estos *bloques de datos* se incorporan a la lógica para almacenar los valores de los datos que se escriben durante la ejecución del programa.

RESUMEN DEL TERCER VÍDEO

Cuando se accede al bloque Main, se pueden ver todas las herramientas para programar la lógica Ladder. Así, se tienen los contactos normalmente abiertos, los contactos normalmente cerrados y las bobinas, además de las herramientas para abrir y cerrar ramas del código.



Interfaz del bloque Main.

A la derecha de la pantalla, se encuentra la lista de instrucciones básicas con: las operaciones lógicas, los temporizadores, los contadores, los instrumentos de comparación y las funciones matemáticas entre muchas otras.

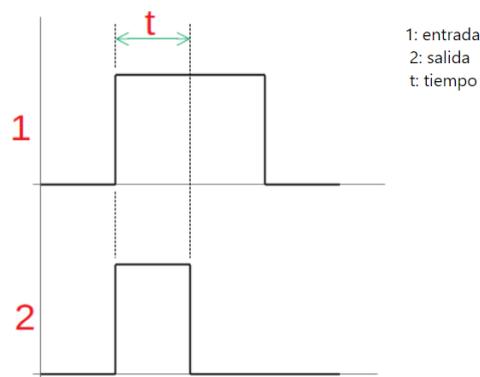


Herramientas para programar el código Ladder.

Dentro de la carpeta de *operaciones lógicas con bits*, se tienen todas las herramientas para crear una lógica de contactos, es decir, se incluyen los contactos normalmente abiertos y cerrados, las bobinas o las bobinas de SET y RESET.

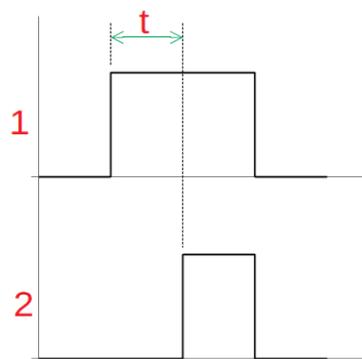
En cuanto a los *temporizadores*, existen cuatro tipos de temporizador, cada uno de ellos con sus características y ventajas.

-Temporizador TP: Se trata de un temporizador cuya salida se activa cuando detecta un flanco ascendente de entrada, y solo se desactiva una vez termina el tiempo predeterminado. Es decir, aunque la entrada pase a ser nula, la salida sigue activa hasta que finaliza el tiempo asignado por el usuario.



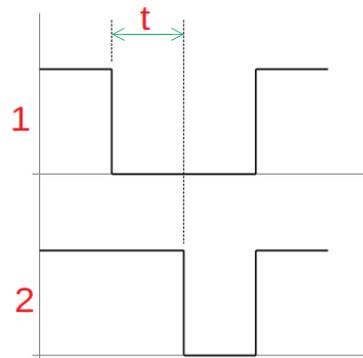
Funcionamiento del Temporizador TP.

-Temporizador TON: Se trata de un temporizador con retardo a la conexión. Su salida se activa cuando transcurre el tiempo programado desde el flanco ascendente, pero se desactiva una vez que la entrada vuelve a ser nula.



Funcionamiento del Temporizador TON.

-Temporizador TOF: Es un temporizador con retardo a la desconexión, siendo su comportamiento opuesto al del temporizador TON. En este caso, solo se desactiva una vez transcurre el tiempo determinado por el usuario.



Funcionamiento del Temporizador TOF.

-Temporizador TONR: También es conocido como acumulador de tiempo, es un temporizador que almacena el tiempo que la entrada está activada. De esta forma, cuando el tiempo acumulado alcanza el valor predeterminado, se activa la salida del temporizador. Para devolver el valor del temporizador a cero, se cuenta con una entrada RESET.

A continuación, se describen los tres tipos de *contadores*:

-Contador Ascendente: cuenta hacia arriba con cada flanco ascendente de entrada. Para devolver el valor del contador a cero, presenta una entrada de RESET.

-Contador Descendente: cuenta hacia abajo con cada flanco ascendente de entrada. Cuando el valor es menor o igual que cero, entonces se activa la salida del contador.

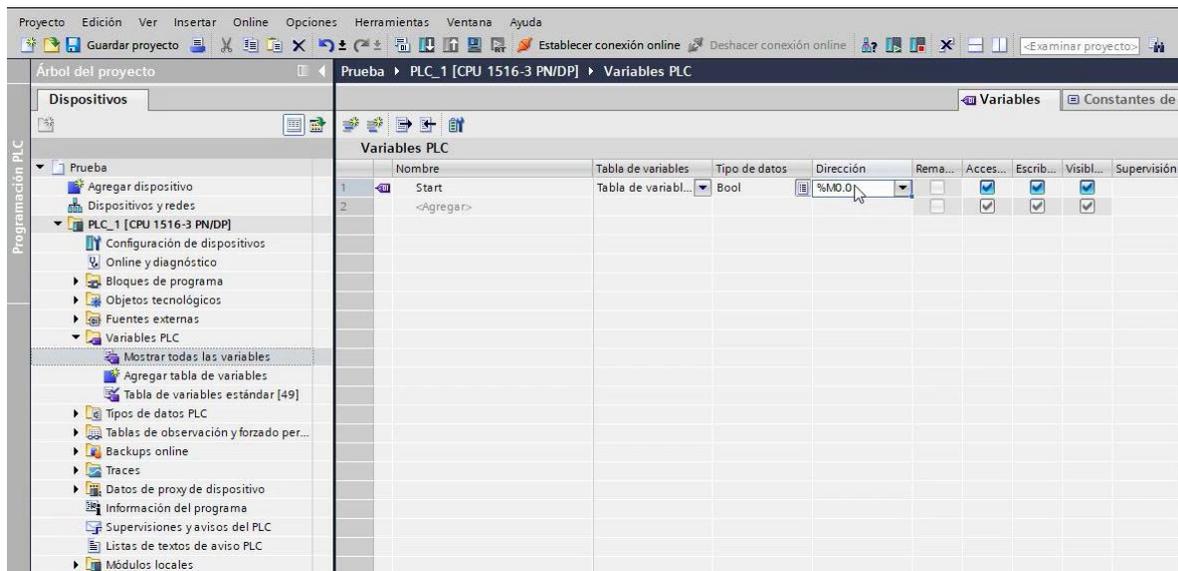
-Contador Ascendente-Descendente: combina los dos contadores anteriores. En este caso se tienen dos entradas y dos salidas, con una marca que almacena el valor actual del contador.

Posteriormente se encuentran los *comparadores*, que son muy útiles para trabajar con temporizadores, contadores y entradas o salidas analógicas. Aquí, se encuentran las funciones igual, distinto, menor o igual, mayor o igual, etc.

Finalmente, se tienen las *funciones matemáticas*, que presentan un bloque “calculadora” donde se introducen las variables de entrada, se define la operación matemática y se devuelve un resultado, que generalmente es una marca analógica.

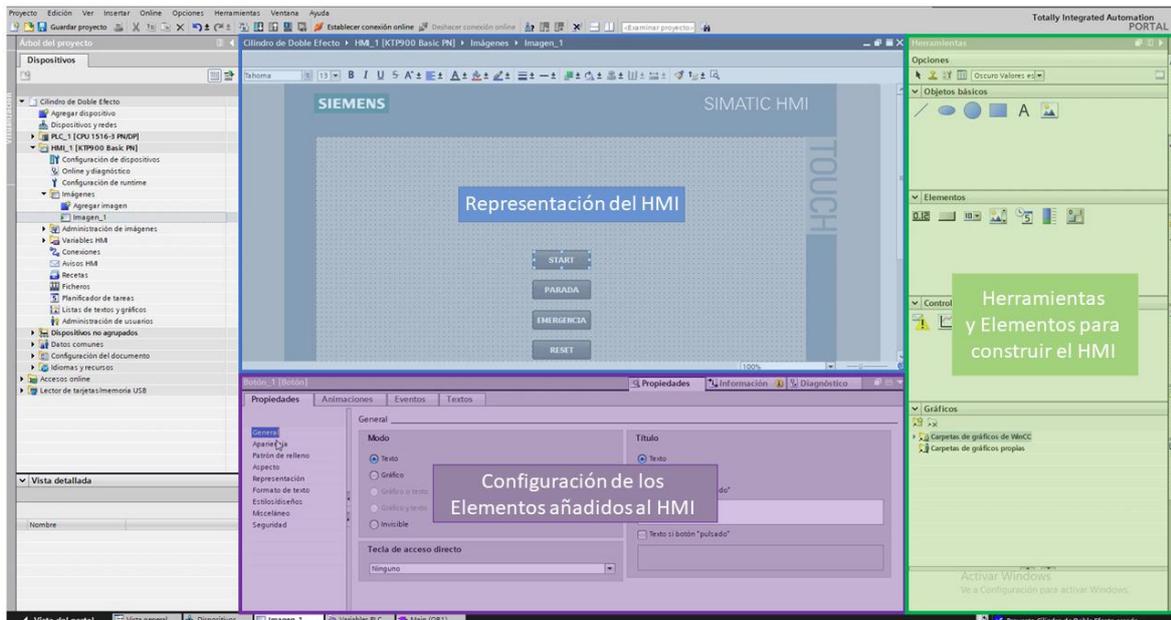
Volviendo a la carpeta del PLC, un poco más abajo, se encuentra la ventana de *variables del PLC*. Aquí, se tiene la pestaña de *Mostrar todas las variables*, en la que se recogen las variables creadas y desde la cual se pueden añadir nuevas variables al sistema.

Para crear una nueva variable se comienza asignándole un nombre, se selecciona el tipo de dato del que se trata y finalmente se establece su dirección. Las variables pueden definirse como una entrada, como una salida o como una marca. Las entradas y salidas se corresponden con las señales que entran y salen del PLC, mientras que las marcas se utilizan para almacenar datos y estados. Estas variables pueden tomar valores de 0 y 1 si el dato es de tipo booleano, o pueden almacenar datos de un byte (8bits), un Word (16bits) o un Double de 32 bits.



Variables del PLC.

Tras analizar las funciones y herramientas para programar el PLC, se procede a estudiar la carpeta del HMI. Para poder configurar la interfaz hombre-máquina, se ha de abrir la carpeta de *imágenes* y posteriormente pinchar sobre la *imagen raíz*. De esta forma se accede a una representación de la pantalla y a un conjunto de elementos con los que personalizar y diseñar la interfaz. Entre estos elementos, destacan: botones, interruptores, campos de entrada y salida, diagramas de barras, cuadros de texto e imágenes entre muchos otros.



Diseño de la interfaz hombre-máquina.

Cuando se añade uno de estos elementos al HMI, aparece en la parte inferior de la pantalla un menú con cuatro pestañas.

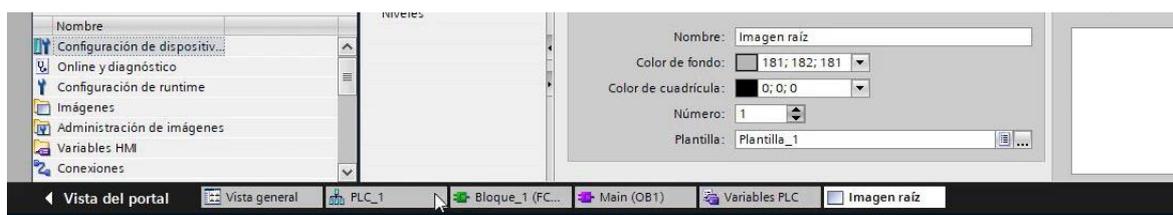
La primera de estas es la ventana *Propiedades*, donde se pueden modificar los ajustes básicos del elemento, tales como su forma o color. Desde la ventana de *Animaciones*, se modifica la apariencia y la visibilidad de los elementos en función de las variables del sistema. Además, desde esta ventana se les pueden asignar una serie de movimientos. Finalmente, en la ventana de *Eventos* se configuran los botones y se determina su mecanismo de funcionamiento. Esto se presenta con mayor detalle en el apartado **¡Error! No se encuentra el origen de la referencia..**

Descritos el PLC y el HMI, se explican los elementos restantes de la interfaz del programa. En primer lugar, se encuentra la herramienta con la cual *establecer la conexión online* entre TIA Portal y el PLC virtual. Mediante esta conexión online, se podrá transferir la lógica creada en TIA Portal al PLC, por medio de la herramienta *cargar en dispositivo*. Del mismo modo, se puede realizar la operación opuesta con la herramienta *cargar desde dispositivo*, con la cual se puede importar la lógica de un PLC al programa de TIA Portal. Finalmente, el botón de *iniciar la simulación* permite simular el comportamiento del PLC y del HMI desde el programa.



Herramientas de TIA Portal.

Por último, resaltar que todas las pestañas abiertas se pueden encontrar en la parte inferior de la pantalla, pudiendo acceder rápidamente a cada una de estas.

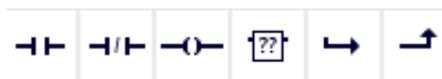


Acceso a las pestañas abiertas.

BREVE INTRODUCCIÓN A LA LÓGICA LADDER

EXTRA

La lógica Ladder es un lenguaje de programación que se utiliza para programar los controladores lógicos programables, o PLCs, utilizados en las aplicaciones industriales. Se trata de un lenguaje visual y fácil de aprender debido a su similitud con los circuitos eléctricos de relés. Por este motivo, se considera un tipo de lenguaje gráfico, pues en vez de incorporar líneas de texto y código, la programación se realiza combinando diferentes elementos gráficos, denominados símbolos. Estos símbolos están diseñados para replicar esquemas y diagramas eléctricos, pues la lógica Ladder fue originalmente concebida para técnicos y electricistas.



Símbolos de la lógica Ladder.

Cada símbolo del código Ladder representa una instrucción lógica que se puede combinar para crear las distintas funciones y tareas dentro del PLC. Entre estos símbolos destacan los contactos normalmente abiertos, los contactos normalmente cerrados y las bobinas, aunque también se pueden incluir otro tipo de elementos como temporizadores, contadores, operaciones matemáticas y comparadores entre muchos otros. Es importante destacar que, aunque se suele trabajar con variables booleanas, que adoptan valores de 0 o 1, también es posible la incorporación de variables analógicas.

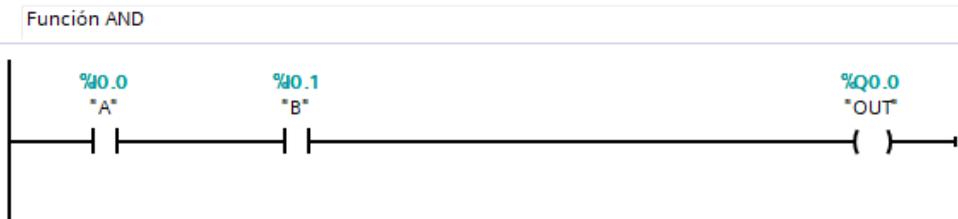
Dado que la enseñanza de la lógica Ladder no es uno de los principales objetivos de esta guía, únicamente se mostrará, a modo de ejemplo, la construcción de operaciones elementales como son las funciones: *NOT*, *AND*, *OR* y *XOR*. Para representar estas funciones por medio de la lógica Ladder, se van a asociar las variables de entrada (A & B) a contactos normalmente abiertos o cerrados, mientras que la variable de salida (OUT) se asocia a una bobina.

- **Función NOT:** La función NOT es la instrucción más simple de todas, pues se utiliza para representar el valor opuesto de una variable. Por ello, esta función únicamente requiere de la incorporación de un contacto normalmente cerrado. De esta forma, la variable asociada a la salida (OUT) siempre presenta un valor opuesto al del contacto (A).



Lógica Ladder de la función NOT.

• **Función AND:** Para que una operación lógica siga una condición AND, se debe cumplir que todos los parámetros involucrados en la operación (A & B) sean verdaderos. Por ello, en este caso, la función AND se construye a partir de contactos normalmente abiertos posicionados en serie, de tal forma que, si uno de ellos es falso, la salida (OUT) asociada a la bobina estará desactivada.



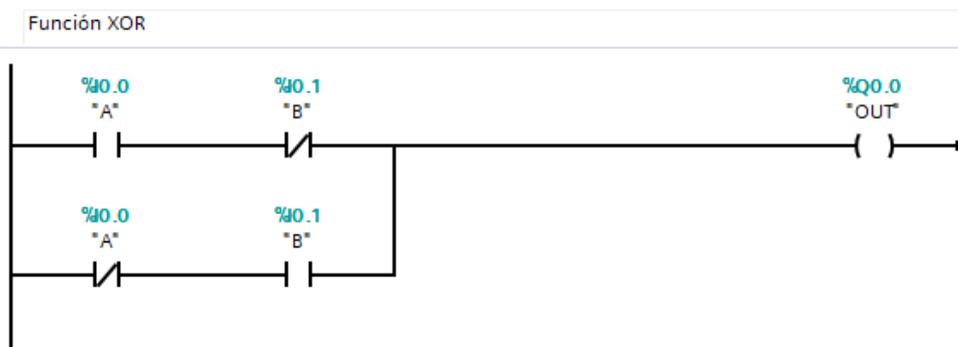
Lógica Ladder de la función AND.

• **Función OR:** La función OR sigue una lógica opuesta a la función AND, pues busca que la salida se active en cuanto una de las variables sea verdadera. Por ello, esta función se construye a partir de contactos normalmente abiertos posicionados en paralelo.



Lógica Ladder de la función OR.

• **Función XOR:** Finalmente, la función XOR se caracteriza por activar la salida (OUT) cuando solo una de las variables se encuentra activada. De esta forma, si ambas variables de entrada (A & B) se encuentran activadas o desactivadas, la salida presentará un valor nulo. Por ello, la lógica Ladder de la función XOR presenta el siguiente aspecto.



Lógica Ladder de la función XOR.