

4 INTERFAZ USUARIO MÁQUINA (HMI)

El segundo hito marcado por el cliente para este proyecto era la realización de una interfaz informática que permitiera la comunicación con el sistema físico. Esta interfaz se conoce como Interfaz Usuario Máquina (HMI, Human Machine Interface).

En el presente proyecto, el objetivo de esta HMI era el dotar al usuario de una herramienta útil para la configuración y testeo del sistema DSA-Antena. Se persigue pues que en la aplicación se tenga acceso tanto al control de los diferentes parámetros y modos de funcionamiento como al estado de los dispositivos y al sistema de testeo integrado, BIT.

Para ello, la HMI llamará a los diferentes servicios que la aplicación DLL provee, de la forma en que se ha detallado en el apartado 3 del presente documento.

4.1 Estructura básica del HMI

La estructura básica de la HMI se puede ver en la siguiente figura. Se presenta desde la ventana principal, desde la que se podrán seleccionar los diferentes modos. En este esquema básico introductorio se resume en control, BIT, y otros modos.

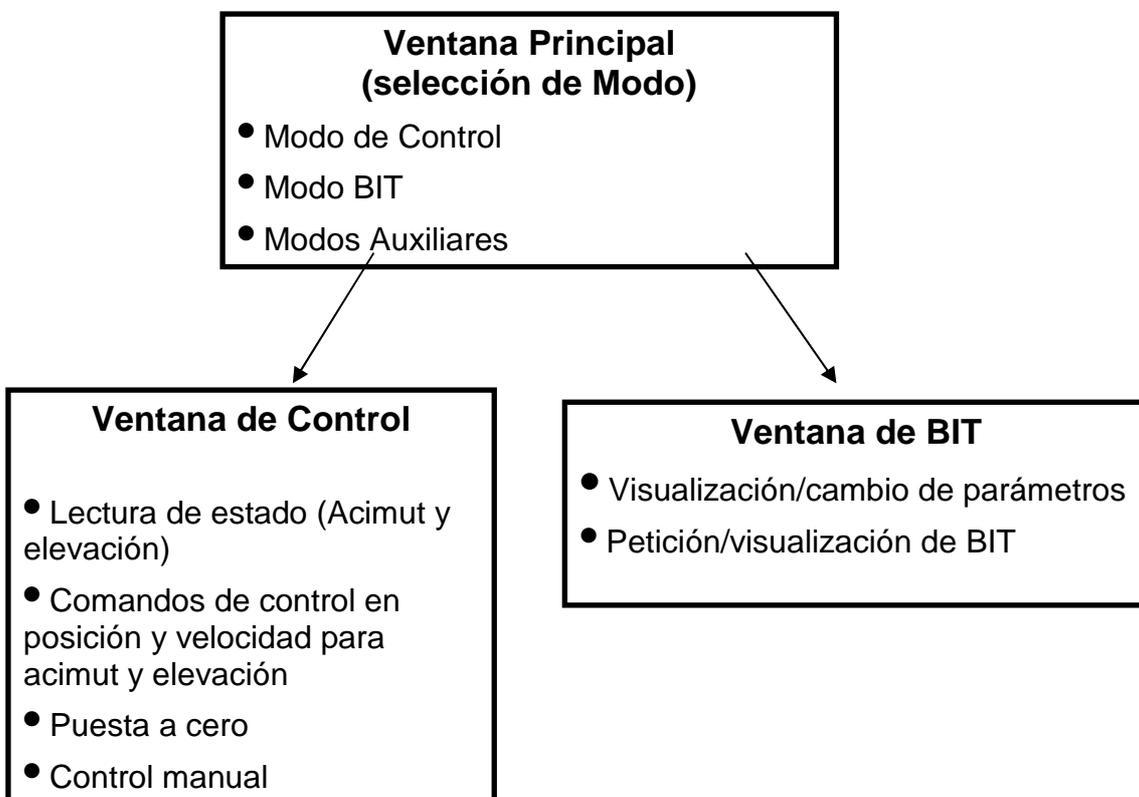


Figura 13: Resumen de la pantalla principal del HMI

La tarea de proveer el método para el control de la antena se realiza principalmente en la ventana de control. En ella se monitorizan los valores de posición de los ejes de acimut y elevación, además de poder dar valores a seguir por parte de dichos ejes. Este seguimiento puede ser tanto en modo posición como en modo velocidad. Se incluye además la posibilidad de realizar un control manual mediante la pulsación de botones tipo flecha, y la posibilidad de llevar la antena a posición cero.

Por otra parte, en la pantalla de BIT, se muestra la situación de las diferentes banderas definidas por el método de testeo incluido. Con ello se provee la posibilidad de un chequeo rápido a posibles errores en el sistema.

En los próximos apartados se detallan más extensamente todas las pantallas del HMI, así como sus parámetros, sus significados y el funcionamiento global de la aplicación.

4.2 Entorno de trabajo.

La implementación de esta interfaz está realizada íntegramente en tecnología Microsoft Visual Basic, en su versión 6.0. La elección de esta herramienta viene justificada por ser petición del cliente. Visual Basic 6.0 estaba desarrollado para conseguir llevar al clásico lenguaje de aprendizaje BASIC a un estadio de desarrollo que le permitiera convertirse en una opción interesante a la hora de desarrollar interfaces para entornos Windows.

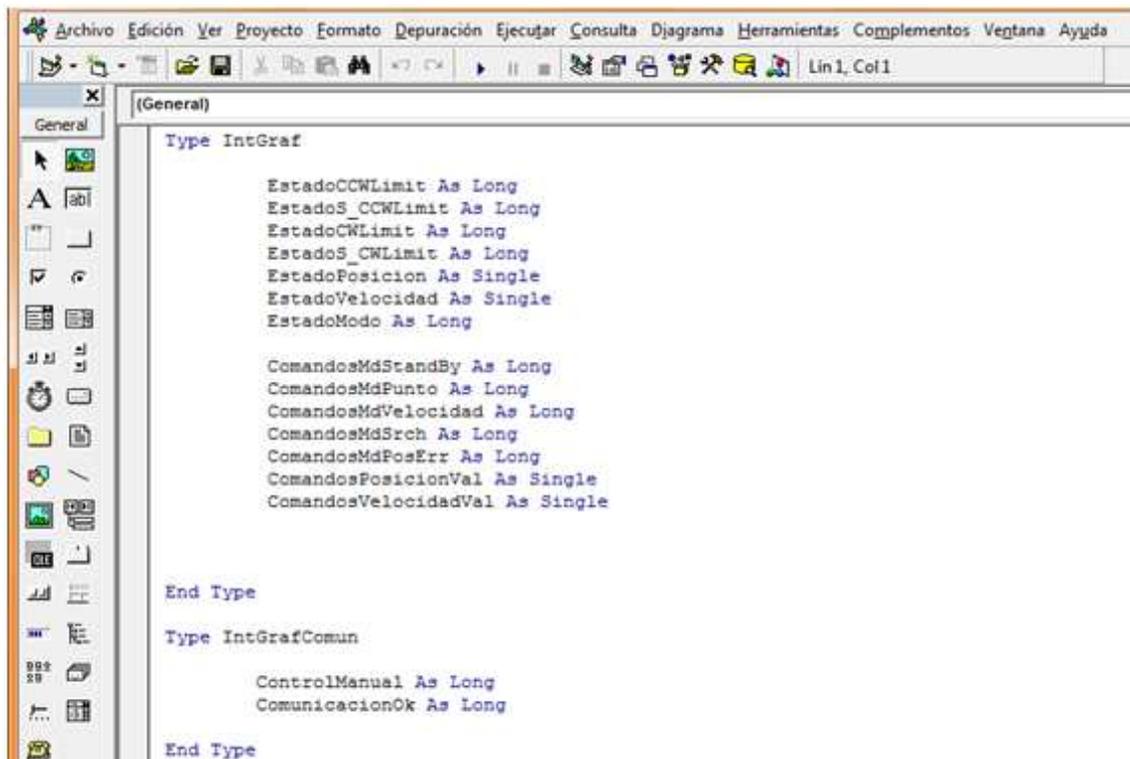


Figura 14: Detalle del código de las estructuras de comunicación en Visual Basic 6.0

De esta forma, las herramientas del entorno Visual Basic 6.0 más importante en el desarrollo de la HMI eran los formularios y los controles asociados a ellos. Si bien este IDE de Microsoft permite desarrollar aplicaciones de usuario más rudimentarias, como aplicaciones orientadas a líneas de comando, la potencia de este entorno viene marcada por el uso de formularios.

Así, Visual Basic es un lenguaje de programación basado en objetos, en el que los objetos principales son los formularios y controles. El trabajo del programador consiste en leer y modificar las propiedades de los objetos, llamar a sus métodos y responder a los sucesos y eventos generados por formularios y controles. [7]

4.3 Las estructuras para la comunicación con la DLL

Como se ha contado el apartado anterior, correspondiente a la aplicación DLL, la comunicación de ésta con aplicaciones externas, y por ello, con la HMI desarrollada en este apartado, se realiza a través de unas estructuras de datos.

En este apartado se presentan dichas estructuras implementadas ahora para la HMI. Como se puede observar, son equivalentes a las presentadas para la aplicación en C++ de la DLL.

Type IntGraf

```
EstadoCCWLimit As Long
EstadoS_CCWLimit As Long
EstadoCWLimit As Long
EstadoS_CWLimit As Long
EstadoPosicion As Single
EstadoVelocidad As Single
EstadoModo As Long

ComandosMdStandBy As Long
ComandosMdPunto As Long
ComandosMdVelocidad As Long
ComandosMdSrch As Long
ComandosMdPosErr As Long
ComandosPosicionVal As Single
ComandosVelocidadVal As Single
```

End Type

Type IntGrafComun

ControlManual As Long
ComunicacionOk As Long

End Type

Type IntGrafSRCH

CentroAz As Single
SectorAz As Single
PasoAz As Single
VelocidadAz As Single

CentroEl As Single
SectorEl As Single
PasoEl As Single
VelocidadEl As Single

Tipo As Long
UpdVia As Long

End Type

Type IntGrafBIT

BITSafeSwitch As Long
BITPhaseCurrentSum As Long
BITHPhaseA As Long
BITHPhaseB As Long
BITHPhaseC As Long
BITMotorCurrent As Long
BITLowBusVoltage As Long
BITHighBusVoltage As Long
BITHighBusCurrent As Long
BITLowBrkCurrent As Long
BITHighBrkCurrent As Long
BITRefVoltage As Long
BITEncoderACount As Long
BITEncoderBCount As Long
BITEncoderAIndex As Long
BITEncoderBIndex As Long
BITHome As Long
BITLimitDown As Long
BITMotorVelocity As Long
BITMotorLoadPos As Long
BITInternalTemperature As Long
BITInternalHumidity As Long

BITMotorTemperature As Long
BITBridgeTemperature As Long
BITMotorBimetal As Long
BITPowerBridgeLockOut As Long
BITInternalPwrSupply As Long
BITFlashError As Long
BITRsrv1 As Long
BITRsrv2 As Long
BITRsrv3 As Long
BITRsrv4 As Long
BITActualizaOk As Long
End Type

Type IntGrafSDO
Eje As Long
Indice As Long
Subindice As Long
TipoDato As Long
ValorLeer As Long
ValorEscribir As Long
Respuesta As Long
End Type

4.4 Ventana Menú Principal

En la ventana principal de la HMI se presentan los diferentes modos de funcionamiento y opciones que se ponen a disposición del usuario. Esta ventana.



Figura 15: Ventana principal de la HMI

Esta ventana no tiene más papel en el desarrollo de la aplicación que el de presentar un menú adecuado para el funcionamiento global de la aplicación. Cada modo de uso tendrá su propia interfaz, y al salir de ellos siempre se vuelve aquí. Al entrar en uno de los modos, se bloquea esta ventana, de forma que nunca podremos entrar en dos modos a la vez.

4.5 Ventana de Control

La ventana de control concentra la mayor parte de funcionalidades de la HMI. En ella se encuentran tanto los estados como las opciones de control de posición y velocidad de los dos DSA, el de acimut y el de elevación. Esta ventana se puede ver en la siguiente figura.

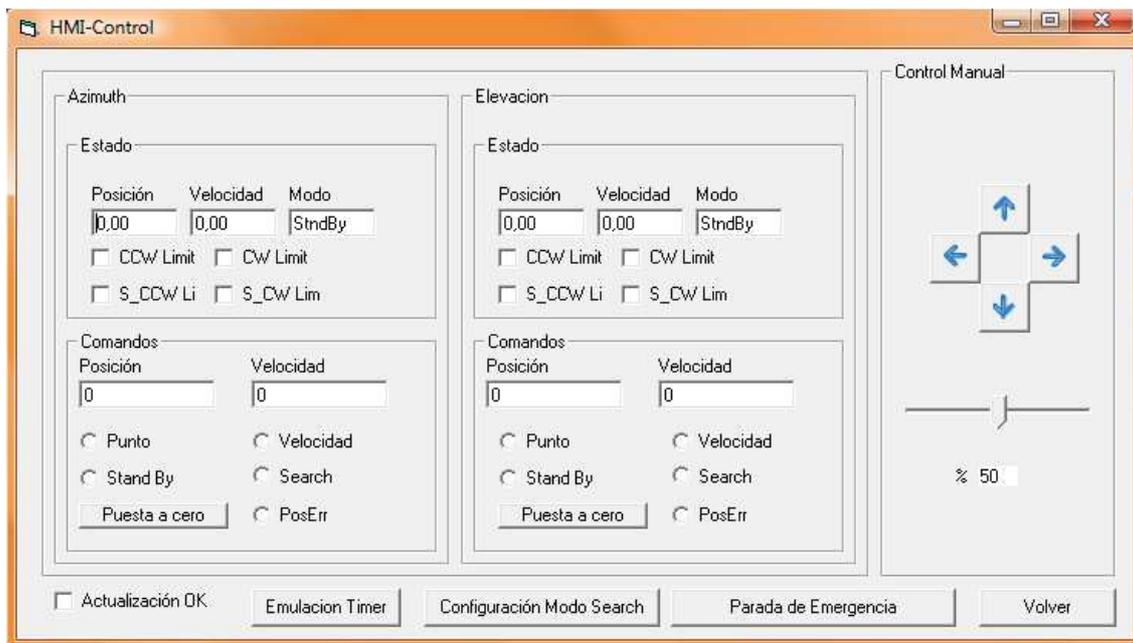


Figura 16: Ventana de control del HMI

Se puede observar que hay dos partes similares, una para cada DSA. En el panel *estado* encontramos tres *textbox*, llamados *posición*, *velocidad* y *modo*. Los dos primeros indican el valor de la posición y de la velocidad de motor controlado por cada DSA, mientras que el tercero indica si el sistema está en modo posición, modo velocidad o en modo Stand By.

Las unidades de las posiciones están en grados. El rango para la componente acimutal está comprendido entre 0° y 360° , mientras que la componente de elevación tiene un rango entre $\pm 180^\circ$. El motivo de que el rango en elevación sea este es que EADS así lo requería, para poder realizar posteriormente el control de la antena por su parte.

Por otra parte, existen también cuatro *checkbox* en los que monitorizan los finales de carrera de los motores. *CW Limit* y *CCW Limit* monitorizan los finales de carrera hardware en el sentido de las agujas de reloj y en el contrario, respectivamente. Por otra parte, *SCW* y *SCCW* son los finales software de los mismos. De esta manera el usuario puede comprobar si alguno de los ángulos ha alcanzado su límite.

En la sección *Comandos*, que podemos ver en la siguiente figura, se introducen las órdenes para los DSA. Los diferentes modos de funcionamiento disponibles

son seleccionados mediante el uso de los *radiobuttons*. Al pinchar sobre ellos, el comando de posición o velocidad introducido en los cuadros de texto son enviados a los DSA.

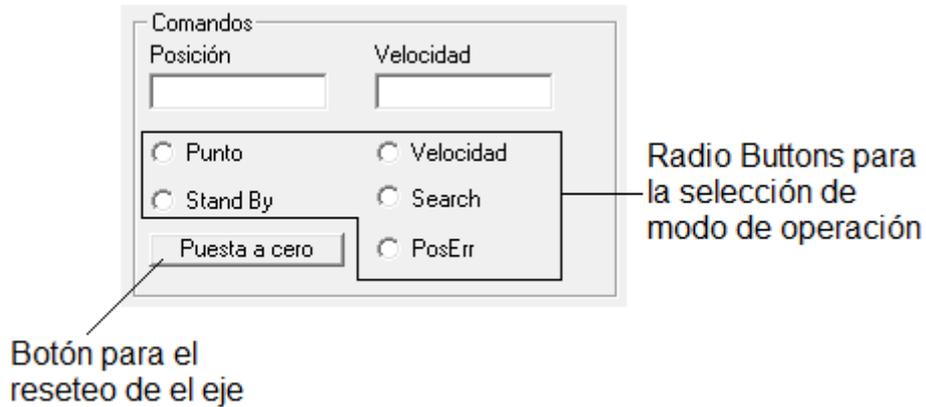


Figura 17: Sección Comandos de la ventana de control

En el modo Stand By, la velocidad de movimiento se reducirá hasta cero, quedando el motor parado en la posición alcanzada en ese momento. El sistema quedará a la espera de nuevos comandos. El modo Search ¿??

Finalmente, el botón de puesta a cero lleva los motores a la posición por defecto.

A la derecha de la ventana se encuentra la sección para el control manual de la antena. En este caso, no se ha realizado por separado el control del eje acimutal y del eje de elevación, sino que, a través de simples flechas, el usuario puede controlar el movimiento de la antena. Se puede observar el detalle en la siguiente figura

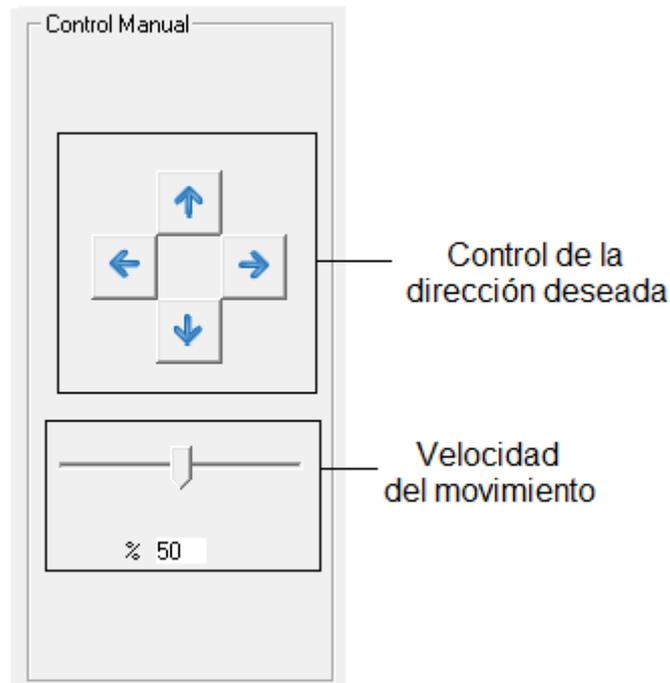


Figura 18: Control manual de la antena

Como se observa, hay una flecha para cada posible dirección de la antena. Este sistema está pensado para posibilitar un apuntamiento específico por parte del usuario, para escenarios de puestas en marcha o pruebas. Junto con la cámara que dispone el sistema, se puede lograr que la antena apunte al lugar de interés.

La barra de desplazamiento que encontramos a la izquierda nos permite definir la velocidad con que la antena se moverá. Hay que señalar que mientras esté pulsado uno de los botones de flecha el motor se moverá a la velocidad indicada por dicha barra, y será al soltarse el botón cuando cese el movimiento.

Para terminar con el repaso a esta ventana de la interfaz, quedan por comentar los botones *Parada de Emergencia* y *Configuración Modo Search*, y el *checkbox* de *Actualiza Ok*. El botón de parada de emergencia detiene por completo el funcionamiento de los motores. Ambos motores son decelerados hasta que se paran, y son posteriormente apagados. La diferencia con el modo Stand By radica en que en el caso de la parada de emergencia se paran ambos motores, mientras que en Stand By quien para es el eje que en ese momento se esté controlando.

Por otro lado, el botón de Configuración de modo search permite al usuario ajustar los parámetros para la ejecución de dicho modo. La ventana de configuración se puede ver en la siguiente figura:



Figura 19: Pantalla para la configuración del modo búsqueda.

Como se puede ver, en la pantalla se irían introduciendo los parámetros necesarios para realizar una búsqueda de la posición deseada. El campo SYNC hace referencia al tipo de algoritmo utilizado para alcanzar la posición deseada.

Por último, el *checkbox* de *Actualiza Ok* aparecerá marcado siempre que la información de la pantalla esté correctamente actualizada, lo que en el fondo significa que la comunicación PC-DSA se está produciendo correctamente.

4.6 Ventana BIT

La ventana BIT monitoriza todos los parámetros del protocolo Built In Test, incluido en el software del pedestal, y que fue comentado en la sección dos de este proyecto. De esta manera, se puede comprobar de un vistazo si existe algún problema en el sistema.



Figura 20: Ventana del BIT de la HMI

De nuevo se tiene un botón para la parada de emergencia en caso de que fuera necesaria esta, y un checkbox para monitorizar que la comunicación PC-DISA (y con ella los datos del BIT mostrados aquí) está actualizada y en correcto funcionamiento. El significado de las distintas alarmas se puede consultar en [2]

Además, aparece el botón limpiar, cuyo propósito es el de eliminar las banderas activadas, de forma que con cada nueva recepción de datos de BIT, la información sea actualizada. Esto es necesario porque la HMI sólo actualiza esta pantalla cuando recibe una alarma, pero no al cesar dicha alarma. Con ello evitamos que haya falsas alarmas activadas.

4.7 Ventana envío SDO

A través del envío de mensajes SDO (Service Data Objects, consultar la sección dos de este proyecto) se puede realizar una escritura de los datos del

diccionario de objetos del DSA que deseemos. Esto permite configurar los parámetros que se consideren interesantes. La ventana de envío se puede ver en la siguiente figura:

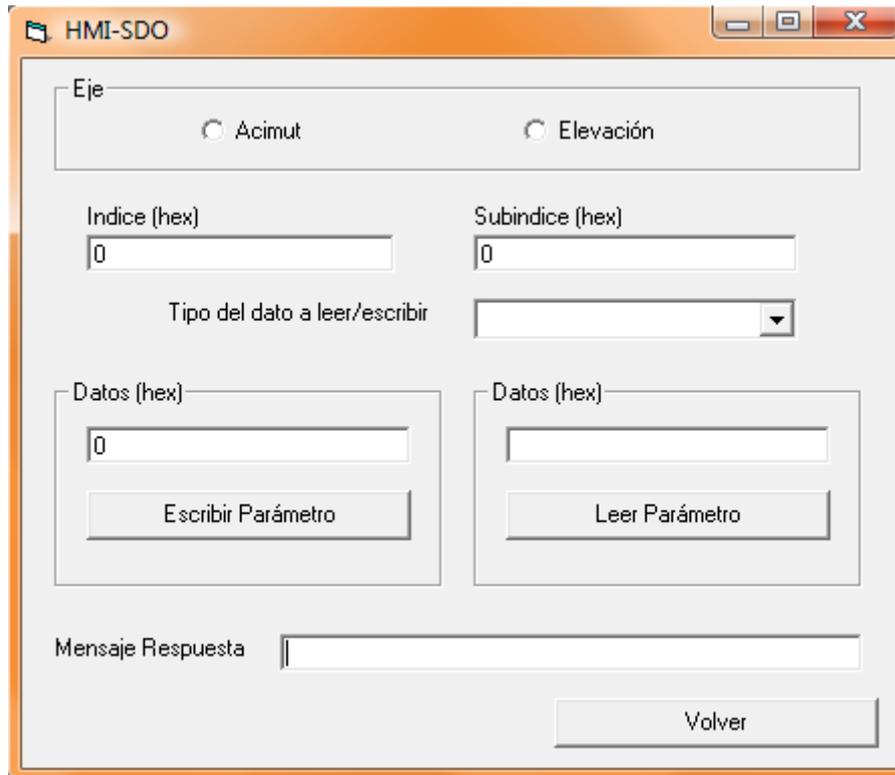


Figura 21: Ventana para el envío de SDO del HMI

Como se ve, se debe elegir entre el eje de acimut o elevación. De esta forma actuaremos sobre uno u otro de los servos. También puede verse que es necesario conocer exactamente el índice del objeto a escribir, así como el valor en hexadecimal que toma el comando, y el tipo de dato en el que se almacena dicho parámetro.

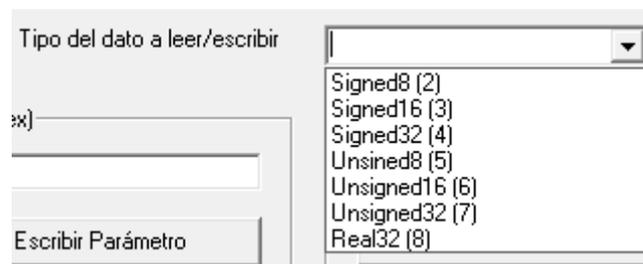


Figura 22: Opciones de datos de SDO

Una vez elegidas estas opciones, podremos tanto escribir el parámetro, como consultarlo mediante una lectura. El mensaje de respuesta se muestra en el último *textbox* que aparece. La información sobre el servicio SDO y los

diferentes objetos del diccionario de objetos se puede consultar en el apartado 2 de este proyecto y en la referencia [2].

4.8 Ventana Configurar Puerto

La funcionalidad de esta ventana viene marcada por la necesidad de realizar, al comienzo de la aplicación gráfica, una conexión a través del puerto serie con el sistema DSA, de forma que se pueda establecer una comunicación. Aunque se comenta detrás de otras ventanas, realmente esta es la primera pantalla con la que se encuentra el usuario. Se puede ver una captura de esta ventana en la siguiente figura:

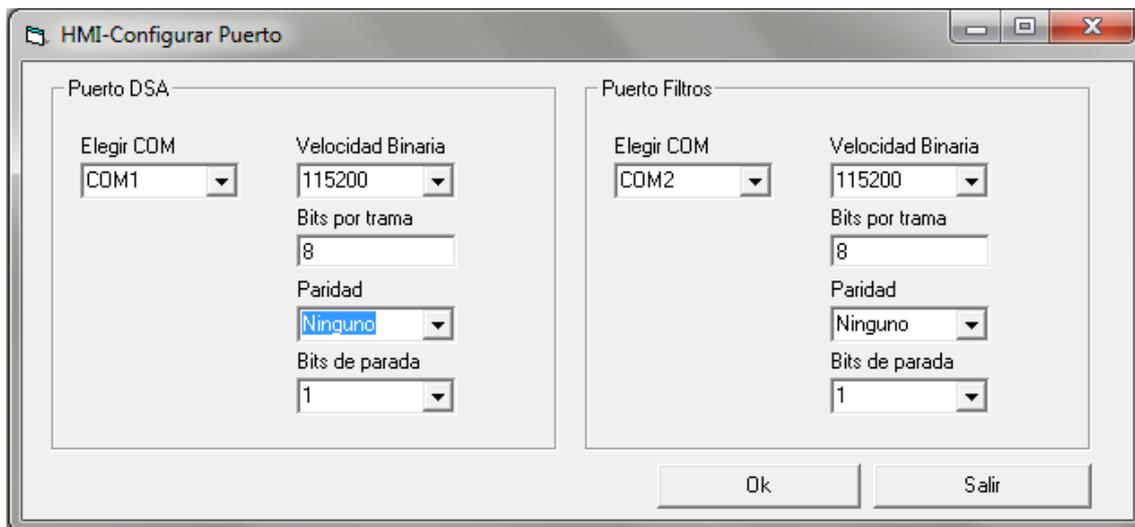


Figura 23: Ventana para la configuración del puerto de comunicaciones del HMI

Se observa que también hay una sección para los filtros. Efectivamente, es necesario configurar también al inicio la conexión con el banco de filtros, si se desea poder interactuar con ellos, mediante la ventana diseñada a tal efecto.

En ambos casos, si falla la conexión, el programa da un mensaje de error y no avanza.

4.9 Ventana Filtros

A través de la ventana de filtros, se puede seleccionar uno de los cuatro filtros del banco de filtros de GIZA TECHNOLOGIES. Dichos filtros no tienen mayor relevancia en este proyecto que el de la necesidad marcada por el cliente de disponer, en la herramienta gráfica, de una pantalla como la que se está presentando para poder seleccionar el filtro deseado. Dicha pantalla se muestra en la siguiente figura:



Figura 24: Ventana Filtros de la HMI