
Impresoras Fiscales HASAR
OCX Fiscal Hasar: Guía de Referencia
Rev. 001



Cía. HASAR SAIC
www.hasar.com

ventas@hasar.com
www.grupohasar.com

Marcos Sastre Esq. José Ingenieros
(B1618CSD) El Talar - Gral. Pacheco
Pcia. de Bs. As - República Argentina

TE: (54)(011) 4727-8900
FAX: (54)(011) 4736-7300

Copyright © 1997/2006 – Cía. HASAR SAIC

El presente documento se halla sujeto a cambios sin previo aviso.

Cía. HASAR SAIC no asume responsabilidad alguna por errores u omisiones contenidas en este documento, ni asume responsabilidad alguna por los daños y/o perjuicios que el uso de esta información pudiera causar.

Este documento no puede ser reproducido, total o parcialmente, ni almacenado para su posterior reproducción por cualquier método o medio, sin autorización escrita de **Cía. HASAR SAIC**.

CONTENIDO GENERAL

OCX PARA IMPRESORAS FISCALES HASAR	6
EVOLUCIÓN DEL OCX FISCAL HASAR	6
<i>Fiscal010724.ocx</i>	7
Registrando <i>fiscal010724.ocx</i>	7
<i>Fiscal051122.ocx</i>	7
Registrando <i>fiscal051122.ocx</i>	9
OCX FISCAL HASAR Y DESARROLLOS DE SOFTWARE	9
<i>Caso Visual Basic</i>	9
<i>Caso Visual FoxPro 6.0</i>	11
OBSERVACIONES GENERALES	12
IMPRESORAS FISCALES HASAR SOPORTADAS	13
MÉTODOS DEL OCX FISCAL HASAR	14
DESCRIPCIÓN DE LOS MÉTODOS DEL OCX FISCAL HASAR	16
<i>Método: Abortar()</i>	17
<i>Método: AbrirCajonDeDinero()</i>	17
<i>Método: AbrirComprobanteFiscal()</i>	18
<i>Método: AbrirComprobanteNoFiscal()</i>	19
<i>Método: AbrirComprobanteNoFiscalEnSlip()</i>	20
<i>Método: AbrirComprobanteNoFiscalHomologado()</i>	20
<i>Método: AutodetectarControlador()</i>	22
<i>Método: AutodetectarModelo()</i>	23
<i>Método: AvanzarPapel()</i>	24
<i>Método: BajaFiscal()</i>	24
<i>Método: CambiarCodigoIngresosBrutos()</i>	25
<i>Método: CambiaFechaInicioActividades()</i>	26
<i>Método: CambiarResponsabilidadIVA()</i>	26
<i>Método: CambiarVelocidad()</i>	27
<i>Método: CancelarComprobante()</i>	28
<i>Método: CancelarComprobanteFiscal()</i>	28
<i>Método: Capacidades()</i>	28
Capacidades soportadas por las impresoras fiscales Hasar	30
<i>Método: CapacidadRestante()</i>	30
<i>Método: CargarLogo()</i>	31
<i>Método: CerrarComprobanteFiscal()</i>	32
<i>Método: CerrarComprobanteNoFiscal()</i>	33
<i>Método: CerrarComprobanteNoFiscalHomologado()</i>	33
<i>Método: Comenzar()</i>	34
<i>Método: ConfigurarControlador()</i>	36
<i>Método: ConfigurarControladorCompleto()</i>	37
<i>Método: ConfigurarControladorPorBloque()</i>	41
<i>Método: CortarComprobanteNoFiscal()</i>	43
<i>Método: DatosCliente()</i>	43
<i>Método: DescripcionStatusFiscal()</i>	45
<i>Método: DescripcionStatusImpresor()</i>	46
<i>Método: DescuentoGeneral()</i>	47

Método: DescuentoUltimoItem()	48
Método: DetalleRecibo()	49
Método: DevolucionDescuento()	49
Método: DNFHFarmacias()	50
Método: DNFHReparto()	51
Método: Enviar()	51
Método: EnviarStatPrn()	53
Método: EspecificarIVANoInscripto()	55
Método: EspecificarNombreDeFantasia()	56
Método: EspecificarPercepcionGlobal()	56
Método: EspecificarPercepcionPorIVA()	57
Método: Finalizar()	57
Método: ImprimirCodigoDeBarras()	58
Método: ImprimirInfoEnDNFH()	59
Método: ImprimirItem()	61
Items negativos	67
Método: ImprimirItemEnCotizacion()	67
Método: ImprimirItemEnCuenta()	68
Método: ImprimirItemEnRemito()	69
Método: ImprimirPago()	69
Método: ImprimirTextoFiscal()	71
Método: ImprimirTextoNoFiscal()	72
Método: ImprimirVoucher()	73
PROPIEDADES DEL OCX FISCAL HASAR	81
EVENTOS QUE GENERA EL OCX FISCAL HASAR	83
EXCEPCIONES QUE GENERA EL OCX FISCAL HASAR	84
DESCRIPCIÓN DE LAS EXCEPCIONES	84
H_ERR_ABORT	84
H_ERR_ALREADYOPEN	85
H_ERR_ATOMIC	85
H_ERR_BIND_SOCKET	85
H_ERR_CF_NOT_DETECTED	85
H_ERR_CONNECTION	86
H_ERR_DEPRECATED_FUNCTION	86
H_ERR_FIELD_INVALID	86
H_ERR_FIELD_NOT_FOUND	86
H_ERR_FIELD_OPTIONAL	87
H_ERR_FILENAME	87
H_ERR_GENERIC	87
H_ERR_HANDLER	88
H_ERR_INIT_SOCKET	88
H_ERR_INVALID_BIT	88
H_ERR_INVALID_BUFFER	88
H_ERR_INVALID_PORT	89
H_ERR_INVALIDPTR	89
H_ERR_INVALID_DATE	89
H_ERR_INVALID_PARAMETER	89
H_ERR_INVALID_STATE	93
H_ERR_INVALID_TIME	94
H_ERR_IP_FORMAT	94
H_ERR_MANDATORY_PARAMETER	94
H_ERR_MAX_RETRIES	94

<i>H_ERR_NO_MORE_ITEMS</i> -----	95
<i>H_ERR_NOMEM</i> -----	95
<i>H_ERR_NOT_EXECUTED</i> -----	95
<i>H_ERR_NOT_OPEN</i> -----	95
<i>H_ERR_NOT_SUPPORTED</i> -----	95
<i>H_ERR_NOTOPENYET</i> -----	96
<i>H_ERR_OPEN</i> -----	96
<i>H_ERR_PRINTER_NOT_FOUND</i> -----	97
<i>H_ERR_READ</i> -----	97
<i>H_ERR_SERIAL_PARAMETERS</i> -----	97
<i>H_ERR_SOCKET_SELECT</i> -----	97
<i>H_ERR_STATPRN</i> -----	97
<i>H_ERR_TIMEOUT</i> -----	98
<i>H_ERR_VERSION_SOCKET</i> -----	98
<i>H_ERR_WRITE</i> -----	98
<i>H_ERR_WRONG_PROTOCOL</i> -----	98

OCX para Impresoras Fiscales HASAR

Hasta el momento existen dos versiones liberadas del OCX Fiscal Hasar (objeto ActiveX -de distribución gratuita-); ambas se documentan en la presente guía de referencia:

Fiscal010724.ocx
Fiscal051122.ocx

Notar que el nombre del OCX Fiscal Hasar es siempre de la forma: *fiscal***AAMMDD**.ocx (**AAMMDD** es la fecha de liberación de la versión alfa del OCX Fiscal Hasar)

Así definidos los nombres para las distintas versiones del OCX Fiscal Hasar los desarrolladores de software de gestión para puntos de venta pueden identificar fácilmente la fecha de su aparición (incluida en el nombre): 24/07/2001 en el primer caso, y 22/11/2005 en el segundo.

En el sitio web del Grupo HASAR (www.grupohasar.com), en particular en la página de Cía.HASAR SAIC (www.hasar.com), los archivos zipeados de drivers fiscales incluyen la última versión liberada del OCX Fiscal Hasar (además de la presente guía de referencia y un material extenso, de interés para desarrolladores de software de gestión para puntos de venta).

Evolución del OCX Fiscal Hasar

Si bien existe una versión primaria llamada simplemente *fiscal.ocx*, se recomienda migrar en una primera etapa al uso del OCX Fiscal Hasar llamado *fiscal010724.ocx* (recomendado y compatible con la versión primaria *fiscal.ocx*).

Independientemente de la versión del OCX Fiscal Hasar en uso, este objeto ActiveX no aplica las restricciones impuestas por la normativa fiscal vigente. Se limita simplemente a generar los strings de comandos que el software de gestión para puntos de venta necesite enviar a las impresoras fiscales a partir de los datos que dicho software le pase, generando posteriormente los eventos y excepciones que surjan por rechazos a dichos comandos, y/u otros motivos.

El desarrollador de software de gestión para puntos de venta debe completar la consulta de esta guía de referencia con la lectura obligatoria del manual de comandos (**publ....pdf**) de cada impresora fiscal Hasar que su software soportare para saber si

hay diferencias en el uso de comandos al pasar de un modelo a otro, si el set de comandos soportados por cada impresora fiscal es diferente, etc.

Es responsabilidad del software de gestión del punto de venta garantizar que los datos resulten válidos para la impresora fiscal, y que los comandos se envíen en el orden correcto de acuerdo al documento que se necesite emitir.

Este objeto ActiveX ofrece elementos de control y consulta suficientes para que el software se entere del rechazo de comandos, del motivo, del estado en que la impresora fiscal Hasar se encuentra, y de cualquiera otra información de interés para el software. Obviamente, el OCX Fiscal Hasar no puede reportar lo que la propia impresora fiscal Hasar no informa.

Fiscal010724.ocx

Se trata de un control OCX desarrollado con tecnología "COM" e insertable en cualquier entorno de 32 bits que soporte objetos ActiveX de este tipo.

Esta versión del OCX Fiscal Hasar funciona como un "wrapper" alrededor de una DLL de 32 bits (*winfis32.dll* -distribuida también gratuitamente por Cía.HASAR SAIC-), motivo por el cual es condición necesaria la presencia de esta DLL tanto para registrar, como para distribuir el software de gestión para puntos de venta.

Registrando fiscal010724.ocx

Como primer paso la DLL (*winfis32*) debe copiarse en *c:\windows\system(32)*, o en la carpeta correspondiente a la versión de Windows® en uso. Esta librería no debe registrarse, solamente debe estar presente.

En segundo lugar debe registrarse el OCX Fiscal Hasar en el sistema por intermedio de algún utilitario de registración apropiado. Por ejemplo, ejecutando sin parámetro alguno la siguiente sentencia:

```
regsvr32  fiscal010724.ocx
```

Fiscal051122.ocx

Esta versión del OCX Fiscal Hasar se distingue de la anterior en muchos aspectos, comenzando por su independencia de la DLL (*winfis32*) evitando así el uso, y consi-

guiente distribución e instalación de la librería. *Fiscal051122.ocx* ahora encapsula una librería estática de clases polimórficas (cada clase representa un modelo de impresora fiscal Hasar, supeditado a una interfaz uniforme, de manera de poder utilizar cada modelo independientemente de sus diferencias).

- ✓ Se amplía a veintiseis (26) la cantidad de modelos de impresoras fiscales Hasar soportadas naturalmente por esta versión del OCX Fiscal Hasar.
 - ✓ Permite determinar el modelo de impresora fiscal Hasar conectada físicamente, sin necesidad de selección manual.
 - ✓ Permite comprobar, para el caso de selección manual, si el modelo de impresora fiscal Hasar elegido por el usuario se corresponde con el conectado físicamente.
 - ✓ Incorpora la multiplicidad de transportes para comunicaciones. Además de un mejorado manejo de puertos serie (RS-232), se puede utilizar *fiscal051122.ocx* para comunicarse con impresoras fiscales Hasar remotas a través del protocolo TCP/IP, sea utilizando TCP o UDP. Además, es posible ajustar las variables del protocolo para optimizar el uso en condiciones atípicas (tiempo para salida por error de "time out", cantidad de reintentos, etc.).
 - ✓ Presenta mejoras en propiedades y métodos, como los utilizados para obtener discriminaciones de documentos, que permiten escribir programas más elegantes.
 - ✓ Ofrece respuestas estructuradas. Toda información devuelta por la impresora fiscal Hasar es retornada como argumento opcional de salida en cada método, eliminando de esta manera la inspección manual de la respuesta cruda y minimizando las posibilidades de error de las aplicaciones.
 - ✓ Nuevas propiedades y métodos que permiten el manejo completo de todos los sets de comandos soportados por la línea de impresoras fiscales Hasar homologadas hasta la aparición de esta versión del OCX Fiscal Hasar.
 - ✓ Permite consultar las capacidades del modelo de impresora fiscal Hasar conectada, pudiendo el software de gestión para puntos de venta decidir en tiempo de ejecución, por ejemplo, qué tipos de documentos es posible emitir con dicha impresora fiscal Hasar.
 - ✓ Provee conversión automática de juegos de caracteres ANSI (utilizado en los sistemas operativos modernos) al juego de caracteres ASCII (utilizado por los controladores fiscales), y viceversa.
 - ✓ Mejoramiento de la generación de eventos y excepciones para facilitar el tratamiento de errores por parte del software de gestión para puntos de venta.
-

Registrando fiscal051122.ocx

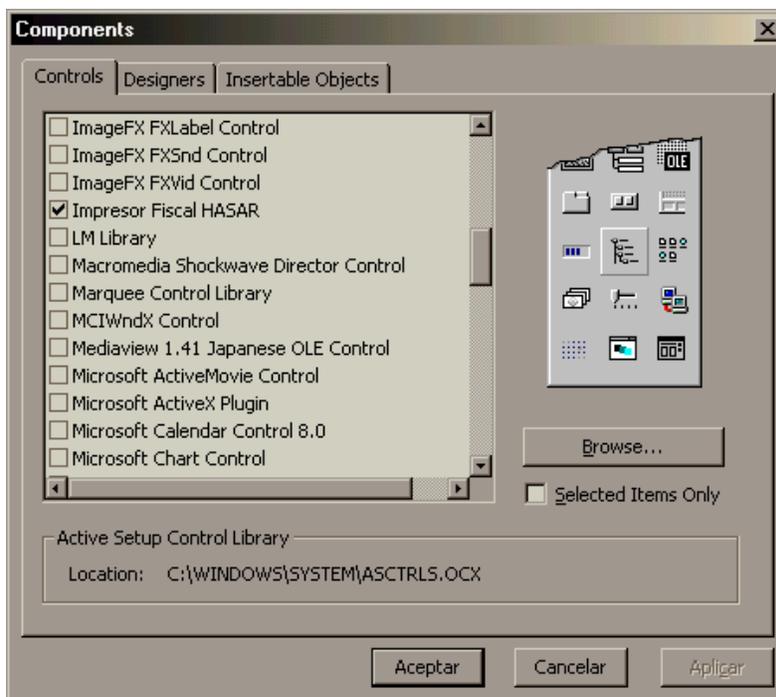
Esta versión del objeto ActiveX no requiere de la DLL (*winfis32*). El OCX Fiscal Hasar solamente debe registrarse en el sistema por intermedio de algún utilitario de registración apropiado. Por ejemplo, ejecutando sin parámetro alguno la siguiente sentencia:

```
regsvr32  fiscal051122.ocx
```

OCX Fiscal Hasar y desarrollos de software

Una vez registrado adecuadamente se puede encontrar al OCX Fiscal Hasar entre los demás objetos ActiveX listados por el entorno de desarrollo e insertarlo en un programa.

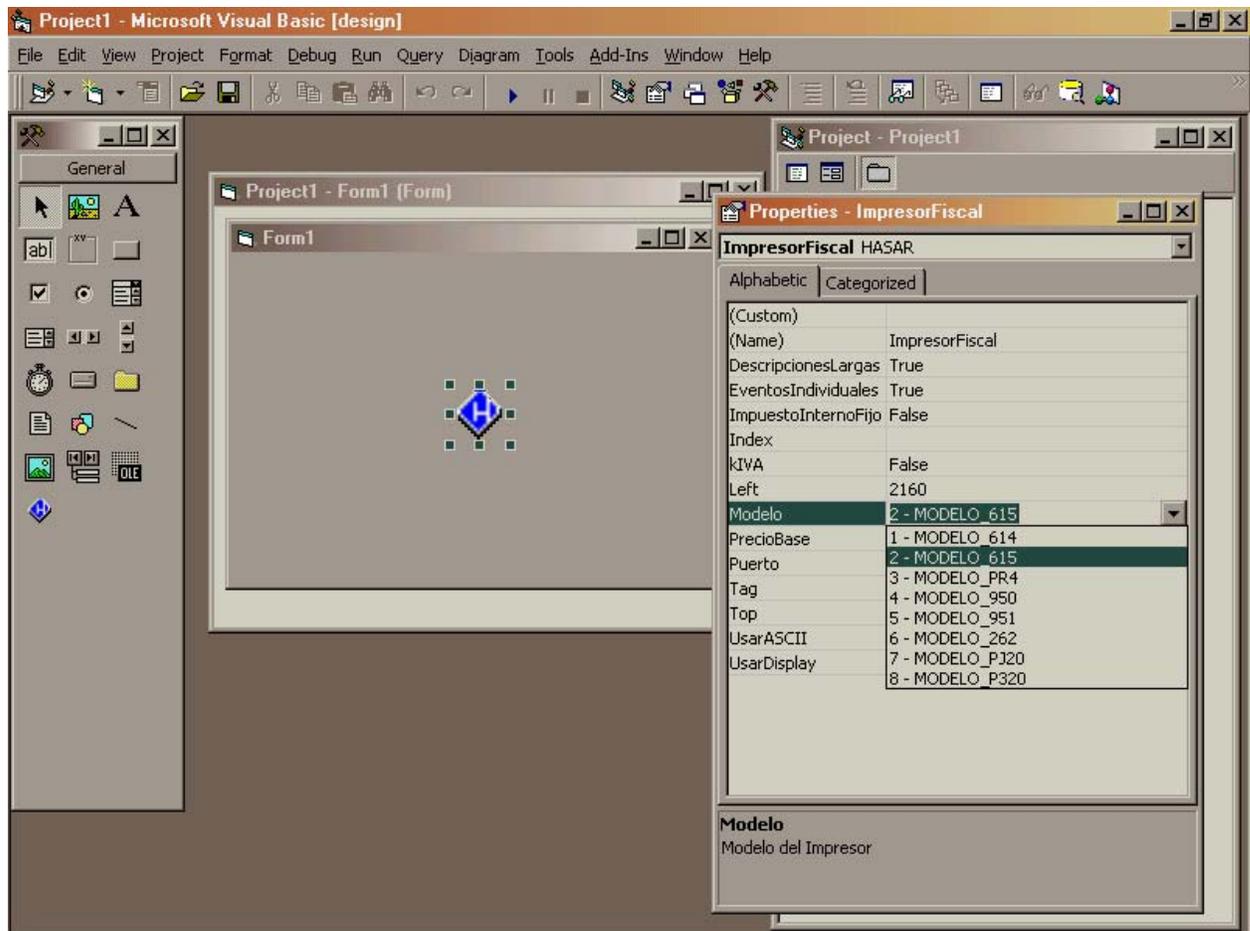
Caso Visual Basic



En este entorno de desarrollo se podrá encontrar al OCX Fiscal Hasar en la ventana de componentes, cuyo aspecto es similar a la imagen de la izquierda.

Si el OCX Fiscal Hasar es seleccionado, podrá encontrárselo en la barra de herramientas de Visual Basic como un ícono más.

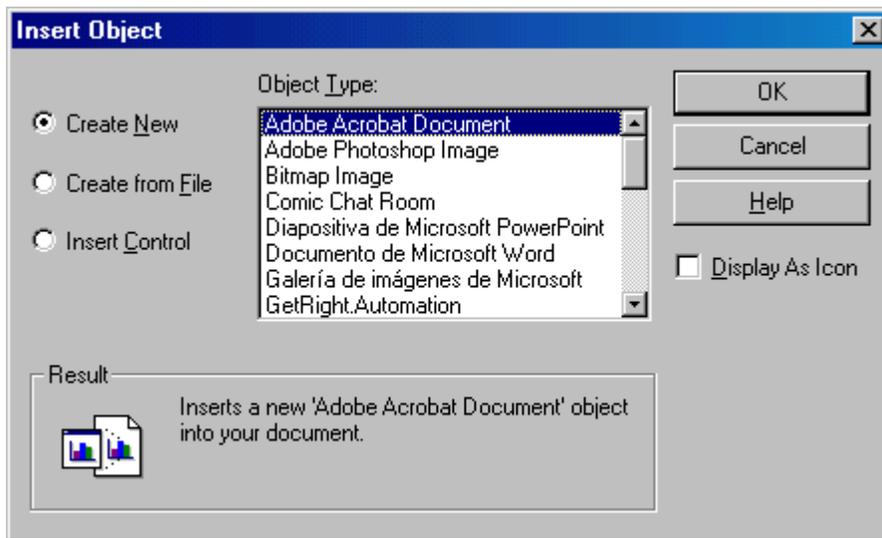
La barra de herramientas mencionada tiene un aspecto similar a la imagen de la página siguiente:



El siguiente es un breve ejemplo que muestra en pocas líneas cómo imprimir un ticket a Consumidor Final, utilizando el OCX Fiscal Hasar.

```
Hasar1.Modelo = MODELO_615  
Hasar1.Puerto = 1  
Hasar1.Comenzar  
Hasar1.TratarDeCancelarTodo  
Hasar1.AbrirComprobanteFiscal TICKET_C  
Hasar1.ImprimirItem "Artículo 1", 1, 100, 21, 0  
Hasar1.Subtotal True  
Hasar1.ImprimirPago "Efectivo", 100  
Hasar1.CerrarComprobanteFiscal  
Hasar1.Finalizar
```

Caso Visual FoxPro 6.0



En primera instancia debe insertarse el OCX Fiscal Hasar en un formulario del proyecto. Luego, debe arrastrarse desde la barra de herramientas (toolbar) el ícono denominado “ActiveX Control (OleControl)” apareciendo así la ventana cuya imagen se ve a la izquierda:

Seleccionando la opción “Insert Control”, luego de unos segundos se mostrará una lista de los objetos ActiveX registrados; entre ellos el componente “Fiscal Class”.

Otra forma de lograr el mismo resultado es a partir de la galería de componentes, eligiendo al OCX Fiscal Hasar desde la carpeta “ActiveX”.

Según el propio help de Visual FoxPro 6.0: “The AutoYield property should be set to False when a form contains an ActiveX control”.

En consecuencia, es importante tener en cuenta que el software de gestión para puntos de venta podrá recibir todos los eventos generados por el OCX Fiscal Hasar (a partir de las respuestas de la impresora fiscal Hasar) siempre y cuando la propiedad *AutoYield* (de Visual FoxPro) se encuentre seteada con el valor **.F.** (falso). De lo contrario, el manejo de errores por parte del software de gestión para puntos de venta se verá complicado. Se trata de una limitación del propio lenguaje de desarrollo.

El siguiente ejemplo permite cumplir con lo expuesto:

```

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 análogo al mostrado en el caso de Visual Basic, para emitir un tique a Consumidor Final, utilizando el OCX Fiscal Hasar.

```

TICKET_C = 84
MODELO_615 = 2

THISFORM.Olecontrol1.Puerto = 1
THISFORM.Olecontrol1.Modelo = MODELO_615
THISFORM.Olecontrol1.Comenzar
THISFORM.OleControl1.TratarDeCancelarTodo
THISFORM.Olecontrol1.AbrirComprobanteFiscal (TICKET_C)
THISFORM.Olecontrol1.ImprimirItem ("Artículo 1", 1, 100, 21, 0)
THISFORM.Olecontrol1.Subtotal (.T.)
THISFORM.Olecontrol1.ImprimirPago ("Efectivo", 100)
THISFORM.Olecontrol1.CerrarComprobanteFiscal
THISFORM.Olecontrol1.Finalizar

```

Observaciones generales

El uso del OCX Fiscal Hasar está limitado al nivel de compatibilidad de cada lenguaje de desarrollo con la norma ActiveX para diseño de objetos. Por ejemplo, el reconocimiento de constantes embebidas en el objeto (TICKET_C, por ejemplo), el soporte de eventos, los tipos especiales (formato de fecha OLE), uso de propiedades con argumentos múltiples, etc., forman parte del grupo de características que pueden no estar soportadas por todos los lenguajes de desarrollo visual. El desarrollador deberá consultar al respecto en la documentación correspondiente a su entorno de desarrollo.

Impresoras fiscales Hasar soportadas

Impresora Fiscal Hasar		Versiones del OCX Fiscal Hasar	
SMH/	firmware	010724	051122
P-950F	1.00	SI	SI
P-951F	1.00	SI	SI
P-614F	1.00	SI	SI
P-615F	1.00	SI	SI
P-715F	1.00	---	SI
""	2.01	---	SI
PT-262F	1.00	SI	SI
PT-272F	1.00	---	SI
P-PR4F	1.00	SI	SI
P-PR5F	1.00	---	SI
""	2.01	---	SI
PL-8F	1.00	---	SI
""	2.01	---	SI
PL-9F	1.00	---	SI
PJ-20F	1.00	SI	SI
P-320F	1.00	SI	SI
P-321F	1.00	---	SI
P-322F	1.00	---	SI
""	2.01	---	SI
P-330F	1.00	---	SI
""	2.01	---	SI
""	2.02	---	SI
P-425F	1.00	---	SI
""	2.01	---	SI
P-435F	1.00	---	SI
""	1.01	---	SI

Los modelos de impresoras fiscales Hasar homologadas y no soportados por el OCX Fiscal Hasar igualmente pueden ser manejados por el software de gestión para puntos de venta mediante este objeto Activex (ver método *Enviar()*).

Métodos del OCX Fiscal Hasar

Métodos del OCX Fiscal Hasar	Versión OCX Fiscal Hasar	
	010724	051122
Abortar	SI	SI
AbrirCajonDeDinero	SI	SI
AbrirComprobanteFiscal	SI	SI
AbrirComprobanteNoFiscal	SI	SI
AbrirComprobanteNoFiscalEnSlip	SI	:::
AbrirComprobanteNoFiscalHomologado	SI	SI
AutodetectarControlador	SI	SI
AutodetectarModelo	---	SI
AvanzarPapel	SI	SI
BajaFiscal	SI	SI
CambiarCodigoIngresosBrutos	---	SI
CambiaFechaInicioActividades	---	SI
CambiarResponsabilidadIVA	---	SI
CambiarVelocidad	SI	SI
CancelarComprobante	SI	SI
CancelarComprobanteFiscal	SI	SI
Capacidades	---	SI
CapacidadRestante	SI	SI
CargarLogo	SI	SI
CerrarComprobanteFiscal	SI	SI
CerrarComprobanteNoFiscal	SI	SI
CerrarComprobanteNoFiscalHomologado	SI	SI
Comenzar	SI	SI
ConfigurarControlador	SI	SI
ConfigurarControladorCompleto	SI	SI
ConfigurarControladorPorBloque	SI	SI
CortarComprobanteNoFiscal	SI	SI
DatosCliente	SI	SI
DescripcionStatusFiscal	SI	SI
DescripcionStatusImpresor	SI	SI
DescuentoGeneral	SI	SI
DescuentoUltimoItem	SI	SI
DetalleRecibo	SI	SI
DevolucionDescuento	SI	SI
DNFHFarmacias	SI	SI
DNFHReparto	SI	SI
Enviar	SI	SI

Métodos del control OCX	Versión control OCX	
	010724	051122
EnviarStatPrn	SI	SI
EspecificarIVANoInscripto	SI	SI
EspecificarNombreDeFantasia	SI	::
EspecificarPercepcionGlobal	SI	SI
EspecificarPercepcionPorIVA	SI	SI
Finalizar	SI	SI
ImprimirCodigoDeBarras	SI	SI
ImprimirInfoEnDNFH	---	SI
ImprimirItem	SI	SI
ImprimirItemEnCotizacion	SI	SI
ImprimirItemEnCuenta	SI	SI
ImprimirItemEnRemito	SI	SI
ImprimirPago	SI	SI
ImprimirTextoFiscal	SI	SI
ImprimirTextoNoFiscal	SI	SI
ImprimirVoucher	SI	SI
LeerIVA	---	SI
LeerMemoriaDeTrabajo	SI	SI
LeerMontoTotal	---	SI
LeerPercepcionesGenerales	---	SI
LeerPercepcionesPorDescripcion	---	SI
LeerPercepcionesPorIVA	---	SI
MostrarEnDisplay	SI	SI
ObtenerCAIUltimoDocumentoA	---	SI
ObtenerConfiguracion	SI	SI
ObtenerConfiguracionCompleta	SI	::
ObtenerDatosDeConfiguracion	SI	::
ObtenerDatosDeInicializacion	SI	SI
ObtenerNombreDeFantasia	SI	::
ObtenerVersionDelControlador	SI	SI
PedidoDeStatus	SI	SI
PedirPrimerIVA	SI	::
PedirSiguienteIVA	SI	::
Recibir	SI	SI
ReComenzar	SI	::
ReImprimirComprobante	SI	SI
ReporteX	SI	SI
ReporteZ	SI	SI
ReporteZIndividualPorFecha	SI	SI
ReporteZIndividualPorNumero	SI	SI
ReporteZPorFechas	SI	SI
ReporteZPorNumeros	SI	SI

Métodos del control OCX	Versión control OCX	
	010724	051122
Subtotal	SI	SI
TratarDeCancelarTodo	SI	SI
VerificarModelo	---	SI

--- Esta indicación en la tabla anterior significa que si se intenta ejecutar una sentencia de programa donde se invoca al método del OCX Fiscal Hasar se producirá una excepción atrapable por el software de gestión para puntos de venta (por ejemplo, en Visual Basic por vía de la sentencia "On Error...").

Código de excepción del entorno: 438 (dec)
1B6 (hex)

Mensaje asociado: "El objeto no admite esta propiedad o método"

Por ejemplo: HASAR1.LeerIVA(...) ; usando *fiscal010724.ocx*

::: Esta indicación en la tabla anterior significa que si se intenta ejecutar una sentencia de programa donde se invoca al método del OCX Fiscal Hasar se producirá una excepción atrapable por el software de gestión para puntos de venta (por ejemplo, en Visual Basic por vía de la sentencia "On Error...").

Código de excepción del OCX Fiscal Hasar: -2147220926 (dec)
80040242 (hex)

Constante embebida: H_ERR_DEPRECATED_FUNCTION
Mensaje asociado: "Función discontinuada"

Por ejemplo: HASAR1.PedirPrimerIVA ; usando *fiscal051122.ocx*

Descripción de los métodos del OCX Fiscal Hasar

La mayor parte de los métodos del OCX Fiscal Hasar generan, a partir de los argumentos pasados por el software de gestión para puntos de venta, un string de comando que también se envía a la impresora fiscal Hasar conectada. El envío del comando es transparente para el software de gestión para puntos de venta.

Es responsabilidad del desarrollador consultar, además de esta guía de referencia, el manual de comandos de la impresora fiscal para informarse de lo esperado por la impresora fiscal Hasar en cada comando, de lo que puede entregar como respuesta,

en que casos puede ser rechazado, del comportamiento de la impresora fiscal Hasar, etc., etc..

Es responsabilidad del software de gestión para puntos de venta garantizar la exactitud de la información pasada en los argumentos, así como el tratamiento de los eventos y excepciones que pudieran generarse con motivo de la invocación a un método del OCX Fiscal Hasar.

- ✓ Esta marca indica que el método está disponible en la versión del OCX Fiscal Hasar especificada.
- ⊖ Esta marca indica que el método no está disponible en la versión del OCX Fiscal Hasar especificada.

Método: *Abortar()*

010724	✓
051122	✓

Aborta la espera de respuesta en curso.

Este método del OCX Fiscal Hasar no genera ningún string de comando destinado a la impresora fiscal Hasar conectada.

HASAR1.Abortar

El software de gestión para puntos de venta puede utilizar *Abortar()* para interrumpir un proceso en curso (espera de respuesta a un comando) cuando la propiedad *ReintentoConstante()* se encuentra seteada con el valor "verdadero", y se produce una situación de "time out" en la comunicación con la impresora fiscal Hasar.

Método: *AbrirCajonDeDinero()*

010724	✓
051122	✓

Abre el cajón de dinero.

Este método del OCX Fiscal Hasar genera y envía el string del comando *OpenBox*.

HASAR1.AbrirCajonDeDinero

Este método es de utilidad para el software de gestión para puntos de venta si desea manejar el cajón de dinero conectado a la impresora fiscal Hasar (no todos los modelos permiten la conexión de tal dispositivo).

Método: *AbrirComprobanteFiscal()*

010724	✓	Abre un documento fiscal.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>OpenFiscalReceipt</i> .

Dim Comprob As DocumentosFiscales

**Comprob = TICKET_C
HASAR1.AbrirComprobanteFiscal Comprob**

El tipo de dato *DocumentosFiscales* puede ser alguna de las siguientes constantes embebidas en el OCX Fiscal Hasar:

Constante	dec	hex
FACTURA_A	48	30
FACTURA_B	49	31
NOTA_DEBITO_A	68	44
NOTA_DEBITO_B	69	45
RECIBO_A	97	61
RECIBO_B	98	62
TICKET_C	84	54
TICKET_FACTURA_A	65	41
TICKET_FACTURA_B	66	42
TICKET_NOTA_DEBITO_A (*)	50	32
TICKET_NOTA_DEBITO_B (*)	51	33
(*) Estas constantes embebidas no existen en la versión 010724 del OCX Fiscal Hasar.		

El documento fiscal TICKET_C es un tique a consumidor final, razón por la cual no se requieren los datos del comprador en forma previa.

Los documentos fiscales "B" en formulario continuo u hoja suelta (y los "B" en formato tique en algunos modelos de impresoras fiscales Hasar) no requieren los datos del comprador en forma previa obligatoria. Sin embargo, la apertura del comprobante en estas condiciones queda limitado a un monto máximo de \$1000.00 debido a exigencias de la normativa fiscal vigente. Para el resto de los comprobantes fiscales es obligatorio invocar en forma previa al método *DatosCliente()*.

Tener presente que los comprobantes fiscales en formato tique pueden alcanzar un monto máximo de \$5000.00 debido a exigencias de la normativa fiscal vigente.

Si el software de gestión para puntos de venta necesitase saber el número del comprobante abierto con el método *AbrirComprobanteFiscal()*, la manera de obtenerlo (si fuese posible) dependerá de la versión de OCX Fiscal Hasar, del modelo de impresora fiscal Hasar conectada, y del tipo de comprobante fiscal que se hubiere abierto.

(051122) **Dim nro_abrir As Long**

HASAR1.AbrirComprobanteFiscal FACTURA_B
nro_abrir = HASAR1.PrimerNumeroDeDocumentoActual

(010724) **Dim nro_abrir As String**
HASAR1.AbrirComprobanteFiscal FACTURA_B
nro_abrir = HASAR1.Respuesta(3)

Método: AbrirComprobanteNoFiscal()

010724	✓	Abre un comprobante no fiscal.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>OpenNonFiscalReceipt.</i> , u <i>OpenNonFiscalSlip</i> , según corresponda.

(051122) **Dim Estac As TiposDeEstacion**
Estac = ESTACION_TICKET
HASAR1.AbrirComprobanteNoFiscal [Estac]

(010724) **HASAR1.AbrirComprobanteNoFiscal**

El uso destinado al documento no fiscal es libre en formato tique. Sin embargo, en formato de hoja suelta o formulario continuo, el uso del documento está restringido a los indicados por la RG AFIP 1198/02.

El tipo de dato *TiposDeEstacion* puede ser alguna de las siguientes constantes embebidas en el OCX Fiscal Hasar:

Constante		dec	hex
ESTACION_SLIP	(*)	83	53
ESTACION_TICKET	(*)	84	54

(*) Estas constantes embebidas no existen en la versión 010724 del OCX Fiscal Hasar.

El argumento *Estac* es opcional y su uso sólo está disponible en la versión 051122 del OCX Fiscal Hasar.

Cuando el argumento opcional es *ESTACION_TICKET* el comprobante no fiscal es abierto en la estación de formato tique. La omisión de este argumento causa que el OCX Fiscal

Hasar asuma *ESTACION_TICKET*. Cuando el argumento opcional es *ESTACION_SLIP* el comprobante no fiscal es abierto en la llamada tercera estación (impresoras fiscales Hasar SMH/P-PRxF, o SMH/P-4x5F).

Si el software de gestión para puntos de venta necesita abrir un comprobante no fiscal en la tercera estación, utilizando la versión 010724 del OCX Fiscal Hasar, deberá recurrir al método *AbrirComprobanteNoFiscalEnSlip()*.

Método: *AbrirComprobanteNoFiscalEnSlip()*

010724	✓	Abre un comprobante no fiscal en la estación de slip.
051122	⊗	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>OpenNonFiscalSlip</i> .

El uso de este método es válido solamente en impresoras fiscales Hasar que dispongan de una tercera estación de impresión (SMH/P-PRxF, o SMH/P-4x5F).

(010724) **HASAR1.AbrirComprobanteNoFiscalEnSlip**

El uso destinado al documento no fiscal es libre en formato tique. Sin embargo, en formato de hoja suelta o formulario continuo, el uso del documento está restringido a los indicados por la RG AFIP 1198/02.

Método: *AbrirComprobanteNoFiscalHomologado()*

010724	✓	Abre un comprobante no fiscal homologado.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>OpenDNFH</i> .

Dim Comprob As DocumentosNoFiscales
Dim nro_dnfh As String

Comprob = ORDEN_SALIDA
nro_dnfh = "00000123"

HASAR1.AbrirComprobanteNoFiscalHomologado Comprob, nro_dnfh

El tipo de dato *DocumentosNoFiscales* puede ser alguna de las siguientes constantes embebidas en el OCX Fiscal Hasar:

Constante		dec	hex
ADELANTO_REMUNERACION	(*)	39	27
CARGO_HABITACION		85	55
CLAUSULA_CREDITO	(*)	58	3A
CLAUSULA_SEGURO	(*)	59	3B
COMUNICACION_CLIENTE	(*)	33	21
COTIZACION		117	75
INGRESO_EGRESO_SUCURSAL	(*)	43	2B
MINUTA_CREDITO	(*)	36	24
NOTA_CREDITO_A		82	52
NOTA_CREDITO_B		83	53
OFRECIMIENTO_CREDITO	(*)	34	22
OFRECIMIENTO_PASAPORTE	(*)	37	25
OFRECIMIENTO_TARJETA	(*)	35	23
ORDEN_SALIDA		115	73
PAGARE	(*)	60	3C

En el caso de apertura de notas de crédito (independientemente de la estación de impresión, y si se trata de "A", o "B") se requiere en forma previa y obligatoria que el software de gestión para puntos de venta invoque al método *DatosCliente()*; y en forma también previa y obligatoria setee la propiedad *DocumentoDeReferencia()* (en el caso de la versión 010724 del OCX Fiscal Hasar se debe setear la propiedad *InformacionRemito()*).

Constante		dec	hex
POLIZA_SEGURO	(*)	61	3D
RECIBO_X		120	78
RECORDATORIO	(*)	62	3E
REMITO		114	72
RENOVACION_CREDITO	(*)	38	26
RESCATE_MERCADERIA	(*)	42	2A
RESUMEN_CUENTA		116	74
SOLICITUD_CLAVE_TARJETA	(*)	41	29
SOLICITUD_CREDITO	(*)	63	3F
SOLICITUD_TARJETA_DEBITO	(*)	40	28
TICKET_NOTA_CREDITO_A	(*)	52	34
TICKET_NOTA_CREDITO_B	(*)	53	35
TICKET_PAGARE	(*)	55	37
TICKET_RECIBO_X	(*)	54	36

(*) No existen en la versión 010724 del OCX Fiscal Hasar.

Para el resto de los comprobantes no fiscales homologados que se pueden abrir, los datos previos al comando de apertura son opcionales.

Tener presente que las notas de crédito en formato tique pueden alcanzar un monto máximo de \$5000.00 debido a exigencias de la normativa fiscal vigente.

El argumento opcional, en la invocación del método, que especifica número a asignar al documento no fiscal homologado puede omitirse solamente en el caso

de notas de crédito y remitos, ya que los mismos debe numerarlos la impresora fiscal Hasar debido a exigencias de la normativa fiscal vigente. En consecuencia, la única numeración que puede resultar de interés obtener, por parte del software de gestión para puntos de venta, es la correspondiente a notas de crédito y/o remitos. La manera de obtener dicha numeración dependerá de la versión de OCX Fiscal Hasar, y del modelo de impresora fiscal Hasar conectada, y del tipo de comprobante no fiscal homologado que se hubiere abierto.

(051122)

Dim Comprob As DocumentosNoFiscales
Dim nro_abrir As Long

Comprob = REMITO

HASAR1.DatosCliente "Nombre Cliente", "99999999995", _
TIPO_CUIT, RESPONSABLE_INSCRIPTO, _
"Domicilio Cliente"

```
HASAR1.AbrirComprobanteNoFiscalHomologado Comprob
nro_abrir = HASAR1.PrimerNumeroDeDocumentoActual
```

```
(010724) Dim Comprob As DocumentosNoFiscales
Dim nro_abrir As String

Comprob = REMITO
HASAR1.DatosCliente "Nombre Cliente", "99999999995", _
TIPO_CUIT, RESPONSABLE_INSCRIPTO, _
"Domicilio Cliente"
HASAR1.AbrirComprobanteNoFiscalHomologado Comprob
nro_abrir = HASAR1.Respuesta(3)
```

Método: AutodetectarControlador()

010724	✓	Autodetecta el controlador fiscal.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>StatusRequest</i> , el cual es inocuo si llega a la impresora fiscal, pero permite determinar si existe comunicación. El envío de este comando es transparente para el software de gestión para puntos de venta, salvo por la excepción que se genera si la impresora fiscal Hasar no es detectada. La excepción generada dependerá del motivo de la no detección.

```
(010724) HASAR1.Puerto = 1
HASAR1.AutodetectarControlador HASAR1.Puerto
```

```
(051122) HASAR1.AutodetectarControlador [Puerto]
```

(010724) En esta versión del OCX Fiscal Hasar el argumento del método *AutodetectarControlador()* es obligatorio.

(051122) En esta versión del OCX Fiscal Hasar el argumento del método *AutodetectarControlador()* es opcional. Si se omite la autodetección se realiza sobre el puerto serie seteado en la propiedad *Puerto()*.

El OCX Fiscal Hasar busca, probando en todas las velocidades posibles, tener comunicación con una impresora fiscal Hasar. Cada intento equivale a invocar al método *Comenzar()* sabiendo de antemano la velocidad a emplear (valor seteado en la propiedad *Baudios()*).

Cada intento del método *AutodetectarControlador()* por establecer conexión con la impresora fiscal Hasar a una nueva velocidad dispara el evento *ProgresoDetección()*.

Si el método *AutodetectarControlador()* no fracasa, la propiedad *Baudios()* quedará seteada con la velocidad de puerto serie a la cual fue detectada.

El uso del método *AutodetectarControlador()* implica que no se necesita invocar al método *Comenzar()*.

Método: *AutodetectarModelo()*

010724	⊗	Autodetecta el modelo de controlador.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>GetPrinterVersion</i> .

El método *AutodetectarModelo()* permite determinar -de ser ello posible- el modelo de impresora fiscal Hasar conectada al puerto serie del host (generalmente la PC).

Debe ser invocado con posterioridad al correcto seteo de las propiedades *Puerto()* y *Baudios()*, y con posterioridad a la finalización exitosa del método *Comenzar()* o del método *AutodetectarControlador()*.

```
(051122) HASAR1.Puerto = 1
HASAR1.AutodetectarControlador
HASAR1.AutodetectarModelo
```

Determinado el modelo de impresora fiscal Hasar, el OCX Fiscal Hasar setea la propiedad *Modelo()* con la constante embebida adecuada.

Para que el método *AutodetectarModelo()* finalice con éxito se requiere que la impresora fiscal Hasar conectada al puerto serie del host contenga en su set al comando *GetPrinterVersion*.

Si el modelo de impresora fiscal Hasar conectada al puerto serie del host no coincide con ninguno de los soportados por el OCX Fiscal Hasar, se generará la excepción identificada con la constante embebida *H_ERR_CF_NOT_DETECTED*.

Si la impresora fiscal Hasar conectada no soporta al comando *GetPrinterVersion*, se producirá la excepción identificada con la constante embebida *H_ERR_NOT_SUPPORTED*.

Si la impresora fiscal Hasar no estuviera conectada al puerto serie del host, se generará la excepción identificada con la constante embebida *H_ERR_CF_NOT_DETECTED*.

Si el modelo de impresora fiscal Hasar no pudo ser identificado por el método *AutodetectarModelo()*, el valor de la propiedad *Modelo()* será determinado, y deberá ser modificado por el software de gestión para puntos de venta. La modificación puede ser a través una rutina propia del software para los modelos de impresoras fiscales Hasar que no soportan al comando *GetPrinterVersion*, o por selección manual por parte del usuario a través de un combo.

Método: AvanzarPapel ()

010724	✓	Avanza el papel de una estación de la impresora.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string de uno de los siguientes comandos: <i>FeedReceipt</i> , <i>FeedJournal</i> , o <i>FeedReceiptJournal</i> . El comando a enviar depende de la indicación del tipo de papel, especificado por el software de gestión para puntos de venta en el argumento correspondiente.

Este método le permite al software de gestión para puntos de venta realizar avances de papel cuando necesite dejar líneas en blanco. Es importante destacar que no es aplicable sobre todos los comprobantes posibles de emitir, ni está disponible en todos los modelos de impresoras fiscales Hasar.

```

Dim papel As TiposDePapel
Dim lineas As Long

papel = PAPEL_DIARIO
lineas = 3
HASAR1.AvanzarPapel papel, lineas
    
```

Ambos argumentos del método *AvanzarPapel()* son obligatorios.

Constante	dec	hex
PAPEL_DIARIO	1	1
PAPEL_TICKET	0	0
PAPEL_TICKET_Y_DIARIO	2	2

El tipo de datos *TiposDePapel* puede ser alguna de las constantes embebidas en el OCX Fiscal Hasar que se listan en la tabla de la izquierda.

Método: BajaFiscal()

010724	✓	Da de baja una memoria fiscal bajo autorización de la AFIP.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>KillEprom</i> .

El software de gestión para puntos de venta debe tener la capacidad de poder ejecutar, bajo estrictos controles de acceso, la baja de la memoria fiscal que posee la impresora fiscal Hasar. Puede ocurrir que en presencia y a pedido de la autoridad fiscal competente, como resultado de controles realizados por dicha autoridad, ésta solicite dar la baja de la memoria fiscal. Si bien existe todo un procedimiento administrativo y técnico para concretar la baja de una memoria fiscal, en casos muy excepcionales el software de gestión para puntos de venta debe estar en condiciones de resolver la parte técnica de dicho procedimiento (a realizar siempre en presencia y a pedido de la autoridad fiscal competente).

(010724) **HASAR1.BajaFiscal**

En la versión indicada del OCX Fiscal Hasar el método *BajaFiscal()* no requiere argumentos.

(051122) **Dim registro**

HASAR1.BajaFiscal [registro]

En la versión indicada del OCX Fiscal Hasar, en el método *BajaFiscal()* existe un argumento opcional: *[registro]*. Ello debido a que el comando *KillEprom* que se genera puede requerirlo en función del modelo de impresora fiscal Hasar conectada al puerto serie del host. De ser requerido el argumento *[registro]* debe pasarse el número de registro con el cual fue inicializada la impresora fiscal; generalmente un string que comienza con "HH", le sigue un tercer caracter y siete dígitos.

Método: *CambiarCodigoIngresosBrutos()*

010724	⊙
051122	✓

Cambia el código de ingresos brutos.

Este método del OCX Fiscal Hasar genera y envía el string del comando *ChangeIBNumber*.

(051122) **Dim codigo As String**

codigo = "IB 1234567"

HASAR1.ReporteZ

HASAR1.CambiarCodigoIngresosBrutos codigo

El software de gestión para puntos de venta debería ofrecer al usuario, mediante estrictos controles de acceso, la posibilidad de modificar su número de inscripción en ingresos brutos, si tal cambio legalmente se produce.

El método *CambiarCodigoIngresosBrutos()* exige haber realizado previamente el cierre diario "Z" (invocar al método *ReporteZ()*).

El método *CambiarCodigoIngresosBrutos()* puede ser utilizado hasta un máximo de 5 (cinco) veces en la vida útil de la memoria fiscal de la impresora fiscal Hasar.

Método: CambiaFechaInicioActividades()

010724	⊙	Cambia la fecha de inicio de actividades.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>ChangeBussinessStartupDate</i> .

(051122) **Dim fecha As Date**

```
fecha = "01/01/2006"
HASAR1.ReporteZ
HASAR1.CambiaFechaInicioActividades fecha
```

El software de gestión para puntos de venta debería ofrecer al usuario, mediante estrictos controles de acceso, la posibilidad de modificar la fecha de inicio de actividades, si tal cambio legalmente se produce. Por ejemplo, al mudar la impresora fiscal Hasar de una sucursal comercial a otra.

El método *CambiaFechaInicioActividades()* exige haber realizado previamente el cierre diario "Z" (invocar al método *ReporteZ()*).

El método *CambiaFechaInicioActividades()* puede ser utilizado hasta un máximo de 5 (cinco) veces en la vida útil de la memoria fiscal de la impresora fiscal Hasar.

Método: CambiarResponsabilidadIVA()

010724	⊙	Cambia la responsabilidad IVA del controlador fiscal.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>ChangeIVAResponsability</i> .

(051122) **Dim categ As TiposResponsabilidadImpresor**

```
categ = CLASE_RESPONSABLE_INSCRIPTO
HASAR1.ReporteZ
HASAR1.CambiarResponsabilidadIVA categ
```

El software de gestión para puntos de venta debería ofrecer al usuario, mediante estrictos controles de acceso, la posibilidad de modificar la responsabilidad frente a IVA del emisor, si tal cambio legalmente se produce.

El método *CambiarResponsabilidadIVA()* exige haber realizado previamente el cierre diario "Z" (invocar al método *ReporteZ()*).

El método *CambiarResponsabilidadIVA()* puede ser utilizado hasta un máximo de 5 (cinco) veces en la vida útil de la memoria fiscal de la impresora fiscal Hasar.

Constante	dec	hex
CLASE_MONOTRIBUTO	77	4D
CLASE_MONOTRIBUTO_SOCIAL (*)	83	53
CLASE_NO_RESPONSABLE	65	41
CLASE_RESPONSABLE_EXENTO	69	45
CLASE_RESPONSABLE_INSCRIPTO	73	49
CLASE_RESPONSABLE_NO_INSCRIPTO (**)	78	4E

El tipo de datos *TiposResponsabilidadImpresor* puede ser alguna de las constantes embebidas en el OCX Fiscal Hasar que se listan en la tabla de la izquierda.

(*) Según la normativa fiscal vigente esta categoría frente a IVA ha sido incorporada entre las que deben ser consideradas.

(**) Según la normativa fiscal vigente esta categoría frente a IVA ha desaparecido. Se mantiene por compatibilidad con viejos modelos de impresoras fiscales Hasar que la soportaban.

Método: CambiarVelocidad()

010724	✓
051122	✓

Cambia la velocidad de comunicación serie.

Este método del OCX Fiscal Hasar genera y envía el string del comando *SetComSpeed*.

Dim veloc As long

```
veloc = 19200
HASAR1.CambiarVelocidad veloc
HASAR1.Finalizar
HASAR1.Baudios = veloc
HASAR1.Comenzar
```

Cuando el software de gestión para puntos de venta invoca al método *CambiarVelocidad()*, a diferencia de la propiedad *Baudios()*, está modificando la velocidad del puerto serie de la impresora fiscal Hasar; no la del host. Como paso posterior se deberá invocar al método *Finalizar()*, y posteriormente setear la propiedad *Baudios()* con la nueva velocidad, para finalmente ejecutar el método *Comenzar()*.

Si no se respetan los pasos descritos el software de gestión para puntos de venta perderá la comunicación con la impresora fiscal Hasar por no coincidir la velocidad de su puerto serie con el valor seteado en la propiedad *Baudios()* del host.

Si bien la impresora fiscal soporta las velocidades: 1.200, 2.400, 4.800, 9.600, 19200, 38400, 57600, y 115200, este último valor no debe setearse con el método

CambiarVelocidad(), ya que posteriormente 115200 no será aceptado para setear la propiedad *Baudios()* por ser una velocidad no soportada por Windows®.

Método: CancelarComprobante()

010724	✓	Cancela un comprobante.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>Cancel</i> .

La invocación a este método cancela cualquier tipo de comprobante en curso, pero no es aplicable a todos los modelos de impresoras fiscales Hasar.

Método: CancelarComprobanteFiscal()

010724	✓	Cancela un comprobante fiscal.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>Cancel</i> o <i>TotalTender</i> , según corresponda al modelo de impresora fiscal hasar conectada al puerto serie del host.

Mediante la invocación a este método el software de gestión para puntos de venta puede cancelar el comprobante fiscal en curso, siempre que el estado de la impresora fiscal Hasar para ese documento lo permita.

Se debe tener presente que la normativa fiscal vigente no permite cancelar comprobantes fiscales sobre los cuales se hubiera registrado un pago parcial.

Metodo: Capacidades()

010724	⊖	Características soportadas por las diferentes familias de controladores fiscales.
051122	✓	Este método del OCX Fiscal Hasar no genera y ni envía el string de comando alguno para las impresoras fiscales hasar.

El software de gestión para puntos de venta puede determinar con este método, que retorna un valor booleano, si determinada característica está disponible en la impresora fiscal Hasar que se encuentra seteada en la propiedad *Modelo()*.

```

Dim capac As CapacidadesSoportadas

capac = CAPACIDAD_CAJON

If ( HASAR1.Capacidades capac [, subcapac] [, resultado] ) then
.....
    
```

End If

El argumento *capac* es de uso obligatorio.

El argumento [, *sucapac*] es optativo, y el tipo de dato a utilizar (como su uso) dependerá del primer argumento con el cual se invoca al método *Capacidades()*.

El argumento [, *resultado*] es optativo y carece de uso por el momento. Está reservado para futuras prestaciones.

Constante	dec	hex
CAPACIDAD_DOCUMENTO_SLIP	0	0
CAPACIDAD_DOCUMENTO_TICKET	1	1
CAPACIDAD_ESTACION	2	2
CAPACIDAD_CAJON	3	3

El tipo de datos *CapacidadesSoportadas* puede ser alguna de las constantes embebidas en el OCX Fiscal Hasar que se listan en la tabla de la izquierda.

Constante	dec	hex
SOPORTA_FACTURA	0	0
SOPORTA_NOTA_CREDITO	1	1

El tipo de datos *TiposDeDocumentoSoportado* puede ser alguna de las constantes embebidas en el OCX Fiscal Hasar que se listan en la tabla de la izquierda.

Este tipo de dato puede utilizarse solamente como segundo argumento del método *Capacidades()* cuando el primer argumento toma el valor CAPACIDAD_DOCUMENTO_SLIP o el valor CAPACIDAD_DOCUMENTO_TICKET.

Dim capac As CapacidadesSoportadas
Dim subcapac As TiposDeDocumentoSoportado

capac = CAPACIDAD_DOCUMENTO_SLIP
subcapac = SOPORTA_FACTURA

If (HASAR1.Capacidades capac , subcapac [, resultado]) then

End If

Constante	dec	hex
ESTACION_SLIP	83	53
ESTACION_TICKET	84	54

El tipo de datos *TiposDeEstacion* puede ser alguna de las constantes embebidas en el OCX Fiscal Hasar que se listan en

la tabla de la izquierda. Este tipo de dato puede utilizarse solamente como segundo argumento del método *Capacidades()* cuando el primer argumento toma el valor CAPACIDAD_ESTACION.

Cuando el primer argumento del método *Capacidades()* toma el valor CAPACIDAD_CAJON no es requerido el segundo argumento.

Capacidades soportadas por las impresoras fiscales Hasar

capac (dec)	sub-capac (dec)	Formato tique SMH/P								
		-614F	-615F	-715F	-PR4F	-PR5F	-950F	-951F	T-262F	T-272F
0	0	---	---	---	---	---	---	---	---	---
	1	---	---	---	---	---	---	---	---	---
1	0	---	SI	SI	SI	SI	---	SI	---	---
	1	---	---	SI	---	SI	---	---	---	---
2	83	---	---	---	SI	SI	SI	SI	---	---
	84	SI	SI	SI	SI	SI	SI	SI	SI	SI
3	---	SI	SI	SI	SI	SI	SI	SI	SI	SI

capac (dec)	sub-capac (dec)	Formato formulario continuo SMH/P						
		J-20F	-320F	-321F	-322F	-330F	L-8F	L-9F
0	0	SI	SI	SI	SI	SI	SI	SI
	1	SI	SI	SI	SI	SI	SI	SI
1	0	---	---	---	---	---	---	---
	1	---	---	---	---	---	---	---
2	83	SI	SI	SI	SI	SI	SI	SI
	84	---	---	---	---	---	---	---
3	---	---	---	---	---	---	---	---

capac (dec)	sub-capac (dec)	CIF SMH/	
		-425F	-435F
0	0	SI	SI
	1	SI	SI
1	0	SI	SI
	1	SI	SI
2	83	SI	SI
	84	SI	SI
3	---	SI	SI

Método: CapacidadRestante()

Obtiene la capacidad de Zetas restantes.

010724	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>HistoryCapacity</i> .
051122	✓	

```
(010724) Dim ztotal As String
          Dim zhechas As String
```

```
HASAR1.CapacidadRestante
ztotal = HASAR1.Respuesta(3)
zhechas = HASAR1.Respuesta(4)
```

En la versión 010724 del OCX Fiscal Hasar el método *CapacidadRestante()* no requiere de argumentos.

```
(051122) Dim ztotal, zhechas
```

```
HASAR1.CapacidadRestante ztotal, zhechas
```

En la versión 051122 del OCX Fiscal Hasar ambos argumentos son opcionales. Notar que se trata de argumentos de salida.

Método: CargarLogo()

010724	✓
051122	✓

Carga un bmp como logo.

Este método del OCX Fiscal Hasar genera y envía el string del comando *StoreLogo* o *ResetLogo*, según corresponda.

```
Dim archbmp As String
```

```
HASAR1.ReporteZ
```

```
'// Se carga el logo
```

```
archbmp = "c:\hasar\img\milogo.bmp"
```

```
HASAR1.CargarLogo archbmp
```

```
HASAR1.ReporteZ
```

```
'// Se elimina el logo
```

```
archbmp = ""
```

```
HASAR1.CargarLogo archbmp
```

El método *CargarLogo()* solamente puede ser invocado con posterioridad a la emisión de un cierre diario "Z" (método *ReporteZ()*).

Mediante este método el software de gestión para puntos de venta puede cargar en la memoria de trabajo de la impresora fiscal Hasar la imagen de un logotipo comercial para que se imprima cada vez que se emite un comprobante.

El argumento *archbmp* debe ser el nombre de un archivo de imágenes (se puede incluir el path correspondiente) en formato BMP que debe respetar la especificación dada en el manual de comandos de la impresora fiscal Hasar. El OCX Fiscal Hasar toma el archivo BMP y convierte la imagen en una lista de comandos (*StoreLogo*) que son enviados a la impresora fiscal Hasar.

Si el argumento *archbmp* es un string vacío, el OCX Fiscal Hasar asume que se quiere eliminar el logo de la memoria de trabajo de la impresora fiscal Hasar; genera y envía entonces el comando *ResetLogo*.

Método: *CerrarComprobanteFiscal()*

010724	✓	Cierra un comprobante fiscal.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>CloseFiscalReceipt</i> .

(010724) **Dim comprob As String**

**HASAR1.CerrarComprobanteFiscal
comprob = HASAR1.Respuesta(3)**

En la versión *010724* del OCX Fiscal Hasar el método *CerrarComprobanteFiscal()* no requiere argumentos.

(051122) **Dim copias, comprob**

**copias = 2
HASAR1.CerrarComprobanteFiscal copias, comprob**

En la versión *051122* del OCX Fiscal Hasar ambos argumentos son opcionales, siendo el primero (*copias*) de entrada, mientras que el segundo (*comprob*) es de salida.

El argumento de entrada *copias* le permite al software de gestión para puntos de venta solicitar a la impresora fiscal Hasar la emisión de ejemplares adicionales del comprobante fiscal cerrado sin tener que enviar otro comando; por ejemplo invocando al método *ReimprimirComprobante()*.

El argumento de salida *comprob* le permite al software de gestión para puntos de venta obtener de la impresora fiscal Hasar el número con el cual se ha cerrado el comprobante fiscal. Este número no tiene porque coincidir con el de apertura. Tener presente que la normativa fiscal vigente exige incrementar el número de comprobante cada vez que se realiza un transporte a la hoja siguiente.

La normativa fiscal vigente no permite cerrar comprobantes fiscales cuyo monto acumulado sea cero.

Método: CerrarComprobanteNoFiscal()

010724	✓	Cierra un comprobante no fiscal.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>CloseNonFiscalReceipt</i> .

Este método cierra el comprobante no fiscal independientemente de la estación de impresión en la hubiera sido abierto.

(010724) **HASAR1.CerrarComprobanteNoFiscal**

En la versión *010724* del OCX Fiscal Hasar el método *CerrarComprobanteNoFiscal()* no requiere argumentos.

(051122) **Dim copias**

copias = 2

HASAR1.CerrarComprobanteNoFiscal copias

En la versión *051122* del OCX Fiscal el argumento de entrada *copias* del método *CerrarComprobanteNoFiscal()* le permite al software de gestión para puntos de venta solicitar a la impresora fiscal Hasar la emisión de ejemplares adicionales del comprobante no fiscal cerrado sin tener que enviar otro comando; por ejemplo invocando al método *ReimprimirComprobante()*.

Método: CerrarComprobanteNoFiscalHomologado()

010724	✓	Cierra un comprobante no fiscal homologado.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>CloseDNFH</i> .

(010724) **Dim comprob As String**

**HASAR1.CerrarComprobanteNoFiscalHomologado
comprob = HASAR1.Respuesta(3)**

En la versión 010724 del OCX Fiscal Hasar el método *CerrarComprobanteNoFiscalHomologado()* no requiere argumentos.

(051122) **Dim copias, comprob**

copias = 2

HASAR1.CerrarComprobanteNoFiscalHomologado copias, comprob

En la versión 051122 del OCX Fiscal Hasar ambos argumentos son opcionales, siendo el primero (*copias*) de entrada, mientras que el segundo (*comprob*) es de salida.

El argumento de entrada *copias* le permite al software de gestión para puntos de venta solicitar a la impresora fiscal Hasar la emisión de ejemplares adicionales del comprobante no fiscal homologado que se ha cerrado sin tener que enviar otro comando; por ejemplo invocando al método *ReimprimirComprobante()*.

El argumento de salida *comprob* le permite al software de gestión para puntos de venta obtener de la impresora fiscal Hasar el número con el cual se ha cerrado el comprobante no fiscal homologado. Este número no tiene porque coincidir con el de apertura. Tener presente que la normativa fiscal vigente exige incrementar el número de comprobante cada vez que se realiza un transporte a la hoja siguiente.

La normativa fiscal vigente no permite cerrar notas de crédito cuyo monto acumulado sea cero.

Método: Comenzar()

010724	✓
051122	✓

Comienza una sesión, inicializando el puerto y el impresor.

Este método del OCX Fiscal Hasar genera y envía el string del comando *StatusRequest*.

Como condición previa a la invocación del método *Comenzar()*, el software de gestión para puntos de venta debe setear adecuadamente las propiedades asociadas a las comunicaciones, según el vínculo a utilizar. El método *Comenzar()* inicializará la comunicación con la impresora fiscal Hasar e intentará comunicarse con ella enviando dos comandos de pedido de status (*StatusRequest*). El OCX Fiscal Hasar generará una excepción (o un evento -según la versión-) si no puede establecer la comunicación deseada.

(010724) (051122)

HASAR1.Puerto = 1

HASAR1.Baudios = 9600

HASAR1.Comenzar

En la versión 010724 del OCX Fiscal Hasar solamente se puede utilizar como vínculo para las comunicaciones al puerto serie. Ello implica setear la propiedad *Puerto()* con el número asociado al puerto serie (RS-232) donde está conectada la impresora fiscal Hasar, y setear la propiedad *Baudios()* con la velocidad a utilizar.

```
(051122)  HASAR1.Transporte = SOCKET_TCP  
HASAR1.Puerto = 7227  
HASAR1.DireccionIP = "127.0.0.1"  
HASAR1.Comenzar
```

Con la versión 051122 del OCX Fiscal Hasar, además de la conexión local a través del puerto serie (RS-232), la impresora fiscal Hasar puede manejarse remotamente mediante protocolo TCP/IP. En este último caso, el software de gestión para puntos de venta debe setear la propiedad *Transporte()* para indicar si se va a usar, o no, UDP. Debe setear, además, la propiedad *Puerto()* para indicar el número de socket a utilizar, y la propiedad *DireccionIP()* para especificar el nodo de la red en el cual se encuentra físicamente conectada la impresora fiscal. El nodo puede ser una PC (requiere de un programa servidor que lea el socket y redireccione al puerto serie de la PC), o puede ser el conversor Hasar CVE-RS232 (la redirección es transparente).

Cuando el software de gestión para puntos de venta invoca al método *Comenzar()* debe tener presente el estado de la propiedad *ReintentoConstante()*. Por ejemplo, al tratar de establecer comunicaciones vía el puerto serie con una impresora fiscal Hasar apagada, puede ocurrir que:

```
HASAR1.Puerto = 1  
HASAR1.Baudios = 9600  
HASAR1.ReintentoConstante = False  
HASAR1.Comenzar
```

En la versión 010724 del OCX Fiscal Hasar se produce la excepción H_ERR_TIMEOUT.

En la versión 051122 del OCX Fiscal Hasar se produce la excepción H_ERR_CF_NOT_DETECTED.

O bien, puede ocurrir que:

```
HASAR1.Puerto = 1  
HASAR1.Baudios = 9600  
HASAR1.ReintentoConstante = True  
HASAR1.Comenzar
```

En la versión 010724 del OCX Fiscal Hasar genera el evento *ImpresoraNoResponde()*. Lo mismo ocurre con la versión 051122.

Algunas observaciones sobre el uso del método Comenzar():

El desarrollador de software para puntos de venta debe tener en cuenta estas observaciones al pasar de una versión a otra del OCX Fiscal Hasar.

1. Se invoca al método *Comenzar()* en forma consecutiva sin invocar al método *Finalizar()* entre una llamada y otra.

<p>..... HASAR1.Comenzar HASAR1.Comenzar </p>	<p>En la versión 010724 del OCX Fiscal Hasar se genera una excepción. En la versión 051122 no se genera ninguna excepción (se resuelve internamente).</p>
---	--

2. Se invoca al método *AutodetectarControlador()* y al método *Comenzar()* en forma consecutiva sin invocar al método *Finalizar()* entre una llamada y otra.

<p>..... HASAR1.AutodetectarControlador HASAR1.Comenzar </p>	<p>En la versión 010724 del OCX Fiscal Hasar se genera una excepción. En la versión 051122 no se genera ninguna excepción (se resuelve internamente).</p>
--	--

3. Se invoca al método *Comenzar()* y al método *Recomenzar()* en forma consecutiva sin invocar al método *Finalizar()* entre una llamada y otra.

<p>..... HASAR1.Comenzar HASAR1.Recomenzar </p>	<p>En la versión 010724 del OCX Fiscal Hasar no se genera ninguna excepción. En la versión 051122 no se genera una excepción por estar discontinuado el método.</p>
---	--

Método: ConfigurarControlador()

010724	✓	Configura parámetros del controlador.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>ConfigureControllerByOne</i> .

Dim cambiar As ParametrosDeConfiguracion
Dim param As String

cambiar = COPIAS_DOCUMENTOS
param = "2"
HASAR1.ConfigurarControlador cambiar, param

El método *ConfigurarControlador()* permite al software de gestión para puntos de venta modificar parámetros de comportamiento de la impresora fiscal Hasar que podrían llegar a modificarse durante la jornada comercial.

Casi la totalidad de los parámetros modificables permiten invocar al método *ConfigurarControlador()* sin que resulte necesario efectuar un cierre diario "Z" (invocar al método *ReporteZ()*). Otros requieren estar fuera de todo documento para proceder al cambio.

Se debe consultar el manual de comandos de la impresora fiscal Hasar para saber cuales parámetros de comportamiento pueden modificarse en cualquier momento, cuales no, y cuales no son soportados por la impresora fiscal Hasar en uso.

Constante	dec	hex
IMPRESION_CAMBIO	52	34
IMPRESION_LEYENDAS	53	35
CORTE_PAPEL	54	36
IMPRESION_MARCO	55	37
REIMPRESION_CANCELADOS	56	38
COPIAS_DOCUMENTOS	57	39
PAGO_SALDO	58	3A
SONIDO_AVISO	(**) 59	3B
ALTO_HOJA	(*) 60	3C
ANCHO_HOJA	(*) 61	3D
ESTACION_REPORTES_XZ	(*) 62	3E
MODO_IMPRESION	(*) 63	3F

El tipo de datos *ParametrosDeConfiguracion* puede ser alguna de las constantes embebidas en el OCX Fiscal Hasar que se listan en la tabla de la izquierda.

El software de gestión para puntos de venta debería ofrecer estas opciones de configuración al usuario final.

(*) Estas constantes embebidas del OCX Fiscal Hasar no existen en la versión 010724.

(**) Si bien el valor de la constante embebida se mantiene, el nombre SONIDO_AVISO solamente es válido en la versión 051122. Para la versión 010724 del OCX Fiscal Hasar el nombre es SONIDO.

Método: ConfigurarControladorCompleto()

010724	✓	Configuración del controlador completo.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>GeneralConfiguration</i> .

Mediante el método *ConfigurarControladorCompleto()* el software de gestión para puntos de venta puede modificar parámetros de comportamiento de la impresora fiscal Hasar que en su mayoría están asociados, directa o indirectamente, a la normativa fiscal vigente.

El software de gestión para puntos de venta podrá invocar al método `ConfigurarControladorCompleto()` si previamente ha realizado un cierre diario "Z" (invocando al método `ReporteZ()`).

El desarrollador del software de gestión para puntos de venta deberá consultar en el manual de comandos de la impresora fiscal Hasar si el comando generado por el método `ConfigurarControladorCompleto()` está soportado por su set de comandos, y cuales de estos parámetros de comportamiento son ignorados y cuales tiene efecto; además de consultar los valores posibles para dichos parámetros.

Con excepción de los dos primeros argumentos del método `ConfigurarControladorCompleto()`, los argumentos son de uso optativo.

En la versión 010724 del OCX Fiscal Hasar todos los argumentos son de tipo string, con excepción de los dos primeros que son booleanos. En esta versión un string vacío ("") significa parámetro opcional omitido.

(010724)

Dim imp_rep As Boolean, defecto As Boolean

**Dim lim_cf As String, lim_tf As String, iva_noi As String
Dim nro_copias As String, cambio As String, leyend As String
Dim corte As String, marco As String, reimp_doc As String
Dim saldo As String, pitido As String**

**imp_rep = True
defecto = True
lim_cf = "1000.00"
lim_tf = "5000.00"
iva_noi = "0.00"
nro_copias = "2"
cambio = "P"
leyend = "P"
corte = "P"
marco = "N"
reimp_doc = "P"
saldo = "Cuenta Corriente"
pitido = "P"**

HASAR1.ReporteZ

**HASAR1. ConfigurarControladorCompleto imp_rep, defecto, lim_cf, _
lim_tf, iva_noi, nro_copias, cambio, leyend, corte, marco, _
reimp_doc, saldo, pitido**

(051122)

Dim imp_rep As Boolean, defecto As Boolean**Dim lim_cf As String, lim_tf As String, iva_noi As String****Dim cambio As String, leyend As String, marco As String, reimp_doc As String****Dim saldo As String, pitido As String****Dim nro_copias As NumeroDeCopias****Dim corte As TipoDeCorteDePapel****Dim alto As TiposDeAltoHoja****Dim ancho As TiposDeAnchoHoja****Dim estacion As TiposDeEstacion****Dim modo As TiposDeModoImpresion****imp_rep = True****defecto = True****lim_cf = 1000****lim_tf = 5000****iva_noi = 0****nro_copias = DUPLICADO****cambio = True****leyend = True****corte = CORTE_PARCIAL****marco = False****reimp_doc = True****saldo = "Cuenta Corriente"****pitido = True****alto = ALTO_A4****ancho = ANCHO_NORMAL****estacion = ESTACION_TICKET****modo = USO_ESTACION_TICKET****HASAR1.ReporteZ****HASAR1. ConfigurarControladorCompleto imp_rep, defecto, lim_cf, _
lim_tf, iva_noi, nro_copias, cambio, leyend, corte, marco, _
reimp_doc, saldo, pitido, alto, ancho, estacion, modo**

Notar que la versión 051122 del OCX Fiscal Hasar dispone de más argumentos que la versión 010724.

En la versión 051122 del OCX Fiscal Hasar los strings vacíos son considerados como tales (un string vacío no es considerado ausente). Si se quiere omitir un argumento se deben colocar las comas seguidas sin valor alguno entre ellas (... , , ...).

imp_rep	Boolean	---	Exigido	---
defecto	Boolean	---	Exigido	---
lim_cf	String	Numérico	Ausente	---
lim_tf	String	Numérico	Ausente	---
iva_noi	String	Numérico	Ausente	---
nro_copias	String	Numérico	Ausente	NumeroDeCopias
cambio	Boolean	---	Ausente	---
leyend	Boolean	---	Ausente	---
corte	String	---	Ausente	TipoDeCorteDePapel
marco	Boolean	---	Ausente	---
reimp_doc	Boolean	---	Ausente	---
saldo	String	---	Ausente	---
pitido	Boolean	---	Ausente	---
alto	String	---	Ausente	TiposDeAltoHoja
ancho	String	---	Ausente	TiposDeAnchoHoja
estacion	String	---	Ausente	TiposDeEstacion
modo	String	---	Ausente	TiposDeModoImpresion

En cuanto a los tipos de datos soportados en cada argumento, vale la tabla de la izquierda.

En la versión 051122 del OCX Fiscal Hasar resulta más versátil la manera de pasar argumentos al método *ConfigurarControladoCompleto()*.

Los argumentos "lim_cf" (monto límite para tiques factura "B" a consumidor final sin datos completos), y "lim_tf" (monto límite para la emisión de tiques factura) tienen un valor establecido por la normativa fiscal vigente que no debe ser modificado, salvo indicación explícita de dicha normativa. El argumento "iva_noi" (sobretasa de iva no inscripto ya no es de aplicación, según la normativa fiscal vigente).

El argumento "imp_rep" indica si se debe imprimir un reporte de configuración cuando el comando es ejecutado por la impresora fiscal Hasar.

Si el argumento "defecto" es verdadero los argumentos omitidos serán llevados a su valor por defecto en la impresora fiscal Hasar.

El argumento "nro_copias" permite establecer la cantidad de ejemplares que permitirá emitir la impresora fiscal Hasar.

El argumento "cambio" indica si se debe realizar la impresión de la leyenda "CAMBIO \$0.00".

El argumento "leyend" indica si deben imprimirse las leyendas opcionales (inscripción en ingresos brutos, etc.).

El argumento "corte" permite establecer el tipo de corte de papel cuando la impresora fiscal Hasar deba realizar tal acción.

El argumento "marco" indica si debe imprimirse el marco de los documentos a emitir.

El argumento "reimp_doc" indica si debe ser reimpresso un comprobante cancelado por corte de energía.

El argumento "saldo" permite establecer el texto que se imprimirá como medio de pago en facturas y tiques factura cuando el software de gestión para puntos de venta cierre el comprobante fiscal sin haber registrado pago alguno.

El argumento "pitido" indica si la impresora fiscal Hasar debe emitir sonido ante falta de papel.

Los argumentos "alto" y "ancho" permiten establecer el tamaño de papel a utilizar en formularios continuos u hojas sueltas.

El argumento "estacion" permite establecer la estación de impresión a utilizar cuando se invoca al método *ReporteZ()*.

El argumento "modo" permite especificar el modo de trabajo de la impresora fiscal Hasar.

Constante	dec	hex
NO_COPIAS	48	30
ORIGINAL	49	31
DUPLICADO	50	32
TRIPLICADO	51	33
CUADRUPLICADO	52	34

El tipo de datos *NumeroDeCopias* puede ser alguna de las constantes embebidas en el OCX Fiscal Hasar que se listan en la tabla de la izquierda. Estas constantes solamente pueden emplearse en la versión 051122 del OCX Fiscal Hasar.

Constante	dec	hex
CORTE_TOTAL	70	46
NO_CORTE	78	4E
CORTE_PARCIAL	80	50

El tipo de datos *TipoDeCorteDePapel* puede ser alguna de las constantes embebidas en el OCX Fiscal Hasar que se listan en la tabla de la izquierda. Estas constantes solamente pueden emplearse en la versión 051122 del OCX Fiscal Hasar.

Constante	dec	hex
ALTO_A4	65	41
ALTO_REDUCIDO	77	4D
ALTO_OFICIO	79	4F

El tipo de datos *TiposDeAltoDeHoja* puede ser alguna de las constantes embebidas en el OCX Fiscal Hasar que se listan en la tabla de la izquierda. Estas constantes solamente pueden emplearse en la versión 051122 del OCX Fiscal Hasar.

Constante	dec	hex
ANCHO_REDUCIDO	77	4D
ANCHO_NORMAL	78	4E

El tipo de datos *TiposDeAnchoDeHoja* puede ser alguna de las constantes embebidas en el OCX Fiscal Hasar que se listan en la tabla de la izquierda. Estas constantes solamente pueden emplearse en la versión 051122 del OCX Fiscal Hasar.

Constante	dec	hex
OCX_FISCAL_SLIP	83	53
ESTACION_TICKET	84	54

alguna de las constantes embebidas en el OCX Fiscal Hasar que se listan en la tabla de la izquierda. Estas constantes solamente pueden emplearse en la versión 051122 del OCX Fiscal Hasar.

Constante	dec	hex
USO_ESTACION_TICKET	65	41
USO_ESTACION_TICKET	77	4D

uede ser alguna de las constantes embebidas en el OCX Fiscal Hasar que se listan en la tabla de la izquierda. Estas constantes solamente pueden emplearse en la versión 051122 del OCX Fiscal Hasar.

El software de gestión para puntos de venta debería ofrecer estas opciones de configuración al usuario final.

Método: ConfigurarControladorPorBloque()

010724 ción del controlador por bloque.

051122 do del OCX Fiscal Hasar genera y envía el string del comando *ConfigureControllerByBlock*.

Mediante el método *ConfigurarControladorPorBloque()* el software de gestión para puntos de venta puede modificar parámetros de comportamiento de la impresora fiscal Hasar que en su mayoría están asociados, directa o indirectamente, a la normativa fiscal vigente.

El software de gestión para puntos de venta podrá invocar al método *ConfigurarControladorporBloque()* si previamente ha realizado un cierre diario "Z" (invocando al método *ReporteZ()*).

El desarrollador del software de gestión para puntos de venta deberá consultar en el manual de comandos de la impresora fiscal Hasar si el comando generado por el método *ConfigurarControladorPorBloque()* está soportado por su set de comandos, y cuales de estos parámetros de comportamiento son ignorados y cuales tiene efecto; además de consultar los valores posibles para dichos parámetros.

```
Dim lim_cf As Double, lim_tf As Double, iva_noi As Double
Dim cambio As Boolean, leyend As Boolean
```

```
Dim nro_copias As NumeroDeCopias
Dim corte As TipoDeCorteDePapel
```

```
lim_cf = 1000
lim_tf = 5000
iva_noi = 0
nro_copias = DUPLICADO
cambio = True
leyend = True
corte = True
```

```
HASAR1.ReporteZ
HASAR1. ConfigurarControladorCompleto lim_cf, lim_tf, iva_noi, nro_copias, _
cambio, leyend, corte
```

Los argumentos "lim_cf" (monto límite para tiques factura "B" a consumidor final sin datos completos), y "lim_tf" (monto límite para la emisión de tiques factura) tienen un valor establecido por la normativa fiscal vigente que no debe ser modificado, salvo indicación explícita de dicha normativa. El argumento "iva_noi" (sobretasa de iva no inscripto ya no es de aplicación, según la normativa fiscal vigente).

El argumento "nro_copias" permite establecer la cantidad de ejemplares que permitirá emitir la impresora fiscal Hasar.

El argumento "*cambio*" indica si se debe realizar la impresión de la leyenda "CAMBIO \$0.00".

El argumento "*leyend*" indica si deben imprimirse las leyendas opcionales (inscripción en ingresos brutos, etc.).

El argumento "*corte*" permite establecer el tipo de corte de papel cuando la impresora fiscal Hasar deba realizar tal acción.

Constante	dec	hex
NO_COPIAS	48	30
ORIGINAL	49	31
DUPLICADO	50	32
TRIPLICADO	51	33
CUADRUPLICADO	52	34

El tipo de datos *NumeroDeCopias* puede ser alguna de las constantes embebidas en el OCX Fiscal Hasar que se listan en la tabla de la izquierda.

Constante	dec	hex
CORTE_TOTAL	70	46
NO_CORTE	78	4E
CORTE_PARCIAL	80	50

El tipo de datos *TipoDeCorteDePapel* puede ser alguna de las constantes embebidas en el OCX Fiscal Hasar que se listan en la tabla de la izquierda.

El software de gestión para puntos de venta debería ofrecer estas opciones de configuración al usuario final.

Método: CortarComprobanteNoFiscal()

010724	✓
051122	✓

Corta el papel de un comprobante no fiscal.

Este método del OCX Fiscal Hasar genera y envía el string del comando *CutNonFiscalReceipt*.

Mediante el método *CortarComprobanteNoFiscal()* el software de gestión para puntos de venta puede provocar el corte de papel (formato tique) cuando esté emitiendo un comprobante no fiscal.

HASAR1.CortarComprobanteNoFiscal

Método: DatosCliente()

010724	✓
051122	✓

Datos del cliente para facturación.

Este método del OCX Fiscal Hasar genera y envía el string del comando *SetCustomerData*.

El método *DatosCliente()* debe ser invocado por el software de gestión para puntos de venta en forma previa al método de apertura de un comprobante fiscal -*AbrirComprobanteFiscal()*-, o al método de apertura de una nota de crédito -*AbrirComprobanteNoFiscalHomologado()*-.

El método *DatosCliente()* puede ser opcionalmente omitido en algunos modelos de impresoras fiscales Hasar, pero solamente en el caso de emisión de facturas o tiques factura "B" a consumidor final. En tal caso, el monto del comprobante queda limitado a \$1000.00 por exigencias de la normativa fiscal vigente.

```

Dim rsocial As String, nro_doc As String, domic As String
Dim tipo_doc As TiposDeDocumento
Dim riva As TiposDeResponsabilidades

rsocial = "Nombre del Cliente..."
nro_doc = "99999999995"
domic = "Domicilio Cliente..."

tipo_doc = TIPO_CUIT
riva = RESPONSABLE_INSCRIPTO

HASAR1.DatosCliente rsocial, nro_doc, tipo_doc, riva, domic
    
```

El argumento "*rsocial*" permite indicar el nombre o la razón social del cliente al que se le emitirá el comprobante.

El argumento "*nro_doc*" permite indicar el número de documento que presenta el cliente. De tratarse de un número de CUIT(L) deben ser 11 dígitos y satisfacer el algoritmo de verificación de AFIP.

El argumento "*tipo_doc*" permite indicar cual es el tipo de documento presentado por el cliente.

El argumento "*riva*" permite indicar cual es la responsabilidad frente a IVA del cliente.

El argumento "*domic*" es opcional. Su presencia depende del comando *SetCustomerData* que intergra el set de comandos de la impresora fiscal Hasar en uso.

Constante	dec	hex
TIPO_NINGUNO	32	20
TIPO_LE	48	30
TIPO_LC	49	31
TIPO_DNI	50	32
TIPO_PASAPORTE	51	33
TIPO_CI	52	34
TIPO_CUIT	67	43
TIPO_CUIL (*)	76	4C

El tipo de datos *TiposDeDocumento* puede ser alguna de las constantes embebidas en el OCX Fiscal Hasar que se listan en la tabla de la izquierda.

(*) Estas constantes embebidas no existen en la versión 010724 del OCX Fiscal Hasar.

Constante	dec	hex
NO_RESPONSABLE	65	41
BIENES_DE_USO	66	42
CONSUMIDOR_FINAL	67	43
RESPONSABLE_EXENTO	69	45
RESPONSABLE_INSCRIPTO	73	49
MONOTRIBUTO	77	4D
RESPONSABLE_NO_INSCRIPTO	78	4E
MONOTRIBUTO_SOCIAL (*)	83	53
NO_CATEGORIZADO	84	54
EVENTUAL (*)	86	56
EVENTUAL_SOCIAL (*)	87	57

El tipo de datos *TiposDeResponsabilidades* puede ser alguna de las constantes embebidas en el OCX Fiscal Hasar que se listan en la tabla de la izquierda.

(*) Estas constantes embebidas no existen en la versión 010724 del OCX Fiscal Hasar.

En la versión 010724 del OCX Fiscal Hasar se produce una excepción ("*Error no identificado*") cuando el software de gestión para puntos de venta indica el domicilio y no corresponde, o cuando lo omite y es obligatorio.

En la versión 051122 del OCX Fiscal Hasar se produce una excepción ("*Parámetro opcional no es válido o está ausente cuando es necesario (Cargar Datos del Cliente)*") cuando el software de gestión para puntos de venta cuando omite el domicilio e indicarlo es obligatorio. En cambio, si indica el domicilio cuando no corresponde se genera el evento *ErrorFiscal()* ("*Campo de datos inválido*").

Método: *DescripcionStatusFiscal()*

010724	✓	Dado un número de bit individual de la palabra de status fiscal, obtiene su descripción. Este método del OCX Fiscal Hasar no genera ni envía ningún string de comando.
051122	✓	

Dim texto As String
Dim param As Long

param = F_INVALID_FIELD_DATA
texto = HASAR1.DescripcionStatusFiscal param

Mediante el método *DescripcionStatusFiscal()* el software de gestión para puntos de venta puede mostrar al usuario final un mensaje asociado a la generación de los eventos *ErrorFiscal()* o *EventoFiscal()*.

Private Sub HASAR1_ErrorFiscal(ByVal Flags As Long)

MsgBox "Error Fiscal: " & HASAR1.DescripcionStatusFiscal(Flags)

End Sub

Si la propiedad *EventosIndividuales()* está seteada con el valor "falso" no se podrá utilizar el número crudo -que ingresa como argumento- generado por los eventos mencionados (se deben separar los bits previamente para procesarlos individualmente).

La excepción *H_ERR_INVALID_BIT* es generada por el OCX Fiscal Hasar cuando el valor especificado para obtener la descripción es inválido.

Constante	dec	hex	Mensaje
F_FISCAL_MEMORY_FAIL	1	1	Falla de memoria fiscal.
F_WORKING_MEMORY_FAIL	2	2	Falla de memoria de trabajo.
F_ALWAYS_ZERO	4	4	
F_UNRECOGNIZED_COMMAND	8	8	Comando no reconocido.
F_INVALID_FIELD_DATA	16	10	Campo de datos inválido.
F_INVALID_COMMAND	32	20	Comando inválido para el estado actual.
F_TOTAL_OVERFLOW	64	40	Rebose de total.
F_FISCAL_MEMORY_FULL	128	80	Memoria fiscal llena, bloqueada o dada de baja.
F_DATE_SET_FAIL	2048	800	Fallo en la puesta de la fecha/hora.
F_LOGIC_OR_0_8	32768	8000	Error fatal fiscal.

Las constantes embebidas que se listan en la tabla anterior están asociadas a la generación del evento *ErrorFiscal()*.

Constante	dec	hex	Mensaje
F_FISCAL_MEMORY_NEAR_FULL	256	100	Memoria fiscal a punto de llenarse.
F_FISCAL_TERMINAL_CERTIFIED	512	200	Terminal fiscal certificada.
F_FISCAL_TERMINAL_FISCALIZED	1024	400	Terminal fiscal inicializada.
F_RECEIPT_SLIP_OPEN	4096	1000	Comprobante fiscal abierto.
F_RECEIPT_OPEN	8192	2000	Comprobante abierto.
F_INVOICE_OPEN	16384	4000	Factura abierta.

Las constantes embebidas que se listan en la tabla anterior están asociadas a la generación del evento *EventoFiscal()*.

Método: *DescripcionStatusImpresor()*

010724	✓
051122	✓

Dado un número de bit individual de la palabra de status de impresora, obtiene su descripción. Este método del OCX Fiscal Hasar no genera ni envía ningún string de comando.

Dim texto As String
Dim param As Long

param = P_OFFLINE
texto = HASAR1.DescripcionStatusFiscal param

Mediante el método *DescripcionStatusImpresora()* el software de gestión para puntos de venta puede mostrar al usuario final un mensaje asociado a la generación de los eventos *ErrorImpresora()* o *EventoImpresora()*.

Private Sub HASAR1_ErrorImpresora(ByVal Flags As Long)

MsgBox "Error Impresora: " & HASAR1.DescripcionStatusImpresora(Flags)

End Sub

Si la propiedad *EventosIndividuales()* está seteada con el valor "falso" no se podrá utilizar el número crudo -que ingresa como argumento- generado por los eventos mencionados (se deben separar los bits previamente para procesarlos individualmente).

La excepción H_ERR_INVALID_BIT es generada por el OCX Fiscal Hasar cuando el valor especificado para obtener la descripción es inválido.

Constante	dec	hex	Mensaje
P_JOURNAL_PAPER_LOW	16	10	Falta papel en diario.
P_RECEIPT_PAPER_LOW	32	20	Falta papel en ticket.
P_BUFFER_EMPTY	128	80	El buffer de transmisión está vacío.
P_DRAWER_CLOSED	16384	4000	El cajón de dinero está cerrado.

(*) No existen en la versión 010724 del OCX Fiscal Hasar.

Las constantes embebidas que se listan en la tabla anterior están asociadas a la generación del evento *ErrorImpresora()*.

Constante	dec	hex	Mensaje
P_PRINTER_ERROR	4	4	Error mecánico de impresora
P_OFFLINE	8	8	Impresora fuera de línea
P_BUFFER_FULL	64	40	El buffer de transmisión está completo.
P_SLIP_PLATEN_OPEN	256	100	La tapa de la impresora está abierta

Las constantes embebidas que se listan en la tabla anterior están asociadas a la generación del evento *EventoImpresora()*.

Si bien la constante embebida P_PRINTER_ERROR identifica un error mecánico de la impresora fiscal Hasar (por ejemplo, si se traba el cabezal de impresión), en algunos modelos este evento se puede generar por un acuse de falta de papel porte de la impresora fiscal Hasar.

Método: DescuentoGeneral()

010724	✓	Realiza un descuento o un recargo sobre el total de la factura.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del co-

mando *GeneralDiscount*.

```

Dim descrip As String
Dim monto As Double
Dim negativo As Boolean

descrip = "Oferta Especial..."
monto = 10.0
negativo = True

HASAR1.DescuentoGeneral descrip, monto, negativo

```

Mediante el método *DescuentoGeneral()* el software de gestión para puntos de venta puede indicar un monto de descuento/recargo que en la impresora fiscal Hasar se traduce en su aplicación sobre el monto total de la factura. Esto significa que el monto del descuento/recargo será prorrateado entre todas las alícuotas de IVA intervinientes en el comprobante fiscal (o nota de crédito) que se está emitiendo.

El método DescuentoGeneral() puede ser invocado más de una vez si fuese necesario, pero una vez invocado no se pueden realizar más ventas (invocar al método ImprimirItem()).

El método DescuentoGeneral() no afecta a los montos de impuestos internos fijos acumulados en el comprobante en curso.

Método: DescuentoUltimoItem()

010724	✓	Efectúa un descuento o un recargo sobre el último ítem vendido.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>LastItemDiscount</i> .

```

Dim descrip As String
Dim monto As Double
Dim negativo As Boolean

descrip = "Oferta sobre Artículo..."
monto = 5.0
negativo = True

HASAR1.DescuentoUltimoItem descrip, monto, negativo

```

Mediante el método *DescuentoUltimoItem()* el software de gestión para puntos de venta puede indicar un monto de descuento/recargo que en la impresora fiscal Hasar se traduce en su aplicación sobre el último ítem vendido. Esto significa que el monto del descuento/recargo será afectado por los mismos impuestos (IVA e internos) que el ítem sobre el cual se aplica.

El método *DescuentoUltimoItem()* puede ser invocado solamente una vez luego de una venta (haber invocado al método *ImprimirItem()*).

El método *DescuentoUltimoItem()* no afecta a los montos de impuestos internos fijos acumulados en el comprobante en curso.

Método: *DetalleRecibo()*

010724	✓	Envía al controlador la información detallada de un recibo.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>ReceiptFiscalText</i> .

Mediante el método *DetalleRecibo()* el software de gestión para puntos de venta puede imprimir líneas de texto en un recibo ("A", "B/C", o "X") que conformarán el "*En concepto de...*" para explicar el monto del documento.

Dim texto As String

```
texto = "Seña TV 29 pulg."
HASAR1.DetalleRecibo texto
```

El método *DetalleRecibo()* separa el texto en dos o más líneas en caso de superar el máximo aceptado por la impresora fiscal Hasar para una línea.

El desarrollador del software de gestión para puntos de venta deberá consultar en el manual de comandos de la impresora fiscal Hasar en uso la longitud máxima para cada línea de texto, y la máxima cantidad de líneas soportadas.

Método: *DevolucionDescuento()*

010724	✓	Efectúa un descuento o un recargo a un IVA, o una devolución de envases. Este método del OCX Fiscal Hasar genera y envía el string del comando <i>ReturnRecharge</i> .
051122	✓	

```
Dim descrip As String
Dim monto As Double, miva As Double, impint As Double
Dim negativo As Boolean
Dim uso As TiposDeDescuentos
```

```
descrip = "Oferta sobre IVA 21%..."
```

```

monto = 5.0
miva = 21.0
impint = 0.0
negativo = True
uso = DESCUENTO_RECARGO
    
```

HASAR1.DevolucionDescuento **descrip, monto, miva, impint, negativo, uso**

Uno de los usos del método *DevolucionDescuento()* permite al software de gestión para puntos de venta indicar un monto de descuento/recargo que en la impresora fiscal Hasar se traduce en su aplicación sobre todos los ítems vendidos que se encuentren afectados por la alícuota de IVA indicada mediante el argumento "miva", y el impuesto interno indicado mediante el argumento "impint".

El otro uso del método *DevoluciónDescuento()* permite el manejo de devolución de envases.

Constante	dec	hex
DESCUENTO_RECARGO	66	42
DEVOLUCION_ENVASES	69	45

El tipo de datos *TiposDeDescuentos* puede ser alguna de las constantes embebidas en el OCX Fiscal Hasar que se

listan en la tabla superior.

El método DevolucionDescuento() puede ser invocado más de una vez si fuese necesario, pero una vez invocado no se pueden realizar más ventas (invocar al método ImprimirItem()).

Método: DNFHFarmacias()

010724	✓
051122	✓

Imprime el Documento No Fiscal Homologado de Farmacias.

Este método del OCX Fiscal Hasar genera y envía el string del comando *DNFHFarmacia*.

Este método se encuentra disponible para el software de gestión de puntos de venta ubicados en farmacias. Su uso no está restringido a otros rubros si resultare de utilidad.

Dim copias As Long

copias = 2

.....

HASAR1.CerrarComprobanteFiscal

HASAR1.DNFHFarmacias copias

Mediante el método *DNFHFarmacias()* el software de gestión para puntos de venta puede emitir un comprobante para obras sociales a partir de la venta realizada en el comprobante fiscal emitido en forma inmediatamente anterior. El software solamente podrá incorporar texto en líneas de encabezado o pie del comprobante.

El argumento "*copias*" permite indicar la cantidad de ejemplares que deberán imprimirse.

Método: *DNFHReparto()*

010724	✓	Imprime el Documento No Fiscal Homologado de Reparto.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>DNFHReparto</i> .

Este método se encuentra disponible para el software de gestión de puntos de venta ubicados en locales comerciales con entrega a domicilio, y su uso está destinado a facilitar el control de lo enviado, transportado, y/o recibido.

Dim copias As Long

copias = 2

.....

HASAR1.CerrarComprobanteFiscal
HASAR1.DNFHReparto copias

Mediante el método *DNFHReparto()* el software de gestión para puntos de venta puede emitir un comprobante de control de mercadería, a partir de la venta realizada en el comprobante fiscal emitido en forma inmediatamente anterior. El software solamente podrá incorporar texto en líneas de encabezado o pie del comprobante.

El argumento "*copias*" permite indicar la cantidad de ejemplares que deberán imprimirse.

Método: *Enviar()*

010724	✓	Envía un comando crudo al impresor.
051122	✓	Este método del OCX Fiscal Hasar no genera ningún string de comando.

El método *Enviar()* se encuentra disponible para ser invocado por el software de gestión para puntos de venta, solamente en aquellos casos que los métodos del OCX re-

sulten insuficientes, o incompatibles, para generar apropiadamente el string de comando requerido por la impresora fiscal Hasar en uso.

El método *Enviar()* permite hacer llegar a la impresora fiscal Hasar un string de comando cualquiera -dentro del set soportado por el equipo- construido por el software de gestión para puntos de venta. Es claro que la construcción de dicho string debe respetar la sintaxis explicada en el manual de comandos de la impresora fiscal Hasar.

Veamos el caso del comando de apertura de un tique nota de crédito "B" utilizando una impresora fiscal Hasar SMH/P-715F. Este ejemplo se ocupa solamente del comando mencionado, y no de los previos -y obligatorios- necesarios para abrir una nota de crédito.

Con la versión *051122* del OCX Fiscal Hasar se resuelve sencillamente del siguiente modo:

```
HASAR1.Modelo = MODELO_715
.....
HASAR1.AbrirComprobanteNoFiscalHomologado TICKET_NOTA_CREDITO_B
```

En cambio, con la versión *010724* del OCX Fiscal Hasar se debe recurrir al método *Enviar()*:

Dim comando As String, FS As String

```
HASAR1.Modelo = MODELO_615
.....
FS = Chr(28)
comando = Chr(128) & FS & "S" & FS & "T"
HASAR1.Envia comando
```

Notar que la versión *010724* del OCX Fiscal Hasar no tiene una constante embebida para identificar a la impresora fiscal Hasar SMH/P-715F. Ello implica setear la propiedad *Modelo()* de manera que se considere a una impresora fiscal Hasar lo más similar posible en cuanto al set de comandos soportados; por ejemplo la SMH/P-615F. Resuelto este primer problema, el software de gestión para puntos de venta se encuentra luego con otro, consecuencia del seteo realizado: no se puede invocar al método *AbrirComprobanteNoFiscalHomologado()* por no estar soportado el comando que se genera (*OpenDNFH*) en la impresora fiscal Hasar SMH/P-615F. Es en este punto donde el desarrollador del software debe programar la construcción del string del comando e invocar al método *Enviar()*.

Método: *EnviarStatPrn()*

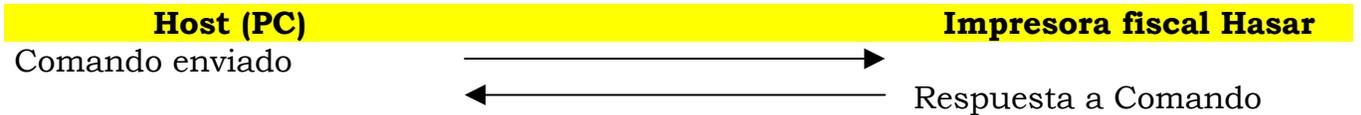
010724	✓	Envía un comando <i>StatPRN</i> .
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>StatPRN</i> .

El método *EnviarStatPrn()* permite realizar consultas de estado intermedio a la impresora fiscal Hasar. Se denomina estado intermedio al periodo de tiempo excedente al que normalmente transcurre entre la llegada de un comando a la impresora fiscal Hasar y la devolución de su correspondiente respuesta. Cuando se inicia la demora en la entrega de la respuesta esperada la impresora fiscal Hasar informa que tal situación ocurre entregando una primera respuesta que corresponde al comando *StatPRN* (como si éste se hubiera enviado). Tal situación se traduce en el OCX Fiscal Hasar seteando la propiedad *HuboStatPrn()* con el valor "verdadero".

La demora en la entrega de la respuesta a un comando puede ocurrir por la propia naturaleza de la ejecución del comando enviado, por falta de papel, etc.

Mientras la respuesta esperada se encuentra demorada se puede consultar a la impresora fiscal Hasar enviando un comando *StatPRN* mediante el método *EnviarStatPrn()*. Si la respuesta que se recibe corresponde al comando *StatPRN* enviado -la propiedad *HuboStatPrn()* se encuentra en "verdadero"- se puede enviar otro comando *StatPRN* (nuevamente con *EnviarStatPrn()*). Si la respuesta recibida es la esperada, entonces se puede enviar el siguiente comando a la impresora fiscal Hasar.

Caso Normal



Caso con consulta de estado intermedio



El manejo de la consulta de estado intermedio puede hacerlo:

1. Directamente el OCX Fiscal Hasar (transparente para el software).
2. El software de gestión para puntos de venta.

En el segundo caso se deben dar una serie de condiciones para que ello sea posible:

- a. La impresora fiscal Hasar debe soportar la consulta de estado intermedio. Con la versión *010724* del OCX Fiscal Hasar no hay forma de saberlo. Se debe consultar el manual de comandos de la impresora fiscal Hasar para saber si el comando *StatPRN* integra el set de comandos soportados. Con la versión *051122* del OCX Fiscal Hasar se puede consultar el estado la propiedad booleana *SoportaStatPRN()*.
- b. Con la versión *010724* del OCX Fiscal Hasar deben estar seteadas las siguientes propiedades del modo que se indica:

```
HASAR1.ReintentoConstante = False
HASAR1.ModoStatPrn = True
```

Con la versión *051122* del OCX Fiscal Hasar deben estar seteadas las siguientes propiedades del modo que se indica:

```
HASAR1.ReintentoConstante = True / False
HASAR1.ModoStatPrn = True
```

El siguiente ejemplo (basado en la versión *010724* del OCX Fiscal Hasar) permite al software de gestión para puntos de venta el manejo de la consulta de estado intermedio:

```
Dim cmd_statprn As String

cmd_statprn = Chr(161)
HASAR1.ReintentoConstante = False
HASAR1.ModoStatPrn = True

.....
HASAR1.AbrirComprobanteFiscal

If (HASAR1.HuboStatPrn) Then
  Do
    HASAR1.Enviar cmd_statprn
  While (Not(HASAR1.HuboStatPrn))
End If
```

El siguiente ejemplo (basado en la versión *051122* del OCX Fiscal Hasar) permite al software de gestión para puntos de venta el manejo de la consulta de estado intermedio:

```

HASAR1.ReintentoConstante = False
HASAR1.ModoStatPrn = True

.....
HASAR1.AbrirComprobanteFiscal

If (HASAR1.HuboStatPrn) Then
  Do
    HASAR1.EnviarStatPrn
  While (Not(HASAR1.HuboStatPrn))
End If

```

Método: *EspecificarIVANoInscripto()*

010724	✓
051122	✓

Usa un IVA no inscripto distinto al calculado por el controlador. Este método del OCX Fiscal Hasar genera y envía el string del comando *ChargeNonRegisteredTax*.

Las impresoras fiscales Hasar cuentan con un parámetro de configuración que es la sobretasa (50%) a aplicar sobre cada una de las alícuotas de IVA intervinientes en el comprobante en emisión, cuando el emisor es un responsable inscripto y el receptor es un responsable no inscripto.

Puede ocurrir que por exigencias de la normativa fiscal vigente el software de gestión para puntos de venta deba calcular (y aplicar) el monto que se origina por dicha sobretasa, y no el que calcula automáticamente la impresora fiscal Hasar (por citar un caso, la venta de combustibles y otros productos en un mismo comprobante).

Dim monto As Double

```

monto = 125.56
HASAR1.EspecificarIVANoInscripto monto

```

En la versión 051122 del OCX Fiscal Hasar el método *EspecificarIVANoInscripto()* se mantiene por compatibilidad para productos software que lo utilicen. Con algunos modelos de impresoras fiscales Hasar la invocación del método *EspecificarIVANoInscripto()* genera una excepción por "*Función no soportada*".

La normativa fiscal vigente ha eliminado las responsabilidades IVA No Inscripto y Venta Bienes de Uso, razón por la cual el método EspecificarIVANoInscripto() no debería ser invocado nunca.

Método: *EspecificarNombreDeFantasia ()*

010724	✓
051122	⊗

El nombre de fantasía se usa en caso de que la razón social no coincida con el nombre comercial de la firma. Este método del OCX Fiscal Hasar genera y envía el string del comando *SetFantasyName*.

En la versión 051122 del OCX Fiscal Hasar el método *EspecificarNombreDeFantasia()* se encuentra "discontinuada" (se debe emplear la propiedad *NombreDeFantasia()*).

El método *EspecificarNombreDeFantasia()* permite al software de gestión para puntos de venta cargar la memoria de trabajo de la impresora fiscal Hasar con dos líneas de texto cuyo contenido suele ser el nombre de fantasía empleado por el comercio, o la empresa, y que no tiene porque coincidir con la razón social. La ubicación de la impresión de las líneas de fantasía, así como la longitud de los textos debe consultarse en el manual de comandos de la impresora fiscal Hasar en uso.

(010724) **Dim linea1 As String, linea2 As String**

linea1 = "Texto 1 fantasía"

linea2 = "Texto 2 fantasía"

HASAR1.EspecificarNombreDeFantasia linea1, linea2

Método: *EspecificarPercepcionGlobal()*

010724	✓
051122	✓

Especifica una percepción no ligada a un IVA en particular.

Este método del OCX Fiscal Hasar genera y envía el string del comando *Perceptions*.

Dim descrip As String

Dim monto As Double

descrip = "Percep. Global 1"

monto = 57.89

HASAR1.EspecificarPercepcionGlobal descrip, monto

Mediante el método *EspecificarPercepcionGlobal()* el software de gestión para puntos de venta puede imprimir en un comprobante fiscal (o una nota de crédito) el detalle

de una retención realizada al cliente. En general se trata de retenciones aplicadas sobre el total del comprobante.

Es responsabilidad del software de gestión para puntos de venta la aplicación (cuando correspondiere) y el cálculo del monto de la percepción, de acuerdo a las indicaciones de la normativa fiscal vigente.

Método: EspecificarPercepcionPorIVA()

010724	✓	Especifica una percepción ligada a un IVA particular.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>Perceptions</i> .

Dim descrip As String

Dim monto As Double, alic As Double

descrip = "Percep. Global 1"

monto = 57.89

alic = 21.0

HASAR1.EspecificarPercepcionGlobal descrip, monto, alic

Mediante el método *EspecificarPercepcionPorIVA()* el software de gestión para puntos de venta puede imprimir en un comprobante fiscal (o una nota de crédito) el detalle de una retención realizada al cliente. En general se trata de retenciones que se aplican sobre ítems afectados por determinada alícuota de IVA.

Es responsabilidad del software de gestión para puntos de venta la aplicación (cuando correspondiere) y el cálculo del monto de la percepción, de acuerdo a las indicaciones de la normativa fiscal vigente.

Método: Finalizar()

010724	✓	Finaliza una sesión cerrando el puerto.
051122	✓	Este método del OCX Fiscal Hasar no genera y no envía ningún string de comando a la impresora fiscal Hasar.

HASAR1.Finalizar

Mediante el método *Finalizar()* el software de gestión para puntos de venta puede dar por terminada una sesión de trabajo, liberando los recursos asociados a las comunicaciones con la impresora fiscal Hasar.

Es responsabilidad del software de gestión para puntos de venta invocar al método *Finalizar()* cada vez que termina la ejecución del programa (cualquiera sea el punto de salida utilizado).

Algunas observaciones sobre el método *Finalizar()*:

Caso 1: Invocar al método *Finalizar()* sin haber invocado al método *Comenzar()*. Tanto en la versión 010724 del OCX Fiscal Hasar, como en la versión 051122 no se genera excepción o evento alguno.

Caso 2: Invocar al método *Finalizar()* luego de haber invocado al método *Finalizar()*. Tanto en la versión 010724 del OCX Fiscal Hasar, como en la versión 051122 no se genera excepción o evento alguno.

Método: *ImprimirCodigoDeBarras()*

010724	✓	Imprime un código de barras.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>BarCode</i> .

```
Dim tipo_cod As TiposDeCodigoDeBarras
Dim codigo As String
Dim impr_nro As Boolean, impr_ya As Boolean
```

```
tipo_cod = CODIGO_TIPO_EAN_13
codigo = "779123456789"
impr_nro = True
impr_ya = False
```

```
HASAR1.ImprimirCodigoDeBarras tipo_cod, codigo, impr_nro, impr_ya
```

Constante	dec	hex
CODIGO_TIPO_EAN_13	49	31
CODIGO_TIPO_EAN_8	50	32
CODIGO_TIPO_UPCA	51	33
CODIGO_TIPO_ITS	52	34

El tipo de datos *TiposDeCodigoDeBarras* puede ser alguna de las constantes embebidas en el OCX Fiscal Hasar que se listan en la tabla de la izquierda.

El argumento "codigo" permite indicar el string del código de barras propiamente dicho, y se deben respetar las siguiente pautas:

EAN 13	12 dígitos exactos (no incluir dígito verificador).
EAN 8	7 dígitos exactos (no incluir dígito verificador).
UPCA	11 dígitos exactos (no incluir dígito verificador).
ITS	Cantidad par de dígitos hasta un máximo de 16 pares (*).

(*) *La cantidad máxima de pares de dígitos puede variar no sólo de un modelo a otro de impresora fiscal Hasar, sino también entre documentos disponibles sobre un mismo modelo.*

El argumento "impr_nro" permite indicar si el string "codigo" debe ser impreso debajo de la impresión de las barras.

El argumeto "impr_ya" permite indicar si el código de barras debe imprimirse al instante, o debe postergarse su impresión hasta su ubicación por defecto en el documento. El valor "verdadero" puede ser ignorado por las impresoras fiscales Hasar en algunos documentos en particular (como ser facturas), por tener éstos un diseño preestablecido por la normativa fiscal vigente. En tales casos, se posterga su impresión hasata llegar a la ubicación por defecto.

El desarrollador del software de gestión para puntos de venta debe tener presente que no todos los modelos de impresoras fiscales Hasar imprimen códigos de barras, que los modelos que sí lo soportan pueden no imprimirlo sobre todos los documentos, o pueden no imprimir todos los códigos de barras mencionados.

Método: ImprimirInfoEnDNFH()

010724	⊙
051122	✓

Imprime información en documento no fiscal homologado.
Este método del OCX Fiscal Hasar genera y envía el string del comando *PrintDNFHInfo*.

Una vez abierto el documento no fiscal homologado sobre el que se imprimirá texto, el método *ImprimirInfoEnDNFH()* debe ser invocado como mínimo una vez en cada área de impresión disponible en el documento abierto.

```
(051122) Dim campo As Long
Dim texto As String
Dim Comprob As DocumentosNoFiscales
Dim nro_dnfh As String
.....
Comprob = COMUNICACION_CLIENTE
nro_dnfh = "00000123"
HASAR1.AbrirComprobanteFiscal Comprob, nro_dnfh
.....
campo = 1
```

texto = "Estimado Cliente: Hoy nos acercamos a Ud....."
HASAR1.ImprimirInfoEnDNFH campo, texto

El argumento "*campo*" puede tomar los valores 1, 2 o 3 para indicar en que área del documento no fiscal homologado se imprimirá el contenido del argumento "*texto*". Dependiendo del documento no fiscal homologado abierto se dispondrá de una, dos, o las tres áreas de impresión. En la página siguiente se muestra una tabla con los distintos documentos sobre los que puede aplicarse el método *ImprimirInfoEnDNFH()*.

El desarrollador de software de gestión para puntos de venta deberá consultar en el manual de comandos de la impresora fiscal Hasar cuantas son las áreas de impresión disponibles en cada uno de los documentos no fiscales homologados listados.

El argumento "*texto*" admitirá una longitud determinada para cada área de impresión y documento no fiscal homologado (listado en la tabla de esta página) que ha sido abierto. En cada área de impresión el método *ImprimirInfoEnDNFH()* podrá ser invocado una cantidad prefijada de veces que puede no ser la misma en todas ellas, no coincidir en distintos documentos no fiscales homologados; e incluso, cuando la impresora fiscal se encuentra en modo entrenamiento, la cantidad de invocaciones admitidas por área puede ser menor a la aceptada cuando el equipo se encuentra fiscalmente inicializado.

El desarrollador de software de gestión para puntos de venta deberá consultar en el manual de comandos de la impresora fiscal Hasar cuantas veces, por área de impresión, es posible invocar al método ImprimirInfoEnDNFH(), así como las longitudes admitidas para los textos.

Constante	dec	hex
COMUNICACION_CLIENTE	33	21
OFRECIMIENTO_CREDITO	34	22
OFRECIMIENTO_TARJETA	35	23
MINUTA_CREDITO	36	24
OFRECIMIENTO_PASAPORTE	37	25
RENOVACION_CREDITO	38	26
ADELANTO_REMUNERACION	39	27
SOLICITUD_TARJETA_DEBITO	40	28
SOLICITUD_CLAVE_TARJETA	41	29
RESCATE_MERCADERIA	42	2A
INGRESO_EGRESO_SUCURSAL	43	2B
CLAUSULA_CREDITO	58	3A
CLAUSULA_SEGURO	59	3B
PAGARE	60	3C
POLIZA_SEGURO	61	3D
RECORDATORIO	62	3E
SOLICITUD_CREDITO	63	3F

El tipo de dato *DocumentosNoFiscales* puede ser alguna de las constantes embebidas en el OCX Fiscal Hasar que se listan en la tabla de la izquierda. Notar que no sobre todos los documentos no fiscales homologados posibles se puede invocar al método *ImprimirInfoEnDNFH()*.

Método: ImprimirItem()

010724	✓	Imprime un ítem de venta.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>PrintLineItem</i> .

Mediante el método *ImprimirItem()* el software de gestión para puntos de venta podrá imprimir un ítem sobre un documento fiscal abierto, una nota de crédito, o un recibo "X" (con objeto de generar un monto para el comprobante).

Es importante que el desarrollador del software de gestión para puntos de venta tenga presente que influyen en el comportamiento del método *ImprimirItem()* -a la hora de generar el string del comando *PrintLineItem*- el valor de determinadas propiedades del OCX Fiscal Hasar y el contenido de los argumentos suministrados por el software de gestión para puntos de venta. Una vez enviado el string de comando, la impresora fiscal se comportará como indica tal comando y teniendo en cuenta el documento que ha sido abierto.

El método *ImprimirItem()* podrá invocarse cuantas veces resulte necesario. En el caso de impresoras fiscales Hasar que emiten facturas y notas de crédito, el transporte a la siguiente hoja es transparente para el software de gestión para puntos de venta. Si el ítem a imprimir no cuenta con espacio para su impresión en la página actual, la impresora fiscal Hasar automáticamente imprime el transporte a la hoja siguiente al pie de la página actual, pasa a la siguiente hoja, imprime el transporte proveniente de la página anterior y luego el ítem que se quería imprimir.

La situación descrita del transporte a la siguiente hoja no es exclusiva del uso del método ImprimirItem(). Puede ocurrir con cualquier comando que genere impresión sobre el documento abierto.

Se debe tener presente que la normativa fiscal vigente le exige a las impresoras fiscales incrementar el número de documento en cada nueva hoja.

```

Dim descrip As String
Dim cant As Double, monto As Double, miva As Double, impint As Double
.....
descrip = "Artículo Uno"
cant = 1
monto = 100
miva = 21
impint = 0

```

```
HASAR1.ImprimirItem descrip, cant, monto, miva, impint
```

El argumento "*descrip*" permite, al software de gestión para puntos de venta, indicar el texto que describe al ítem a detallar en el comprobante abierto. La longitud de este texto varía entre modelos de impresoras fiscales y entre documentos soportados por el mismo modelo.

El software de gestión para puntos de venta puede extender la descripción del ítem a imprimir mediante dos mecanismos:

- a. Invocación del método *ImprimirTextoFiscal()* del OCX Fiscal Hasar.
- b. Asignándole a la propiedad *DescripcionesLargas()* el valor "verdadero". Propiedad, ésta, del OCX Fiscal Hasar.

Adaptando el ejemplo visto al caso **a.**, resulta:

```

Dim descrip As String, previo1 As String, previo2 As String
Dim cant As Double, monto As Double, miva As Double, impint As Double
.....
previo1 = "Descripción I..."
HASAR1.ImprimirtextoFiscal previo1

previo2 = "Descripción II..."
HASAR1.ImprimirtextoFiscal previo2

descrip = "Artículo Uno"
cant = 1
monto = 100
miva = 21
impint = 0

HASAR1.ImprimirItem descrip, cant, monto, miva, impint

```

Adaptando el ejemplo anterior al caso **b.**, resulta:

```

Dim descrip As String
Dim cant As Double, monto As Double, miva As Double, impint As Double
.....
HASAR1.DescripcionesLargas = True

descrip = "Descripción I...Descripcion II...Artículo Uno"
cant = 1
monto = 100
miva = 21
impint = 0

HASAR1.ImprimirItem descrip, cant, monto, miva, impint

```

El argumento "*cant*" permite, al software de gestión para puntos de venta, indicar la cantidad de unidades del ítem a imprimir. Se pueden usar hasta 10 dígitos decimales. Sin embargo, la cantidad de dígitos en parte entera varía entre distintos modelos de impresoras fiscales Hasar: máximo de 3, 6 u 8 dígitos. Si el argumento "*cant*" es distinto de 1, la impresora fiscal Hasar resuelve la multiplicación de "*cant*" por "*monto*". Si el argumento "*cant*" es un número que supera la cantidad de dígitos admitidos en parte entera por la impresora fiscal Hasar en uso, la impresión del ítem puede resolverse del siguiente modo:

```

Dim descrip As String, auxtext As String
Dim cant As Double, monto As Double, miva As Double, impint As Double
.....

auxtext = "Cant= 1500 ; P.Unit= $ 0.50"
HASAR1.ImprimirtextoFiscal auxtext

descrip = "Artículo Uno"
cant = 1
monto = 0.50 * 1500
miva = 21
impint = 0

HASAR1.ImprimirItem descrip, cant, monto, miva, impint

```

El argumento "*monto*" permite, al software de gestión para puntos de venta, indicar el precio unitario del ítem a imprimir. Se pueden usar hasta 6 dígitos en parte entera. Sin embargo, la cantidad de dígitos en la parte decimal varía entre distintos modelos de impresoras fiscales Hasar: máximo 2 o 4 dígitos. Si el argumento "*monto*" es un número que supera la cantidad de dígitos admitidos en parte decimal por la impresora fiscal Hasar en uso, la impresión del ítem puede resolverse de manera similar al ejemplo anterior:

```

Dim descrip As String, auxtext As String
Dim cant As Double, monto As Double, miva As Double, impint As Double
.....

auxtext = "Cant= 20 P.Unit= $ 0.8174"
HASAR1.ImprimirtextoFiscal auxtext

descrip = "Artículo Uno"
cant = 1
monto = 16.35
miva = 21

```

'// 0.8174 * 20 = 16.348

impint = 0

HASAR1.ImprimirItem descrip, cant, monto, miva, impint

El valor que el software de gestión para puntos de venta pasa como argumento "monto", al invocar al método *ImprimirItem()*, debe ser coherente con el valor asignado a la propiedad *PrecioBase()* del Ocx Fiscal Hasar. Si esta propiedad tiene como valor "verdadero" el argumento "monto" no debe incluir los montos generados por la aplicación de los impuestos indicados en los argumentos "miva" (IVA) e "impint" (impuestos internos).

El argumento "miva" permite, al software de gestión para puntos de venta, indicar la alícuota de IVA que afecta al ítem, independientemente del hecho que el monto originado por su aplicación se incluya, o no, en el argumento "monto" y del valor asignado a la propiedad *PrecioBase()* del OCX Fiscal Hasar. Si no se requiere generar monto de IVA el argumento "miva" debe ser cero.

El argumento "impint" permite, al software de gestión para puntos de venta, indicar el impuesto interno que afecta al ítem, independientemente del hecho que el monto originado por su aplicación se incluya, o no, en el argumento "monto" y del valor asignado a la propiedad *PrecioBase()* del OCX Fiscal Hasar. Si no se requiere generar monto de impuesto interno el argumento "impint" debe ser cero.

El OCX Fiscal Hasar, a la hora de generar el string del comando *PrintLineItem*, interpretará el contenido del argumento "impint" de acuerdo a los valores asignados a las propiedades: *KIVA()*, *ImpuestoInternoFijo()* e *ImpuestoInternoPorMonto()*. Resulta evidente que el software de gestión para puntos de venta debe especificar el contenido del argumento "impint" de manera coherente al valor de las propiedades mencionadas.

Propiedades OCX Fiscal HASAR	Valor asignado	Interpretación
<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>	False False False	Porcentaje Variable
<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>	False True True	Monto Fijo
<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>	True False False	Factor K Variable
<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>	True True False	Factor K Fijo

La normativa fiscal vigente exige que para los impuestos internos fijos, su monto acumulado no puede ser afectado por descuentos y/o recargos. Las impresoras fiscales Hasar cumplen con dicha exigencia.

Los montos generados por aplicación de impuestos internos variables sobre los ítems impresos son afectados por la aplicación de descuentos y/o recargos.

Caso: Porcentaje Variable

```

Dim descrip As String
Dim cant As Double, monto As Double, miva As Double, impint As Double
.....
HASAR1.KIVA = False
HASAR1.ImpuestoInternoFijo = False
HASAR1.ImpuestoInternoPorMonto = False
HASAR1.PrecioBase = True

descrip = "Artículo Uno"
cant = 1
monto = 100
miva = 21
impint = 10.0                                     '// 10% sobre precio base

HASAR1.ImprimirItem descrip, cant, monto, miva, impint

```

Caso: Monto Fijo

```

Dim descrip As String
Dim cant As Double, monto As Double, miva As Double, impint As Double
.....
HASAR1.KIVA = False
HASAR1.ImpuestoInternoFijo = True
HASAR1.ImpuestoInternoPorMonto = True
HASAR1.PrecioBase = True

descrip = "Artículo Uno"
cant = 1
monto = 100
miva = 21
impint = 10.0                                     '// $ 10.00 de monto fijo

HASAR1.ImprimirItem descrip, cant, monto, miva, impint

```

Notar en este caso que el valor pasado como argumento "*impint*" es el mismo que para el caso anterior. Sin embargo, será tratado de manera diferente por el estado en que se encuentran las propiedades involucradas (que difieren respecto del caso anterior).

Caso: Factor K Variable

```

Dim descrip As String
Dim cant As Double, monto As Double, miva As Double, impint As Double
.....
HASAR1.KIVA = True
HASAR1.ImpuestoInternoFijo = False

```

```

HASAR1.ImpuestoInternoPorMonto = False
HASAR1.PrecioBase = True

descrip = "Artículo Uno"
cant = 1
monto = 100
miva = 21
impint = 0.90909090 // 10% sobre precio base

HASAR1.ImprimirItem descrip, cant, monto, miva, impint

```

Caso: Factor K Fijo

```

Dim descrip As String
Dim cant As Double, monto As Double, miva As Double, impint As Double

.....
HASAR1.KIVA = True
HASAR1.ImpuestoInternoFijo = True
HASAR1.ImpuestoInternoPorMonto = False
HASAR1.PrecioBase = True

descrip = "Artículo Uno"
cant = 1
monto = 100
miva = 21
impint = 0.90909090 // $10.0 de monto fijo

HASAR1.ImprimirItem descrip, cant, monto, miva, impint

```

Notar en este caso que el valor pasado como argumento "*impint*" es el mismo que para el caso anterior. Sin embargo, será tratado de manera diferente por el estado en que se encuentran las propiedades involucradas (que difieren respecto del caso anterior).

El Factor K es un coeficiente cuya determinación emana de la normativa fiscal vigente. Como es tratado dicho coeficiente por la impresora fiscal Hasar se describe, para quien desee ahondar en el tema, en el archivo *tips.pdf* incluido en el zip de drivers fiscales que se puede descargar gratuitamente de www.hasar.com.

El Factor K es un valor que debe variar entre cero (0) y uno (1). La impresora fiscal Hasar soporta hasta 8 dígitos decimales en la indicación de este dato, y que el software de gestión para puntos de venta puede calcular del siguiente modo:

Supongamos 10% de impuesto interno variable para un producto con un

precio base de \$ 100.00.

Entonces: $ii = 0.10$

Luego: $impint = 1/(1+ ii) = 1/1.1 = 0.90909090$

Supongamos \$ 10.00 de impuesto interno fijo para un producto con un precio base de \$ 100.00.

Entonces: $ii = 10.00/100.00 = 0.10$

Luego: $impint = 1/(1+ ii) = 1/1.1 = 0.90909090$

Obviamente, en ambos casos, las propiedades involucradas del OCX Fiscal Hasar deben tener los valores asignados apropiadamente, según lo visto.

Items negativos

La normativa fiscal vigente impide la venta negativa de ítems que no se hubieran registrado previamente, en positivo, en el comprobante en curso. En consecuencia, la impresión de un ítem negativo tiene como finalidad posibilitar la corrección de una venta errónea, pero no la realización de canjes y/o devoluciones (recordar que la emisión de comprobantes a través de una impresora fiscal debe hacerse en forma "concomitante" -exigencia la normativa fiscal vigente-).

Se podrá hacer una venta negativa del ítem, siempre y cuando su cantidad no supere a la cantidad de la venta positiva del mismo ítem (deben coincidir: descripción, precio e impuestos).

Para que la venta resulte negativa alcanza con indicar el valor del argumento "monto" precedido del signo '-' (menos).

Método: *ImprimirItemEnCotizacion()*

010724	✓	Imprime un ítem en cotización.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>PrintQuotationItem</i> .

Mediante el método *ImprimirItemEnCotización()* el software de gestión para puntos de venta podrá imprimir una línea de detalle de una cotización.

Dim detalle As String

.....

```
detalle = "las autoridades de....."
HASAR1.ImprimirItemEnCotizacion detalle
```

El argumento "*detalle*" permite, al software de gestión para puntos de venta, imprimir una de las líneas que conforman el documento que se está emitiendo.

Método: ImprimirItemEnCuenta()

010724	✓	Imprime un ítem en resumen de cuenta.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>PrintAccountItem</i> .

Mediante el método *ImprimirItemEnCuenta()* el software de gestión para puntos de venta podrá imprimir una línea de detalle de un resumen de cuenta, o cargo a la habitación (ambos, documentos no fiscales homologados).

Dim fecha As Date

Dim nro_doc As String, descrip As String

Dim debe As Double, haber As Double

.....

fecha = "06/02/2006"

nro_doc = "0001-00000765"

descrip = "Combustible del mes..."

debe = 20.00

haber = 0.0

.....

HASAR1.ImprimirItemEnCuenta fecha, nro_doc, descrip, debe, haber

El argumento "*fecha*" permite, al software de gestión para puntos de venta, indicar la fecha (debe respetar formato estándar OLE) del comprobante indicado con argumento "*nro_doc*".

El argumento "*nro_doc*" permite, al software de gestión para puntos de venta, asentar en el resumen de cuenta, o cargo a la habitación, un número de comprobante como referencia para el detalle indicado bajo el argumento "*descrip*".

El argumento "*descrip*" permite, al software de gestión para puntos de venta, indicar un texto a modo de detalle para inferir el motivo por el cual el comprobante "*nro_doc*" se incluye en el documento que se está emitiendo.

El argumento "*debe*" permite, al software de gestión para puntos de venta, indicar el monto del comprobante que se está detallando como argumento "*nro_doc*".

El argumento "*haber*" permite, al software de gestión para puntos de venta, indicar un monto correspondiente a un adelanto, seña, o pago a cuenta, por parte del cliente.

Método: *ImprimirItemEnRemito()*

010724	✓	Imprime un ítem en remito.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>PrintEmbarkItem</i> .

Mediante el método *ImprimirItemEnRemito()* el software de gestión para puntos de venta podrá imprimir una línea de detalle de un remito (documento no fiscal homologado).

```

Dim descrip As String
Dim cant As Double
.....
descrip = "Bultos x 48 Producto A..."
cant = 5.0
.....
HASAR1.ImprimirItemEnRemito descrip, cant

```

El argumento "*descrip*" permite, al software de gestión para puntos de venta, indicar un texto a modo de detalle del ítem que se está remitiendo.

El argumento "*cant*" permite, al software de gestión para puntos de venta, indicar la cantidad que se remite de dicho ítem

Método: *ImprimirPago()*

010724	✓	Imprime un medio de pago.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>TotalTender</i> .

Mediante el método *ImprimirPago()* el software de gestión para puntos de venta podrá imprimir en el comprobante fiscal el detalle de un pago realizado por el cliente.

El desarrollador deberá consultar en el manual de comandos de la impresora fiscal Hasar en uso respecto de la máxima cantidad de pagos que se pueden detallar.

La normativa fiscal vigente exige que en las facturas se detalle como mínimo un medio de pago. Si el software de gestión para puntos de venta cierra el comprobante fiscal (invocando al método CerrarComprobanteFiscal()) sin haber indicado pago alguno, la impresora fiscal Hasar imprimirá automáticamente un pago cuyo texto por defecto es "Cuenta Corriente" -texto que puede ser modificado por el software de gestión para puntos de venta-.

De manera análoga, si se cierra el comprobante fiscal sin que el monto de lo pagado alcance a saldar el total acumulado en el comprobante -según la impresora fiscal Hasar- se imprimirá automáticamente "Cuenta Corriente" y el monto será el que falte saldar.

```
(010724)  Dim descrip As String
          Dim monto As Double
          .....
          descrip = "Efectivo Pesos"
          monto = 100.0
          .....

          HASAR1.ImprimirPago descrip, monto
          .....
          MsgBox "(+)Falta pagar/(-)Vuelto: " & HASAR1.Respuesta(3)
```

```
(051122)  Dim descrip As String
          Dim monto As Double
          Dim resto
          .....
          descrip = "Efectivo Pesos"
          monto = 100.0
          .....
          HASAR1.ImprimirPago descrip, monto, resto
          .....
          MsgBox "(+)Falta pagar/(-)Vuelto: " & resto
```

El argumento "*descrip*" permite, al software de gestión para puntos de venta, indicar un texto descriptivo para el medio de pago que se va a imprimir.

El argumento "*monto*" permite, al software de gestión para puntos de venta, indicar el monto de pago que se detalla como el argumento "*descrip*".

El argumento "*resto*" existe solamente en la versión 051122 del OCX Fiscal Hasar. Notar que se trata de un argumento de salida que le permite al software de gestión para puntos de venta obtener información relativa al pago: cuanto falta pagar, o si se trata del vuelto.

Método: ImprimirTextoFiscal()

010724	✓	Imprime un texto fiscal.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>PrintFiscalText</i> .

Mediante el método *ImprimirTextoFiscal()* el software de gestión para puntos de venta puede, en forma previa a la impresión de un ítem, imprimir texto libre como una extensión de la descripción del ítem posterior, o como leyendas adicionales relacionados con dicho ítem.

Se puede imprimir más de una línea de texto fiscal asociado a un ítem. Se deberá consultar en el manual de comandos de la impresora fiscal Hasar la cantidad de líneas posibles, ya que varía de un modelo a otro e incluso entre documentos fiscales.

La impresión de texto fiscal (se admite doble ancho) obliga al software de gestión para puntos de venta a realizar una venta posterior.

```
(010724) Dim texto1 As String, texto2 As String
.....
texto1 = "En Oferta hasta: 28/02/2006"
texto2 = Chr(244) & "Desc. 10% sobre precio de lista"
.....
HASAR1.ImprimirTextoFiscal texto1
HASAR1.ImprimirTextoFiscal texto2
HASAR1.ImprimirItem "Producto -A-", 1, 100, 21, 0
```

```
(051122) Dim texto1 As String, texto2 As String
.....
texto1 = "En Oferta hasta: 28/02/2006"
texto2 = HASAR1.DobleAncho & "Desc. 10% sobre precio de lista"
.....
HASAR1.ImprimirTextoFiscal texto1
HASAR1.ImprimirTextoFiscal texto2
HASAR1.ImprimirItem "Producto -A-", 1, 100, 21, 0
```

Si el texto a imprimir excede la capacidad de la línea de texto fiscal (sin incluir blancos intermedios), el OCX Fiscal Hasar lo trunca contando caracteres desde el final, y reemplaza los tres primeros caracteres del string truncado por puntos.

Supongamos una línea de texto fiscal que admite 28 caracteres, y que el texto a imprimir es el siguiente:

"Meestoypasandoenlalogitudadmitidaparaestestringsinblancos"

Entonces, se imprime: **"...tidaparaestestringsinblancos"**

El desarrollador del software de gestión para puntos de venta puede optar por manejar directamente la impresión del texto fiscal, mediante el método *ImprimirtextoFiscal()*, o dejar que lo maneje el OCX Fiscal Hasar. En el segundo caso, es necesario asignar a la propiedad *DescripcionesLargas()* el valor "verdadero", e indicar la descripción del ítem con textos que excedan su capacidad.

Por ejemplo, en el caso de impresoras fiscales emisoras de tiques y/o tiques factura:

```
HASAR1.DescripcionesLargas = True
HASAR1.ImprimirItem "Esta descripción es más larga de lo permitido", 1, 100, 21, 0
```

Método: ImprimirTextoNoFiscal()

010724	✓	Imprime un texto no fiscal.
051122	✓	Este método del OCX Fiscal Hasar genera y envía el string del comando <i>PrintNonFiscalText</i> .

Mediante el método *ImprimirtextoNoFiscal()* el software de gestión para puntos de venta puede generar un documento no fiscal con texto libre. Cada línea de ese documento debe imprimirse con este método.

La normativa fiscal vigente obliga a la impresora fiscal a imprimir la leyenda " No Fiscal ***" cada cuatro líneas de impresión.***

El uso destinado al documento no fiscal es libre en formato tique. Sin embargo, en formato de hoja suelta o formulario continuo, el uso del documento está restringido a los indicados por la RG AFIP 1198/02.

La impresión del texto no fiscal puede hacerse en doble ancho.

```
(010724) Dim texto As String
.....
texto = Chr(244) & "Línea de texto libre..."
.....
HASAR1.ImprimirtextoNoFiscal texto
```

```
(051122) Dim texto As String
```

```

.....
texto = HASAR1.DobleAncho & "Línea de texto libre..."
.....
HASAR1.ImprimirtextoNoFiscal texto

```

El argumento "texto" le permite, al software de gestión para puntos de venta, imprimir en un documento no fiscal el texto indicado.

Método: ImprimirVoucher()

010724	✓	Imprime un voucher de tarjeta de crédito o débito.
051122	✓	Este método del OCX Fiscal Hasar genera y envía tres strings de comandos: <i>SetVoucherData1</i> , <i>SetVoucherData2</i> y <i>PrintVoucher</i> .

Mediante el método *ImprimirVoucher()* el software de gestión para puntos de venta podrá imprimir vouchers por operaciones que involucran transacciones con tarjetas de crédito/débito (compras, devoluciones, cancelaciones, etc.)

```

(010724) Dim cliente As String, tarjeta As String, nro_tarj As String
Dim comercio As String, monto As String
Dim vencim As Date
Dim cuotas As Long, terminal As Long, lote As Long, cupon As Long
Dim autoriz As Long, copias As Long, comprob As Long
Dim voucher As TiposDeVouchers
Dim clase_tarj As TiposDeTarjetas
Dim ingreso As TiposDeIngresoDeTarjeta

Dim oper As TiposOperacionDeTarjeta
.....
cliente = "Nombre del Cliente..."
tarjeta = "PLASTI CARD"
voucher = VOUCHER_DE_COMPRA
nro_tarj = "1234567890123456"
vencim = "31/12/2007"
clase_tarj = TARJETA_CREDITO
cuotas = 1
comercio = "0908"
terminal = 321
lote = 100
cupon = 1234
ingreso = INGRESO_DE_TARJETA_AUTOMATIZADO
oper = OPERACION_TARJETA_ONLINE
autoriz = 4567
monto = "187.65"

```

comprob = 78956**copias = 2**

.....

**HASAR1.ImprimirVoucher cliente, tarjeta, voucher, nro_tarj, vencim, _
 clase_tarj, cuotas, comercio, terminal, lote, _
 cupon, ingreso, oper, autoriz, monto, comprob, _
 copias**

(051122) **Dim cliente As String, tarjeta As String, nro_tarj As String
 Dim comercio As String, monto As String
 Dim vencim As Date
 Dim cuotas As Long, terminal As Long, lote As Long, cupon As Long
 Dim autoriz As Long, copias As Long
 Dim voucher As TiposDeVouchers
 Dim clase_tarj As TiposDeTarjetas
 Dim ingreso As TiposDeIngresoDeTarjeta
 Dim oper As TiposOperacionDeTarjeta
 Dim comprob, vendedor, estacion

 cliente = "Nombre del Cliente..."
 tarjeta = "PLASTI CARD"
 voucher = VOUCHER_DE_COMPRA
 nro_tarj = "1234567890123456"
 vencim = "31/12/2007"
 clase_tarj = TARJETA_CREDITO
 cuotas = 1
 comercio = "0908"
 terminal = 321
 lote = 100
 cupon = 1234
 ingreso = INGRESO_DE_TARJETA_AUTOMATIZADO**

**oper = OPERACION_TARJETA_ONLINE
 autoriz = 4567
 monto = "187.65"
 comprob = 78956
 copias = 2
 vendedor = "015: JRG"
 estacion = ESTACION_TICKET

 HASAR1.ImprimirVoucher cliente, tarjeta, voucher, nro_tarj, vencim, _
 clase_tarj, cuotas, comercio, terminal, lote, _
 cupon, ingreso, oper, autoriz, monto, comprob, _
 copias, vendedor, estacion**

El argumento "*cliente*" permite, al software de gestión para puntos de venta, indicar el nombre del titular de la tarjeta de crédito/débito involucrada en la transacción.

El argumento "*tarjeta*" permite, al software de gestión para puntos de venta, indicar el nombre que identifica comercialmente a la tarjeta de crédito/débito.

El argumento "*voucher*" permite, al software de gestión para puntos de venta, indicar la operación de la transacción por la cual se emite el voucher de tarjeta.

Constante	dec	hex
VOUCHER_DE_CANCELACION_DEVOLUCION	65	41
VOUCHER_DE_COMPRA	67	43
VOUCHER_DE_DEVOLUCION	68	44
VOUCHER_DE_CANCELACION_COMPRA	86	56

El tipo de dato *TiposDeVouchers* puede ser alguna de las constantes embebidas en el OCX Fiscal Hasar que se listan en la tabla de la izquierda.

El argumento "*nro_tarj*" permite, al software de gestión para puntos de venta, indicar los 16 dígitos del número de la tarjeta de crédito/débito involucrada en la transacción.

El argumento "*vencim*" permite, al software de gestión para puntos de venta, indicar la fecha de vencimiento de la tarjeta de crédito/débito involucrada en la transacción.

El argumento "*clase_tarj*" permite, al software de gestión para puntos de venta, indicar si la tarjeta presentada para la transacción es de crédito o de débito.

Constante	dec	hex
TARJETA_CREDITO	67	43
TARJETA_DEBITO	68	44

El tipo de dato *TiposDeTarjetas* puede ser alguna de las constantes embebidas en el OCX Fiscal Hasar que se listan en la tabla de la izquierda.

El argumento "*cuotas*" permite, al software de gestión para puntos de venta, indicar en cuantas cuotas desea el cliente realizar su pago.

El argumento "*comercio*" permite, al software de gestión para puntos de venta, indicar el número de comercio que lo identifica ante la entidad emisora de la tarjeta, o ante quien autorice la transacción.

El argumento "*terminal*" permite, al software de gestión para puntos de venta, indicar el número de terminal donde se realizó la transacción. Dicho número de terminal es una identificación ante la entidad emisora de la tarjeta, o ante quien autorice la transacción.

El argumento "*lote*" permite, al software de gestión para puntos de venta, indicar el número de lote al cual pertenece la transacción realizada.

El argumento "*cupon*" permite, al software de gestión para puntos de venta, indicar el número que identifica al voucher emitido por la transacción realizada.

El argumento "*ingreso*" permite, al software de gestión para puntos de venta, indicar si el ingreso del número de la tarjeta de crédito/débito se realizó por digitación en te

clado, o por lectura de la correspondiente banda magnética.

Constante	dec	hex
INGRESO_DE_TARJETA_AUTOMATIZADO	32	20
INGRESO_DE_TARJETA_MANUAL	42	2A

Fiscal Hasar que se listan en la tabla de la izquierda.

El tipo de dato *TiposDeIngresoDeTarjeta* puede ser alguna de las constantes embebidas en el OCX

(010724)

El tipo de operación deberá comprender alguno de los siguientes valores: OPERACION_TARJETA_ONLINE o OPERACION_TARJETA_OFFLINE.

(051122)

ImprimirVoucher()

En el OCX original el parámetro 'NumeroComprobanteAsociado' era solo numérico y en el nuevo puede ser string.

En el nuevo OCX se agregan 2 nuevos parámetros opcionales al final, 'Vendedor' de tipo string y 'Estación de Impresión' de tipo ENUM TiposDeEstacion.

BSTR NumeroDeTarjeta,

DATE FechaDeVencimiento,
 enum TiposDeTarjetas TipoTarjeta,
 long Cuotas,
 BSTR CodigoDeComercio,
 long NumeroDeTerminal,
 long NumeroDeLote,
 long Numero,
 enum TiposDeIngresoDeTarjeta TipoIngreso,
 enum TiposOperacionDeTarjeta TipoOperacion,
 long NumeroAutorizacion,
 BSTR Monto,
 VARIANT NumeroComprobanteAsociado,
 long Copias,
 VARIANT vVendedor,
 VARIANT vEstacion)

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.

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.

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.

TratarDeCancelarTodo()

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

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.

EspecificarPercepcionPorIVA (Nombre, Monto, IVA)

Imprime una percepción relacionada con un IVA en particular.

ReimprimirComprobante()

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

InformacionRemito [Propiedad tipo Cadena]

Devuelve o actualiza la información de remitos para facturas o notas de crédito.
Métodos de Reportes/Obtención de Datos:

ReporteX(), ReporteZ()

Efectúa un reporte X ó Z en el impresor fiscal.

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.

ObtenerNombreDeFantasia (PrimeraLinea, SegundaLinea)

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

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.

ObtenerConfiguracionCompleta ()

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

Propiedades del OCX Fiscal Hasar

Propiedad del OCX Fiscal	Versión OCX Fiscal Hasar	
	010724	051122
Baudios	SI	SI
CajonAbierto	SI	SI
ContadorImpresoraOcupada	SI	SI
DescripcionDocumentoEnCurso	SI	SI
DescripcionesLargas	SI	SI
DescripcionEstadoControlador	SI	SI
DireccionIP	---	SI
DobleAncho	---	SI
DocumentoDeReferencia	---	SI
Encabezado	SI	SI
EstadoControlador	SI	SI
EventosIndividuales	SI	SI
FechaHoraFiscal	SI	SI
HuboErrorFiscal	SI	SI
HuboErrorMecanico	SI	SI
HuboFaltaPapel	SI	SI
HuboStatPrn	SI	SI
ImpuestoInternoFijo	SI	SI
ImpuestoInternoPorMonto	SI	SI
IndicadorFiscal	SI	SI
IndicadorImpresora	SI	SI
InformacionRemito	SI	---
KIVA	SI	SI
Modelo	SI	SI
ModoStatPrn	SI	SI
NombreDeFantasia	---	SI
PaginasDeUltimoDocumento	---	SI
PrecioBase	SI	SI
PrimerNumeroDeDocumentoActual	---	SI
Puerto	SI	SI
ReintentoConstante	SI	SI
Reintentos	---	SI
Respuesta	SI	SI
SoportaStatPrn	---	SI
TiempoDeEspera	---	SI
Transporte	---	SI
UltimaFactura	SI	---

Propiedad del OCX Fiscal	Versión OCX Fiscal Hasar	
	010724	051122
UltimaNotaCreditoA	---	SI
UltimaNotaCreditoBC	---	SI
UltimoDocumentoFiscalA	---	SI
UltimoDocumentoFiscalBC	---	SI
UltimoDocumentoFueCancelado	---	SI
UltimoRemito	---	SI
UltimoTicket	SI	---
UsarASCII	SI	SI
UsarDisplay	SI	SI
VerificacionCompletaDeErrores	---	SI
Version	---	SI

--- Esta indicación en la tabla anterior significa que si se intenta ejecutar una sentencia de programa donde se lee o modifica la propiedad del OCX Fiscal Hasar se producirá una excepción atrapable por el software de gestión del punto de venta (por ejemplo, en Visual Basic por vía de la sentencia "*On Error...*").

Código de excepción del entorno: 438 (dec)
1B6 (hex)

Mensaje: "*El objeto no admite esta propiedad o método*"

Por ejemplo: HASAR1.DireccionIP = "192.0.0.1" ; usando *fiscal010724.ocx*
Por ejemplo: nro = HASAR1.UltimoTicket ; usando *fiscal051122.ocx*

Eventos que genera el OCX Fiscal Hasar

Eventos que genera el OCX Fiscal Hasar	Versión OCX Fiscal Hasar	
	010724	051122
ErrorFiscal	SI	SI
ErrorImpresora	SI	SI
EventoCajon	SI	SI
EventoFiscal	SI	SI
EventoImpresora	SI	SI
FaltaPapel	SI	SI
ImpresoraNoResponde	SI	SI
ImpresoraOcupada	SI	SI
ImpresoraOK	SI	SI
ProgresoDeteccion	SI	SI

Algunos de los eventos que aparecen en la tabla anterior pueden ser generados por el OCX Fiscal Hasar por más de un motivo (como es, por dar un ejemplo, el caso del evento *ErrorFiscal*). Para este tipo de eventos existe un conjunto de constantes embebidas asociadas al evento que permiten distinguir la causa que dió origen al mismo. Las constantes mencionadas se detallan en la descripción de cada evento, en los casos que corresponda.

Excepciones que genera el OCX Fiscal Hasar

En este apartado se detalla la lista de constantes embebidas en el OCX Fiscal Hasar asociadas a la generación de excepciones. El desarrollador de software de gestión para puntos de venta deberá atrapar (y tratar convenientemente) estas excepciones; como hacerlo puede consultarlo en la documentación correspondiente a su entorno de desarrollo.

Por ejemplo, en Visual Basic pueden atraparse excepciones mediante la sentencia "On Error...".

```
Public Sub Ejemplo( )
    On Error GoTo Tratar_On_Error
    .....
    .....
Exit Sub

Tratar_On_Error:
    Debug.Print "On ERR: " & Err.Number & " = " & Err.Description

End Sub
```

Descripción de las excepciones

- ✓ Esta marca indica que la excepción puede ser generada por la versión del OCX Fiscal Hasar especificada.
- ⊙ Esta marca indica que la excepción no es generada por la versión del OCX Fiscal Hasar especificada.

H ERR ABORT

(dec = -2147220982) (hex = 8004020A)

010724	✓	El proceso en curso fue interrumpido por el software de gestión del punto de venta. Esta excepción se generará por el OCX Fiscal Hasar cuando, estando la propiedad 'ReintentoConstante' seteada con valor 'Verdadero', se produzca una situación de 'time out' en la comunicación con la impresora fiscal Hasar, y el software de gestión del punto de venta decida interrumpirla utilizando el método 'Abortar'.
051122	✓	

Mensajes asociados:

- (010724) *Aborción provocada por el usuario.*
 (051122) *Operación abortada.*

H ERR ALREADYOPEN

(dec = -2147220987) (hex = 80040205)

010724	✓
051122	✓

El puerto serie (RS-232) ya estaba abierto y el software de gestión para puntos de venta intentó abrirlo nuevamente. Una posibilidad es que el propio software ya lo tenga abierto.

Mensajes asociados:

- (010724) *El puerto seleccionado ya estaba abierto.*
 (051122) *El dispositivo ya estaba abierto.*

H ERR ATOMIC

(dec = -2147220989) (hex = 80040203)

010724	✓
051122	✓

El software de gestión para puntos de venta intentó enviar un comando a la impresora fiscal Hasar cuando el OCX Fiscal Hasar todavía no ha recibido la respuesta del comando inmediatamente anterior.

Mensajes asociados:

- (010724) *Intento de operar cuando se estaba procesando un comando anterior.*
 (051122) *Error de reentrancia en la ejecución.*

H ERR BIND SOCKET

(dec = -2147220967) (hex = 80040219)

010724	⊘
051122	✓

Mensaje asociado:

- (051122) *Error de bind().*

H ERR CF NOT DETECTED

(dec = -2147220931) (hec = 8004023D)

010724	⊗
051122	✓

Mensaje asociado:

(051122) *Controlador no detectado.*

H ERR CONNECTION

(dec = -2147220966) (hex = 8004021A)

010724	⊗
051122	✓

Mensaje asociado:

(051122) *Error de conexión.*

H ERR DEPRECATED FUNCTION

(dec = -2147220926) (hex = 80040242)

010724	⊗
051122	✓

Esta excepción se produce cuando el software de gestión para puntos de venta intenta invocar un método no disponible en la versión del OCX Fiscal Hasar en uso. Por ejemplo, con la versión 051122 cuando se intenta ejecutar *Recomenzar()*.

Mensaje asociado:

(051122) *Función Discontinuada.*

H ERR FIELD INVALID

(dec = -2147220933) (hex = 8004023B)

010724	✓
051122	✓

Un argumento tiene un valor no esperado.

Mensajes asociados:

(010724) *Parámetro no valido.*

(051122) *Parámetro no valido.*

H ERR FIELD NOT FOUND

(dec = -2147220942) (hex = 80040232)

010724	✓
051122	✓

Esta excepción se produce cuando el software de gestión para puntos de venta invoca la propiedad *Respuesta()* solicitando un campo que no existe en el sting recibido desde la impresora fiscal Hasar, como respuesta al último comando enviado. Considerar que los strings de respuesta varían según el modelo del impresora fiscal Hasar, y si el comando fue aceptado o rechazado.

Mensajes asociados:

(010724) *Campo de respuesta no encontrado.*

(051122) *Campo de respuesta no encontrado.*

H ERR FIELD OPTIONAL

(dec = **-2147220934**) (hex = **8004023A**)

010724	✓
051122	✓

Un argumento declarado como opcional en un método es requerido.

Mensajes asociados:

(010724) *Un parámetro opcional no es válido o está ausente cuando es necesario.*

(051122) *Un parámetro opcional no es válido o está ausente cuando es necesario.*

H ERR FILENAME

(dec = **-2147220935**) (hex = **80040239**)

010724	✓
051122	✓

Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *CargarLogo()* utilizando como argumento un nombre de archivo inexistente, o el archivo indicado no puede ser abierto.

Mensajes asociados:

(010724) *El nombre de archivo no es válido.*

(051122) *El nombre de archivo no es válido.*

H ERR GENERIC

(dec = **-2147220991**) (hex = **80040201**)

010724	✓
051122	⊗

Esta excepción se produce cuando hay una desinteligencia entre el OCX Fiscal Hasar y la DLL Fiscal Hasar (*winfis32.dll*), o cuando el error no adjudicable a algún motivo específico.

Mensaje asociado:

(010724) *Error general en DLL fiscal.*

H ERR HANDLER

(dec = **-2147220990**) (hex = **80040202**)

010724	✓
051122	⊗

Mensaje asociado:

(010724) *Handler inválido.*

H ERR INIT SOCKET

(dec = **-2147220968**) (hex = **80040218**)

010724	⊗
051122	✓

Mensajes asociados:

(021122) *Error de inicialización de socket.*

O el que el sistema operativo (*Windows*) retorne en las siguientes circunstancias:
??

H ERR INVALID BIT

(dec = **-2147220940**) (hex = **80040234**)

010724	✓
051122	✓

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.

Mensajes asociados:

(010724) *Identificador de bit no válido.*

(051122) *Identificador de bit no válido.*

H ERR INVALID BUFFER

(dec = **-2147220941**) (hex = **80040233**)

010724	✓
051122	⊗

Mensaje asociado:

(010724) *Buffer inválido o insuficiente.*

H ERR INVALID PORT

(dec = -2147220936) (hex = 80040238)

010724	✓
051122	⊗

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().

Mensaje asociado:

(010724) *El puerto especificado es inválido.*

H ERR INVALIDPTR

(dec = -2147220984) (hex = 80040208)

010724	✓
051122	⊗

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

Mensaje asociado:

(010724) *Error de excepción: dirección inválida o lugar insuficiente.*

H ERR INVALID DATE

(dec = -2147220930) (hex = 8004023E)

010724	⊗
051122	✓

Mensaje asociado:

(051122) *Fecha no válida.*

H ERR INVALID PARAMETER

(dec = -2147220925) (hex = 80040243)

010724	⊖
051122	✓

Esta excepción se produce cuando el software de gestión invoca con parámetros incorrectos a un método o propiedad del OCX Fiscal Hasar.

Mensajes asociados:

- (051122) *El parámetro de puerto debe ser numérico.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *AutoDetectarControlador()* con un parámetro no numérico.
- (051122) *El identificador de sub-capacidad es inválido.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *Capacidades()* con una constante de subcapacidad inválida (por ejemplo, el software intentaba determinar si era posible emitir tickets factura "A" con la impresora fiscal Hasar conectada).
- (051122) *El identificador de capacidad es inválido.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *Capacidades()* con una constante de capacidad inválida (por ejemplo, el software intentaba determinar si era posible manejar un cajón de dinero con la impresora fiscal Hasar conectada).
- (051122) *El parámetro de número de serie debe ser un string.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *BajaFiscal()* indicando incorrectamente el número de serie de la impresora fiscal Hasar a dar de baja.

La baja de una impresora fiscal solamente puede ser realizada por el Servicio Técnico Autorizado, la Autoridad Fiscal competente o, ha pedido y en presencia de ésta última, por el software de gestión del punto de venta.

- (051122) *El parámetro de dirección debe ser un string.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *DatosCliente()* pasando incorrectamente el tipo de dato que como argumento correspondiente al domicilio del comprador.
- (051122) *El límite de consumidor final debe ser un número de punto flotante.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *ConfigurarControladorCompleto()* pasando incorrectamente el tipo de dato que como argumento correspondiente al monto límite para comprobantes "B" a consumidores finales.

Este monto límite es un valor establecido por la normativa fiscal vigente. No debe modificarse salvo que legalmente (nueva resolución general del organismo oficial competente) se hubiera modificado.

- (051122) *El límite de ticket factura debe ser un número de punto flotante.*
-

Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *ConfigurarControladorCompleto()* pasando incorrectamente el tipo de dato que como argumento correspondiente al monto límite para comprobantes "A" en formato tique.

Este monto límite es un valor establecido por la normativa fiscal vigente. No debe modificarse salvo que legalmente (nueva resolución general del organismo oficial competente) se hubiera modificado.

- (051122) *El porcentaje de IVA no inscripto debe ser un número de punto flotante.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *ConfigurarControladorCompleto()* pasando incorrectamente el tipo de dato que como argumento correspondiente a la sobretasa de IVA no inscripto.

Actualmente la normativa fiscal vigente indica no aplicar esta sobretasa, dado que desaparecido la categoría Responsable IVA No Inscripto.

- (051122) *El número de copias máximo es inválido.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *ConfigurarControladorCompleto()* pasando incorrectamente la cantidad máxima de ejemplares que se podrán emitir de un comprobante, con la impresora fiscal Hasar.

- (051122) *El número de copias máximo debe estar comprendido entre 0 y 4.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *ConfigurarControladorCompleto()* pasando incorrectamente la cantidad máxima de ejemplares que se podrán emitir de un comprobante, con la impresora fiscal Hasar.

- (051122) *El parámetro *ImprimeCambio* debe ser un número booleano.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *ConfigurarControladorCompleto()* pasando incorrectamente el argumento que configura a la impresora fiscal Hasar para imprimir la leyenda "CAMBIO \$0.00".

- (051122) *El parámetro *ImprimeLeyendasOpcionales* debe ser un número booleano.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *ConfigurarControladorCompleto()* pasando incorrectamente el argumento que configura a la impresora fiscal Hasar para no imprimir las leyendas opcionales (si datos fiscales están preimpresos al dorso de los rollos de papel).

- (051122) *El parámetro *TipoDeCorte* es incorrecto.*
-

Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *ConfigurarControladorCompleto()* pasando incorrectamente el argumento que configura a la impresora fiscal Hasar respecto de la manera en que debe realizar los cortes de papel.

- (051122) *El parámetro ImprimeMarco debe ser un número booleano.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *ConfigurarControladorCompleto()* pasando incorrectamente el argumento que configura a la impresora fiscal Hasar para imprimir el marco de los documentos que emite.
- (051122) *El parámetro ReImprimeDocumentos debe ser un número booleano.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *ConfigurarControladorCompleto()* pasando incorrectamente el argumento que configura a la impresora fiscal Hasar para la impresión de ejemplares adicionales.
- (051122) *La descripción del medio de pago debe ser un string.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *ConfigurarControladorCompleto()* pasando incorrectamente el argumento que corresponde al texto para saldo, que se imprime cuando el software de gestión para puntos de venta no completa el monto de pago de uan factura/tique factura.
- (051122) *El parámetro Sonido debe ser un número booleano.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *ConfigurarControladorCompleto()* pasando incorrectamente el argumento que configura a la impresora fiscal Hasar para la emisión de pitidos por falta de papel.
- (051122) *El parámetro AltoHoja es incorrecto.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *ConfigurarControladorCompleto()* pasando incorrectamente el argumento que configura a la impresora fiscal Hasar respecto al tamaño de papel a considerar.
- (051122) *El parámetro AnchoHoja es incorrecto.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *ConfigurarControladorCompleto()* pasando incorrectamente el argumento que configura a la impresora fiscal Hasar respecto al tamaño de papel a considerar.
- (051122) *El parámetro EstacionImpresionReportesXZ es incorrecto.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *ConfigurarControladorCompleto()* pasando incorrectamente el argumento que configura a la impresora fiscal Hasar res-
-

pecto de la estación de impresión a utilizar por defecto cuando se emitan reportes "X" o cierres diarios "Z".

- (051122) *El parámetro ModoImpresion es incorrecto.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *ConfigurarControladorCompleto()* pasando incorrectamente el argumento que configura a la impresora fiscal Hasar respecto del modo de trabajo (sólo usar estación slip, o modo mixto).
- (051122) *El parámetro de copias debe ser numérico.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *CerrarComprobante....()* pasando incorrectamente el argumento que indica la cantidad de ejemplares a emitir.
- (051122) *El parámetro Estación es incorrecto.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *AbrirComprobanteNoFiscal()* pasando incorrectamente el argumento que indica en que estación de impresión se emitirá el comprobante.
- (051122) *El parámetro de identificación debe ser un string.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *AbrirComprobanteNoFiscalHomologado()* pasando incorrectamente el argumento que indica el número que se asignará al comprobante a emitir.
- (051122) *El parámetro de vendedor debe ser un string válido.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *ImprimirVoucher()* pasando incorrectamente el argumento que identifica al vendedor a consignar en el voucher de tarjetas a emitir.
- (051122) *El parámetro de ComprobanteAsociado debe ser o un string válido o un número.*
Esta excepción se produce cuando el software de gestión para puntos de venta invoca al método *ImprimirVoucher()* pasando incorrectamente el argumento que identifica al número de comprobante fiscal a consignar en el voucher de tarjetas a emitir.

H ERR INVALID STATE

(dec = **-2147220927**) (hex = **80040241**)

010724	⊗
051122	✓

Mensaje asociado:

(051122) *Estado no válido para el requerimiento solicitado.*

H ERR INVALID TIME

(dec = -2147220929) (hex = 8004023F)

010724	⊗
051122	✓

Mensaje asociado:

(051122) *Hora no válida.*

H ERR IP FORMAT

(dec = -2147220963) (hex = 8004021D)

010724	⊗
051122	✓

Mensaje asociado:

(051122) *Error de formato en dirección de IP.*

H ERR MANDATORY PARAMETER

(dec = -2147220928) (hex = 80040240)

010724	⊗
051122	✓

Mensaje asociado:

(051122) *Un parámetro obligatorio no ha sido suministrado.*

H ERR MAX RETRIES

(dec = -2147220962) (hex = 8004021E)

010724	⊗
051122	✓

Mensaje asociado:

(051122) *Se han agotado la cantidad máxima de reintentos.*

H ERR NO MORE ITEMS

(dec = -2147220924) (hex = 80040244)

010724	⊗
051122	✓

Mensaje asociado:

(051122) ??

H ERR NOMEM

(dec = -2147220986) (hex = 80040206)

010724	✓
051122	⊗

No hay suficiente memoria para completar la operación.

Mensaje asociado:(010724) *Falta de memoria.***H ERR NOT EXECUTED**

(dec = -2147220932) (hex = 8004023C)

010724	⊗
051122	✓

Mensaje asociado:(051122) *Función no ejecutada con éxito.***H ERR NOT OPEN**

(dec = -2147220937) (hex = 80040237)

010724	✓
051122	⊗

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.

Mensaje asociado:(010724) *No ha sido abierto el puerto.***H ERR NOT SUPPORTED**

(dec = -2147220938) (hex = 80040236)

010724	✓
051122	✓

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.

Mensajes asociados:

(010724) *Función no soportada en versión de controlador fiscal seleccionada.*

(051122) *Función no soportada en versión de controlador fiscal seleccionada.*

(051122) *Autodetección sólo soportado para puerto serie.*

Esta excepción se produce, por ejemplo, cuando el software de gestión para puntos de venta ha seleccionado como transporte SOCKET_TCP, o SOCKET_UDP, e intenta ejecutar el método *AutoDetectarControlador()*.

(051122) *El modo StatPRN es dependiente del modelo.*

Esta excepción se produce, por ejemplo, cuando el software de gestión para puntos de venta intenta setear la propiedad *ModoStatPrn()* como verdadera para manejar directamente las consultas de estado intermedio, cuando el modelo de impresora fiscal conectada no soporta dichas consultas.

H ERR NOTOPENYET

(dec = -2147220985) (hex = 80040207)

010724	✓
051122	✓

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.

Mensajes asociados:

(010724) *Intento de usar un puerto que no estaba abierto.*

(051122) *No se ha inicializado un transporte.*

H ERR OPEN

(dec = -2147220971) (hex = 80040215)

010724	⊗
051122	✓

Mensaje asociado:

(051122) *Error de apertura.*

H ERR PRINTER NOT FOUND

(dec = -2147220939) (hex = 80040235)

010724	✓
051122	✓

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

Mensaje asociado:(010724) *El impresor solicitado no existe.***H ERR READ**

(dec = -2147220970) (hex = 80040216)

010724	⊗
051122	✓

Mensaje asociado:(051122) *Error de lectura.***H ERR SERIAL PARAMETERS**

(dec = -2147220972) (hex = 80040214)

010724	⊗
051122	✓

Mensaje asociado:(051122) *Error de parámetros de puerto serie.***H ERR SOCKET SELECT**

(dec = -2147220965) (hex = 8004021B)

010724	⊗
051122	✓

Mensaje asociado:(051122) *Error en función select().***H ERR STATPRN**

(dec = -2147220983) (hex = 80040209)

010724	✓
051122	⊗

Mensaje asociado:

(010724) *Error STAT_PRN.*

H ERR TIMEOUT

(dec = **-2147220988**) (hex = **80040204**)

010724	✓
051122	✓

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.

Mensajes asociados:

(010724) *Error de comunicaciones. El controlador no está respondiendo.*

(051122) *Tiempo de espera agotado.*

H ERR VERSION SOCKET

(dec = **-2147220964**) (hex = **8004021C**)

010724	⊗
051122	✓

Mensaje asociado:

(051122) *Error de versión de socket.*

H ERR WRITE

(dec = **-2147220969**) (hex = **80040217**)

010724	⊗
051122	✓

Mensaje asociado:

(051122) *Error de escritura.*

H ERR WRONG PROTOCOL

(dec = **-2147220961**) (hex = **8004021F**)

010724	⊘
051122	✓

Mensaje asociado:

(051122) *El impresor conectado no corresponde al modelo seleccionado.*



GrupoHasar

Febrero 20, 2006 – Rev. 001

Ricardo D. Cárdenes

© **Cía. HASAR SAIC**

Marcos Sastre y José Ingenieros

(B1618CSD) El Talar - Gral. Pacheco - Pcia. de Bs. As. - Argentina

Teléfono: (54)(011) 4727-8900 • Fax: (54)(011) 4736-7300

ventas@hasar.com

www.hasar.com

www.grupohasar.com



CompañíaHasar