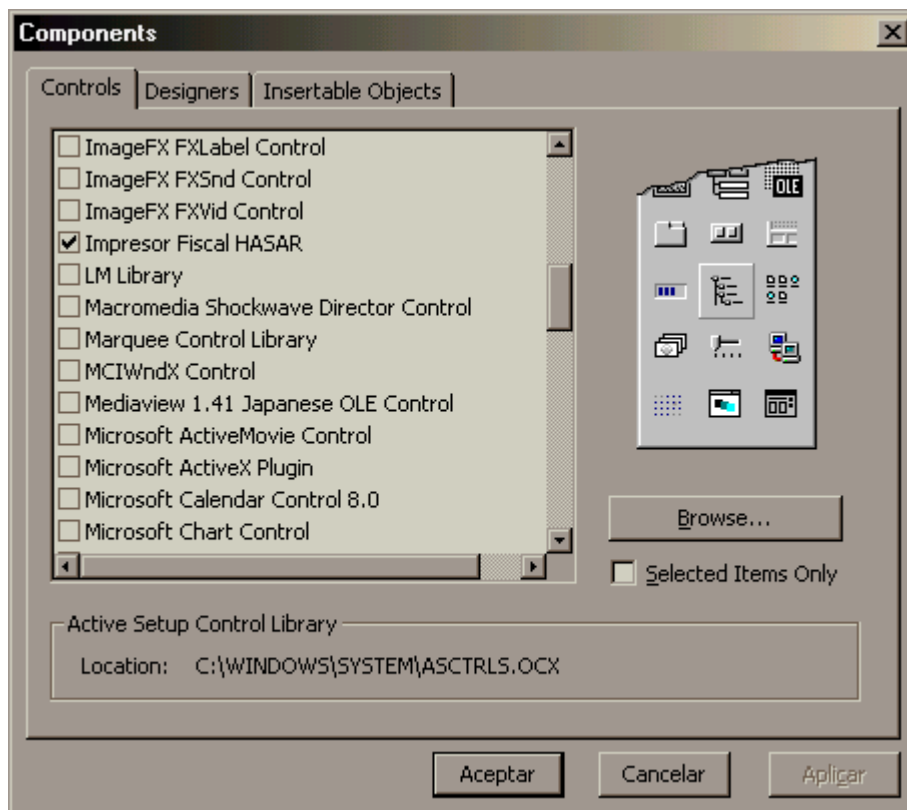


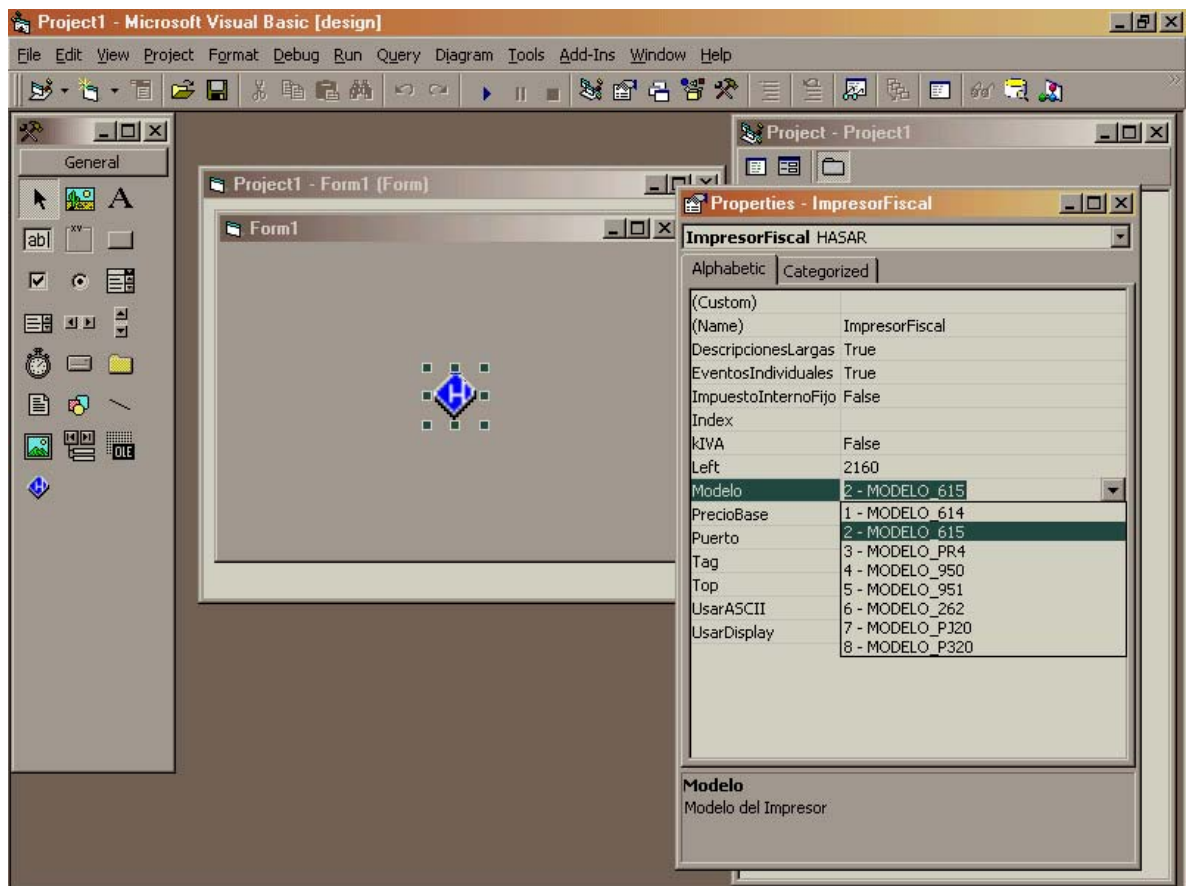
Control ActiveX para Impresor Fiscal

El control ActiveX es un objeto COM insertable en cualquier entorno de 32 bits que soporte controles de este tipo. Como primera medida, ha de registrarse el objeto apropiadamente en el sistema por intermedio de algún utilitario de registración como el RegSvr32.Exe. Una vez registrado se lo puede encontrar entre los demás controles ActiveX registrados e insertarlo en un programa. Este control funciona como un wrapper alrededor de la DLL de 32 bits (winfis32.dll), por lo cual es condición necesaria la presencia de esta DLL tanto al registrarlo como al distribuirlo. Las restricciones de uso de los comandos se aplican según la ley fiscal correspondiente, y no son controladas en forma particular por el objeto.

Una vez registrado, en Visual Basic, por ejemplo, se lo podrá encontrar en la ventana de componentes:



Si se le selecciona en esta pantalla, podrá encontrárselo en la barra de herramientas de Visual Basic como un ícono más:



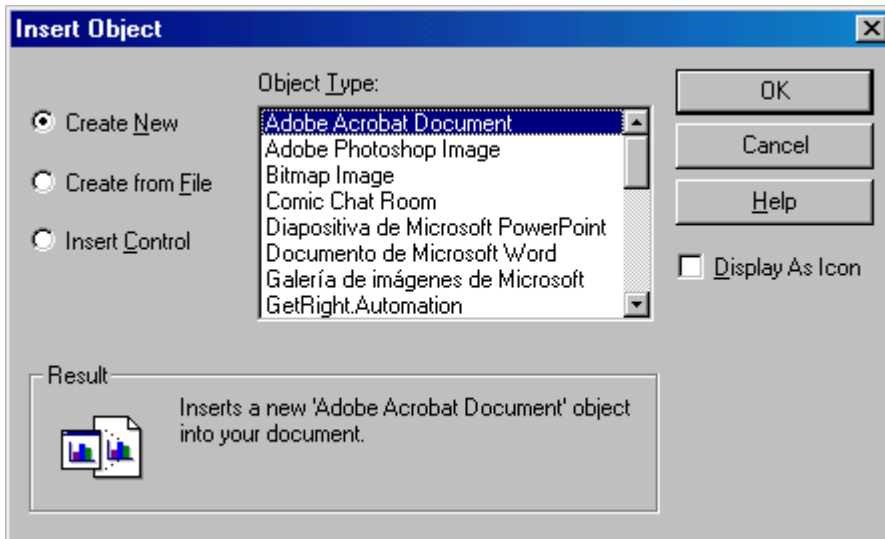
Y se comporta como cualquier otro control de **Visual Basic**.

El siguiente es un breve ejemplo que demuestra en pocas líneas cómo imprimir un ticket:

```
Hasar1.Puerto = 1
Hasar1.Comenzar
Hasar1.TratarDeCancelarTodo
Hasar1.AbrirComprobanteFiscal TICKET_C
Hasar1.ImprimirItem "Item 1", 10, 1, 21, 0
Hasar1.Subtotal True
Hasar1.ImprimirPago "Efectivo", 100
Hasar1.CerrarComprobanteFiscal
Hasar1.Finalizar
```

Visual FoxPro 6.0

En primera instancia debe insertarse en un formulario un componente Active X. Para ésto debe arrastrarse desde la barra de herramientas (toolbar) el icono denominado "Active X Control (OleControl)" apareciendo así la siguiente ventana:



Seleccionando la opción "Insert Control", luego de unos segundos se mostrará una lista de los objetos Active X registrados, entre ellos el componente "Fiscal Class". Otra forma es a partir de la galería de componentes, eligiendo desde el folder "ActiveX". Una cosa importante es que para recibir eventos del controlador (errores, falta papel, etc.) se debe poner a **.F.** la propiedad `AutoYield`. Esto es una limitación de FoxPro y tiene sus consecuencias (consultar el help), pero si se quieren usar controles ActiveX, debe usarse de esta forma, según el help mismo indica (*"The AutoYield property should be set to False when a form contains an ActiveX control"*).

Para asegurar que se procesan todos los eventos de controles ActiveX, establezca la propiedad `AutoYield` del objeto `Application` de Visual FoxPro como falsa (.F.).

Por ejemplo, del siguiente modo:

```
PROCEDURE Unload
    IF This.Tag = "AUTOYIELD_YES"
        _VFP.AutoYield = .T.
    ENDIF
    SYS(2333,1)
ENDPROC
```

```
PROCEDURE Load
    This.Tag = IIF( _VFP.AutoYield, ;
        "AUTOYIELD_YES", ;
        "AUTOYIELD_NO" )
    _VFP.AutoYield = .F.
    SYS(2333,0)
ENDPROC
```

El siguiente es un ejemplo de cómo llamar a los métodos:

```
TICKET_C = 84
THISFORM.Olecontrol1.Puerto = 1
THISFORM.Olecontrol1.Comenzar
```

```
THISFORM.OleControl1.TratarDeCancelarTodo  
THISFORM.Olecontrol1.AbrirComprobanteFiscal (TICKET_C)  
THISFORM.Olecontrol1.ImprimirItem ("Item 1", 10, 1, 21, 0)  
THISFORM.Olecontrol1.Subtotal (.T.)  
THISFORM.Olecontrol1.ImprimirPago ("Efectivo", 100)  
THISFORM.Olecontrol1.CerrarComprobanteFiscal  
THISFORM.Olecontrol1.Finalizar
```

Algunos comentarios importantes

En general, el uso del control ActiveX está limitado al nivel de compatibilidad de cada lenguaje con la norma ActiveX. El reconocimiento de constantes embebidas ("TICKET_C", por ejemplo), el soporte de eventos, los tipos especiales (Fecha OLE, por ejemplo) forman parte del grupo de características que pueden no estar soportadas por todos los lenguajes. En caso que las constantes embebidas no sean tomadas automáticamente por el lenguaje, al final se expone la lista de ellas. A continuación se ofrece una breve descripción de los métodos, eventos y propiedades del objeto. Cada comando corresponde a uno o más comandos del controlador; para mayor información acerca del comportamiento de los comandos y el parser fiscal, se recomienda leer el manual del controlador correspondiente.

Propiedades

1. Propiedades de Configuración del Controlador:

- `PrecioBase` [Booleana]

Define el modo de trabajo de los precios. Si este valor es verdadero, se trabaja con base imponible, caso contrario con precio final (impuestos incluidos)

- `EventosIndividuales` [Booleana]

Determina si los eventos de estado se generan por cada uno de los componentes del estado, o todos juntos en un valor comprendiendo todos los bits que forman el estado. Trabajar según la primer modalidad es más cómoda para procesar los diferentes estados, ya que no requiere lógica de bits, pero la segunda es más rápida, y consume menos recursos.

- `UsarDisplay` [Booleana]

Determina si se usará el display (si el impresor posee uno) en los comandos que soportan uso automático de display.

- `kIVA` [Booleana]

Determina si se usará kIVA en vez de porcentaje de impuestos internos. El controlador trabaja directamente con kIVA; si esta propiedad tiene el valor Falso (*default*), se realizará una conversión interna para convertir de porcentaje de impuestos internos a kIVA. Esto no afecta en absoluto si los impuestos internos son fijos.

- `ImpuestoInternoFijo` [Booleana]

Indica si la próxima venta tendrá impuestos internos fijos no afectado por descuentos/recargos. Luego de la operación de venta siguiente vuelve a su valor default (False: impuestos internos porcentuales).

- `ImpuestoInternoPorMonto` [Booleana]

Indica si la próxima venta tendrá impuestos internos fijos indicados por coeficiente o por monto. Solo tiene efecto cuando la propiedad anterior, `ImpuestoInternoFijo`, indique que la venta siguiente tendrá impuestos internos fijos. Luego de la operación de venta siguiente vuelve a su valor default (False: impuestos internos fijos expresados por coeficiente).

- `UsarASCII` [Booleana]

Si esta propiedad es verdadera, no se realizará internamente la conversión para compatibilizar el juego de caracteres de Windows con el de la impresora fiscal. El default es *false*.

- `ReintentoConstante` [Booleana]

Si esta propiedad es verdadera, el número de reintentos en la transmisión de paquetes hacia el Controlador Fiscal cuando el mismo no responde (error de Time Out) se hace infinito. En tal caso, cada vez que se produzca un error de Time Out, este se reportará por medio del evento '*Impresora No Responde*'. El proceso de reintento constante podrá ser interrumpido por medio del método '*Abortar*'. El default es *false*.

- `Modelo` [Tipo ModelosDeImpresoras]

Indica el modelo de impresora a usar. Las posibilidades son: `MODELO_614`, `MODELO_615`, `MODELO_PR4`, `MODELO_950`, `MODELO_951`, `MODELO_262`, `MODELO_PJ20` y `MODELO_P320`. Esta elección condicionará los comandos según estén soportados por las distintas reglamentaciones a las que se somete cada modelo.

- `ModoStatPrn` [Booleana]

Indica si se va a usar la modalidad StatPRN o no, es decir, si el impresor retorna inmediatamente al ver que estando ocupado, demoraría cierto tiempo en entregar la respuesta. Este modo es útil para controlar falta de papel o demoras sin trabajar con eventos. Ver la propiedad `HuboStatPrn` y el comando `EnviarStatPrn`. Sólo funciona en los nuevos controladores.

- `DescripcionesLargas` [Booleana]

Este valor decide si las descripciones de los artículos son truncadas al superar el tamaño máximo soportado por el impresor (Falso), o extendidas usando el comando de impresión de texto fiscal (Verdadero, *default*).

- `Puerto` [Entero Largo]

Indica el puerto físico donde está conectado el impresor fiscal. Debe ser especificado antes de cualquier operación que implique comandos al impresor.

- `Baudios`

Velocidad que se espera encontrar al controlador. Debe ponerse antes de `Comenzar()`.

2. Propiedades del Impresor Fiscal:

- `FechaHoraFiscal` [Tipo DATE (fecha OLE)]

La Fecha/Hora del impresor en un momento dado. Si se quiere fijar la fecha/hora sólo será posible después de una Z y antes del primer comprobante de la jornada fiscal.

- `Encabezado(n)` [Cadena]

Dado un número (n) se recupera o se especifica una línea del encabezado del ticket/factura. Si el número de línea se refiere a las diez primeras, se trata del encabezado en sí. Si está en el rango de once a veinte, es el pie.

- Respuesta(n) [Cadena]

Se refiere a la respuesta del controlado fiscal, relacionada con el último comando enviado. Dado un número de campo (n), se obtiene la porción de la respuesta requerida. Si el número de campo es 0 (cero), se obtiene la respuesta cruda.

3. Propiedades de Estado:

- UltimaFactura [Entero Largo]

Indica el número de la última factura A emitida, si el impresor soporta el uso de facturas. Sólo lectura.

- UltimoTicket [Entero Largo]

Indica el número del último ticket o factura tipo B/C. Sólo lectura.

- CajonAbierto [Booleana]

Esta propiedad es sólo lectura, y refleja el estado del cajón de dinero. Si posee un valor verdadero, el cajón se encuentra abierto.

- HuboErrorFiscal [Booleana]

Refleja si hubo un error de controlador fiscal en el último comando. Sólo lectura.

- HuboErrorMecanico [Booleana]

Refleja si hubo un error mecánico al finalizar el último comando enviado. Sólo lectura.

- HuboFaltaPapel [Booleana]

Refleja si hubo falta de papel al finalizar el último comando enviado. Sólo lectura.

- HuboStatPrn [Booleana]

Refleja si el impresor está ocupado imprimiendo, y no ha finalizado aún el comando enviado. Se puede volver a chequear el estado del impresor usando el comando `EnviarStatPrn`, ya que mientras este flag esté en True, la respuesta no corresponde al comando original enviado sino al estado actual del impresor (ocupado).

- ContadorImpresoraOcupada [Entero Largo]

Muestra la cantidad de veces que el impresor se ha reportado como ocupado durante la ejecución del último comando. Asociado al evento `ImpresoraOcupada`.

- `IndicadorFiscal (Flag) [Booleana]`

Constata si el bit requerido (flag) se halla encendido en la última respuesta fiscal. Los valores posibles son aquellos manejados en los eventos fiscales (`F_FISCAL_MEMORY_FAIL`, etc.). Devuelve `True` o `False` dependiendo si está o no señalado.

- `IndicadorImpresora (Flag) [Booleana]`

Constata si el bit requerido (flag) se halla encendido en la última respuesta fiscal. Los valores posibles son aquellos manejados en los eventos de impresora (`P_PRINTER_ERROR`, etc.). Devuelve `True` o `False` dependiendo si está o no señalado.

- `EstadoControlador [Entero Largo]`

Devuelve el estado extendido del controlador fiscal, si el modelo que se está usando lo soporta. Este estado es un número que indica el estado exacto en el que está el controlador, entre los siguientes:

`S_NONFORMATTED_MEMORY` (Memoria fiscal no formateada)
`S_NONINITIALIZED_MEMORY` (Memoria fiscal no inicializada)
`S_RECEIPT_NOT_OPENED` (No hay ningún comprobante abierto)
`S_FISCAL_RECEIPT_OPENED` (Un comprobante fiscal se encuentra abierto)
`S_FISCAL_TEXT_ISSUED` (Comprobante fiscal abierto. Se acaba de imprimir un texto fiscal)
`S_NONFISCAL_RECEIPT_OPENED` (Un comprobante no fiscal se encuentra abierto)
`S_TENDER` (Comprobante fiscal abierto. Se realizó al menos un pago)
`S_TENDER_CLEARED` (Comprobante fiscal abierto. Se saldó el monto)
`S_PERCEPTION` (Comprobante fiscal abierto. Se ha emitido el comando de percepciones)
`S_KILLED` (El controlador ha sido dado de baja)
`S_RETURN_RECHARGE` (Comprobante fiscal abierto. Se realizó un descuento/recargo general)
`S_DISCOUNT_CHARGE` (Comprobante fiscal abierto. Se realizó una bonificación/recargo/devolución de envases)
`S_RECEIPT_CONCEPT` (Recibo fiscal abierto. Se acaba de imprimir una línea con el concepto del recibo)
`S_CREDIT_NOTE` (Una nota de crédito se encuentra abierta. Se puede hacer un crédito)
`S_CREDIT_NOTE_DISCOUNT` (Nota de crédito abierta. Se realizó un descuento/recargo general)
`S_CREDIT_NOTE_RETURN` (Nota de crédito abierta. Se realizó una bonificación/recargo/devolución de envases)
`S_CREDIT_NOTE_PERCEPTION` (Nota de crédito abierta. Se ha emitido el comando de percepciones)
`S_CREDIT_NOTE_TEXT` (Nota de crédito abierta. Se acaba de escribir una línea de texto)
`S_INTERNAL_USE_RECEIPT` (Recibo de uso interno abierto. Se acaba de imprimir una línea con el concepto del recibo)
`S_QUOTATION` (Una cotización se encuentra abierta)
`S_EMBARK` (Un remito/orden de salida se encuentra abierto)
`S_ACCOUNT` (Un resumen de cuenta / cargo a la habitación se encuentra abierto)

- `DescripcionEstadoControlador [Cadena]`

Devuelve un string conteniendo la descripción del estado actual del controlador, si el modelo que se está usando lo soporta.

Métodos:

1. Métodos de Inicialización/Comandos crudos:

- `Comenzar()`

El primer método a llamarse. Como condición se ha de especificar la propiedad de puerto de comunicaciones asociado antes de llamar a este método. Iniciará la comunicación con el impresor y generará un error atrapable si hay problemas físicos.

Nota: si en el momento de ejecutar este método con la propiedad '*ReintentoConstante*' en True se produce un Error de Time Out (el Controlador Fiscal no responde), automáticamente se generará el evento '*ImpresoraNoResponde*', señalando de este único modo la situación anómala.

- `ReComenzar()`

Verifica si el puerto especificado en la propiedad 'Puerto' ya se encontraba abierto. Generará un error atrapable si el puerto especificado no se encontraba abierto.

Nota: este método no modifica el estado de ninguna de las propiedades del objeto en relación al valor que tenían al momento de ejecutar el método 'Comenzar' previo.

- `Finalizar()`

Cierra el puerto de comunicaciones y libera los recursos asociados. Recomendable llamarlo a la salida del programa.

- `CambiarVelocidad(Velocidad)`

Cambia la velocidad de transmisión del controlador fiscal. Los valores posibles (en baudios) son: 1.200, 2.400, 4.800, 9.600, 19200, 38400 y 57600. A diferencia de la propiedad *Baudios*, este comando le cambia la velocidad al controlador, debiendo ya estar comunicado con él para ejecutarlo. Una vez finalizado el comando, se deberá `Finalizar()` y volver a `Comenzar()` con la velocidad elegida en la propiedad *Baudios*.

- `AutodetectarControlador(Puerto)`

Busca en todas las velocidades posibles un controlador fiscal. Equivale a un `Comenzar()` sabiendo de antemano la velocidad. Si este comando no fracasa, en *Baudios* estará la velocidad detectada, y ya no se necesita usar `Comenzar()`.

Nota: Cada vez que en medio de la ejecución de este método se intenta realizar la conexión con el Controlador Fiscal a una nueva velocidad se dispara el evento '*ProgresoDetección*'.

- `Enviar (Comando)`

Envía un comando crudo al controlador. Se usará sólo en caso que los métodos propuestos por el objeto sean insuficientes.

- `Recibir (Respuesta)`

Recibe la respuesta referente al último comando enviado al controlador fiscal. No es necesario si se usa la propiedad `Respuesta`.

- `PedidoDeStatus ()`

Fuerza un pedido de estado al controlador. Genera todos los eventos asociados y actualiza la respuesta.

- `DescripcionStatusImpresor(Numero), DescripcionStatusFiscal(Numero)`

Dado un número de estado obtenido por intermedio de los eventos `ErrorImpresora` y `EventoImpresora` (`DescripcionStatusImpresor`) o `ErrorFiscal` y `EventoFiscal` (`DescripcionStatusFiscal`), devuelve un string asociado. Si `EventosIndividuales` es `Falso` no se podrá usar el número crudo generado por el evento (hay que separar los bits individuales previamente).

- `Abortar()`

Se utiliza para interrumpir el proceso en curso (interrupción por parte del usuario). Típicamente se utiliza cuando, estando la propiedad `'ReintentoConstante'` seteada en `True`, se produzca una situación de `Time Out` en la comunicación con el Controlador Fiscal y el usuario desee interrumpirla luego de aguardar un número de reintentos por el decidido.

- `EnviarStatPrn()`

Cuando el impresor responde `StatPrn` (reflejado en la propiedad `HuboStatPrn`), la respuesta que volvió no corresponde al comando pedido sino que se informa la situación del impresor en este momento, que está en espera sea porque el comando tardó en ejecutarse por su naturaleza, o sea porque el impresor tiene algún problema de índole mecánica. Se puede volver a preguntar entonces con este comando si la situación ha cambiado, hasta que el flag `HuboStatPrn` no esté más puesto, y se pueda leer la respuesta verdadera.

2. Métodos de Documentos Fiscales:

- `AbrirComprobanteFiscal (Tipo)`

Abre un documento fiscal. El tipo permitido es alguno de los siguientes: `TICKET_C`, `TICKET_FACTURA_A`, `TICKET_FACTURA_B`, `FACTURA_A`, `FACTURA_B`, `RECIBO_A`, `RECIBO_B`, `NOTA_DEBITO_A`, `NOTA_DEBITO_B`. Este comando está condicionado por el modelo de impresor que se posea. Para que funcione en caso de tickets-factura o factura, debe usarse antes el comando `DatosCliente`.

- `ImprimirTextoFiscal (Texto)`

Imprime un texto relacionado con un ítem de venta en un comprobante fiscal. Debe ir obligatoriamente seguido de otro comando de impresión de texto fiscal o de un comando de venta.

- `ImprimirItem (Descripcion, Cantidad, Monto, IVA, ImpuestosInternos)`

Imprime una venta en un comprobante fiscal. La descripción se comportará de acuerdo a la propiedad `DescripcionesLargas`, y los impuestos internos de acuerdo a la propiedad `kIVA`.

- `Subtotal (Imprime)`

Genera un comando de subtotal para el controlador. Si el parámetro booleano `Imprime` es verdadero, se imprime el ticket el subtotal. En cualquier caso, los valores de subtotal pueden ser consultados usando la propiedad `Respuesta`.

- `ImprimirPago (Descripcion, Monto)`

Imprime un pago en el ticket, con los parámetros pasados a esta función. Consultar en el manual del controlador por límites en la cantidad de pagos que se pueden usar.

- `CerrarComprobanteFiscal()`

Cierra un comprobante fiscal. Consultar el manual por restricciones a este comando (el total debe ser positivo, no puede haber contadores de IVA en negativo, etc.)

- `CancelarComprobanteFiscal()`

Cancela un comprobante fiscal, si el estado del controlador lo permite.

- `TratarDeCancelarTodo()`

Vuelve el impresor a un estado conocido, no importa dónde se encuentre. Lo hace por intermedio de cancelaciones y cierres.

- `CancelarComprobante()`

Cancela cualquier tipo de comprobante en curso. Sólo soportado en algunos modelos de controladores.

- `DescuentoUltimoItem (Descripcion, Monto, EnNegativo)`

Realiza un descuento o un recargo sobre el último ítem. El monto se ve afectado por el parámetro booleano `EnNegativo`, que si es verdadero, invertirá el signo del monto. Si el monto resulta ser positivo, se tratará de un recargo, mientras si es negativo será un descuento. Está supeditado al modelo del impresor.

- `DescuentoGeneral (Descripcion, Monto, EnNegativo)`

Realiza un descuento o un recargo sobre el total del comprobante. Verificar anterior comando para su uso.

- `DevolucionDescuento (Descripcion, Monto, IVA, ImpuestosInternos, EnNegativo, Tipo)`

Devuelve envases o genera una venta negativa especial, según lo indicado en el parámetro `Tipo`, que puede asumir alguno de los siguientes valores: `DESCUENTO_RECARGO` o `DEVOLUCION_DE_ENVASES`. El comportamiento es idéntico al comando `DescuentoUltimoItem`.

- `PedirPrimerIVA()`, `PedirSiguienteIVA()`

Recuperan los valores de IVA del último ticket. Consultar la documentación para saber en qué modelos está soportado este comando. Los valores resultantes se recuperan con la propiedad `Respuesta`.

3. Métodos de Facturas Fiscales:

- `DatosCliente (Nombre, NroDocumento, TipoDocumento, ResponsabilidadIVA, Dirección)`

Especifica el cliente relacionado con el próximo ticket-factura a abrir. En `NroDocumento` se espera un número de CUIT o documento (sin espacios ni guiones), cuyo tipo está especificado en el siguiente parámetro según los siguientes valores: `TIPO_CUIT`, `TIPO_LE`, `TIPO_LC`, `TIPO_DNI`, `TIPO_PASAPORTE`, `TIPO_CI` o `TIPO_NINGUNO`. La responsabilidad frente al IVA será alguna de las siguientes: `RESPONSABLE_INSCRIPTO`, `RESPONSABLE_NO_INSCRIPTO`, `RESPONSABLE_EXENTO`, `NO_RESPONSABLE`, `CONSUMIDOR_FINAL`, `BIENES_DE_USO`, `MONOTRIBUTO` o `NO_CATEGORIZADO`. Dirección especificará el domicilio comercial del cliente (es obligatorio en algunos modelos de impresores). Este comando es mandatorio para la apertura de un ticket factura. No todos los tipos de clientes están soportados en todos los controladores.

Nota: el campo de Dirección es obligatorio para los modelos `MODELO_P320` y `MODELO_PJ20`, y no debe cargarse para el resto de los modelos. De no cumplirse cualquiera de estas situaciones según el modelo de controlador, se producirá un error de “*Parámetro opcional no válido o ausente cuando es necesario*”.

- `EspecificarPercepcionPorIVA (Nombre, Monto, IVA)`

Imprime una percepción relacionada con un IVA en particular.

- `EspecificarPercepcionGlobal (Nombre, Monto)`

Imprime una percepción no relacionada con un IVA en particular.

- `EspecificarIVANoInscripto (Monto)`

Cambia el valor de IVA no inscripto calculado por el controlador fiscal.

- `DetalleRecibo (Detalle)`

Ingresa el detalle de un recibo, separándolo en líneas distintas en caso de superar el máximo aceptado por una línea sola.

- `ReimprimirComprobante ()`

Reimprime el último comprobante realizado. Disponible sólo en algunos modelos de controlador.

4. Métodos de Documentos No Fiscales:

- `AbrirComprobanteNoFiscal ()`

Abre un comprobante no fiscal en la estación de tickets.

- `AbrirComprobanteNoFiscalEnSlip ()`

Abre un comprobante no fiscal en la estación de slip, si la impresora tiene dicha estación.

- `ImprimirTextoNoFiscal (Texto)`

Imprime texto no fiscal, atendiendo a la estación donde se haya abierto el comprobante no fiscal.

- `CerrarComprobanteNoFiscal ()`

Cierra el comprobante no fiscal, sea donde fuera que esté abierto.

- `ImprimirVoucher (NombreCliente, NombreTarjeta, Tipo, NumeroDeTarjeta, FechaDeVencimiento, TipoTarjeta, Cuotas, CodigoDeComercio, NumeroDeTerminal, NumeroDeLote, Numero, TipoIngreso, TipoOperacion, NumeroAutorizacion, Monto, NumeroComprobanteAsociado, Copias)`

Imprime un voucher de tarjeta. En `NombreCliente` se espera el nombre del cliente; en `NombreTarjeta`, la marca de tarjeta manejada, mientras que en `Tipo` se especifica el tipo de voucher relacionado con la transacción de tarjeta, entre `VOUCHER_DE_COMPRA`, `VOUCHER_DE_CANCELACION_COMPRA`, `VOUCHER_DE_DEVOLUCION` y `VOUCHER_DE_CANCELACION_DEVOLUCION`. La fecha de vencimiento de la tarjeta va en `FechaDeVencimiento`; se espera el tipo de tarjeta entre los siguientes valores: `TARJETA_CREDITO` y `TARJETA_DEBITO`. El tipo de ingreso puede ser `INGRESO_DE_TARJETA_MANUAL` o `INGRESO_DE_TARJETA_AUTOMATIZADO`. El tipo de operación deberá comprender alguno de los siguientes valores: `OPERACION_TARJETA_ONLINE` o `OPERACION_TARJETA_OFFLINE`. Consultar el manual del controlador para más información acerca de este comando.

- `DNFHReparto(Copias)`

Imprime la cantidad de copias especificada del Documento No Fiscal Homologado de Reparto, según especificación.

- `DNFHFarmacias (Copias)`

Imprime la cantidad de copias especificada del Documento No Fiscal Homologado de Farmacias, según especificación.

- `CortarComprobanteNoFiscal()`

Corta el papel de un comprobante no fiscal.

- `AvanzarPapel (TiposDePapel Papel, Lineas)`

Avanza el papel del impresor. El papel seleccionado será uno de los siguientes: `PAPEL_TICKET`, `PAPEL_DIARIO` o `PAPEL_TICKET_Y_DIARIO`.

- `AbrirComprobanteNoFiscalHomologado(Tipo, Identificacion)`

Abre un comprobante homologado entre los siguientes `NOTA_CREDITO_A`, `NOTA_CREDITO_B`, `REMITO`, `ORDEN_SALIDA`, `RESUMEN_CUENTA`, `CARGO_HABITACION`, `COTIZACION`, o `RECIBO_X`. Sólo disponible en controladores que soporten este tipo de documentos.

- `ImprimirItemEnRemito(Descripcion, Cantidad)`

Imprime un ítem en un remito según descripción y cantidad. Sólo soportado en algunos modelos.

- `ImprimirItemEnCuenta(FechaComprobante, Comprobante, Descripcion, Debe, Haber)`

Imprime un ítem en un documento de habitación o resumen de cuenta. La fecha debe estar en formato estándar OLE. Sólo soportado en algunos modelos.

- `ImprimirItemEnCotizacion(BSTR Detalle)`

Imprime un ítem en un documento de cotización. Sólo soportado en algunos modelos.

- `CerrarComprobanteNoFiscalHomologado()`

Cierra un comprobante abierto con `AbrirComprobanteNoFiscalHomologado()`.

- `InformacionRemito [Propiedad tipo Cadena]`

Devuelve o actualiza la información de remitos para facturas o notas de crédito.

5. Métodos de Reportes/Obtención de Datos:

- `ReporteX()`, `ReporteZ()`

Efectúa un reporte X ó Z en el impresor fiscal.

- `CapacidadRestante()`

Manda un comando de obtener la capacidad de cierres restante al controlador. Consultar la documentación del comando acerca de los campos de la respuesta.

- `ObtenerDatosDeInicializacion()`

Envía el comando al controlador fiscal para obtener en la respuesta los datos de inicialización. Este comando sólo es válido en ciertos controladores.

- `ObtenerDatosDeConfiguracion()`

Envía el comando al controlador fiscal para obtener en la respuesta los datos de configuración del controlador.

- `ReporteZPorNumeros (Desde, Hasta, ReporteGlobal)`

Imprime en el impresor fiscal un reporte de los cierres diarios efectuados entre los números de cierre especificados en los parámetros `Desde` y `Hasta`. En el parámetro booleano `ReporteGlobal` se especifica el deseo de obtener los datos discriminados por cada cierre o globalmente.

- `ReporteZPorFechas (Desde, Hasta, ReporteGlobal)`

Similar a `ReporteZPorNumeros` sólo que está delimitado por dos fechas.

- `ReporteZIndividualPorNumero (Numero)`

Imprime en el impresor fiscal un reporte del cierre diario correspondiente al número dado.

- `ReporteZIndividualPorFecha (Fecha)`

Similar al `ReporteZIndividualPorNumero` solo que esta especificado por fecha.

- `ObtenerVersionDelControlador()`

Devuelve la versión del controlador, si el comando responde a este comando.

- `LeerMemoriaDeTrabajo()`

Devuelve el estado actual de los acumuladores y contadores del controlador. Consultar el manual del controlador para más información acerca de este comando.

6. Métodos Varios:

- `EspecificarNombreDeFantasia (PrimerLinea, SegundaLinea)`

Especifica el nombre de fantasía de la empresa en caso que se desee imprimir otro nombre adicional a la razón social. Se dispone de dos líneas.

- `ObtenerNombreDeFantasia (PrimeraLinea, SegundaLinea)`

Devuelve en `PrimeraLinea` y `SegundaLinea` el nombre de fantasía de la empresa.

- `ImprimirCodigoDeBarras (Tipo, Codigo, ImprimeNumeros, ImprimeAhora)`

Imprime un código de barras tipo `CODIGO_TIPO_EAN_13`, `CODIGO_TIPO_EAN_8`, `CODIGO_TIPO_UPCA` o `CODIGO_TIPO_ITS`. En `Codigo` va el código en sí; en `ImprimeNumeros` se especifica si imprime los números después del código de barras o no, y en `ImprimeAhora` si ha de imprimirse en este momento o al final del ticket. Este comando no está disponible en todos los modelos de impresores, ni los impresores que los soportan soportan todos los tipos: consultar documentación.

- `AbrirCajonDeDinero()`

Abre el cajón de dinero (si está disponible) adjunto al impresor fiscal.

- `MostrarEnDisplay (Texto, Destino)`

Muestra en el display del impresor (si está disponible) un mensaje. Los valores posibles para el parámetro `Destino` son los siguientes: `LINEA_SUPERIOR`, `LINEA_INFERIOR` y `SECCION_DE_REPETICION`. Este último implica la porción de display donde se muestra el contador de repeticiones de artículos, esto es, en la esquina inferior derecha.

- `CargarLogo (NombreArchivo)`

Carga el logo en el controlador, si el modelo lo soporta. El parámetro debe ser el nombre de un archivo de formato bmp según especificación en el manual de cada controlador que sea compatible con este comando. Si el parámetro es una cadena vacía, se asume que se quiere eliminar el logo del controlador.

- `CambiarResponsabilidadIVA (ResponsabilidadIVA)`

La responsabilidad frente al IVA será alguna de las siguientes: `RESPONSABLE_INSCRIPTO`, `RESPONSABLE_NO_INSCRIPTO`, `RESPONSABLE_EXENTO`, `NO_RESPONSABLE`, `CONSUMIDOR_FINAL`, `BIENES_DE_USO`, `MONOTRIBUTO` o `NO_CATEGORIZADO`.

- `ConfigurarControlador (Parametro, Valor)`

El parámetro será alguno de los siguientes: IMPRESION_CAMBIO, IMPRESION_LEYENDAS, CORTE_PAPEL, IMPRESION_MARCO, REIMPRESION_CANCELADOS, COPIAS_DOCUMENTOS, PAGO_SALDO, SONIDO. No todos los impresores todos los tipos: consultar documentación.

- `ConfigurarControladorCompleto (Imprimir, Defaults, [LimiteConsumidorFinal], [LimiteTicketFactura], [PorcentajeIVANoInscripto], [NumeroDeCopias], [ImprimeCambio], [ImprimeLeyendasOpcionales], [TipoDeCorte], [ImprimeMarco], [ReImprimeDocumentos], [DescripcionDelMedioDePago])`

Configura el controlador de forma completa. Los argumentos entre corchetes son opcionales. El primer argumento indica si se debe imprimir un reporte de configuración o no. Si el segundo argumento es verdadero los parámetros omitidos serán reseteados a su valor por defecto, en caso contrario, su valor no será cambiado. Los siguientes argumentos deben ser del tipo string e indican para cada parámetro su valor correspondiente de acuerdo al manual del controlador. Si se quiere omitir un argumento se debe pasar un string vacío. Este comando sólo es soportado por algunos modelos de controladores.

- `ObtenerConfiguracionCompleta ()`

Devuelve los parámetros de configuración. Sólo es soportado por algunos modelos de controladores.

Eventos:

- EventoImpresora (Flags)

Este evento se genera cuando uno de los siguientes eventos está indicado en la respuesta:

P_PRINTER_ERROR : Error mecánico de impresora.
P_OFFLINE : Impresora fuera de línea.
P_BUFFER_FULL : Sin uso/obsoleto.
P_SLIP_PLATEN_OPEN: La tapa de la impresora está abierta.

Nota: Solo para el modelo de Controlador Fiscal P320F, cuando el evento acuse en la respuesta que se produjo por P_PRINTER_ERROR, esto podrá deberse a dos causas reales, Error mecánico de impresión o Falta de Papel .

- EventoFiscal (Flags)

Se genera cuando alguno de las siguientes advertencias del controlador se produce:

F_FISCAL_MEMORY_NEAR_FULL: La memoria está a punto de llenarse.
F_FISCAL_TERMINAL_CERTIFIED: El controlador está inicializado.
F_FISCAL_TERMINAL_FISCALIZED: El controlador está inicializado.
F_RECEIPT_SLIP_OPEN: Hay un recibo abierto en el slip.
F_RECEIPT_OPEN: Hay un recibo abierto.
F_INVOICE_OPEN: Hay una factura abierta.

- ErrorFiscal (Flags)

Se genera al producirse un error irreparable que hizo que el comando recién enviado se rechazara. Las posibles causas son las siguientes:

F_FISCAL_MEMORY_FAIL: Fallo de memoria fiscal.
F_WORKING_MEMORY_FAIL: Fallo de memoria de trabajo.
F_UNRECOGNIZED_COMMAND: Comando no reconocido.
F_INVALID_FIELD_DATA: Campo de datos inválido.
F_INVALID_COMMAND: Comando inválido para el estado fiscal.
F_TOTAL_OVERFLOW: Rebose de totalizadores.
F_FISCAL_MEMORY_FULL: Memoria fiscal llena.
F_DATE_SET_FAIL: Error de puesta de fecha/hora.

- ErrorImpresora (Flags)

Se genera al producirse un error irreparable a cargo de la Impresora que hizo que el comando recién enviado se rechazara. Actualmente no existe ninguna causa de Error de parte de la impresora capaz de generar este evento. Se reserva para aplicaciones futuras.

- EventoCajon (Opened)

Se dispara cuando cambia el estado del cajón de dinero (si está disponible). En el parámetro Opened se verifica si viene abierto (distinto de cero) o cerrado (cero).

- `ImpresoraOcupada()`

Ciertos comandos generan una espera que se ve reflejada cada determinado tiempo con este evento. Esta espera puede deberse a una impresión larga o a que la impresora está parada en un estado que no puede continuar y devolver el control, por ejemplo, un papel enredado en mitad de un comando. Se puede consultar las veces que se ha producido la señal de espera con la propiedad `ContadorImpresoraOcupada`, y mostrar un mensaje adecuado en caso de que esta señal se prolongue por tiempo indeterminado.

- `ImpresoraNoResponde(CantidadReintentos)`

Se dispara cuando, estando la propiedad `'ReintentoConstante'` seteada en `True`, se produce una situación de Time Out en la comunicación con el Controlador Fiscal. `'CantidadReintentos'` indica la cantidad de veces que la comunicación ha fracasado hasta el momento en que ocurrió el evento.

- `ImpresoraOK()`

Se dispara al finalizar la ejecución de algún método en el que en medio de su ejecución se haya producido alguno de los siguientes eventos: `'ImpresoraOcupada'` – `'ImpresoraNoResponde'` – `'FaltaPapel'`.

El fin de este evento es indicar que luego de ocurrido alguno de los eventos arriba señalados, se pudo continuar y concluir con la ejecución del método en curso.

- `FaltaPapel()`

Se genera al acabarse el papel *durante* una impresión.

- `ProgresoDeteccion(Puerto, Velocidad)`

Se dispara cada vez que en medio de la ejecución del método `'AutodetectarControlador'` se intenta realizar la conexión a una nueva velocidad.

Errores Atrapables:

- H_ERR_GENERIC, H_ERR_HANDLER

Error interno de operación con la DLL. Se devuelve cuando hay una desinteligencia entre el componente ActiveX y la DLL fiscal, o cuando el error es genérico y no adjudicable a algún error específico.

- H_ERR_ATOMIC

Intento de ejecutar un comando cuando se estaba ejecutando otro. Esto ocurre cuando antes de esperar el fin de la ejecución del comando se ejecuta otro, típicamente en entornos orientados a eventos.

- H_ERR_TIMEOUT

Error de comunicaciones con el controlador fiscal. Se produce cuando el puerto no tiene un controlador fiscal conectado, el puerto no está funcionando correctamente, el cable no es correcto, el controlador está apagado, etc.

- H_ERR_ALREADYOPEN

El puerto ya estaba abierto y se intentó abrirlo nuevamente. Una posibilidad es que la misma aplicación ya lo tenga abierto y otra es que otra aplicación lo esté usando actualmente.

- H_ERR_NOMEM

No hay suficiente memoria para completar la operación.

- H_ERR_NOTOPENYET

El puerto no ha sido abierto y se le ha enviado un comando. Típicamente esto ocurre cuando no se ha enviado el comando `Comenzar()` y se están ejecutando comandos hacia el controlador.

- H_ERR_INVALIDPTR

El buffer de respuesta es inválido. Este error es interno del ActiveX y no debería ocurrir en condiciones normales.

- H_ERR_ABORT

El proceso en curso fue interrumpido por el usuario. Este error ocurrirá cuando, estando la propiedad *'ReintentoConstante'* seteada en True, se produzca una situación de Time Out en la comunicación con el Controlador Fiscal y el usuario decida interrumpirla utilizando el método *'Abortar'*.

- `H_ERR_FIELD_NOT_FOUND`

El campo de respuesta especificado no existe. Se ha usado la propiedad `Respuesta` con un campo que no existe para la respuesta del último comando. Tener en cuenta que las respuestas varían según el modelo del controlador.

- `H_ERR_INVALID_BIT`

El valor especificado para obtener la descripción de un error o estado no existe. Este error se produce con las propiedades y métodos `DescripcionStatusFiscal`, `DescripcionStatusImpresor` y `DescripcionEstadoControlador`, que reciben un número para devolver una descripción a cambio.

- `H_ERR_PRINTER_NOT_FOUND`

El impresor especificado no existe. Asociado a la propiedad `Modelo`.

- `H_ERR_NOT_SUPPORTED`

Ese método no está soportado por el modelo de impresor seleccionado. Muchas características están ligadas a un modelo de impresor en particular y no funcionan en todos. El intento de usar un comando no compatible con el impresor seleccionado con la propiedad `Modelo` produce este error.

- `H_ERR_NOT_OPENED`

El puerto no ha sido abierto/inicializado aún. No se ha llamado a `Comenzar()` y se ha intentado operar, o se ha ignorado un error de apertura en este mismo comando.

- `H_ERR_INVALID_PORT`

El puerto no ha sido especificado o es inválido. Llamando a `Comenzar()`, el número de puerto no es un número válido o no ha sido especificado en absoluto. Este valor ha de especificarse obligatoriamente antes de llamar a `Comenzar()`.

- `H_ERR_FILENAME`

El nombre de archivo especificado no existe o no puede ser abierto. Este error está asociado a comandos que reciben un nombre de archivo para realizar una tarea, como `CargarLogo()`.

- `H_ERR_FIELD_OPTIONAL`

Un argumento declarado como opcional en un método es requerido.

- `H_ERR_FIELD_INVALID`

Un argumento tiene un valor no esperado.

Constantes:

Durante la descripción de los métodos y eventos se hizo referencia a constantes. Ciertos lenguajes las pueden tomar desde el objeto directamente; otros necesitan declararlas. Esta es la lista de las constantes:

- Eventos Fiscales:

F_FISCAL_MEMORY_FAIL	= 1
F_WORKING_MEMORY_FAIL	= 2
F_ALWAYS_ZERO	= 4
F_UNRECOGNIZED_COMMAND	= 8
F_INVALID_FIELD_DATA	= 16
F_INVALID_COMMAND	= 32
F_TOTAL_OVERFLOW	= 64
F_FISCAL_MEMORY_FULL	= 128
F_FISCAL_MEMORY_NEAR_FULL	= 256
F_FISCAL_TERMINAL_CERTIFIED	= 512
F_FISCAL_TERMINAL_FISCALIZED	= 1024
F_DATE_SET_FAIL	= 2048
F_RECEIPT_SLIP_OPEN	= 4096
F_RECEIPT_OPEN	= 8192
F_INVOICE_OPEN	= 16384

- Eventos de Impresora:

P_PRINTER_ERROR	= 4
P_OFFLINE	= 8
P_JOURNAL_PAPER_LOW	= 16
P_RECEIPT_PAPER_LOW	= 32
P_BUFFER_FULL	= 64
P_BUFFER_EMPTY	= 128
P_SLIP_PLATEN_OPEN	= 256
P_DRAWER_CLOSED	= 16384

- Modelos de Impresoras:

MODELO_614	= 1
MODELO_615	= 2
MODELO_PR4	= 3
MODELO_950	= 4
MODELO_951	= 5
MODELO_262	= 6
MODELO_PJ20	= 7
MODELO_P320	= 8

- Tipos de Documentos Fiscales:

TICKET_C	= 84
TICKET_FACTURA_A	= 65
TICKET_FACTURA_B	= 66
FACTURA_A	= 48
FACTURA_B	= 49
RECIBO_A	= 97
RECIBO_B	= 98
NOTA_DEBITO_A	= 68
NOTA_DEBITO_B	= 69

- Tipos de Documentos No Fiscales:

NOTA_CREDITO_A	= 82
NOTA_CREDITO_B	= 83
REMITO	= 114
ORDEN_SALIDA	= 115
RESUMEN_CUENTA	= 116
CARGO_HABITACION	= 85
COTIZACION	= 117
RECIBO_X	= 120

- Estado de controlador:

S_NONFORMATTED_MEMORY	= 0
S_NONINITIALIZED_MEMORY	= 1
S_RECEIPT_NOT_OPENED	= 2
S_FISCAL_RECEIPT_OPENED	= 3
S_FISCAL_TEXT_ISSUED	= 4
S_NONFISCAL_RECEIPT_OPENED	= 5
S_TENDER	= 6
S_TENDER_CLEARED	= 7
S_PERCEPTION	= 10
S_KILLED	= 11
S_RETURN_RECHARGE	= 10
S_DISCOUNT_CHARGE	= 11
S_RECEIPT_CONCEPT	= 12
S_CREDIT_NOTE	= 13
S_CREDIT_NOTE_DISCOUNT	= 14
S_CREDIT_NOTE_RETURN	= 15
S_CREDIT_NOTE_PERCEPTION	= 16
S_CREDIT_NOTE_TEXT	= 17
S_INTERNAL_USE_RECEIPT	= 18
S_QUOTATION	= 19
S_EMBARK	= 20
S_ACCOUNT	= 21
S_BLOCKED	= 22
S_NONE	= 255

- Tipos de Documentos de Clientes:

TIPO_CUIT	= 67
TIPO_LE	= 48
TIPO_LC	= 49
TIPO_DNI	= 50
TIPO_PASAPORTE	= 51
TIPO_CI	= 52
TIPO_NINGUNO	= 32

- Responsabilidades de Clientes:

RESPONSABLE_INSCRIPTO	= 73
RESPONSABLE_NO_INSCRIPTO	= 78
RESPONSABLE_EXENTO	= 69
NO_RESPONSABLE	= 65
CONSUMIDOR_FINAL	= 67
BIENES_DE_USO	= 66
MONOTRIBUTO	= 77
NO_CATEGORIZADO	= 84

- Líneas de Display:

LINEA_SUPERIOR	= 76
LINEA_INFERIOR	= 75
SECCION_DE_REPETICION	= 78

- Tipos de Papel:

PAPEL_TICKET	= 0
PAPEL_DIARIO	= 1
PAPEL_TICKET_Y_DIARIO	= 2

- Tipos de Vouchers:

VOUCHER_DE_COMPRA	= 67
VOUCHER_DE_CANCELACION_COMPRA	= 86
VOUCHER_DE_DEVOLUCION	= 68
VOUCHER_DE_CANCELACION_DEVOLUCION	= 65

- Tipos de Tarjetas:

TARJETA_CREDITO	= 67
TARJETA_DEBITO	= 68

- Tipos de Ingreso de Tarjetas:

INGRESO_DE_TARJETA_MANUAL	= 42
INGRESO_DE_TARJETA_AUTOMATIZADO	= 32

- Tipos de Operaciones de Tarjetas:

OPERACION_TARJETA_ONLINE	= 78
--------------------------	------

OPERACION_TARJETA_OFFLINE = 70

- Tipos de Descuento:

DESCUENTO_RECARGO = 66
DEVOLUCION_DE_ENVASES = 69

- Tipos de Parámetros de configuración:

IMPRESION_CAMBIO = 52
IMPRESION_LEYENDAS = 53
CORTE_PAPEL = 54
IMPRESION_MARCO = 55
REIMPRESION_CANCELADOS = 56
COPIAS_DOCUMENTOS = 57
PAGO_SALDO = 58
SONIDO = 59

- Errores: (expresados en hexadecimal)

H_ERR_GENERIC = 80040201
H_ERR_HANDLER = 80040202
H_ERR_ATOMIC = 80040203
H_ERR_TIMEOUT = 80040204
H_ERR_ALREADYOPEN = 80040205
H_ERR_NOMEM = 80040206
H_ERR_NOTOPENYET = 80040207
H_ERR_INVALIDPTR = 80040208
H_ERR_ABORT = 8004020A
H_ERR_FIELD_NOT_FOUND = 80040232
H_ERR_INVALID_BUFFER = 80040233
H_ERR_INVALID_BIT = 80040234
H_ERR_PRINTER_NOT_FOUND = 80040235
H_ERR_NOT_SUPPORTED = 80040236
H_ERR_NOT_OPENED = 80040237
H_ERR_INVALID_PORT = 80040238
H_ERR_FILENAME = 80040239
H_ERR_FIELD_OPTIONAL = 80040240
H_ERR_FIELD_INVALID = 80040241

- Tipos de Códigos de Barras:

CODIGO_TIPO_EAN_13 = 49
CODIGO_TIPO_EAN_8 = 50
CODIGO_TIPO_UPCA = 51
CODIGO_TIPO_ITS = 52