

INTERFACES DE USUARIO MÁS COMUNES EN LOS SISTEMAS AUTOMATIZADOS

Introducción

En cualquier sistema automatizado es necesario disponer de algún tipo de dispositivo que permita al usuario interactuar con el mismo para configurarlo, ponerlo en marcha, pararlo, etc. Desde un punto de vista teórico esta funcionalidad en el esquema general de un sistema automatizado recibe el nombre de “supervisión y explotación del sistema”. (Ver Ilustración 1)

A los dispositivos que permiten llevar a cabo la supervisión de un sistema automatizado se les conoce con el nombre genérico de HMI del Inglés “*Interfaces Hombre Máquina*”. Los dispositivos HMI permitirán dos acciones fundamentalmente:

- ❑ Por un lado representar la evolución de los distintos componentes del proceso: si se activan, si se desactivan, si están llenos o vacíos, abiertos o cerrados, etc.
- ❑ Por otro lado, brindarán al usuario algún mecanismo que le permita dar consignas de evolución al proceso, generalmente por medio de pulsadores. Estos al ser accionados por el usuario estarán indicando al sistema automatizado que debe llevar a cabo una acción determinada: abrir o cerrar, conectar o desconectar, subir o bajar, fijar un horario, etc.

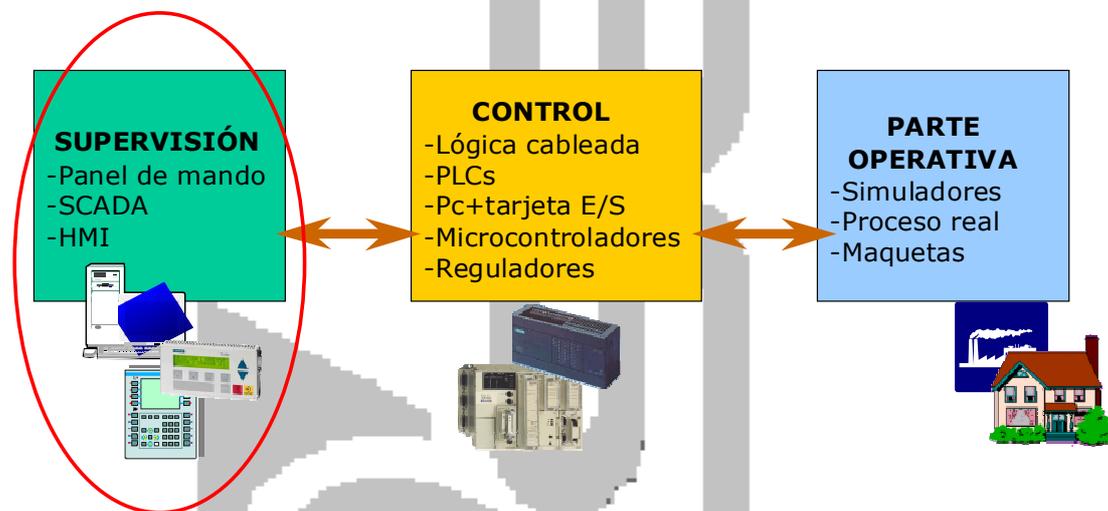


Ilustración 1. Esquema General de un Sistema Automatizado

De lo expuesto anteriormente se deduce que debe existir algún tipo de mecanismo que permita el intercambio de información entre los dispositivos HMI y el sistema automatizado. Este mecanismo será por lo general una conexión mediante algún tipo de red (bus de campo, conexión serie RS232, Ethernet, etc.) entre la parte HMI y el equipo de control. De esta forma la información acerca del estado en que se encuentra el proceso, que estará almacenada en la memoria del equipo de control, quedará accesible para ser representada en los dispositivos de supervisión. Y a su vez, las interacciones del usuario con los elementos HMI serán transmitidas al equipo de control para que este a su vez las transforme en órdenes coherentes para los preaccionadores y accionadores del proceso.

Es de destacar que el trasiego de información entre el proceso y la parte HMI siempre se producirá a través del equipo de control que será el encargado de filtrar la información para evitar incoherencias en el modo de funcionamiento del proceso.

Existe gran variedad de dispositivos que permiten implementar la funcionalidad HMI de un sistema automatizado tal y como se muestra en la Ilustración 2, a saber:

- ❑ *Paneles de operador.* Pueden ser gráficos o de texto, monocromo o a color, de membrana o táctiles. Se caracterizan por tener una pantalla que permite representar la situación en la que se encuentran los componentes del proceso. Para la interacción con el usuario suelen presentar algún tipo de botón o pulsador real o representado gráficamente que al ser accionado envía una orden al equipo de control.
- ❑ *Teléfono.* Recientemente y sobre todo en el ámbito de la automatización de viviendas y edificios es muy común la posibilidad de emplear el teléfono, y sobre todo el móvil, como elemento HMI para recibir información sobre el estado de la vivienda, y para comandar el funcionamiento de los distintos dispositivos de la misma.
- ❑ *Aplicaciones Software.* En algunas ocasiones y dependiendo de las peculiaridades del proceso que se pretende supervisar, puede ser necesario desarrollar una aplicación software que permita implementar

la parte HMI. Serán aplicaciones con un marcado carácter gráfico y diseñadas a medida de las necesidades del usuario. Cuando esas necesidades no son muy distintas a las necesidades medias de la mayoría de los usuarios, se suele recurrir a aplicaciones SCADA (Supervisión de Control y Adquisición de Datos). Estas aplicaciones pueden ser configuradas para permitir al usuario sacar el máximo beneficio de la instalación automatizada. Otra de las razones que puede llevar a un cliente a decantarse por una aplicación llave en mano frente a la parametrización de un SCADA es el coste.

- *Internet.* Por último una posibilidad que cada día tiene más aceptación, es el empleo de Internet como medio de implementar la parte HMI. Para ello es necesario un módulo servidor web que publique la página web a la que el usuario accederá con un navegador estándar. La ventaja fundamental es la ubicuidad del cliente, al poder llevar a cabo la supervisión del sistema desde cualquier equipo con un conexión a Internet mediante un navegador cualquiera. La mayor desventaja es que requiere de algún dispositivo con la capacidad de funcionar como servidor web, y la necesidad de incrementar el nivel de seguridad para impedir los accesos indeseados.

Los más comúnmente empleados cuando el equipo de control es un PLC son los paneles de mando y operador para supervisión a pie de máquina o proceso, y los sistemas SCADA para supervisión remota o a distancia.

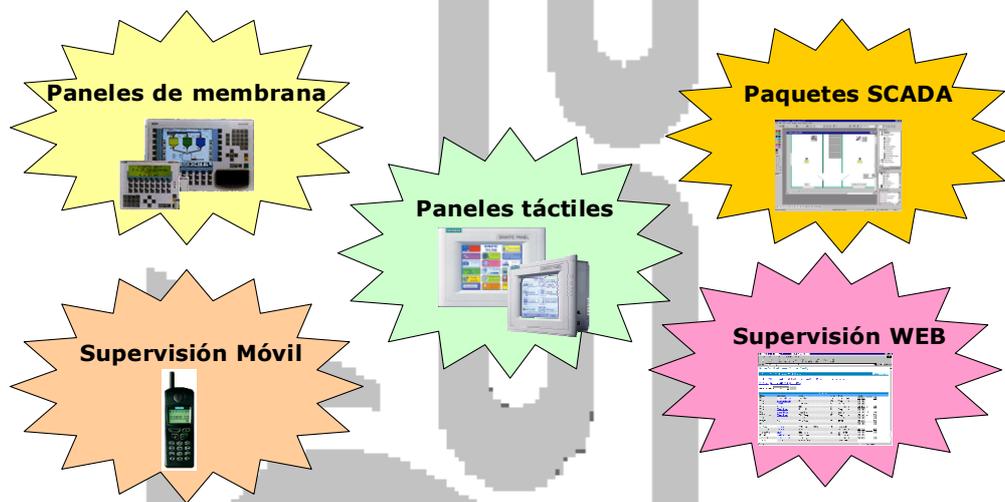
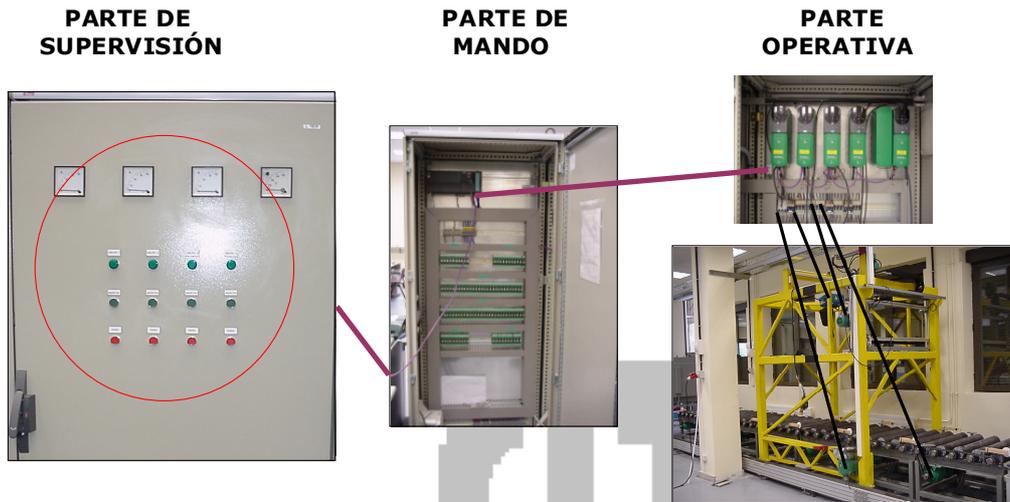


Ilustración 2: Herramientas de Supervisión y Explotación

A continuación se verán las características generales de los sistemas de supervisión más empleados cuando el equipo de control utilizado es un controlador lógico programable.

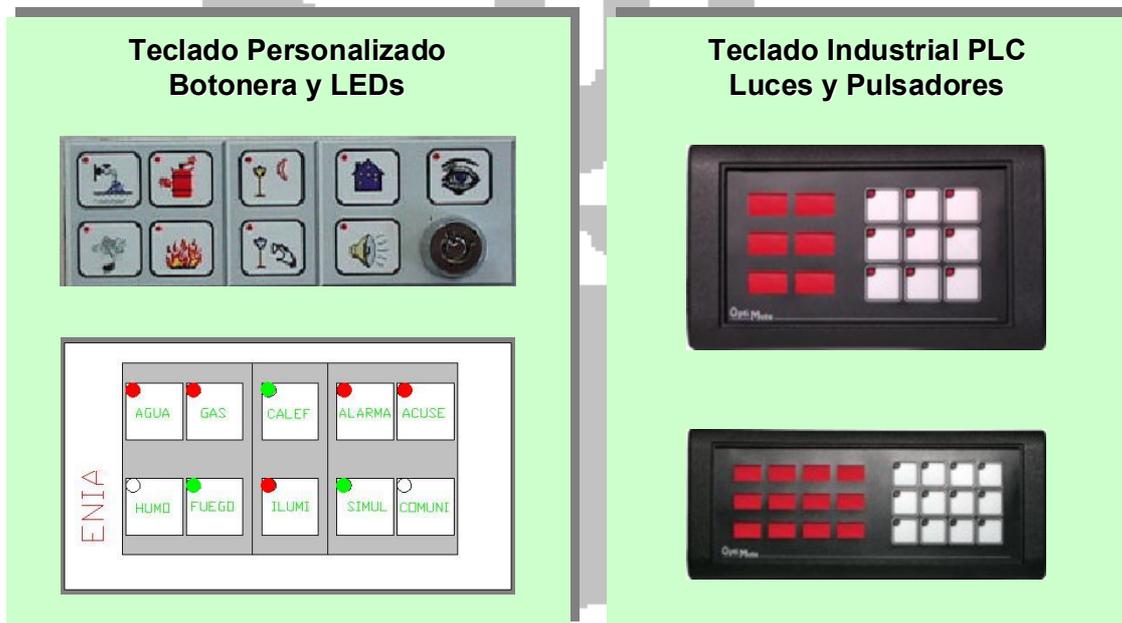
Paneles de Mando

La forma más sencilla y barata de construir un sistema HMI es mediante paneles de mando. Estos suelen estar contruidos a base de pulsadores y lámparas (ver círculo en Ilustración 3) etiquetados con nombres que hacen referencia a las partes del proceso que representan. Con pulsadores y lámparas se pueden representar únicamente señales digitales, pero cuando se requiere representar señales analógicas es necesario recurrir a instrumentos más elaborados como amperímetros, voltímetros, manómetros, etc. Este tipo de instrumentación no suele ser necesaria en la automatización de una vivienda o edificio.

**Ilustración 3: Panel de Mando**

A nivel industrial el acabado de estos paneles no es muy importante mientras su funcionalidad sea la adecuada. Sin embargo, cuando se trata de una instalación automatizada en una vivienda o edificio la estética cobra un papel relevante.

Este tipo de interface está indicado para sistemas domóticos sencillos, donde el nivel de interacción con del usuario con el mismo no es excesivo. En la Ilustración 4 en la parte de la izquierda se pueden observar dos tipos de paneles de mando para una instalación domótica con los que a priori sólo sería posible configurar las alarmas técnicas, las alarmas de intrusión y la iluminación. Además esta configuración se limita únicamente a la conexión y desconexión de dichas funcionalidades, no siendo posible por ejemplo configurar horarios, bloquear el sistema, subordinar acciones a sensores de ningún tipo, etc.

**Ilustración 4: Tipos de Paneles de Mando**

Independientemente del tipo, cada uno de los pulsadores estará cableado a una entrada digital del equipo de control. De esta forma al ser pulsado el equipo detectará su activación y actuará en consecuencia. Por su parte cada una de las lámparas o leds estará conectado a una salida digital del equipo de control. Por lo general esa salida será la misma que active o desactive el accionador correspondiente del proceso en caso de haberlo, por ejemplo una luz. En caso de no haberlo será necesario emplear una salida adicional.

Algunos fabricantes como Siemens proporcionan componentes que pueden ser combinados de la forma que el usuario desee como si de un mecánico se tratara para confeccionar el panel de mando más adecuado a sus necesidades. Para evitar al ingeniero tener que cablear cada uno de los componentes del panel, se

proporcionan un conector que permite conectar fácilmente el panel al autómatas mediante un bus de campo o a través del puerto de programación.

Estos dispositivos tienen las siguientes características:

- ❑ Preequipado con teclas de carrera corta.
- ❑ LEDs planos de varios colores de larga vida en todas las teclas de carrera corta.
- ❑ Dotado de entradas y salidas digitales adicionales para ampliaciones.
- ❑ Dotado de recortes ya perforados para elementos estándar adicionales de 22,5mm tales como pulsadores, pilotos, etc.
- ❑ Provisto de una entrada de habilitación central para impedir que se efectúen entradas.
- ❑ Con test de pilotos y pulsadores integrado.
- ❑ Equipado con una función de diagnóstico.
- ❑ La rotulación de las teclas de carrera corta y larga se realiza mediante unas tiras insertables por detrás del frontal que ofrece protección IP 65.
- ❑ Su parametrización es guiada por medio de menú en una pantalla situada en el lado posterior y dotada de un pequeño teclado.
- ❑ Es posible seleccionar los colores de los leds por ejemplo: rojo, verde, anaranjado, rojo parpadeante, verde parpadeante, anaranjado parpadeante, etc.
- ❑ Las frecuencias de parpadeo integradas son 0,5 Hz y 2 Hz para todas las salidas adicionales
- ❑ Las teclas de carrera corta que sirven de entradas digitales, pueden ser parametrizadas individualmente para trabajar como elementos biestables (interruptores).
- ❑ Es posible parametrizar la duración de impulso para teclas de carrera corta y entradas digitales hasta un máximo de 1000 ms.
- ❑ Facilidad de servicio, ya que todos los estados operativos y los avisos se visualizan en texto claro en la pantalla posterior.
- ❑ Fácil mantenimiento, puesto que no necesitan baterías
- ❑ Todos los parámetros están almacenados en un cartucho de memoria que se puede cambiar fácilmente.

1. Unpack



2. Install



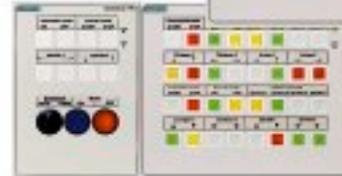
3. Connect



4. Ready!



5. Flexible expansion



Paneles de Operador de Texto

Una forma más elaborada de representar la información procedente del sistema automatizado es mediante paneles de operador de texto (ver **¡Error!No se encuentra el origen de la referencia.**). Estos paneles presentan unas pocas líneas de texto que permiten representar frases informativas en cualquier idioma, lo cual sin duda representa una mejoría frente a los paneles de mando anteriormente vistos.

Dependiendo del tipo de equipo de control empleado estos paneles de texto pueden ser dispositivos independientes que son conectados mediante algún tipo de cable a aquel, o pueden venir unidos físicamente al controlador formando una unidad como se muestra en la figura adjunta.

Estos dispositivos se caracterizan por:

- ❑ Disponer de algún tipo de cable que permite su conexión al equipo de control. Por lo general se suele emplear una conexión RS232 por su bajo coste.

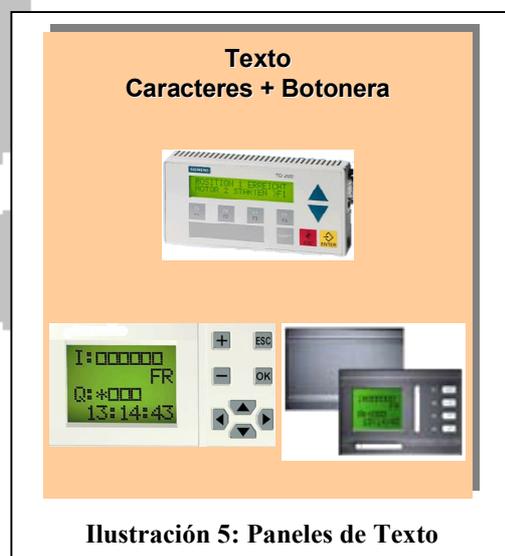


Ilustración 5: Paneles de Texto

- Presentar una carcasa robusta de plástico, con un alto grado de protección IP.
- Mayor estanqueidad al agua gracias a la ausencia de rendijas para tiras de rotulación.
- Poder ser montados sin necesidad de más accesorios, en armarios eléctricos o paneles de mando o usarlo como terminal de mano.
- Poseer una pantalla de cristal líquido retroiluminada lo que favorece su lectura incluso en condiciones de luz desfavorables.
- Tener un teclado estructurado ergonómicamente con teclas de función programables situadas debajo.
- Presentar un puerto integrado para la conexión del cable de conexión.
- Disponer de una conexión para una fuente de alimentación opcional externa opcional. Por defecto pueden tomar la alimentación del equipo de control a través del cable de conexión. Esto sólo es válido cuando la distancia es inferior a 2,5 m
- En algunos modelos es posible su conexión a través de un bus de campo.
- Posibilidad de intercambiar su carátula frontal o de poner tiras de rotulación específicas del usuario.



Ilustración 6: Panel de Texto y Controlador

Un ejemplo de este tipo de dispositivo es el panel TD 200 de Siemens cuyo aspecto se puede ver en la Ilustración 7. Para la navegación entre los distintos mensajes del TD-200 se utilizan las teclas de función. Estas son las siguientes:

- F1: Pone a 1 el bit de Marcas Mx.0
- F2: Pone a 1 el bit de Marcas Mx.1
- F3: Pone a 1 el bit de Marcas Mx.2
- F4: Pone a 1 el bit de Marcas Mx.3

- SHIFT + F1: Pone a 1 el bit de Marcas Mx.4
- SHIFT + F2: Pone a 1 el bit de Marcas Mx.5
- SHIFT + F3: Pone a 1 el bit de Marcas Mx.6
- SHIFT + F4: Pone a 1 el bit de Marcas Mx.7

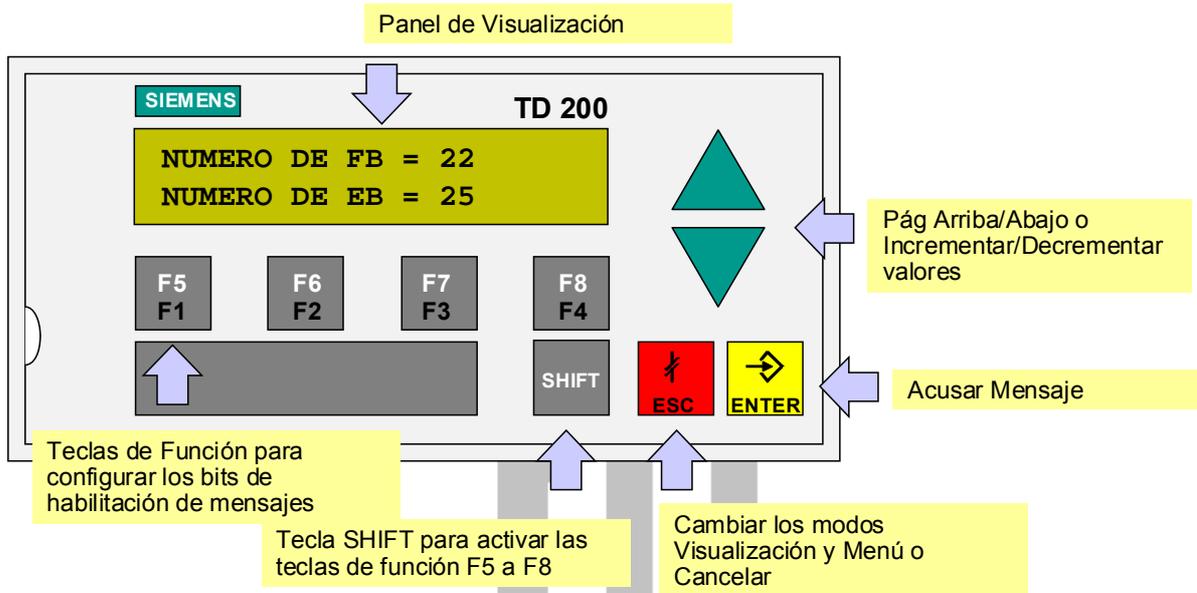


Ilustración 7: TD 200 de Siemens

Este dispositivo funciona de la siguiente forma. Una vez conectado al S7-200, el TD-200 lee el bloque de parámetros de la CPU. Entonces se comprueba si es válido el contenido de todos los parámetros. En caso afirmativo, el TD-200 comienza a consultar los bits de habilitación de mensajes para determinar qué mensaje debe aparecer, lee el mensaje de la CPU y lo visualiza a continuación.

Los mensajes que se visualizan en el TD-200 están almacenados en la CPU. Dichos mensajes contienen texto ASCII, valores intercalados e información sobre el formato.

La CPU habilita los mensajes valiéndose de una tabla de bits de habilitación de mensajes. Tiene que haber un bit de mensaje en la memoria V por cada mensaje configurado. Cuando el programa de control en la CPU decida visualizar el mensaje, activará el respectivo bit de habilitación de mensajes, y si uno de ellos está activado, el TD-200 lee el mensaje correspondiente de la CPU y lo escribe en el display.

El bloque de parámetros del TD-200 contiene la información sobre la configuración de este último y se debe crear en el área de la memoria de datos de la CPU para poder establecer un enlace entre la CPU y el TD-200.

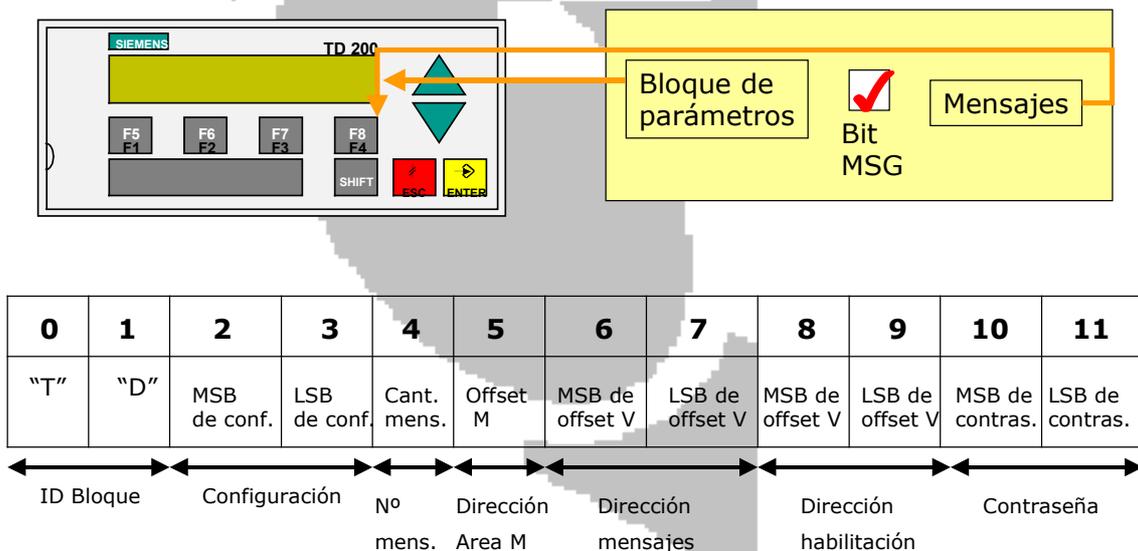


Ilustración 8: Funcionamiento del TD 200

Los mensajes tienen un formato como el que se muestra en la Ilustración 9. Los mensajes pueden contener varios campos de texto, formatos y variables. El TD 200 permite tamaños de mensaje de 20 ó 40 caracteres. El registro en el bloque de parámetros define el tamaño del mensaje para todos los mensajes.

Poniendo a 1 el bit 3 del byte de formato 0, se especifica que debe introducirse una contraseña para editar variables en el TD 200.

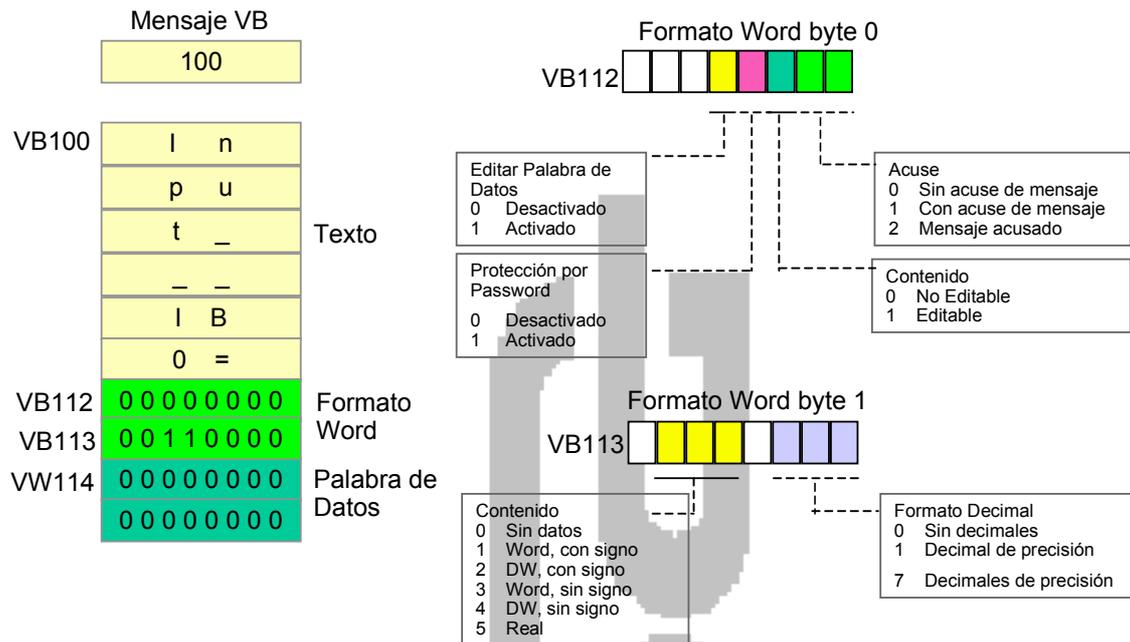


Ilustración 9: Formato de los mensajes del TD 200

Con estos dispositivos, el usuario recibe la información acerca de estado del proceso mediante mensajes de texto fijos o parpadeantes. Este podrá configurar o parametrizar ciertos componentes mediante pulsaciones sobre las teclas que posee el panel en su frontal. Estas teclas no tienen una funcionalidad predefinida sino que dependiendo del mensaje que aparezca en la pantalla su pulsación puede significar una cosa u otra.

Es por tanto necesario disponer de algún tipo de software que permita por un lado configurar qué mensajes de texto y con qué formato deben ser visualizados en cada momento, y por otro lado, definir qué significado tendrá la pulsación de una tecla determinada en cada momento.

En el caso del panel de operador TD200 de Siemens el cual puede ser empleado para la implementar el interface entre un PLC Simatic S7-00 y el usuario, software de programación del propio PLC presenta un asistente que permite llevar a cabo estas configuraciones fácilmente tal y como se muestra a continuación.



Ilustración 10: Asistente de configuración del TD200

El TD200 tiene un menú por defecto que permite configurar ciertas partes de su funcionamiento, como por ejemplo fijar la fecha y la hora. Lo primero que debe seleccionar el usuario es el idioma en el que se representará ese menú. También es posible seleccionar el tipo de fuente que se empleará a la hora de representar los mensajes en la pantalla.

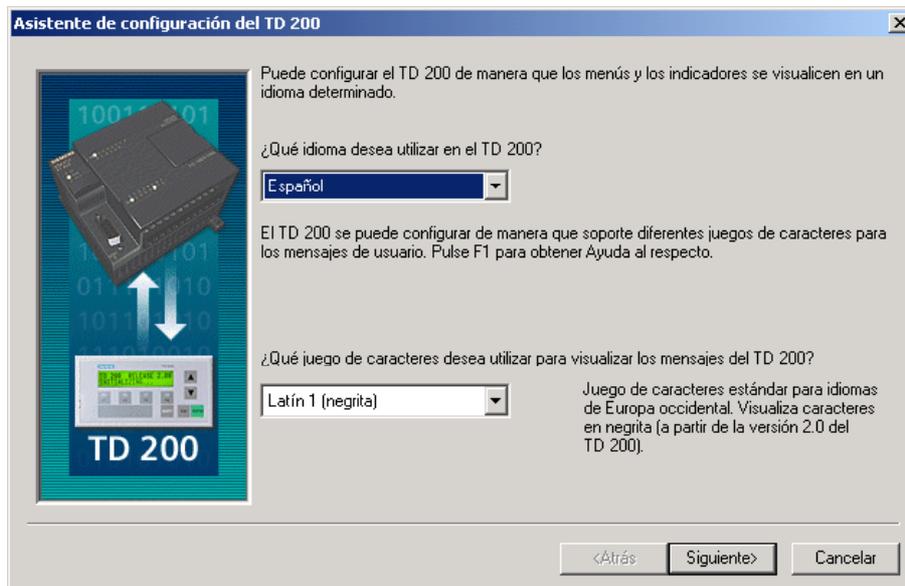


Ilustración 11: Selección del idioma

Algunas partes del menú que por defecto presenta el TD200 pueden ser añadidas o no según desee el usuario. Por ejemplo, puede haber aplicaciones en las que no sea necesario proteger el acceso al dispositivo por medio de un número clave.



Ilustración 12: Partes del menú

Cuando el operario pulsa una tecla del panel la forma en que este se comunica con el equipo de control es mediante una zona de memoria que ambos comparten. Dado que una tecla puede estar en uno de dos posibles estados, activa o no, será suficiente con disponer de un único bit en esa zona de memoria asociado a cada tecla. Cuando el operario pulse una tecla, el TD200 asignará un "1" lógico en el bit asociado con esa tecla. Cuando el operario libere la tecla, se asignará un "0" lógico. El ingeniero puede determinar qué zona de memoria utilizar para llevar a cabo esa comunicación.

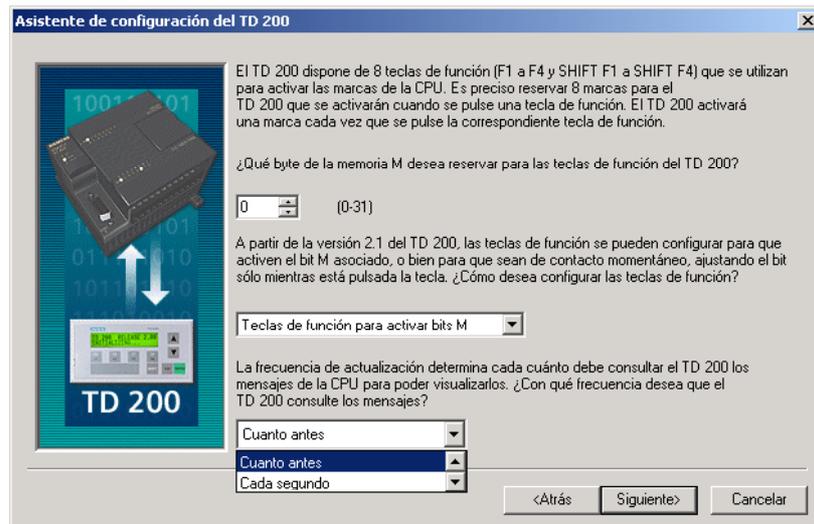


Ilustración 13: Zona de memoria compartida

El tamaño máximo de los mensajes es fijo y el ingeniero sólo puede seleccionar de entre dos posibles: 20 o 40 caracteres. Esta limitación viene dada por la longitud de cada una de las líneas de la pantalla. Existe una segunda limitación relativa al número de mensajes que se pueden llegar a visualizar, siendo 80 el número máximo.

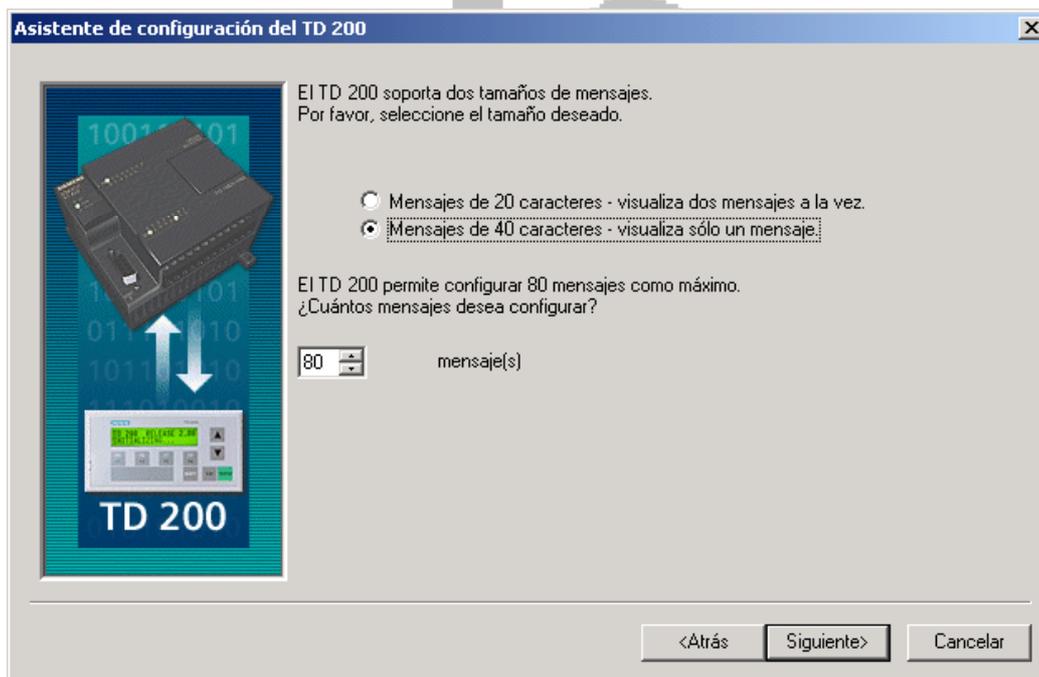


Ilustración 14: Configurando los mensajes

Es de destacar que en el caso del TD200 el cuerpo de los mensajes de texto es almacenado en la memoria del PLC, es decir, que el TD200 no dispone de memoria propia para almacenar información, es simplemente un visualizador de mensajes. Por tanto es necesario que el ingeniero defina en qué zona de la memoria del PLC están libres y pueden ser empleadas para almacenar esos mensajes teniendo en cuenta que el programa de control emplea zonas de memoria.

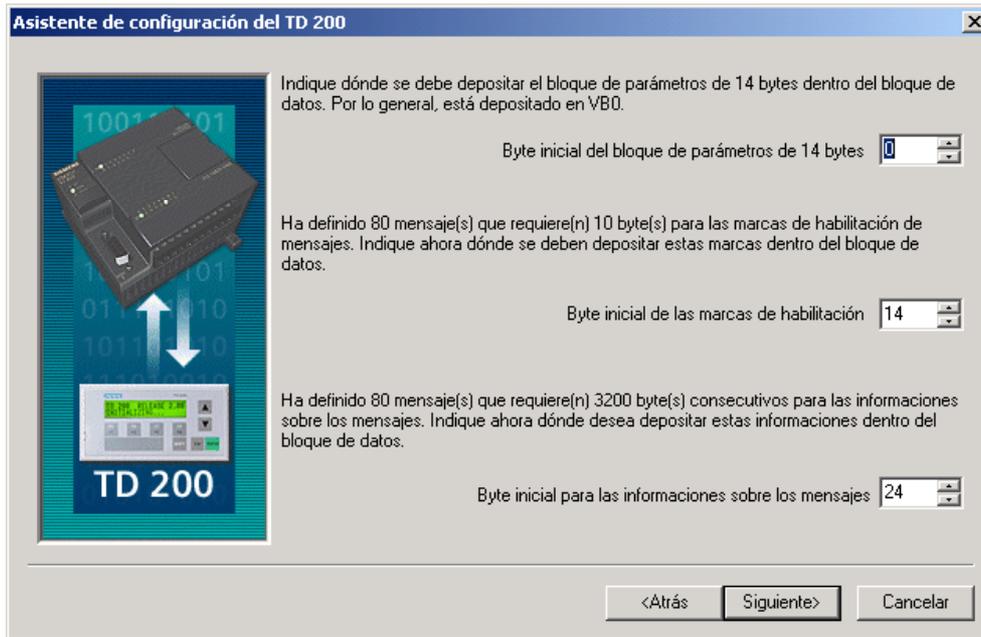


Ilustración 15: Configurando la memoria de los mensajes

Será el ingeniero el que a través del programa de control del PLC determinará qué mensaje se visualizará en cada momento tal y como se muestra en la Ilustración 16. Para lograrlo, deberá asignar un “1” en un bit de la memoria acordado con anterioridad. Al hacer esto, el visualizador accederá a la zona en de la memoria en la que está almacenado el cuerpo del mensaje y lo representará en la pantalla. Si se habilitasen más mensajes (poniendo más bits a 1) de los que se pueden visualizar en la TD 200, se muestra una flecha en la parte inferior derecha del display. En ese momento se pueden utilizar las flechas Arriba/Abajo para ir mostrando todos los mensajes.

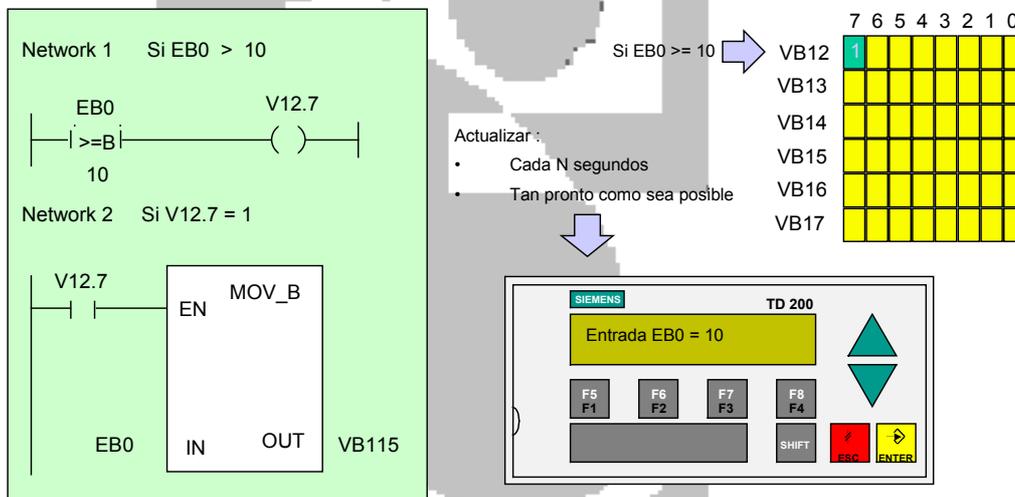


Ilustración 16: Mostrando mensajes

En lo relativo a los mensajes sólo resta por configurar el cuerpo de los mismos.



Ilustración 17: Configurando el cuerpo de los mensajes

Para cada mensaje, el ingeniero puede determinar su modo de funcionamiento, como por ejemplo:

- Definir si un mensaje es editable, es decir, si es posible introducir valores numéricos con desde el mismo.
- Determinar si la edición anterior está protegida con una contraseña o no.
- Si el mensaje debe ser confirmado o no.
- Etc.

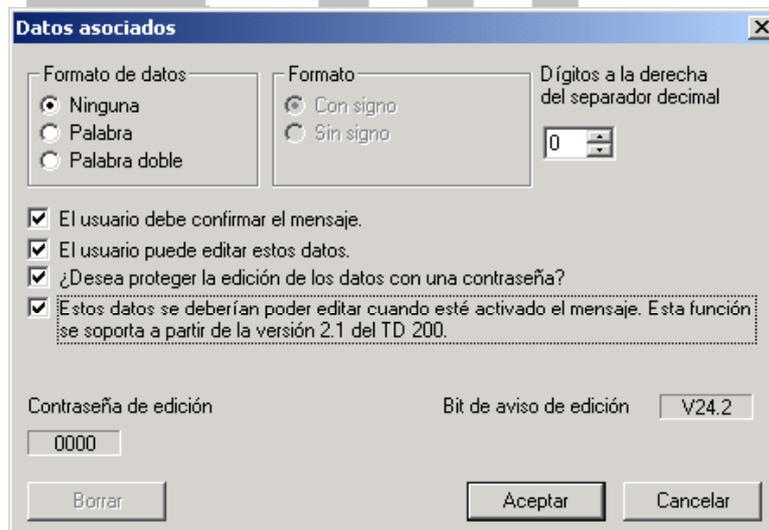


Ilustración 18: Configurando datos asociados a los mensajes

Toda esta información queda almacenada en un módulo especial del programa de control del PLC denominado bloque de datos (DB1). De esta forma cada vez que el equipo se ponga en modo run la memoria del PLC es inicializada con la información introducida anteriormente mediante el asistente y puede comenzar de inmediato el intercambio de información entre el visualizador y el PLC.

```
//  
//COMENTARIOS DEL BLOQUE DE DATOS  
//  
//Pulse F1 para obtener ayuda y un bloque de datos de ejemplo.  
//  
//BEGIN TD200_BLOCK -----  
VB0 'TD'  
VB2 16#50 //Ajustar idioma a Español, ajustar actualización cuanto antes  
VB3 16#F1 //Ajustar la visualización al modo 40; tecla INTRO V3.1;  
//Tecla arriba V3.2; Tecla abajo V3.3 ;  
VB4 80 //Ajustar el número de mensajes  
VB5 16#00 //Ajustar los bits de aviso de las teclas de función de MO.0 - MO.7,  
//teclas de función configuradas para activar bits M  
VW6 24 //Ajustar la dirección inicial de mensajes en VW24  
VW8 14 //Ajustar la dirección inicial de los bits de habilitación de mensajes en VW14  
VW10 0 //Contraseña global (si está habilitada)  
VW12 2 //Juego de caracteres = Latín 1 (negrita)  
//MESSAGE 1  
//Bit de habilitación de mensajes V14.7  
VB24 'Alarma! Tanque 1 Presión > 6 bar '  
//MESSAGE 2  
//Bit de habilitación de mensajes V14.6  
VB64 '  
//MESSAGE 3  
//Bit de habilitación de mensajes V14.5  
VB104 '  
  
//RESTO DE MENSAJES HASTA EL MENSAJE 80  
  
//END TD200_BLOCK -----
```

Ilustración 19: El DB1

La Ilustración 20 presenta un ejemplo que muestra un mensaje en el TD 200 empleado para captar del usuario un valor numérico.

El formato del dato asociado debe incluir un bit que indica si se permite la edición del valor asociado o no.(V106.4)

Cuando la entrada E 0.0 = 1 se habilita el bit de mensajes M12.7 (es puesto a 1). Esto hace que se visualice el mensaje “Val. Medida: nnn” (Mensaje 1) en el TD 200. Para modificar el valor “nnn” de este mensaje el usuario debe seguir los siguientes pasos:

1. Pulsar la tecla ENTER. El cursor se posicionará en el dígito de menor peso del dato asociado.
2. Pulsar una de las teclas de dirección. Al pulsar las teclas de dirección, el dato asociado cambia su valor.
 - Flecha ARRIBA incrementa el valor asociado en 1 unidad
 - Flecha ABAJO decrementa el valor asociado en 1 unidad

Una vez se ha obtenido el valor deseado:

3. Pulsar la tecla ENTER.
4. Pulsar la tecla ENTER de nuevo para copiar el valor modificado en el área de memoria que apunta al valor asociado (VW108).

El programa en la CPU debe ser capaz de detectar que el valor ha sido editado (V106.2). Será entonces cuando el programa procese el valor actual (MOVE_W). Una vez procesado, pondrá a 0 el bit de datos.

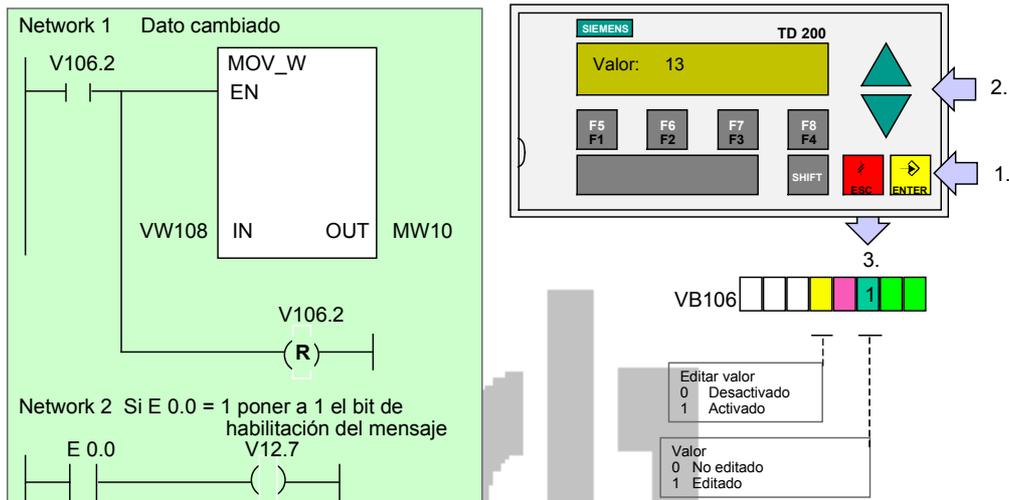
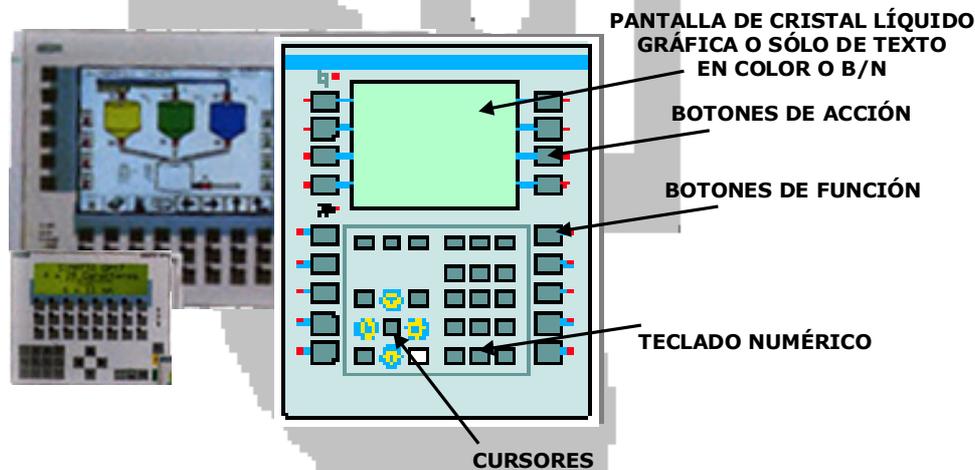


Ilustración 20: Editando mensajes

Paneles de Operador de Membrana

Este tipo de dispositivos se caracteriza por combinar dos elementos: una pantalla gráfica y una serie de teclas de función. Esta es la causa fundamental de que su precio sea sensiblemente mayor que el de los paneles de texto. Hay una gran variedad de formatos debido a los distintos tamaños de pantallas y número de teclas que presentan.

La pantalla que puede ser monocromo o a color, se emplea para representar texto y gráficos indicativos de la situación en la que se encuentra el proceso en cada momento. Las teclas, al igual que ocurría en los paneles de texto, sirven para introducir consignas de funcionamiento y pueden ser programadas de forma que su significado dependa de la pantalla que se esté mostrando.



Un ejemplo de este tipo de productos son los paneles Magelis de Telemecanique del grupo Schneider. Estos presentan las siguientes características:

- ❑ Conformidad con las normas IEC 61131-2, IEC 801-2 nivel 3, IEC 801-3 y IEC 801-4 nivel 3, IEC 68-2-6, IEC 68-2-27, UL 508, CSA.
- ❑ Homologaciones CE, UL, CSA.
- ❑ Temperatura Funcionamiento 0...+ 45 °C
- ❑ Almacenamiento - 20...+ 60 °C
- ❑ Humedad relativa 0...85% (sin condensación)
- ❑ Grado de protección de la parte frontal IP 65, según IEC 529, Nema 4. La parte posterior IP 20, según IEC 529.
- ❑ Resistencia a los choques Según IEC 68-2-27; impulso semisinusoidal 11 ms, 15 gn en los 3 ejes.

- ❑ Montaje y fijación empotrado mediante grapas de resorte (suministradas) montadas a presión (sobre panel de 1,6 a 6 mm de grosor).
- ❑ El cuadro frontal está fabricado en óxido de polifenilo con un 10% de fibra de vidrio (PPO GFN1 SE 1).
- ❑ El teclado es de poliéster endurecido con tratamiento anti UV (Autoflex EB AG).
- ❑ El cuerpo es de óxido de polifenilo con un 10% de fibra de vidrio (PPO GFN1 SE 1).
- ❑ Presenta 8 teclas dinámicas con LED y 10 teclas estáticas con LED con etiquetas modificables.
- ❑ Pantalla LCD retroiluminada monocroma o de 256 colores de 7,4", 9,5" ó 10,4".
- ❑ Definición 320 x 240 píxeles ó 640 x 480 píxeles.
- ❑ Se alimenta a 24 V DC sin aislar.
- ❑ Límites 18...30 V, ondulación máxima del 5% incluida, microcortes 1 ms como máximo.
- ❑ Protecciones contra las inversiones de polaridad y contra las sobrecargas.
- ❑ Consumo 35 W.
- ❑ Señalización mediante 1 LED control de comunicación y 1 LED indicador de teclado en uso (o de teclado táctil en uso) y 11 LED asociados.
- ❑ Sistema operativo Magelis
- ❑ Memoria RAM dinámica 2,5 Mbytes
- ❑ Memoria aplicación en tarjeta PCMCIA tipo II de 4 Mbytes (suministrada) ampliable a 8 ó 10 Mbytes.
- ❑ Número máximo 50 a 450 páginas de aplicación, de alarma o de ayuda.
- ❑ Es posible representar hasta 16 curvas en tiempo real 16.
- ❑ Recetas de producción 5.000 valores de parámetros máx., en 125 grabaciones de entradas máximo.
- ❑ Las conexiones al autómatas se realizan mediante conexión aislada RS 232 C/RS 422/485 con protocolos de comunicación transferibles.
- ❑ Presenta una ranura de emplazamiento para tarjeta de comunicación de formato PCMCIA tipo III según modelo para protocolos de comunicación tipo Modbus.
- ❑ Reloj calendario propio y acceso al reloj calendario del autómatas.
- ❑ Proporciona un relé de alarma con 1 contacto "NA" libre de potencial, máx. 0,5 A a/c 24 V.

Paneles de Operador Táctiles

Este tipo de paneles son el siguiente paso en la evolución natural de los paneles de membrana, en los cuales desaparecen las teclas de función programables, y son sustituidas por teclas representadas en la pantalla que ahora es táctil. Es decir que el ingeniero a la hora de llevar a cabo el diseño de las pantallas, no sólo debe insertar los gráficos que permitirán representar la evolución del proceso, sino que además debe incluir gráficos "especiales" que representen teclas.



Ilustración 21: Paneles Táctiles

La pantalla podrá ser en color o monocromo, pero siempre será táctil para poder recoger el punto sobre el cual el usuario está tocando con su dedo. La propia pantalla se encarga de comprobar si el usuario ha pulsado sobre la superficie de una tecla gráfica y en caso de ser cierto se encargará de representarla como pulsada mientras el dedo esté sobre la misma y como liberada cuando ya no esté. Además, se encargará

de asignar un “1” lógico en un bit de la memoria del equipo de control cuando la tecla esté pulsada, y un “0” cuando esté liberada.

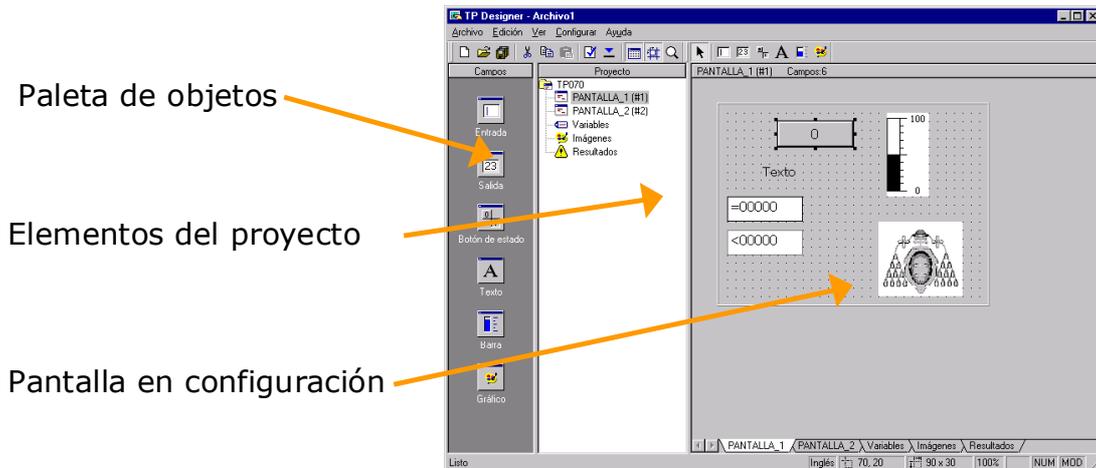


Ilustración 22: Software de configuración

Ese bit de la memoria ha debido ser previamente configurado por el ingeniero mediante el software de programación de las pantallas. Por tanto, este software debe permitir al ingeniero no sólo hacer el diseño gráfico de las distintas pantallas, sino también, establecer la forma de navegar entre las distintas pantallas, diseñar las teclas de función y configurar la zona de memoria del equipo de control que van a afectar, etc. Dado que la complejidad de estos paneles es sensiblemente superior a las de texto y a las de membrana, su sistema operativo también necesita serlo. Por ejemplo los paneles de la serie TP 170 de Siemens emplean el sistema operativo Windows CE.

En la siguiente figura se muestra el cuadro de diálogo que permite configurar el modo de funcionamiento de un botón o tecla. El botón es el objeto que más posibilidades da en lo que a su programación se refiere. Se puede seleccionar cómo se desea que funcione:

- Como Interruptor: de este modo mantendrá un aspecto cuando se encuentre pulsado y otro distinto cuando se encuentre deshabilitado. Es decir, mantiene el último estado.
- Como Pulsador: no mantiene el estado, es decir, siempre va a mostrar el mismo aspecto.

Es posible asociar una variable a un botón, de este modo, se puede conseguir que el botón muestre un aspecto u otro en función del estado de esa variable. En cuanto al aspecto, un botón puede mostrar un texto o una imagen.

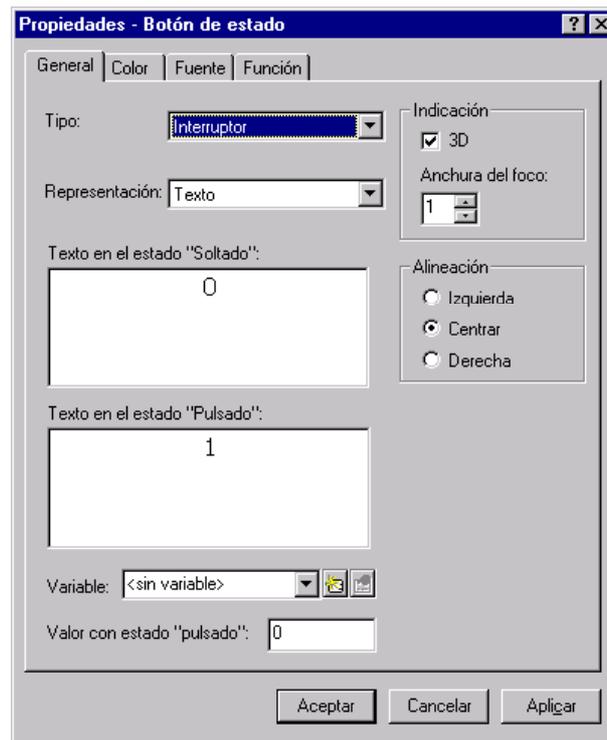


Ilustración 23: Configuración básica de un botón

Además de poder configurar el aspecto y el modo de funcionamiento, el ingeniero puede/debe configurar las variables a las que afecta. En la siguiente figura se puede ver el cuadro de diálogo que permite llevar a cabo esta configuración.

Como se puede apreciar en la Ilustración 24 es posible asociar un evento con acciones. Los eventos que un botón tiene asociados son: la selección, la deselección, la pulsación, la liberación y la modificación. A cada uno de estos eventos el ingeniero puede asociar una acción de entre las siguientes: cambiar el modo de funcionamiento del TP, invertir el valor de un bit, asignar un valor a un bit, cambiar de pantalla, etc.

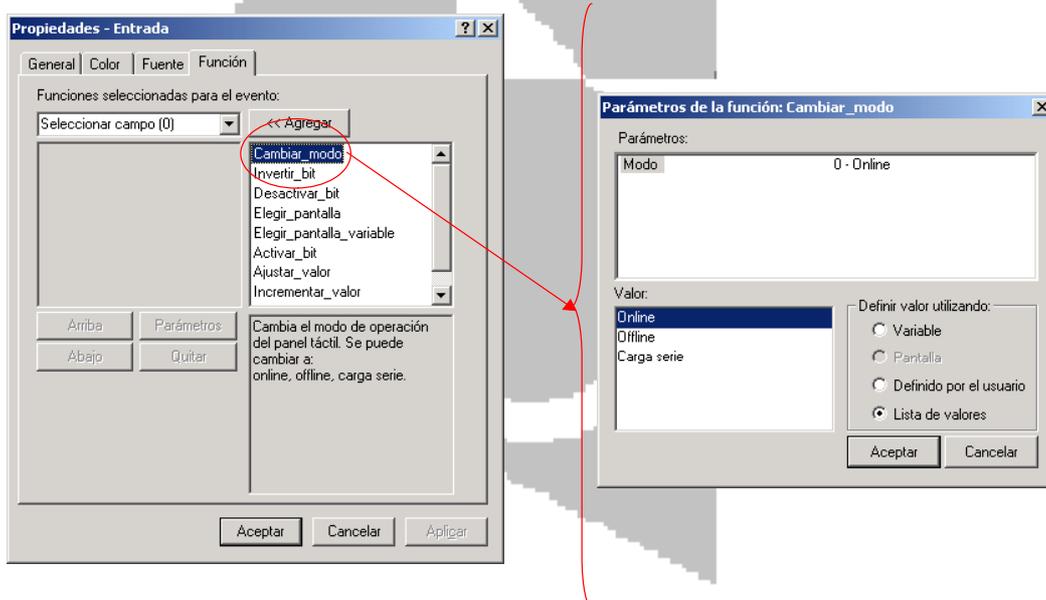


Ilustración 24: Configuración de la acción “Cambiar_modo”

Para configurar la acción “cambiar_modo” el ingeniero debe seleccionar si desea pasar el panel a modo offline u online.

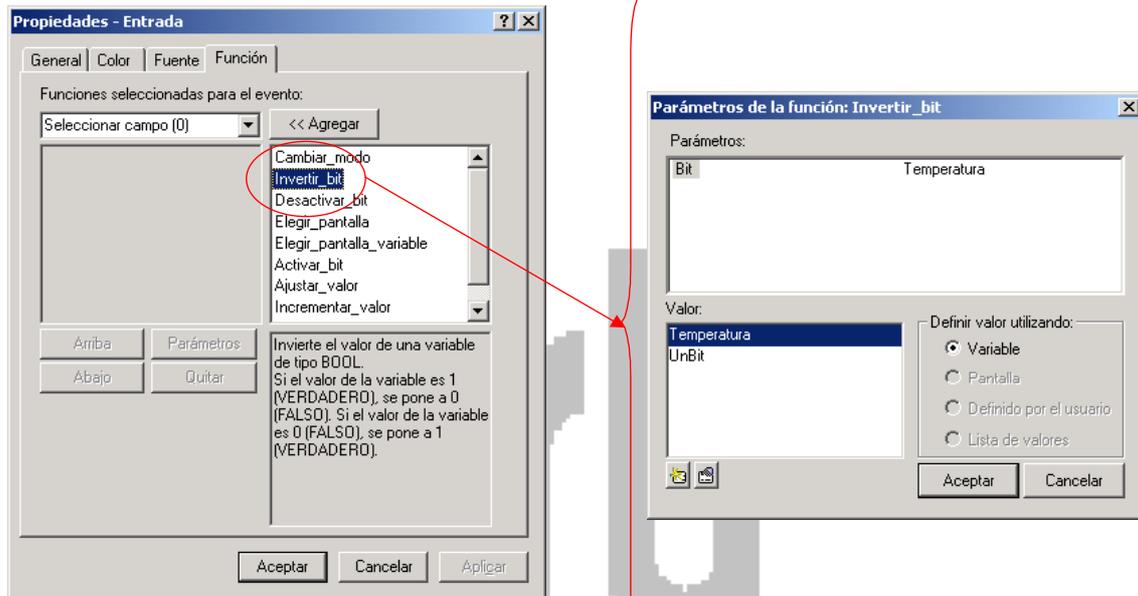


Ilustración 25: Configuración de la acción "Invertir_bit"

Si el ingeniero ha seleccionado la acción invertir bit, debe seleccionar de entre todas las variables definidas hasta el momento y que son de tipo BOOL, cuál de ellas desea que sea invertida cuando el usuario pulse el botón.

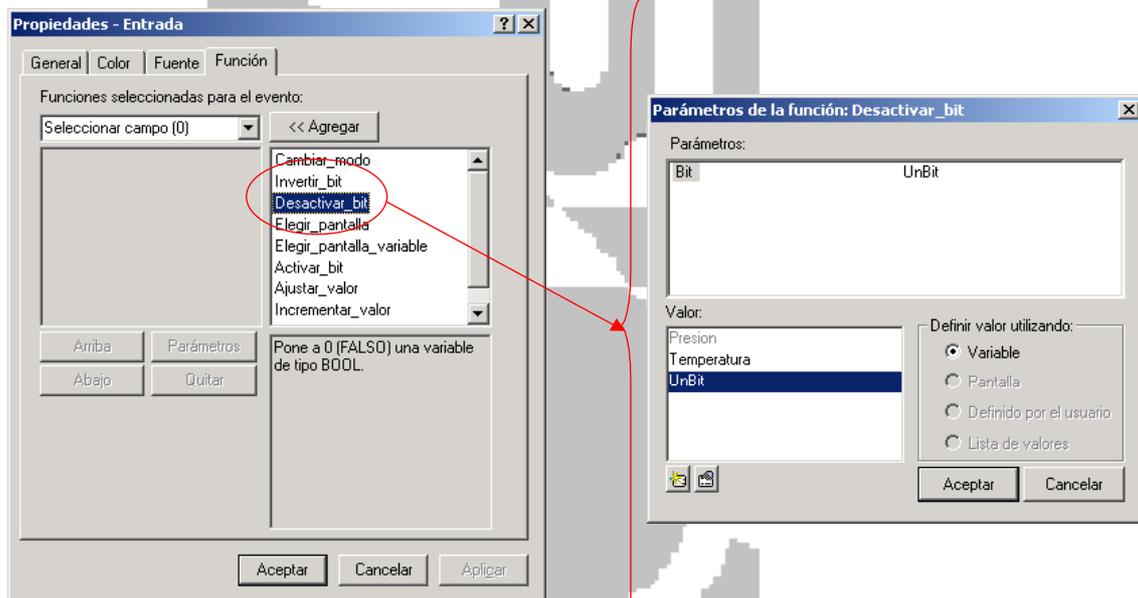


Ilustración 26: Configuración de la acción "Desactivar_bit"

El ingeniero procederá de forma análoga a lo dictado para la configuración de la acción invertir bit, cuando lo que desee sea configurar las acciones desactivar o activar bit.

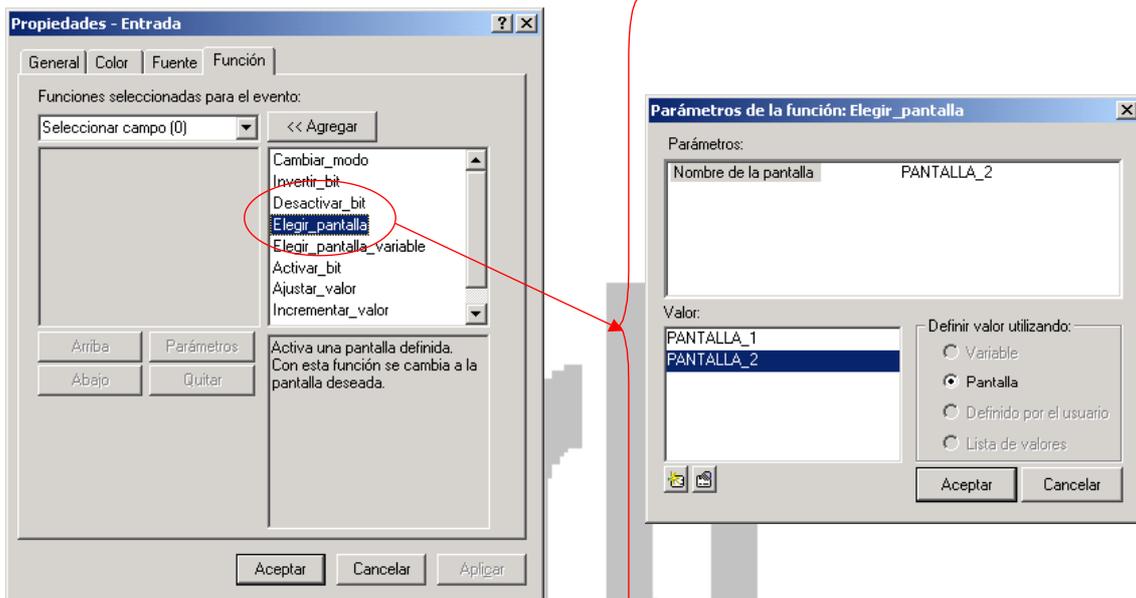


Ilustración 27: Configuración de la acción "Elegir_pantalla"

Al pulsar un botón el ingeniero podrá determinar que la acción a llevar a cabo sea cambiar la pantalla que se esté visualizando en ese momento en el panel. Para ello el ingeniero previamente deberá seleccionar de entre todas las pantallas diseñadas hasta el momento a cuál cambiar al pulsar el botón.

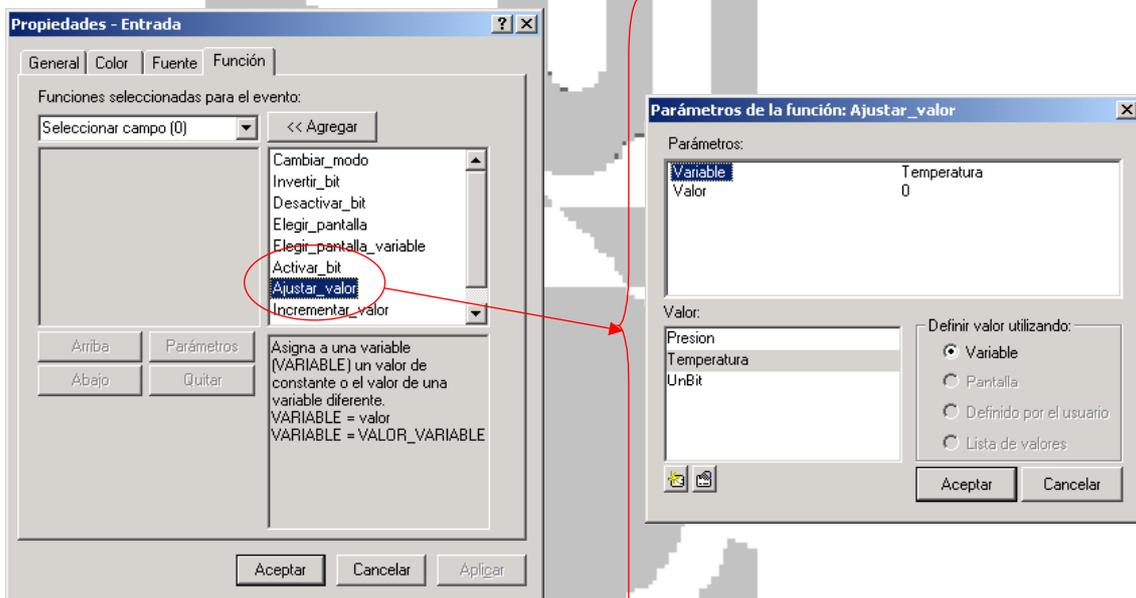


Ilustración 28: Configuración de la acción "Ajustar_valor"

Otra de las acciones que pueden ser llevadas a cabo al pulsar un botón es asignar un valor a una variable que no sea de tipo bit. Para ello el ingeniero previamente debe seleccionar de entre todas las variables de tipo entero definidas hasta el momento cuál desea ajustar. Cuando el usuario pulse el botón correspondiente aparecerá en pantalla un campo de edición que le permitirá editar e introducir un valor.

Previamente a la configuración de las acciones asociadas a un botón el ingeniero debe crear las variables a las que va a afectar. Para ello el ingeniero empleará un cuadro de diálogo como el que se muestra a continuación.

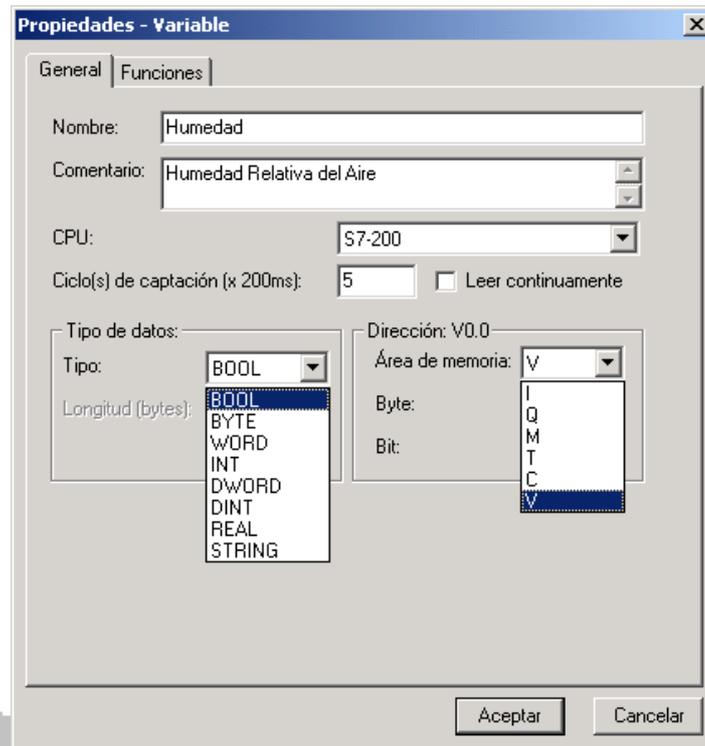


Ilustración 29: Crear variables

Siempre será posible comprobar qué variables han sido creadas hasta el momento mediante una ventana como la que se puede apreciar en la siguiente figura.

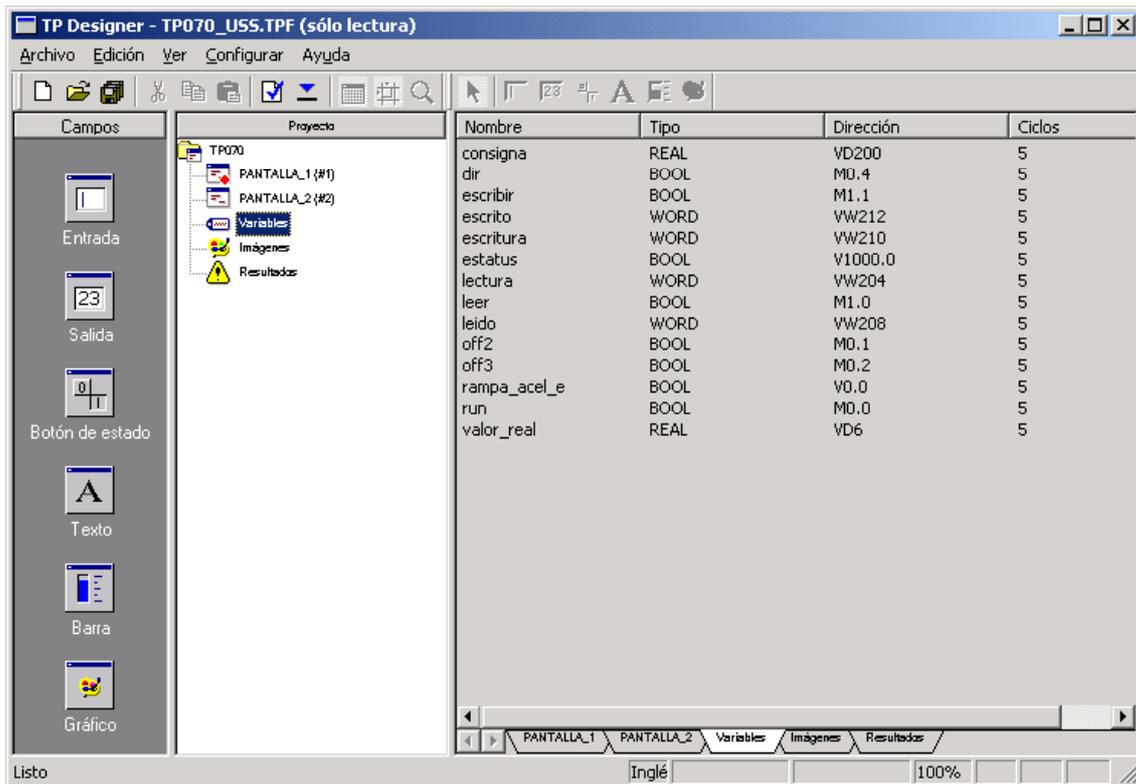


Ilustración 30: Lista de variable creadas

A continuación se enumeran las características de la gama de pantallas táctiles de la serie TP 170 de Siemens las cuales pueden servir de ejemplo para hacerse una idea de las peculiaridades de este tipo de productos.

- ❑ Gran disponibilidad de puertos de comunicación lo que permite su conexión con una gran cantidad de equipos de control no sólo de Siemens sino de muchos otros fabricantes, por ejemplo:
 - ❑ SIMATIC S7-200/-300/-400.
 - ❑ PLC software/slot-PLC de la gama SIMATIC WinAC.
 - ❑ SIMATIC S5 (se precisa convertidor TTY-RS232 opcional).
 - ❑ SIMATIC 505.
 - ❑ SIMATIC S7/WinAC también vía PROFIBUS-DP.
 - ❑ Allen Bradley
 - ❑ Mitsubishi FX
 - ❑ Télémecanique (Uni-Telway)
 - ❑ Modicon (Modbus)
 - ❑ Omron (Host-/Multilink)
 - ❑ GE Fanuc
 - ❑ LG GLOFA GM

TP 170B y OP 170B ofrecen además posibilidades de conexión con:

- ❑ SIMATIC S5/505 también vía PROFIBUS-DP
- ❑ SINUMERIK
- ❑ SIMOTION
- ❑ Impresora

Es de destacar que los drivers de comunicación necesarios para establecer el correcto intercambio de datos entre las pantallas y los equipos de control enumerados anteriormente vienen de serie con las propias pantallas, por lo que el usuario no necesita más que hacer las conexiones oportunas.

- ❑ Posibilidad de descarga remota de la configuración con detección automática de la transferencia
- ❑ Configurable con SIMATIC ProTool/Lite o superior o con SIMATIC WinCC flexible Compact o superior.
- ❑ Funcionalidad completa para tareas sencillas.

- Librería gráfica.
- Aplicable en todo el mundo, hasta 5 idiomas conmutables online (p.ej. cirílico, chino, taiwanés) y hasta 32 idiomas configurables en un proyecto.
- Servicio técnico simplificado gracias a su construcción exenta de mantenimiento y a la larga vida útil de la retroiluminación.
- Conexión integrada para impresora en TP 170B.
- Compact Flash Card, para los registros de recetas y el back-up de la configuración y los datos de sistema en TP 170B (backup/restore).
- Robusto para aplicación industrial.
- Gracias a su reducida profundidad de montaje y su robusto frontal con grado de protección IP65 los paneles son ideales para su aplicación a pie de máquina.
- La cubierta opcional para los paneles táctiles ofrece protección adicional contra suciedad/arañazos y permite protección NEMA 4/12.
- Funcionalidad convincente. Gracias a su funcionalidad adaptada a la práctica los TP 170 y TP 170B resuelven perfectamente las tareas planteadas. El ingeniero puede configurar las pantallas con elementos de tipo gráficos, barras, campos de entrada y salida de datos, botones en pantalla, pilotos luminosos, protección por contraseña, textos fijos escalables, avisos y en el TP 170B, curvas, recetas, listas de textos y gráficos, y mucho más.
- Configuración eficiente. Los paneles SIMATIC de la serie 170 forman parte de la gama de paneles homogénea y escalonada SIMATIC HMI ®. La configuración se realiza con el software SIMATIC ProTool (ProTool/Lite o superior) o con la innovada herramienta de ingeniería SIMATIC WinCC flexible (Compact o superior). SIMATIC WinCC flexible representa la evolución consecuente de la probada familia ProTool. El paso de la familia ProTool a WinCC flexible es muy simple ya que se puede adoptar la completa configuración realizada con ProTool. La nueva familia WinCC flexible ofrece un salto cualitativo importante en términos de eficiencia de configuración así como funcionalidad aumentada y mayor capacidad funcional para los TP 170.

Funciones adicionales en la configuración con WinCC flexible:

- Clases de avisos definibles (comportamiento de acuse y representación configurables).
- Gráficos y textos conmutables en función del idioma.
- Ventana de parametrización y uso de plantillas para crear pantallas.
- Histórico de avisos, ahora también en el TP 170 A.
- Conmutación entre 5 idiomas online, incluido ideogramas asiáticos y caracteres cirílicos ahora también en el TP 170A.

En el TP 170B, además:

- Avisos analógicos (avisos de rebase de límite) complementando a los avisos de bit.
- Administración de usuarios (seguridad).
- Autenticación por identificador de usuario y contraseña.
- Derechos específicos para determinados grupos de usuarios.

SCADAs

SCADA viene de las siglas de “Supervisory Control And Data Acquisition”. Traducido literalmente quiere decir “Supervisión de control y adquisición de datos” pero comúnmente se traduce por “Supervisión, Control y Adquisición de Datos”.

Se trata de una aplicación software especialmente diseñada para funcionar sobre ordenadores de control de producción, proporcionando comunicación con los dispositivos de campo (controladores autónomos, autómatas programables, etc.) y supervisando el correcto funcionamiento del proceso automatizado desde la pantalla del ordenador. Además, provee de toda la información que se genera en el proceso productivo a diversos usuarios, tanto del mismo nivel como de otros superiores dentro de la empresa: control de calidad, gestión, mantenimiento, etc.

Existen dos posibilidades a la hora de desarrollar este tipo de aplicaciones:

- Productos software llave en mano. Se trata de aplicaciones software con las funcionalidades propias de una aplicación SCADA enumeradas más adelante especialmente diseñadas para un tipo de proceso determinado.
- Aplicaciones SCADAs propiamente dichas. Son herramientas de autor que permiten al ingeniero desarrollar tantos sistemas de supervisión como desee.

La diferencia fundamental entre las primeras y las segundas es el carácter abierto y escalable que presentan las últimas frente a las primeras que son cerradas. Así las últimas suelen presentar dos modos de funcionamiento: edición y ejecución. En el primer modo el ingeniero desarrolla las pantallas necesarias para poder supervisar el proceso. En el segundo modo se establece la conexión con el sistema de control y la información adquirida cíclicamente es representada en las distintas pantallas previamente diseñadas según corresponda.

La decisión final sobre por cuál de los dos tipos decantarse será un compromiso entre el precio que se desee pagar (más caras las últimas que las primeras por lo general) y la libertad para no depender de un único proveedor. En el caso de decidir desarrollar una aplicación de tipo llave en mano, la experiencia nos demuestra que aunque nos entreguen el código fuente de la misma, siempre se dependerá en mayor o menor medida de la empresa desarrolladora del producto.

Cuando se habla de herramientas SCADA siempre se refiere al segundo tipo de productos.

Todos los programas necesarios, y en su caso el hardware adicional que se necesite se denomina en general sistema SCADA.

Un SCADA debe cumplir varios objetivos para que su instalación sea perfectamente aprovechada:

- ❑ Deben ser sistemas de arquitectura abierta, capaces de crecer o adaptarse según las necesidades cambiantes de la empresa.
 - ❑ Deben comunicar con total facilidad y de forma transparente al usuario con el equipo de planta y con el resto de la empresa (redes locales y de gestión).
 - ❑ Deben ser programas sencillos de instalar, sin excesivas exigencias de hardware, y fáciles de utilizar, con interfaces amigables con el usuario.



Ilustración 31: SCADA. Supervisory Control and Data Acquisition

Un SCADA debe cumplir tres funciones principales como se indica en la Ilustración 32:

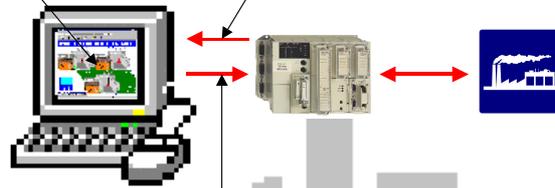
- ❑ Adquisición de datos para recoger, procesar y almacenar la información procedente del proceso.
- ❑ Supervisión, para observar desde el monitor la evolución de las variables del proceso.
- ❑ Control para modificar la evolución del proceso, actuando bien sobre los reguladores autónomos básicos (consignas, alarmas, menús, etc.), bien directamente sobre el proceso mediante las salidas conectadas. Esto último no suele ser lo más común debido a los riesgos que entraña por lo que por lo general se suele optar por enviar las consignas al equipo de control y que sea este el que actúe en el proceso.

SUPERVISIÓN

Representar en el Monitor la Evolución de la Variables del Proceso

ADQUISICIÓN DE DATOS

Recoger, Procesar, Adaptar y Almacenar la Información Recibida

**CONTROL**

Comandar o Modificar la Evolución del Proceso

Ilustración 32: Funciones básicas

Un sistema SCADA puede proporcionar funciones más especializadas gracias a la potencia de procesamiento que proporcionan los ordenadores de hoy en día, a saber:

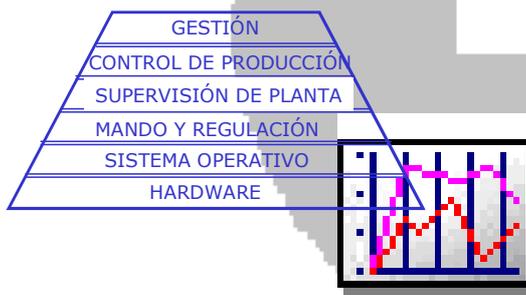
- Transmisión de información entre dispositivos de campo y otros equipos de la red corporativa de la empresa.
- Base de Datos. Capacidad de gestión de una base de datos para almacenamiento de señales del proceso.
- Representación gráfica de datos.
- Explotación e interpretación de los datos adquiridos: gestión de calidad, control estadístico, gestión administrativa y financiera.

TRANSMISIÓN DE INFORMACIÓN

Pasarela de Red entre el Nivel de Mando y Regulación y de Control de Producción

ALMACENAMIENTO

Capacidad de Gestión de una Base de Datos para Almacenamiento de los Valores de las Variables del Proceso

**EXPLORACIÓN DE DATOS**

Ayuda a la Interpretación de los Datos Adquiridos: Tratamiento Estadístico, Gestión Administrativa, Gestión Financiera, etc



Ilustración 33: Funciones avanzadas

Para poder proporcionar estas funcionalidades básicas y avanzadas es necesario que los software SCADA vengan equipados con una serie de componentes.

Configuración

Este módulo permitirá al usuario definir el entorno de trabajo de su aplicación, según la disposición de pantallas requerida y los niveles de acceso para los distintos usuarios, pudiendo establecerse contraseñas de acceso a las distintas pantallas de la aplicación final.

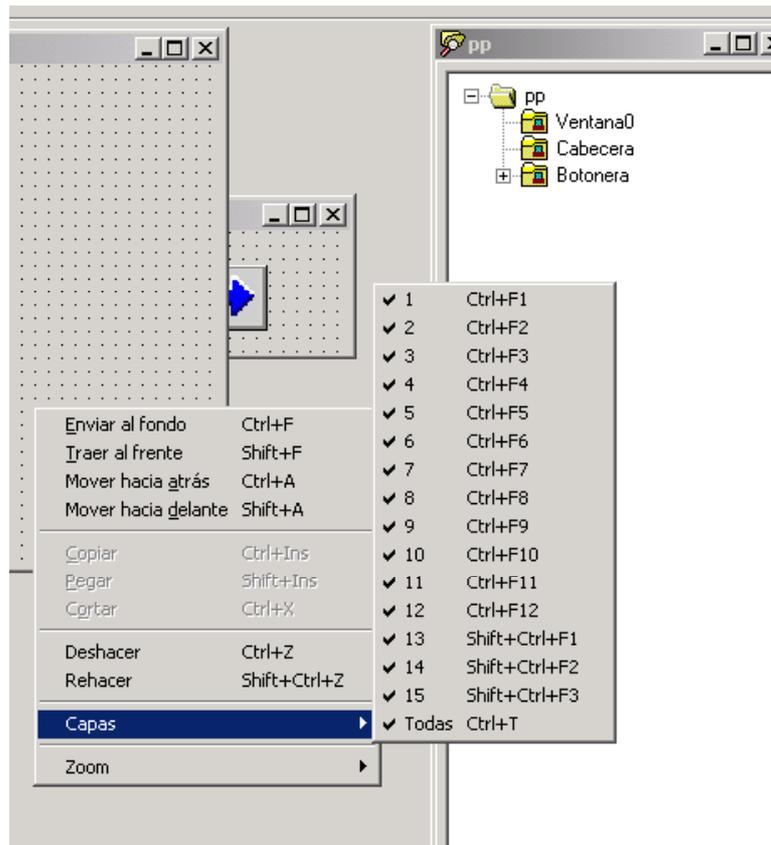


Ilustración 34: Módulo de configuración

Editor gráfico

Este módulo proporciona al ingeniero todas las herramientas necesarias para diseñar las distintas pantallas de supervisión necesarias. Estas pantallas contendrán típicamente sinópticos que en modo de ejecución representarán gráficamente la evolución del proceso en base a la información adquirida a través del equipo de control.

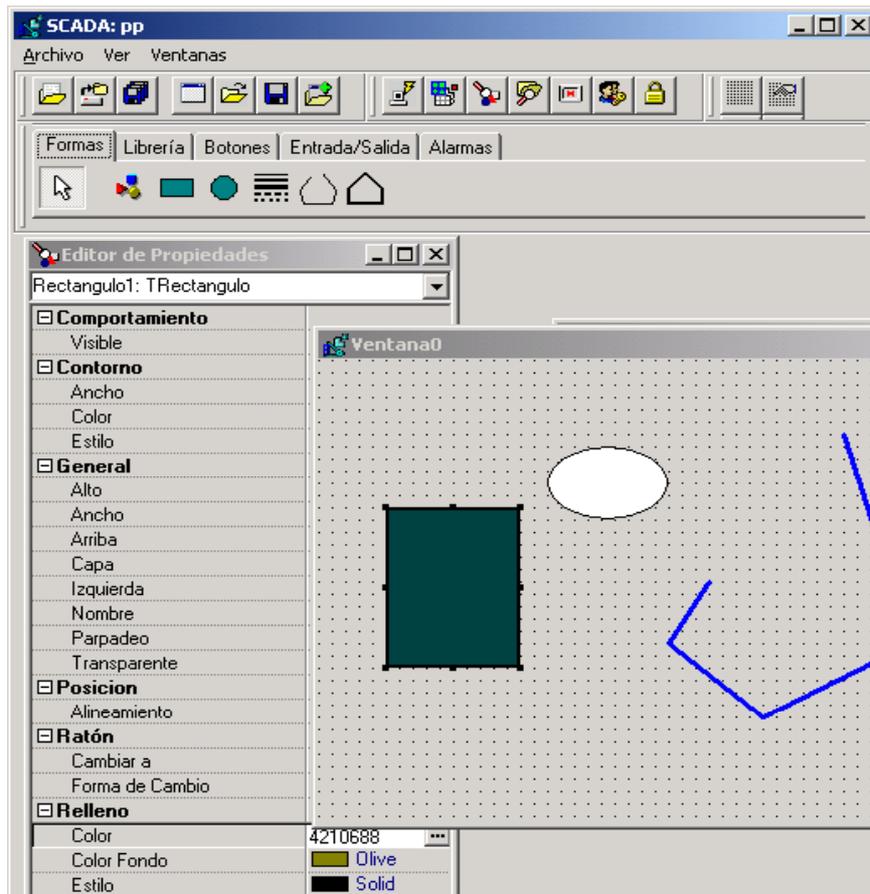


Ilustración 35: Módulo de edición gráfica

Módulo de proceso

En algunas ocasiones para poder llevar a cabo la correcta explotación del sistema automatizado es necesario elaborar acciones de control de alto nivel que no están programadas en el equipo de control. Es el caso en el que es necesario realizar cálculos numéricos complejos o la ejecución de algoritmos que requieren de estructuras de datos complicadas.

Para ello los SCADAs proporcionan al ingeniero en el modo de edición un editor de programas de tipo SCRIPT y un compilador. En el modo de ejecución estos pequeños programas son ejecutados cuando corresponde provocando la activación o no de variables que finalmente son transferidas al equipo de control para que este actúe sobre el proceso.

```
#include "apdefap.h"
void OnLButtonDown(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName, UINT nFlg)
{
    /* Archivo de cabecera que contiene las declaraciones de constantes necesarias
    para el programa de control del movimiento de las mesas de rodillos. */
    #include "rodillos.h"

    // WINCC:TAGNAME_SECTION_START
    // syntax: #define TagNameInAction "DHTagName"
    #define sentido "rod_sentido"
    #define sentido_ext "rod_sentido_ext"
    #define modo "rod_modo"
    // next TagID : 1
    // WINCC:TAGNAME_SECTION_END

    // WINCC:PICNAME_SECTION_START
    // syntax: #define PicNameInAction "PictureName"
    #define rodillos_pdl "rodillos.pdl" // declaracion de las pantallas empleadas
    // next PicID : 1
    // WINCC:PICNAME_SECTION_END

    short int aux = 0;

    // Si al pulsar se esta en modo manual se pasa a modo automatico.
    if ( GetTagBit( modo ) == MANUAL) {
        SetTagBit( modo , AUTOMATICO);
        printf(" Rodillos: modo manual -> modo automatico \n"); // salida en la ventana de diagnostico
    }

    /* Análogamente se procede en modo automático*/
    else {
        // Para evitar incongruencias de sentido, se asigna al sentido externo, el de desplazamiento actual
        aux = GetTagBit( sentido );
        SetTagBit(sentido_ext, aux);

        SetTagBit( modo , MANUAL);
        printf(" Rodillos: modo automatico -> modo manual \n"); // salida en la ventana de diagnostico
    }
}
```

Ilustración 36: Lenguaje de SCRIPTS

Comunicaciones

Uno de los pilares fundamentales de una aplicación SCADA (si propio nombre así lo indica) es el módulo de adquisición de datos o comunicaciones. Este debe estar dotado con el mayor número posible de drivers de comunicación que permitan la conexión del SCADA con la mayor variedad posible de equipos de control de distintos fabricantes.

Como el lector podrá intuir esta labor es ingente debido a la gran cantidad de posibles equipos de control que existe en el mercado y a la gran diversidad de buses de campo para comunicación industrial existentes. Debido a esto, cualquier empresa diseñadora de una aplicación SCADA se verá obligada a realizar una inversión económica brutal para tratar de garantizar la conectividad de su herramienta con el proceso. Para tratar de paliar esta situación se ha desarrollado una solución estándar llamada OPC (Ole for Process Control) cuyas características se detallan más adelante en este capítulo.

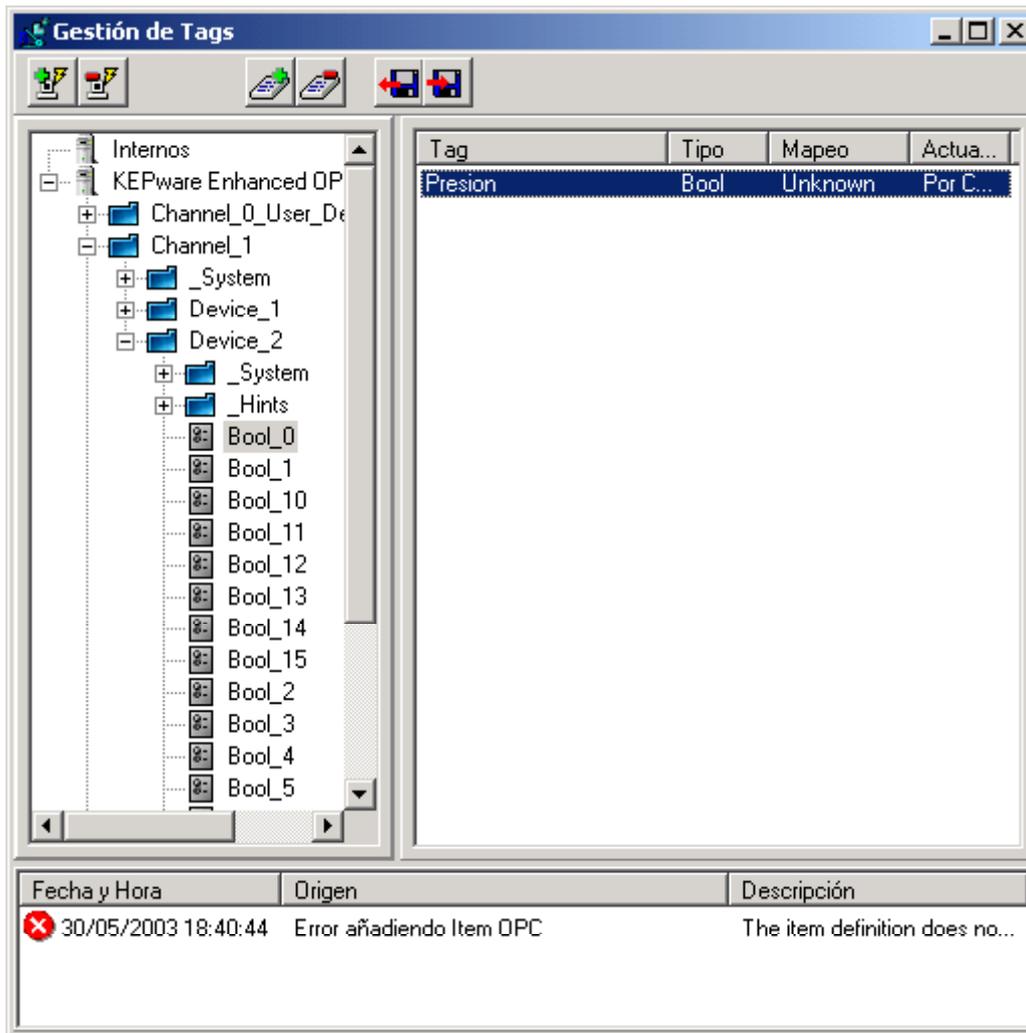


Ilustración 37: Módulo de comunicaciones

Módulo de generación de históricos de señales

Este módulo permite al usuario crear gráficos de tendencias con la evolución en el tiempo de las señales del proceso. En modo edición el usuario podrá definir qué variables desea representar en cada gráfica, el color de la línea y el estilo (sólida, punteada,...) de representación, la escala de la gráfica, etc.

En modo ejecución el eje X representará la evolución del tiempo y el Y representará los valores que van tomando las señales.

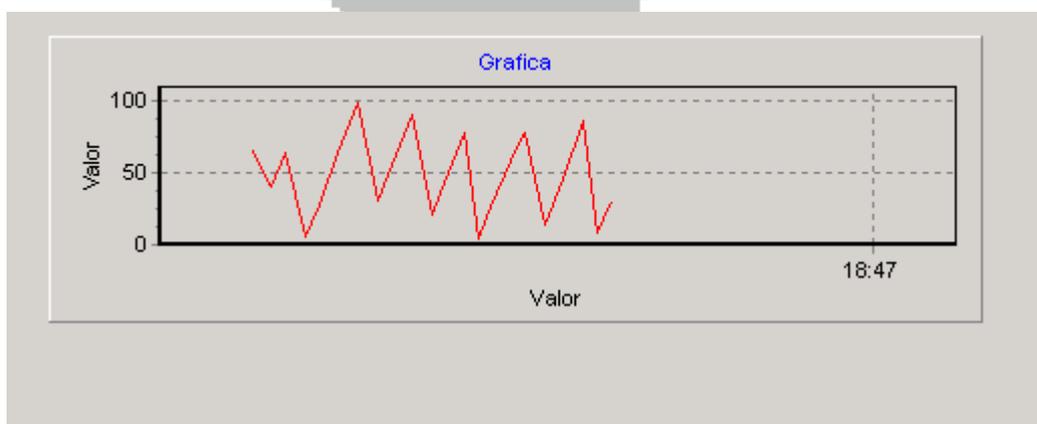


Ilustración 38: Generación de históricos

Módulo de gestión de alarmas

Una de las funcionalidades más importantes que cualquier sistema de supervisión debe proporcionar al ingeniero es una gestión eficiente de aquellas situaciones potencialmente peligrosas que comprometan la integridad de las personas y del propio sistema automatizado.

Por lo general los SCADAs proporcionan al ingeniero un módulo de alarmas que permite gestionar estas situaciones de una forma sencilla e intuitiva. En modo de edición el ingeniero podrá determinar qué variables deben ser observadas con mayor atención y cuáles son las situaciones que pueden ser indicativas de una situación anómala o de alarma, como por ejemplo que el valor de una variable se salga de un determinado rango, que su valor cambie muy rápidamente o con incrementos superiores o inferiores a una cantidad fijada.

En modo ejecución este módulo comprobará si se produce alguna de las situaciones definidas y en caso de producirse llevará a cabo las acciones asociadas por el ingeniero que podrán ir desde mostrar un mensaje, actuar sobre el proceso para tratar de paliar la situación de alarma, enviar un mensaje de emergencia, y por su puesto señalar la situación de alarma en la pantalla del SCADA. El operador podrá entonces llevar a cabo las acciones reparadoras que fuese necesario bien sea a través del SCADA o sobre el proceso directamente.



Ilustración 39: Módulo de gestión de alarmas

Conclusiones

Los sistemas SCADA como se ha podido apreciar son aplicaciones muy potentes específicamente diseñadas para permitir al ingeniero crear una potente herramienta de diálogo con el sistema automatizado.

Estas herramientas están en constante evolución en lo que a las funcionalidades que brindan al usuario se refiere. Los fabricantes de las mismas invierten gran cantidad de recursos en mantenerlas actualizadas y su gran demanda hace que exista una gran competencia en el mercado. Son ejemplos de estas herramientas:

- CIRNET, de CIRCUTOR S.A.
- InTouch, de Wonderware.
- WinCC, de Siemens.
- SYSMAC SCS, de Omron.
- FIXDMACS, de Intellution.
- RS-VIEW32 de Rockwell
- GENESIS32 de Iconics
- Etc.

OPC

Introducción al problema

En el mundo industrial, la integración de componentes pertenecientes a diversos fabricantes ha resultado frecuentemente una tarea difícil, e incluso lo sigue siendo en la actualidad en muchos casos. Las aplicaciones, como operaciones de control y monitorización o procesamiento de datos de gestión, requieren de un driver para cada componente de automatización a cuyos datos se quieren acceder.

Además, los sistemas de automatización pueden únicamente operar con aplicaciones para las que se han desarrollado drivers específicos tal y como se muestra en Ilustración 40.

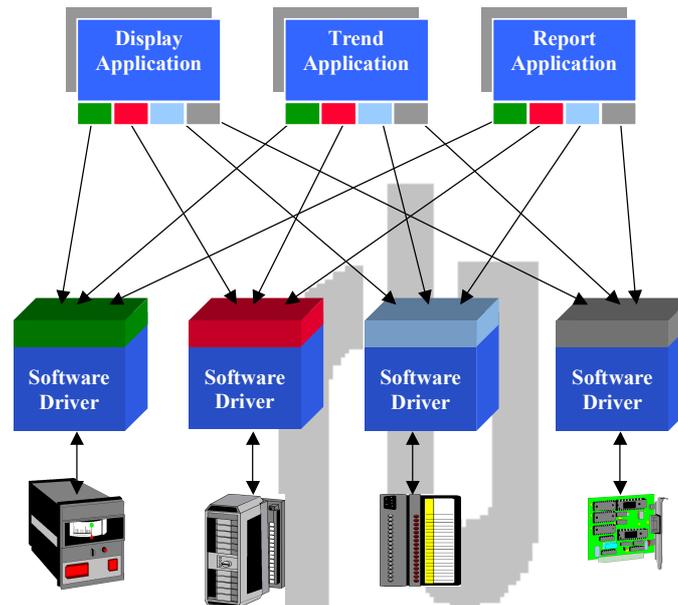


Ilustración 40: Problemática de los sistemas automatizados

Como se observa en el diagrama anterior, cada aplicación necesita utilizar distintos drivers y métodos de acceso para acceder a distintos dispositivos. Además los drivers son altamente dependientes del hardware del dispositivo (con los problemas que esto conlleva), es decir, un cambio en las especificaciones del dispositivo implicará el tener que retocar el driver y quizás el interfaz con el exterior.

Los fabricantes se enfrentan a una ardua tarea para integrar los datos de los procesos industriales en la gestión de su empresa. Principalmente lo que dificulta esta tarea es la gran variedad de dispositivos presentes en el mercado y que cada uno de ellos utiliza sus propios mecanismos de conexión y de comunicación entre diversos equipos a través de redes industriales.

Esta situación es análoga a la que se presentaba en los primeros días de la industria de los computadores cuando era imposible comunicar ordenadores de distintos fabricantes como IBM y DEC, y conlleva los siguientes problemas:

- ❑ Duplicación del esfuerzo, ya que cada fabricante necesita desarrollar sus propios drivers.
- ❑ Inconsistencias entre fabricantes, no todos los drivers se comportan igual ni soportan las mismas características hardware.
- ❑ Problemas de compatibilidad, un pequeño cambio de las capacidades y/o características del hardware puede dejar inservibles muchos drivers.
- ❑ Conflictos de acceso, dos módulos no pueden acceder simultáneamente al mismo dispositivo ya que ambos usan distintos drivers.

OPC. La solución

OPC surge como respuesta a la necesidad de disponer de una manera de acceder a los datos de forma genérica. OPC es la forma abreviada de “OLE for Process Control” y significa tecnología OLE para control de procesos. OPC es un interfaz estándar basado en los requerimientos de la tecnología OLE/COM y DCOM de Microsoft que facilita el intercambio de datos en forma estandarizada y simple entre aplicaciones de control y automatización, entre dispositivos y sistemas de campo y entre aplicaciones administrativas de oficina.



Ilustración 41: Logotipo de OPC

Pero para entender la verdadera motivación para la creación de la especificación OPC es necesario remontarse al año 1990.

La aparición de Windows 3.11 TM y su nueva tecnología DDE (Dynamic Data Exchange) para intercambio dinámico de datos comenzaron a posibilitar la integración de los datos de procesos industriales en aplicaciones de propósito general como Microsoft Excel. Sin embargo pronto resultó obvio que DDE no ofrecía el rendimiento necesario y a pesar de que aparecieron nuevas revisiones como NetDDE y FastDDE pertenecientes a empresas privadas el verdadero interés se volvió a despertar con la aparición de OLE. De esta manera apareció un pequeño grupo de empresas con intereses comunes que se unieron para desarrollar lo que ya en 1996 sería la primera versión de la especificación OPC.

Estas empresas (entre las que ya estaba Microsoft) constituyeron el núcleo de la Fundación OPC. Hoy en día siguen trabajando en la especificación manteniendo el objetivo de disponer de un estándar para el acceso a datos eficiente que sea operativo tanto a nivel industrial como en el resto de niveles de la empresa.

Como muestra el siguiente diagrama, OPC puede funcionar como interfaz genérica entre los datos y la manera de acceder a ellos:

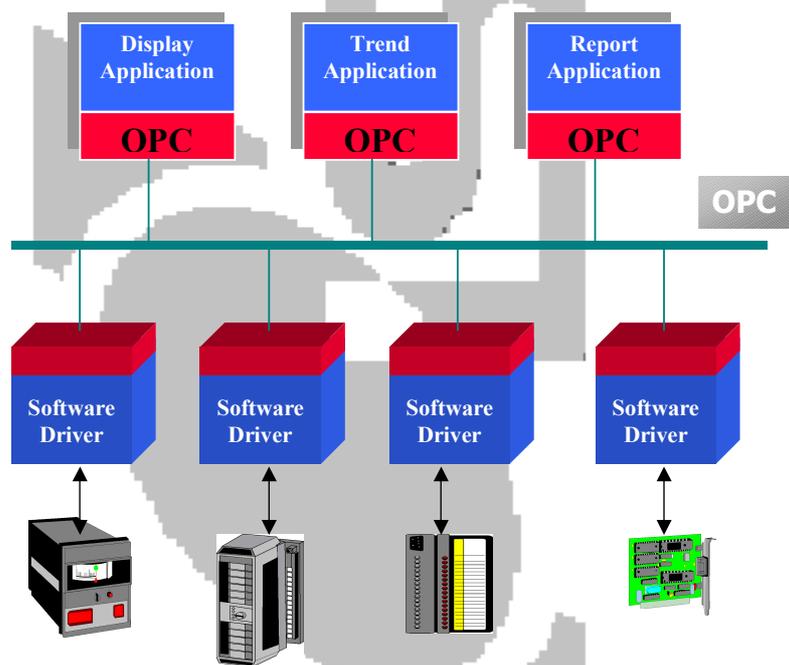


Ilustración 42: OPC soluciona el acceso a datos genérico

OPC proporciona un mecanismo para acceder a datos independientemente de su origen y entregárselos a cualquier cliente, con esto diferencia perfectamente la labor de los desarrolladores de hardware y de software aportando una solución a los problemas mencionados anteriormente:

- ❑ Los desarrolladores de hardware tan solo necesitan desarrollar un paquete de drivers. Estos se comunican siguiendo el interfaz OPC, de tal manera que las aplicaciones que los usen no deberán preocuparse más por posibles cambios en el hardware, ya que el acceso a los datos desde la aplicación a través de OPC se seguirá realizando del mismo modo. Con esto los desarrolladores podrán centrarse más en los datos y su tratamiento y menos en la manera en

la que se obtienen. Gracias al interfaz de acceso a datos de OPC hay compañías que esperan reducir entre un 60% y un 75% el esfuerzo destinado al desarrollo de drivers.

- Los desarrolladores de software no necesitan reescribir los drivers por mejoras y/o cambios en las características del hardware. Como se ha comentado anteriormente el interfaz de acceso a datos soluciona completamente este problema.
- Los usuarios obtienen un mayor número de herramientas entre las que elegir sin preocuparse de los problemas de integración. Efectivamente gracias a OPC el concepto de PLUG & PLAY está más cercano en el mundo de la automatización, ya no es necesario equipar un proceso entero con productos de la misma casa. Incluso podría usarse OPC para comunicar directamente dos aplicaciones de distintas marcas si fuera necesario (siempre que los requisitos de rendimiento lo permitan).

La fundación OPC™ es una organización global, independiente y sin ánimo de lucro que fue creada para extender las especificaciones OPC. Una de sus premisas es su conformidad con la tecnología Windows de Microsoft (uno de sus miembros) y su visión de una automatización estandarizada entre hardware y software basada precisamente en tecnología Windows (de todos modos en los últimos tiempos OPC se está implantando también en otras plataformas como Unix para aumentar su presencia en el mercado).

Actualmente diversas compañías líderes industriales son miembros activos de la fundación OPC: Toshiba, Siemens, Intellution, Rockwell Software, Fisher-Rosemount Systems, HoneyWell y Opto22 son algunas de ellas.



Ilustración 43: Logotipo de la Fundación OPC

La fundación OPC cuenta con una subsidiaria independiente que representa los intereses de todos los miembros Europeos y que adicionalmente se encarga del soporte técnico y de la adaptación a diversos idiomas.

El grupo estratégico Europeo está formado por las siguientes compañías: Intellution, National Instruments, Phoenix Contact, Rockwell Software y Siemens.

La especificación OPC

Con la introducción del interfaz OPC basado en las tecnologías OLE/COM y DCOM de Microsoft, los problemas de conexión vistos anteriormente han quedado atrás. La prioridad principal está en proporcionar un acceso flexible, poderoso y, en particular, simple a los datos.

A continuación se muestra un típico caso de uso de OPC en un entorno industrial automatizado:

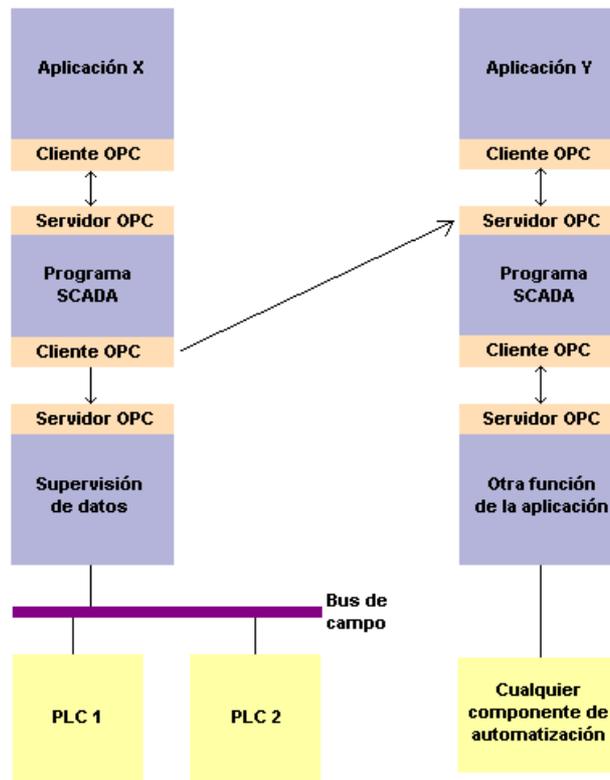


Ilustración 44: Uso de OPC en el ámbito industrial

OPC impone una arquitectura Cliente / Servidor en la que el servidor ofrece ciertos datos y el cliente accede a dichos datos. Obviamente una misma aplicación puede desempeñar ambos roles y funcionar tanto como cliente como servidor OPC.

La relación Cliente / Servidor es de muchos a muchos. El siguiente diagrama muestra un caso posible de conexión entre clientes y servidores OPC que ilustra este concepto.

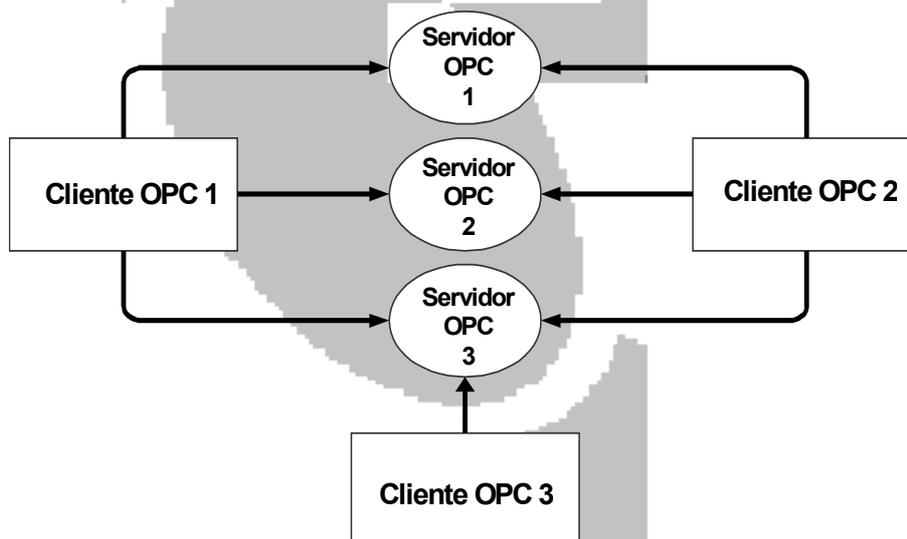


Ilustración 45: Arquitectura cliente/servidor para OPC

Es importante notar que se ha asumido en todo momento que OPC es un interfaz de acceso a datos, cuando realmente la especificación OPC cuenta además de los siguiente interfaces:

- ❑ Acceso a datos. Interfaz de configuración.

- ❑ Acceso a datos. Interfaz de automatización.
- ❑ Alarmas y Eventos. Interfaz de configuración.
- ❑ Alarmas y Eventos. Interfaz de automatización.
- ❑ Procesamiento por lotes(Batch). Interfaz de configuración.
- ❑ Procesamiento por lotes(Batch). Interfaz de automatización.
- ❑ Acceso a Históricos de datos.
- ❑ Seguridad.

La diferencia entre un interfaz de configuración y otro de automatización reside en que este último se desarrolla como un objeto de automatización OLE, pensado para ser utilizado principalmente desde aplicaciones desarrolladas con lenguajes como Visual Basic. El interfaz de configuración está pensado para ser desarrollado en lenguajes como C++ y se espera en general que se utilice desde aplicaciones que necesiten incluir un servidor OPC con el mayor rendimiento posible. Estos conceptos se resumen en el siguiente esquema:

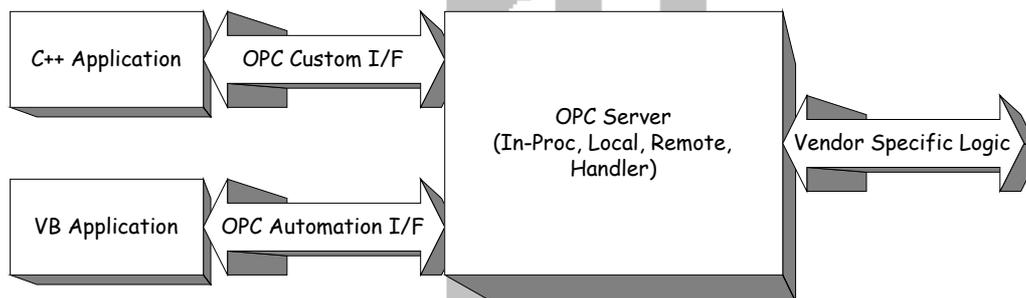


Ilustración 46: Interfaces de automatización y de configuración

NOTA: En el resto del documento se seguirá asumiendo igual que se ha hecho hasta ahora que cuando hablemos de la especificación OPC estaremos hablando del interfaz de acceso a datos, en particular del interfaz de configuración, no del de automatización.

El modelo de objetos componentes COM

COM es un modelo (no hay que confundirlo con un lenguaje) de programación orientado a objetos para crear aplicaciones de software formadas por componentes modulares. COM permite que diferentes módulos de software, programados sin información sobre los demás, funcionen conjuntamente como una única aplicación.

Las características más relevantes de COM son las siguientes:

- ❑ Es orientado a objetos, precisamente podemos hablar indistintamente de objeto que de componente.
- ❑ Proporciona un interfaz binario (se aplica al programa una vez que es compilado a código máquina), gracias a esto es independiente de la plataforma.
- ❑ Es un estándar (modelo o especificación si se prefiere), y como tal no indica como debe implementarse. Así, y ya que proporciona un interfaz binario, es independiente del lenguaje de programación (sirven todos los que puedan definir estructuras y tengan punteros, así sirven C++, JAVA, C, Pascal, Ada, Java e incluso Visual Basic)
- ❑ COM permite que los componentes de software tengan acceso a servicios de software proporcionados por otros componentes, independientemente de si implican llamadas locales a funciones, llamadas a sistemas operativos o comunicaciones de red.

NOTA: Aunque se ha afirmado que COM es un modelo y por lo tanto no indica cómo debe implementarse un objeto COM, COM ofrece un pequeño API que se incluye en la librería COM y se encarga de rutinas como la carga y descarga o localización de objetos COM. Así la implementación de cualquier objeto COM se divide en la implementación genérica aportada por la librería COM y la implementación particular que se haya hecho para ese objeto COM que dependerá del lenguaje y del uso que se le quiera dar.

COM y OPC

COM es el modelo utilizado para definir la especificación OPC. La especificación es desde el punto de vista del programador un conjunto de interfaces que es necesario implementar.

¿Pero, qué es exactamente un interfaz? Un interfaz es un conjunto de funciones relacionadas (normalmente semánticamente) que el objeto COM expone como un grupo, se define a nivel binario como una tabla de punteros a punteros (estos últimos son las direcciones de los métodos que posee dicho interfaz).

Un interfaz es la manera en la que un objeto COM puede exponer sus métodos.

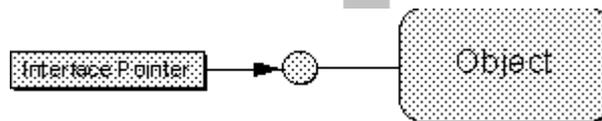


Ilustración 47: Interfaces COM

El convenio establece que los objetos COM deben representarse como a continuación se indica.

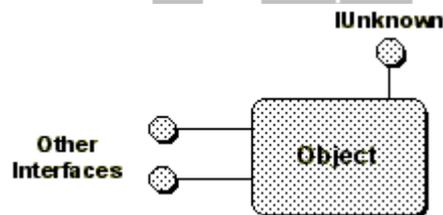


Ilustración 48: Imagen de objeto COM genérico según convenio

COM distribuido

Es un protocolo de objeto que permite a los componentes COM comunicarse directamente y de manera transparente entre ellos a través de una red. DCOM es un lenguaje neutro, de manera que cualquier lenguaje que produzca componentes COM también puede producir aplicaciones DCOM.