

Impresoras Fiscales HASAR

OCX FISCAL HASAR

v150423 y anteriores

(Argentina)

Para Impresoras Fiscales 1ra. Generación (IF 1G)
(vieja tecnología)



OCX FISCAL HASAR

v150423 y anteriores

OCX FISCAL HASAR (ARGENTINA)	7
INTRODUCCIÓN	7
EVOLUCIÓN DEL OCX FISCAL HASAR.....	8
<i>El OCX Fiscal HASAR v010724.....</i>	8
<i>El OCX Fiscal HASAR v051122.....</i>	9
<i>El OCX Fiscal HASAR v110423.....</i>	9
<i>El OCX Fiscal HASAR v150423.....</i>	9
COMO REGISTRAR EL OCX FISCAL HASAR	10
INCORPORANDO EL OCX FISCAL HASAR A UN PROYECTO.....	10
<i>Caso: VC# Visual Studio 2010 Professional.....</i>	10
<i>Caso: Visual Basic 6.0</i>	12
<i>Caso: Visual Fox Pro 6.0.....</i>	14
EL OCX FISCAL HASAR Y WINDOWS®	17
IMPRESORAS FISCALES HASAR (IFH 1G) SOPORTADAS	18
MÉTODOS DEL OCX FISCAL HASAR	20
MÉTODO: ABORTAR()	22
MÉTODO: ABRIRCAJONDEDINERO()	23
MÉTODO: ABRIRCOMPROBANTEFISCAL()	23
MÉTODO: ABRIRCOMPROBANTENoFISCAL()	25
MÉTODO: ABRIRCOMPROBANTENoFISCALENSLIP().....	26
MÉTODO: ABRIRCOMPROBANTENoFISCALHOMOLOGADO().....	27
MÉTODO: ABRIRDNFH().....	27
MÉTODO: APAGARÍNDICADORDETAMBOR()	29
MÉTODO: AUTODETECTARCONTROLADOR()	29
MÉTODO: AUTODETECTARMODELO().....	31
MÉTODO: AVANZARPAPEL()	31
MÉTODO: BAJAFISCAL()	32
MÉTODO: BORRARDOCUMENTODEREFERENCIA().....	33
MÉTODO: BORRARFANTASIAENCABEZADOCOLA()	34
MÉTODO: CAMBIARCÓDIGOINGRESOSBRUTOS()	35
MÉTODO: CAMBIARFECHAÍNICIOACTIVIDADES()	36
MÉTODO: CAMBIARRESPONSABILIDADIVA()	37
MÉTODO: CAMBIARVELOCIDAD()	38
MÉTODO: CANCELARCOMPROBANTE()	39
MÉTODO: CANCELARCOMPROBANTEFISCAL()	40
MÉTODO: CAPACIDADES()	40
<i>Subcapacidades asociadas</i>	41
MÉTODO: CAPACIDADRESTANTE()	42
MÉTODO: CARGARLOGO()	44
MÉTODO: CERRARCOMPROBANTEFISCAL()	45
MÉTODO: CERRARCOMPROBANTENoFISCAL()	46
MÉTODO: CERRARCOMPROBANTENoFISCALHOMOLOGADO()	47
MÉTODO: CERRARDNFH()	47
MÉTODO: COMENZAR()	48
MÉTODO: CORTARCOMPROBANTENoFISCAL()	50
MÉTODO: CONFIGURARCONTROLADOR()	50
<i>Valores asociados a los parámetros de comportamiento</i>	51

MÉTODO: CONFIGURARCONTROLADORCOMPLETO()	54
MÉTODO: CONFIGURARCONTROLADORPORBLOQUE()	59
MÉTODO: DATOSCLIENTE()	61
MÉTODO: DESCRIPCIONSTATUSFISCAL()	64
MÉTODO: DESCRIPCIONSTATUSIMPRESOR()	65
MÉTODO: DESCUENTOGENERAL()	66
MÉTODO: DESCUENTOULTIMOITEM()	67
MÉTODO: DETALLERECIBO()	68
MÉTODO: DEVOLUCIONDESCUENTO()	68
MÉTODO: DEVOLVERPAGO()	70
MÉTODO: DNFHFARMACIAS()	72
MÉTODO: DNFHREPARTO()	72
MÉTODO: ENVIAR()	73
MÉTODO: ENVIARSTATPRN()	74
MÉTODO: ESPECIFICARIVANoINSCRIPTO()	75
MÉTODO: ESPECIFICARNOMBREDEFANTASIA()	76
MÉTODO: ESPECIFICARPERCEPCIONGLOBAL()	77
MÉTODO: ESPECIFICARPERCEPCIONPORIVA()	78
MÉTODO: ESTADOINTERNOIMPRESOR()	78
MÉTODO: FINALIZAR()	79
MÉTODO: IMPRIMIRCÓDIGODeBARRAS()	80
MÉTODO: IMPRIMIRFIRMAOPCIONALENCIERREDNFH()	81
MÉTODO: IMPRIMIRINFOENDNFH()	82
MÉTODO: IMPRIMIRITEM()	83
MÉTODO: IMPRIMIRITEMENCOTIZACION()	85
MÉTODO: IMPRIMIRITEMENCUENTA()	86
MÉTODO: IMPRIMIRITEMEnREMITO()	87
MÉTODO: IMPRIMIRPAGO()	88
MÉTODO: IMPRIMIRTEXTOFISCAL()	89
MÉTODO: IMPRIMIRTEXTONoFISCAL()	90
MÉTODO: IMPRIMIRVOUCHER()	91
MÉTODO: IMPRIMIRVOUCHERCONFECHAOCULTA()	94
MÉTODO: LEERIVA()	97
MÉTODO: LEERMEMORIADETRABAJO()	99
MÉTODO: LEERMONTOTOTAL()	103
MÉTODO: LEERPERCEPCIONESGENERALES()	104
MÉTODO: LEERPERCEPCIONESPORDESCRIPCION()	105
MÉTODO: LEERPERCEPCIONESPORIVA()	106
MÉTODO: MOSTRAREndDISPLAY()	108
MÉTODO: OBTENERCAIULTIMODOCUMENTOA()	109
MÉTODO: OBTENERCONFIGURACION()	110
MÉTODO: OBTENERCONFIGURACIONCOMPLETA()	113
MÉTODO: OBTENERDATOSDeCONFIGURACION()	113
MÉTODO: OBTENERDATOSDeINITIALIZACION()	114
MÉTODO: OBTENERNOMBREDEFANTASIA()	116
MÉTODO: OBTENERVERSIONDELCONTROLADOR()	116
MÉTODO: PEDIDODeSTATUS()	117
MÉTODO: PEDIRPRIMERIVA()	117
MÉTODO: PEDIRSIGUIENTEIVA()	118
MÉTODO: RECIBIR()	119
MÉTODO: RECOMENZAR()	120
MÉTODO: REIMPRIMIRCOMPROBANTE()	120
MÉTODO: REPORTEX()	121
MÉTODO: REPORTEZ()	125
MÉTODO: REPORTEZINDIVIDUALPORFECHA()	129

MÉTODO: REPORTEZINDIVIDUALPORNUMERO()	133
MÉTODO: REPORTEZPORFECHAS()	136
MÉTODO: REPORTEZPORNUMEROS()	137
MÉTODO: SUBTOTAL()	138
MÉTODO: TRATARDECANCELARTODO()	139
MÉTODO: VERIFICARMODELO()	140
PROPIEDADES DEL OCX FISCAL HASAR	141
PROPIEDAD: BAUDIOS()	142
PROPIEDAD: CAJONABIERTO()	143
PROPIEDAD: CONTADORIMPRESORAOCUPADA()	143
PROPIEDAD: DESCRIPCIONESLARGAS()	144
PROPIEDAD: DESCRIPCIONESTADOCONTROLADOR()	145
PROPIEDAD: DESCRIPCIONDOCUMENTOENCURSO()	145
PROPIEDAD: DIRECCIONIP()	146
PROPIEDAD: DOBLEANCHO()	146
PROPIEDAD: DOCUMENTODEREFERENCIA()	147
PROPIEDAD: DOCUMENTOENCURSO()	148
PROPIEDAD: ENCABEZADO()	148
PROPIEDAD: ESTADOCONTROLADOR()	149
PROPIEDAD: EVENTOSINDIVIDUALES()	150
PROPIEDAD: FECHAHORAFISCAL()	150
PROPIEDAD: HUBOERRORFISCAL()	151
PROPIEDAD: HUBOERRORMECANICO()	151
PROPIEDAD: HUBOFALTAPAPEL()	152
PROPIEDAD: HUBOSTATPRN()	152
PROPIEDAD: IMPUESTOINTERNOFIJO()	154
PROPIEDAD: IMPUESTOINTERNOPORMONTO()	155
PROPIEDAD: INFORMACIONREMITO()	156
PROPIEDAD: INDICADORFISCAL()	157
PROPIEDAD: INDICADORIMPRESORA()	158
PROPIEDAD: INTERLINEADO()	158
PROPIEDAD: KIVA()	159
PROPIEDAD: MODELO()	160
PROPIEDAD: MODOSTATPRN()	161
PROPIEDAD: NOMBREARCHIVOLOG()	161
PROPIEDAD: NOMBREDEFANTASIA()	162
PROPIEDAD: PÁGINASDEULTIMODOCUMENTO()	163
PROPIEDAD: PRECIOBASE()	164
PROPIEDAD: PRIMERNUMERODEDOCUMENTOACTUAL()	164
PROPIEDAD: PUERTO()	165
PROPIEDAD: REINTENTOCONSTANTE()	166
PROPIEDAD: REINTENTOS()	166
PROPIEDAD: RESPUESTA()	167
PROPIEDAD: RESUMENIVA()	167
PROPIEDAD: SOPORTASTATPRN()	168
PROPIEDAD: TIEMPODEESPERA()	169
PROPIEDAD: TRANSPORTE()	169
PROPIEDAD: ULTIMANOTACREDITOA()	170
PROPIEDAD: ULTIMANOTACREDITOBC()	170
PROPIEDAD: ULTIMODOCUMENTOFISCALA()	170
PROPIEDAD: ULTIMODOCUMENTOFISCALBC()	171
PROPIEDAD: ULTIMODOCUMENTOFUECANCELADO()	171
PROPIEDAD: ULTIMAFACTURA()	171
PROPIEDAD: ULTIMOTICKET()	172

PROPIEDAD: ULTIMOREMITO().....	172
PROPIEDAD: USARASCII().....	172
PROPIEDAD: USARDISPLAY()	173
PROPIEDAD: VERIFICACIONCOMPLETADEERRORES().....	174
PROPIEDAD: VERSION().....	174
CONSTANTES EMBEBIDAS EN EL OCX FISCAL HASAR	175
ENUM: MODELOSDEIMPRESORAS	175
ENUM: TIPOSDEDOCUMENTO	176
ENUM: TIPOSDERESPONSABILIDADES.....	177
ENUM: TIPOSRESPONSABILIDADIMPRESOR.....	177
ENUM: DOCUMENTOSFISCALES (DF)	178
ENUM: DOCUMENTOSNOFISCALES (DNFH).....	179
ENUM: TIPOSDEDISPLAY	180
ENUM: TIPOSDEPAPEL (ROLLO)	180
ENUM: TIPOSDECODIGOORBARRAS.....	181
ENUM: TIPOSDEVOUCHERS.....	182
ENUM: TIPOSDETARJETAS.....	182
ENUM: TIPOSDEINGRESODETARJETA	183
ENUM: TIPOSDEOPERACIONDETARJETA	183
ENUM: TIPOSDEDESCUENTOS.....	184
ENUM: PARAMETROSDECONFIGURACION	184
ENUM: NUMERODECOPIAS.....	186
ENUM: TIPODECORTEDEPAPEL.....	186
ENUM: DOCUMENTOS.....	187
ENUM: TIPOSDEALTOHOJA	188
ENUM: TIPOSDEANCHOHOJA.....	189
ENUM: TIPOSDEESTACION	189
ENUM: TIPOSDEMODOIMPRESIÓN	190
ENUM: CAPACIDADESSOPORTADAS.....	190
ENUM: TIPOSDEDOCUMENTOSOPORTADO.....	191
ENUM: TIPOSDERESUMENIVA	192
ENUM: DOCUMENTOSIVA	193
ENUM: TIPOSDETRANSPORTE	193
ESTADOS DE LA IMPRESORA FISCAL HASAR (IFH 1G).....	195
ENUM STATENIBBLE.....	195
EVENTOS DEL OCX FISCAL HASAR	198
EVENTOS POSIBLES DE SER GENERADOS	198
COMO ATRAPAR EVENTOS	198
<i>Caso: Visual Basic 6.0</i>	199
<i>Caso: Visual C#</i>	199
LOS EVENTOS DEL OCX FISCAL HASAR.....	200
<i>ErrorFiscal()</i>	200
<i>ErrorImpresora()</i>	200
<i>EventoCajon()</i>	200
<i>EventoFiscal()</i>	201
<i>EventoImpresora()</i>	201
<i>FaltaPapel()</i>	202
<i>ImpresoraNoResponde()</i>	202
<i>ImpresoraOcupada()</i>	202
<i>ImpresoraOK()</i>	203
<i>ProgresoDeteccion()</i>	203
EXCEPCIONES DEL OCX FISCAL HASAR	204

ENUM: ERRORESFISCALES	204
COMO ATRAPAR EXCEPCIONES.....	205
<i>Caso: Visual Basic 6.0</i>	205
<i>Caso: Visual C#</i>	205
MENSAJES ASOCIADOS A LAS EXCEPCIONES	206
APÉNDICE: GENERACIÓN DE COMPROBANTES.....	218
GENERALIDADES	218
(DF) DOCUMENTOS FISCALES	219
(DNFH) TIQUES NOTAS DE CRÉDITO / NOTAS DE CRÉDITO.....	221
(DF) RECIBOS FISCALES	223
(DNFH) RECIBOS 'X'	225
(DNFH) REMITO 'X' / ORDEN DE SALIDA.....	226
(DNFH) RESUMEN DE CUENTA / CARGO A LA HABITACIÓN	227
(DNFH) COTIZACIÓN	228
(DNFH) EN GENERAL	229
COMANDOS DE INICIALIZACIÓN, BAJA Y CONFIGURACIÓN	230
COMANDOS DE DIAGNÓSTICO Y CONSULTA	231
COMANDOS DE CONTROL FISCAL.....	232
COMANDOS DE USO GENERAL	233

OCX Fiscal HASAR (Argentina)

Introducción

Debido a que *Grupo HASAR*, a través de *Compañía HASAR SAIC*, ha homologado equipos fiscales en diferentes países de Latinoamérica, y debido a que las normas fiscales vigentes difieren entre países, se hace necesaria la provisión de herramientas fiscales que respondan a cada normativa fiscal de aplicación. Por ello hay una distribución del **OCX Fiscal HASAR** para cada país. La presente documentación está referida al **OCX Fiscal HASAR** de uso exclusivo en *Argentina*.

El **OCX Fiscal HASAR** permite al software de gestión para puntos de venta dialogar con las **Impresoras Fiscales HASAR 1ra. Generación (IFH 1G, vieja tecnología)** de manera eficiente, y ofrece a los desarrolladores de la aplicación una manera amigable de programación.

Hasta la fecha, en *Argentina*, se han liberado cuatro versiones del **OCX Fiscal HASAR**:

v010724	<i>fiscal010724.ocx</i> Primera versión del OCX Fiscal HASAR . Requiere la librería <i>winfis32.dll</i> (también provista por <i>Compañía HASAR SAIC</i>).
v051122	<i>fiscal051122.ocx</i> A partir de esta versión no se requiere de una librería externa al OCX Fiscal HASAR .
v110423	<i>fiscal110423.ocx</i> Incorpora nuevas IFH 1G homologadas.
V150423	<i>Fiscal150423.ocx</i> Última versión del OCX Fiscal HASAR . Incorpora nuevas IFH 1G homologadas.

La incorporación del **OCX Fiscal HASAR** a cualquier proyecto de desarrollo de software de gestión para puntos de venta está limitada, solamente, por el nivel de compatibilidad de cada lenguaje de programación con:

- La norma *Activex* para desarrollo de objetos.
- Objetos *Activex* desarrollados bajo tecnología COM (*Component Object Model*).
- Los estándares del sistema operativo Windows®.

El **OCX Fiscal HASAR** permite:

- Generar strings de comandos y enviarlos a la **IFH 1G**; a partir de la información suministrada por el software de gestión para puntos de venta.
- Obtener la información contenida en los strings de respuesta de la **IFH 1G**, correspondientes a cada comando enviado.
- Control de errores mediante la generación de excepciones.
Es responsabilidad del software de gestión para puntos de venta atrapar y procesar adecuadamente las excepciones posibles de ser generadas por el **OCX Fiscal HASAR**.

- Control de errores mediante la generación de eventos.
Es responsabilidad del software de gestión para puntos de venta atrapar y procesar adecuadamente los eventos posibles de ser generados por el **OCX Fiscal HASAR**.
- Su inserción en proyectos sobre plataformas 32/64 bits.

El **OCX Fiscal HASAR** ofrece, entre muchas otras, las siguientes ventajas destacables:

- Soporte a todos los modelos y versiones de **IFH 1G** homologadas en *Argentina*, hasta la fecha.
- Identificación de la **IFH 1G** con la que se establezca conexión.
- Multiplicidad de transportes: protocolo serie RS-232, serie USB, TCP/IP, y/o UDP/IP; según el vínculo de conexión con la **IFH 1G**.
- Es de distribución gratuita. Puede descargarse de la web oficial de *Grupo HASAR*: www.grupohasar.com

IMPORTANTE:

Como suplemento a esta documentación, el desarrollador del software de gestión para puntos de venta debe consultar el *Manual de Comandos de la IFH 1G* a soportar con su producto.

Existe, generalmente, una relación uno a uno entre los métodos/propiedades del **OCX Fiscal HASAR** y los strings de comandos soportados por las **IFH 1G**.

Es responsabilidad del software de gestión para puntos de venta garantizar que los datos que se suministren al **OCX Fiscal HASAR** resulten válidos (aceptables por la **IFH 1G**) y que los strings de comandos se generen y envíen en el orden correcto, de acuerdo al comprobante que se desee emitir.

Evolución del OCX Fiscal HASAR

El **OCX Fiscal HASAR** evoluciona en función de:

- La homologación de nuevas **IFH 1G**.
- Cambios en la normativa fiscal vigente.
- Necesidades de los desarrolladores de software de gestión para puntos de venta.
- Valores agregados por *Compañía HASAR SAIC* para ofrecer una herramienta más robusta, eficaz, eficiente, y con más y mejores prestaciones.

El OCX Fiscal HASAR v010724

Es la primera versión oficial del **OCX Fiscal HASAR**. Esta versión funciona como una especie de *wrapper* alrededor de la librería *winfis32.dll* (también provista por *Compañía HASAR SAIC*, gratuitamente). Es condición necesaria para la registración del **OCX Fiscal HASAR** la presencia de la DLL mencionada. En la distribución del software de gestión para puntos de venta que utiliza esta versión del **OCX Fiscal HASAR** debe incluirse, además del OCX, también a la librería *winfis32.dll* (esta librería no debe ser registrada).

Esta versión del **OCX Fiscal HASAR** soporta naturalmente los primeros 8 modelos homologados de **IFH 1G**.

El OCX Fiscal HASAR v051122

A partir de esta versión del **OCX Fiscal HASAR** no se requiere de ninguna DLL externa. Internamente se encapsula una librería estática de clases polimórficas (cada clase representa un modelo de **IFH 1G**, supeditado a una interfaz uniforme, de manera de poder utilizar cada modelo de estos equipos, independientemente de las diferencias que entre ellos existiesen).

Otras modificaciones, respecto de la versión anterior de **OCX Fiscal HASAR**, son las siguientes:

- Se amplía a 35 la cantidad de modelos de **IFH 1G** soportadas naturalmente por esta versión del **OCX Fiscal HASAR**. Es decir, hasta el modelo: SMH/P-1120F v01.00.
- Incorpora un método que permite la auto detección del modelo y versión de la **IFH 1G** con la cual se ha establecido conexión.
- Incorpora un método que permite, en caso de selección manual del modelo y versión de la **IFH 1G**, verificar si hay coincidencia con el que se ha establecido conexión.
- Incorpora nuevos protocolos de comunicación: TCP/IP, y UDP/IP.
- Incorpora la posibilidad de ajustar variables que afectan al protocolo de comunicación para optimizar el uso en condiciones atípicas (tiempo para salida por error de *time out*, cantidad de reintentos, etc.).
- Nuevos métodos/propiedades para emplear a la hora de obtener el desglose impositivo de cada comprobante que se emita.
- Permite consultar las capacidades del modelo de **IFH 1G** conectado, pudiendo el software de gestión para puntos de venta decidir -en tiempo de ejecución-, por ejemplo, qué tipos de documentos son posibles de emitir con dicha **IFH 1G**.
- Se mejora la entrega de información proveniente de la **IFH 1G**, dotando a la mayoría de los métodos del **OCX Fiscal HASAR** de argumentos opcionales de salida. Se crea la tendencia a la eliminación del uso de la propiedad *Respuesta()*, mostrando la intención de deprecar dicha propiedad en alguna versión futura del **OCX Fiscal HASAR**.

El OCX Fiscal HASAR v110423

Las modificaciones, respecto de la versión anterior de **OCX Fiscal HASAR**, son las siguientes:

- Se amplía a 38 la cantidad de modelos de **IFH 1G** soportadas naturalmente por esta versión del **OCX Fiscal HASAR**. Es decir, hasta el modelo: SMH/PT-272F v02.01.
- Se amplia la consulta de capacidades del modelo de **IFH 1G** conectada.

El OCX Fiscal HASAR v150423

Las modificaciones, respecto de la versión anterior de **OCX Fiscal HASAR**, son las siguientes:

- Se amplía a 41 la cantidad de modelos de **IFH 1G** soportadas naturalmente por esta versión del **OCX Fiscal HASAR**. Se incorporan los modelos: SMH/P-441F V02.01, SMH/P-715F v05.04, y SMH/P-340F v01.00.

Como registrar el OCX Fiscal HASAR

El **OCX Fiscal HASAR** no escapa a las generales de la ley. Se registra siguiendo los mismos pasos empleados para registrar cualquier otro objeto de características similares.

- Para poder registrar el **OCX Fiscal HASAR** se debe poseer permiso de administrador.
- Copiar el **OCX Fiscal HASAR** (y la librería *winfis32.dll* -la cual no se registra- en el caso de la versión *v010724* únicamente) a la carpeta *c:\windows\system32*, *c:\windows\syswow64*, o a la que corresponda a la versión de Windows® en uso.
- Ejecutar *cmd.exe* (ejecutable de Windows®) como administrador para abrir una sesión MS-DOS® de Windows®.
- En la sesión MS-DOS® ejecutar *regsvr32.exe* (ejecutable de Windows®), ubicado en *c:\windows\system32*, *c:\windows\syswow64*, o la carpeta que corresponda a la versión de Windows® en uso.

Por ejemplo:

```
[path]\regsvr32 [path]\fiscal\nnnnnn.ocx
```

Donde 'nnnnnn' es la versión del **OCX Fiscal HASAR**.

Una vez registrado el **OCX Fiscal HASAR** es posible incorporarlo al proyecto de desarrollo del software de gestión para puntos de venta.

En la distribución del software de gestión para puntos de venta se podrá emplear cualquier software de instalación, siempre y cuando tal producto instale y registre adecuadamente el **OCX Fiscal HASAR**. Por ejemplo: *Inno Setup*, *InstallShield*, *NSIS*, *Visual Installer*, *Windows Installer*, etc.; por mencionar sólo algunos.

Incorporando el OCX Fiscal HASAR a un proyecto

Nadie mejor que el propio desarrollador del software de gestión para puntos de venta para incorporar el **OCX Fiscal HASAR** a su entorno de desarrollo. Debido a ello, y a la variedad de entornos de desarrollo de software, se mostrarán algunos casos a modo de ejemplo. El mejor lugar para obtener información sobre el tema que nos ocupa, es la documentación propia del entorno de desarrollo de software.

Caso: VC# Visual Studio 2010 Professional

En la primera imagen de la página siguiente puede apreciarse como el examinador de objetos (*Object Browser*) lista aquellos *Namespaces* que integran el proyecto sobre el que estamos trabajando. En la lista puede apreciarse la presencia de **FiscalPrinterLib** (el cual corresponde al **OCX Fiscal HASAR**).

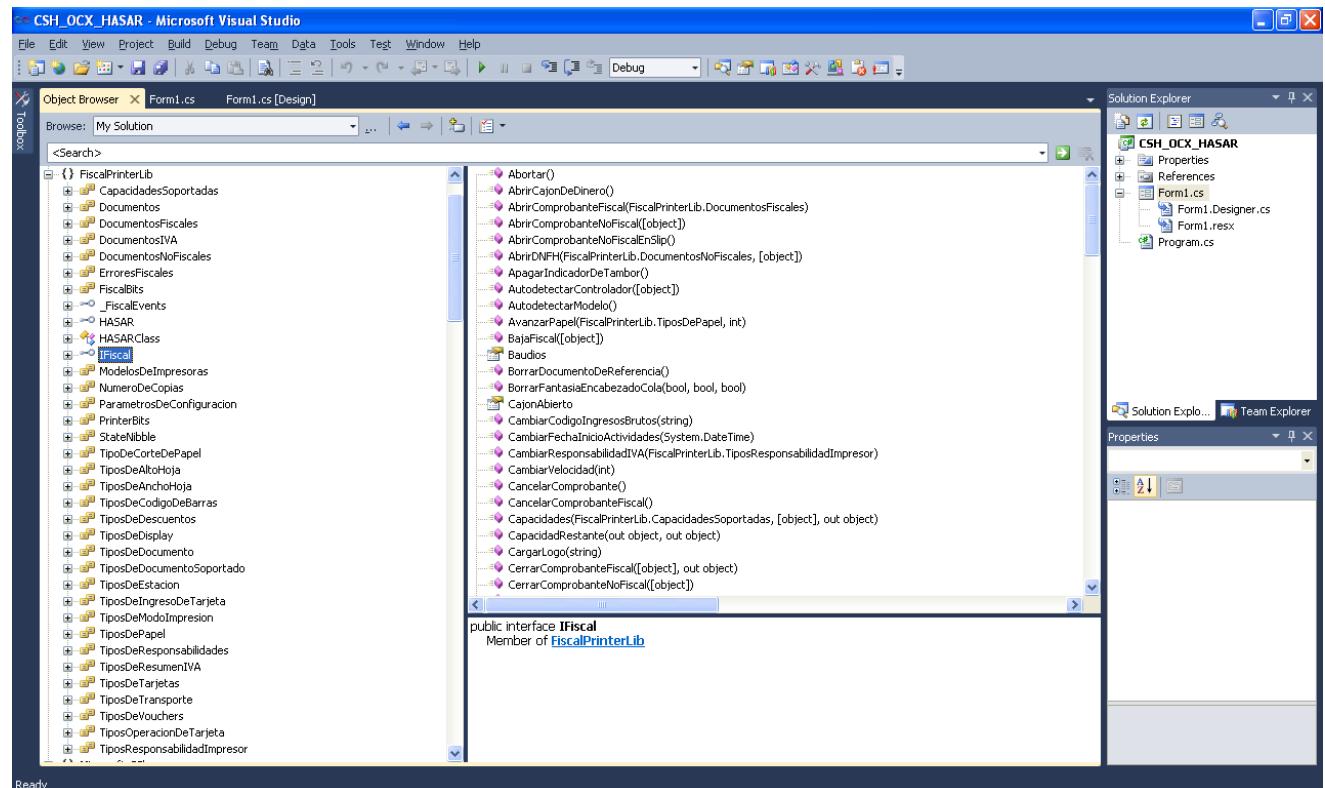
El namespace *FiscalPrinterLib* contiene:

HASAR

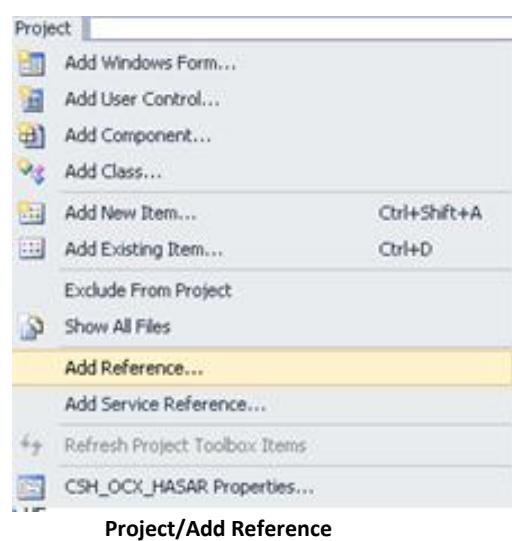
Clase pública que define al **OCX Fiscal HASAR**.

- IFiscal* Interfaz empleada para los métodos y propiedades del **OCX Fiscal HASAR**.
- _FiscalEvents* Interfaz empleada para los eventos que genera el **OCX Fiscal HASAR**.
- ... Constantes embebidas.

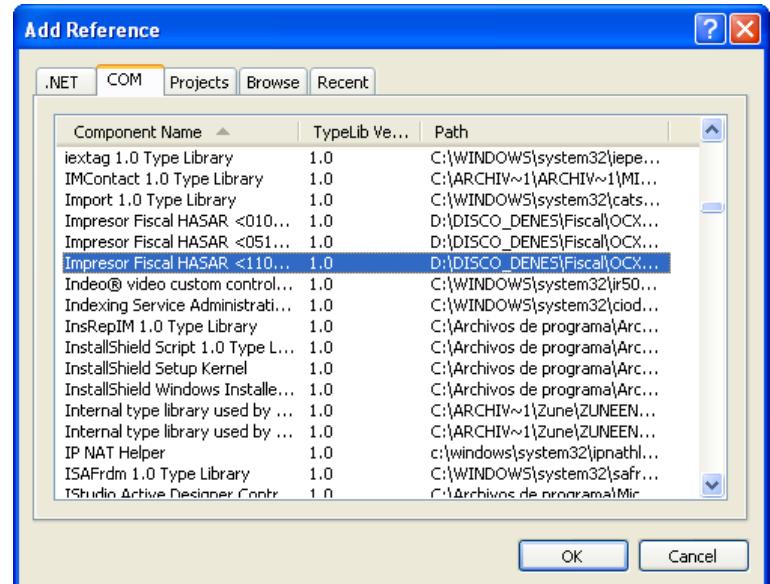
La incorporación del **OCX Fiscal HASAR** al proyecto puede hacerse a través de la opción *Project/Add Reference* de la barra de tareas del entorno de desarrollo (ver imagen inferior izquierda).



Object Browser



Project/Add Reference



Solapa COM

Al hacer clic sobre la opción *Add Reference* se abre una ventana de diálogo (ver imagen anterior -a derecha-), la cual ofrece varias solapas de acceso. A continuación se debe hacer clic sobre la solapa *COM*. Como resultado, el examinador de objetos despliega la lista de componentes registrados por nombre.

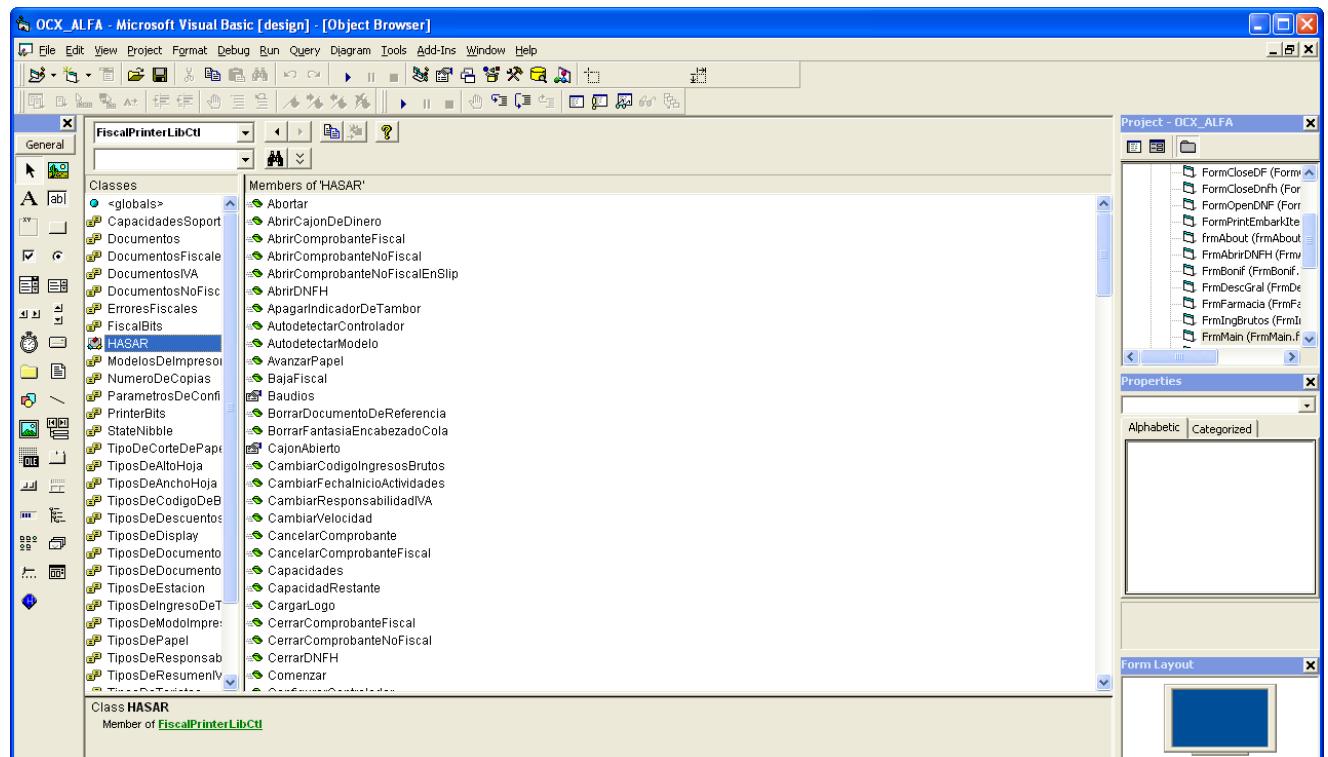
Finalmente, se debe hacer clic sobre el nombre del **OCX Fiscal HASAR**, que aparece como: *Impresor Fiscal HASAR <110423>* y realizando un clic sobre el botón *OK*. Una vez completados los pasos indicados anteriormente se está en condiciones de utilizar el **OCX Fiscal HASAR** en el proyecto de interés.

El ejemplo siguiente se muestra como comenzar a utilizar los métodos y propiedades disponibles en el **OCX Fiscal HASAR**. En particular, se configura la propiedad *Modelo()* para indicar que se va a trabajar con la **IFH 1G** modelo SMH/P-441F v01.00.

```
namespace CSH_OCX_HASAR
{
    public partial class Form1 : Form
    {
        FiscalPrinterLib.HASAR ofh = new FiscalPrinterLib.HASAR();

        private void button1_Click(object sender, EventArgs e)
        {
            ofh.Modelo = FiscalPrinterLib.ModelosDelImpresoras.MODELO_P441;
            ....
        }
    }
}
```

Caso: Visual Basic 6.0

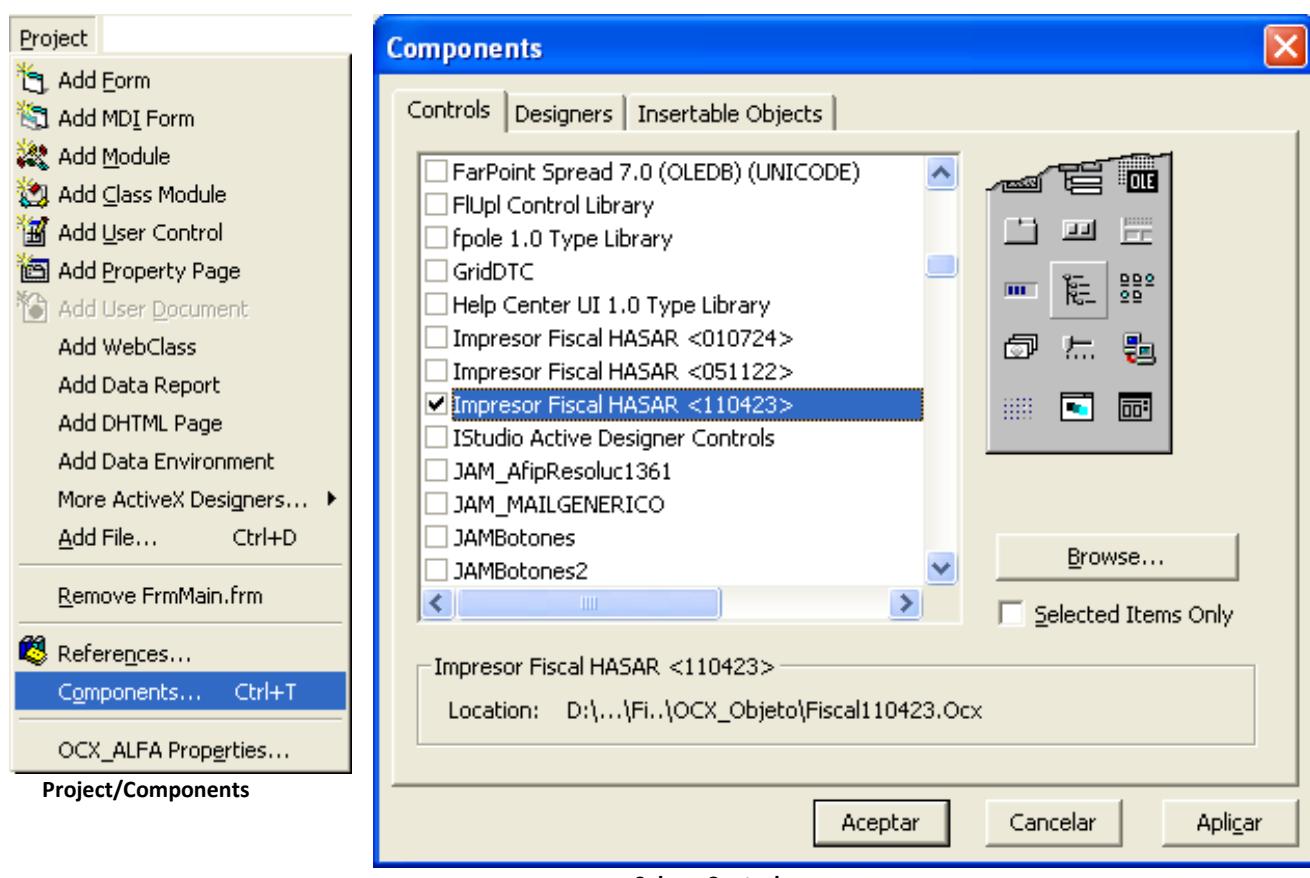


En la figura anterior puede apreciarse como el examinador de objetos (*Object Browser*) lista las librerías y clases que integran el proyecto sobre el que estamos trabajando, entre las que se encuentra *FiscalPrinterLibCtl* (la cual corresponde al **OCX Fiscal HASAR**).

La librería *FiscalPrinterLibCtl* contiene:

- HASAR Clase pública que define al **OCX Fiscal HASAR**.
... Constantes embebidas.

La incorporación del **OCX Fiscal HASAR** al proyecto de interés puede hacerse a través de la opción *Project/Components* de la barra de tareas del entorno de desarrollo (ver imagen siguiente -a la izquierda-). Al hacer clic sobre la opción *Project/Components* se abre una ventana de diálogo con distintas solapas de acceso (ver imagen siguiente -a la derecha-). A continuación se debe hacer clic sobre la solapa *Controls*. Como resultado el examinador de objetos despliega la lista de componentes por nombre.



Solapa Controls

Finalmente se debe hacer clic sobre el nombre del **OCX Fiscal HASAR** y se tilda al componente, que aparece como: *Impresor Fiscal HASAR <110423>*. Para finalizar, realizar un clic sobre el botón *OK*. Una vez completados los pasos indicados anteriormente se está en condiciones de utilizar el **OCX Fiscal HASAR** en el proyecto deseado.

El ejemplo siguiente muestra como comenzar a utilizar los métodos y propiedades disponibles en el **OCX Fiscal HASAR**. En particular, se configura la propiedad *Modelo()* para indicar que se debe trabajar con la **IFH 1G** modelo SMH/P-441F v01.00.

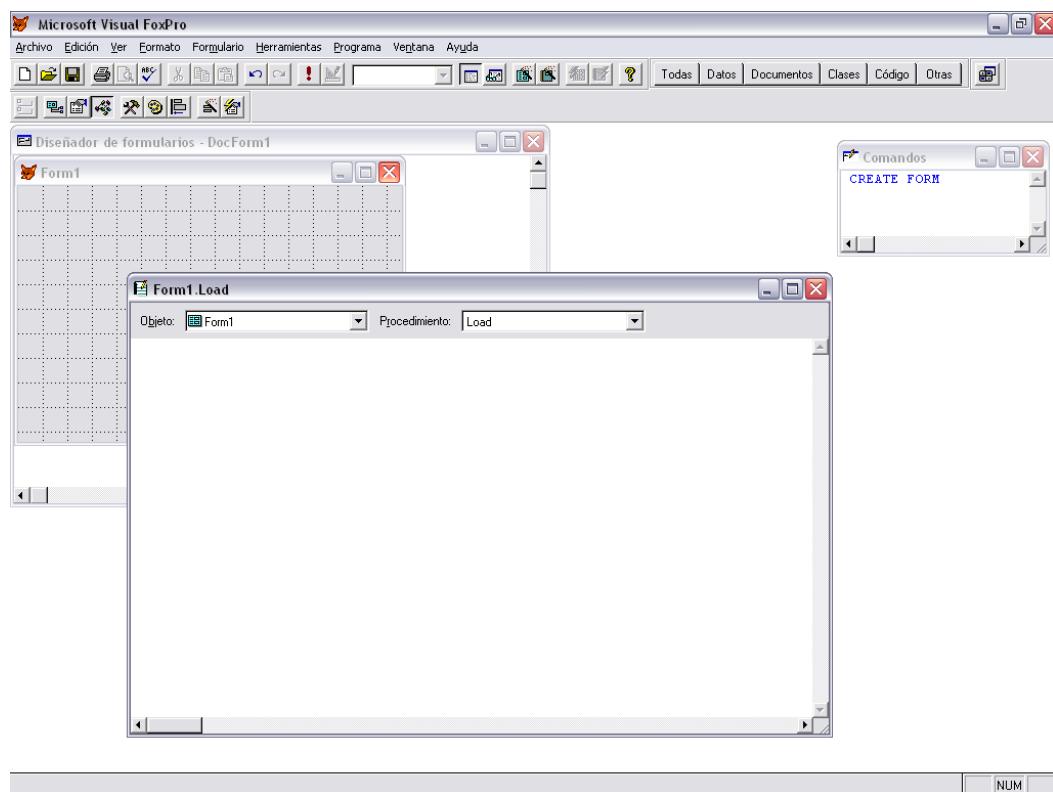
```
Public Sub Dialogar()
    HASAR1.Modelo = MODELO_P441
    '// otras instrucciones de interés
    '// ...
```

End Sub

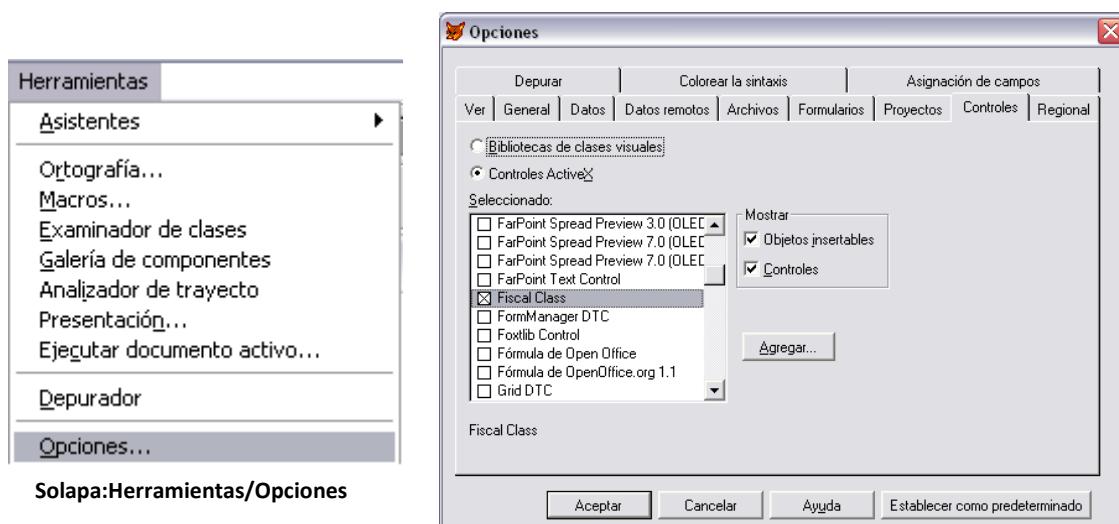
Caso: Visual Fox Pro 6.0

La incorporación del OCX Fiscal HASAR al proyecto puede hacerse a través de la opción *Herramientas/Opciones* del entorno de desarrollo (ver imagen -a la izquierda- al pie de esta la página). Al hacer clic sobre la opción *Herramientas/Opciones* se abre una ventana de diálogo (ver imagen -a la derecha- en la página siguiente), con varias solapas de acceso. A continuación se debe hacer clic sobre la solapa *Controles*.

Asegurarse de haber seleccionado la opción *Controles Activex*, tildando en el marco *Mostrar las búsquedas por Objetos insertables y Controles*.



Entorno de desarrollo



Controles

Como resultado de lo expuesto, el entorno de desarrollo despliega una lista de componentes por nombre. La acción siguiente es tildar *Fiscal Class*, para finalizar con un clic sobre el botón *Aceptar*.

Según el propio help de Visual FoxPro 6.0: "**The AutoYield property should be set to False when a form contains an ActiveX control**" (La propiedad *AutoYield* se debe establecer en .F. <falso> cuando un formulario contiene un control ActiveX -como es el caso del **OCX Fiscal HASAR**-).

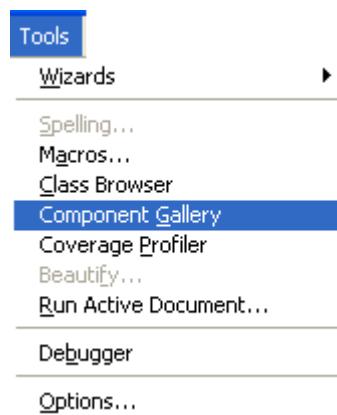
En consecuencia, es importante tener en cuenta que el software de gestión para puntos de venta podrá recibir todos los eventos generados por el **OCX Fiscal HASAR** (a partir de las respuestas de la **IFH 1G**) siempre y cuando la propiedad *AutoYield* (de Visual Fox Pro 6.0) se encuentre seteada con el valor .F. (falso). De lo contrario, el manejo de errores por parte del software de gestión para puntos de venta se verá exigido al empleo de otros métodos más elaborados para obtener el mismo resultado. Se trata de una limitación del propio lenguaje de desarrollo.

El siguiente ejemplo permite cumplir con lo expuesto:

```
PROCEDURE Load
    SYS(2333,0)
    _VFP.AutoYield = .F.  && Para detectar los eventos del OCX Fiscal HASAR debe ser false
    RETURN
ENDPROC

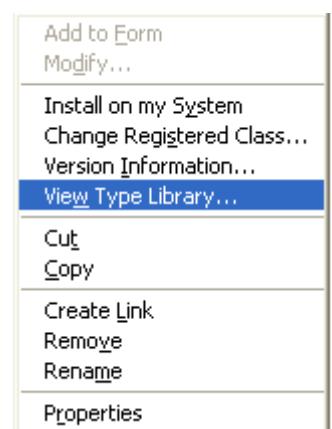
PROCEDURE Unload
    _VFP.AutoYield = .T.
    RETURN
ENDPROC
```

Mediante la barra de tareas del entorno de desarrollo se puede acceder a la galería de componentes (catálogo) y buscar el **OCX Fiscal HASAR** entre los controles instalados. Para ello, hacer clic en la opción *Tools/Component Gallery*.



Una vez visualizada la galería de componentes, expandir *Activex Catalog/Controls* y seleccionar *Installed Controls*. En la lista que se despliega a derecha el **OCX Fiscal HASAR** es mostrado como *Fiscal Class* (ver imagen en la página siguiente).

Para examinar los elementos que componen al **OCX Fiscal HASAR** se debe seleccionar de la lista a *Fiscal Class* y hacer clic con el botón derecho del mouse. Con esta acción emerge un menú donde debe seleccionarse la opción *View Type Library* (ver imagen de la derecha).



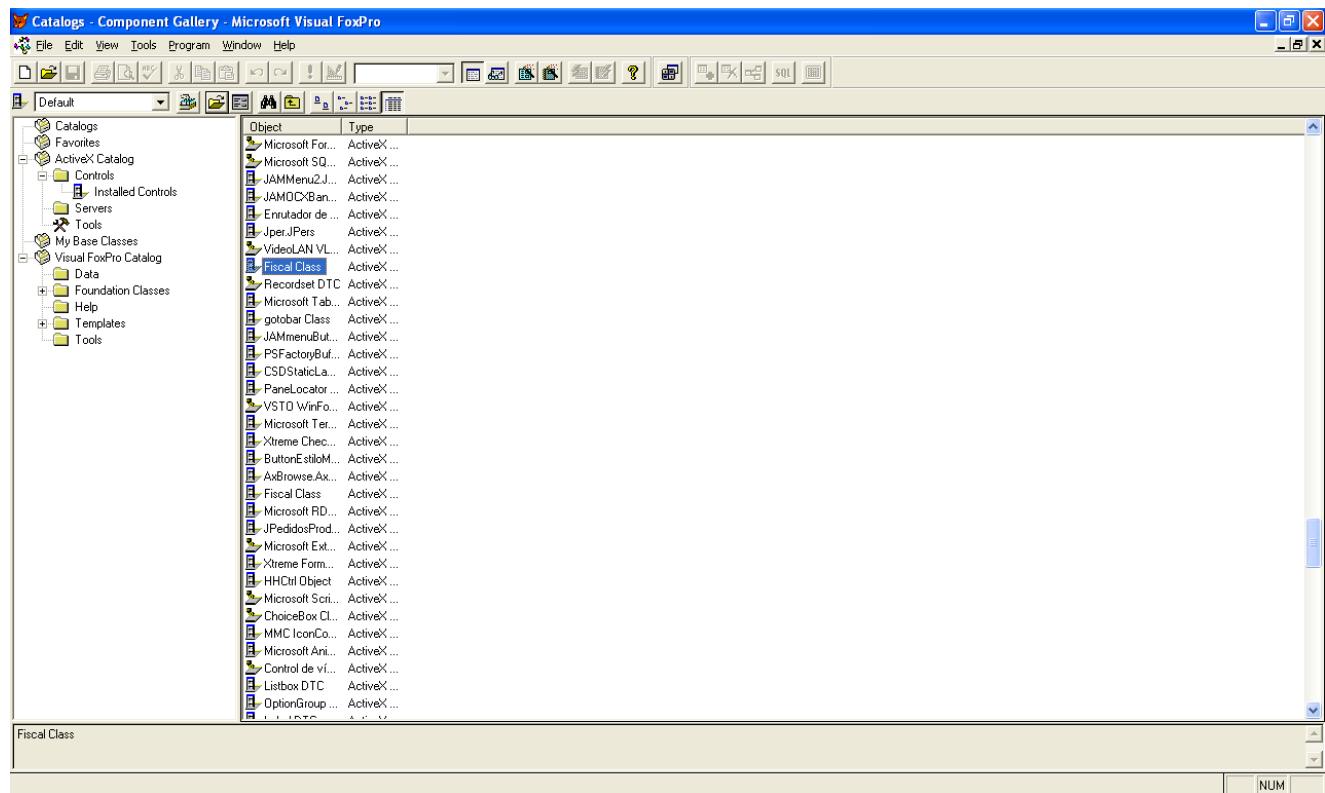
Como resultado de esta última acción, se muestra la lista de miembros de *Fiscal Class* (ver imagen en la página siguiente). En esta lista podrá consultarse:

HASAR

Clase del **OCX Fiscal HASAR**.

IFiscal Métodos y propiedades que componen la interfaz del **OCX Fiscal HASAR**.

_FiscalEvents Eventos que puede generar el **OCX Fiscal HASAR**.



Galería de Componentes instalados

FiscalPrinterLib	
StateNibble	
Documentos	
DocumentosFiscales	
DocumentosNoFiscales	
TiposDeDocumento	
TiposDeResponsabilidades	
TiposResponsabilidadImpresor	
TiposDeEstacion	
TiposDePapel	
TiposDeDisplay	
TiposDeCodigoDeBarras	
TiposDeDescuentos	
TiposDeVouchers	
TiposDeTarjetas	
TiposDelIngresoDeTarjeta	
TiposOperacionDeTarjeta	
ParametrosDeConfiguracion	
NumerosDeCopias	
TipoDeCorteDePapel	
TiposDeAltoHoja	
TiposDeAnchoHoja	
TiposDeModoImpresion	
CapacidadesSoportadas	
TiposDeDocumentoSoportado	
TiposDeResumenIVA	
ModelosDeImpresoras	
FiscalBits	
PrinterBits	
ErroresFiscales	
TiposDeTransporte	
DocumentosIVA	
IFiscal	
_FiscalEvents	
HASAR	

El resto de los elementos de la lista (ver imagen de la izquierda) son las constantes embebidas del **OCX Fiscal HASAR**. Por limitaciones propias del lenguaje, no es posible ver sus nombres y valores (algo posible con otros lenguajes de desarrollo).

Todos estos elementos, incluidas las constantes embebidas del **OCX Fiscal HASAR**, se encuentran definidas en el archivo con extensión TLB. En el caso del **OCX Fiscal HASAR v110423**, dicho archivo se llama *fiscal110423.tlb*.

El ejemplo que se brinda a continuación muestra como comenzar a utilizar los métodos y propiedades disponibles en el **OCX Fiscal HASAR**. En particular, se configura la propiedad *Modelo()* para indicar que se debe trabajar con la **IFH 1G** modelo SMH/P-441F v01.00.

```
DEFINE CLASS frmComprobantes AS FORM
...
This.AddObject('oFiscal', 'frmDialogar')
...
ENDDEFINE
```

```
DEFINE CLASS frmDialogar AS OleControl
...

```

```
OleClass = "hasar.fiscal.1"

WITH ThisForm.oFiscal
...
    .Modelo = 32      && Constante MODELO_P441 = 32
...
ENDWITH
...
ENDDEFINE
```

El OCX Fiscal HASAR y Windows®

El **OCX Fiscal HASAR** puede incluirse en desarrollos 32/64 bits, e incluso registrarse en PCs de 64 bits. La única recomendación es no utilizar el **OCX Fiscal HASAR v010724** en versiones de Windows® posteriores a XP.

Impresoras Fiscales HASAR (IFH 1G) soportadas

En la web oficial de *Grupo HASAR* se encuentra publicada la lista de **IFH 1G** homologadas en *Argentina*. En dicha lista se menciona la *Resolución General* de *AFIP* por la cual cada producto fue homologado y, se indica además, cuando fue publicada en el *Boletín Oficial de la República Argentina* (BORA).

Debido a que hay varias versiones liberadas del **OCX Fiscal HASAR** hasta la fecha, a continuación se brinda la lista de modelos de **IFH 1G** soportadas naturalmente por cada una de ellas.

Modelo SMH/	Constantes Embebidas		OCX Fiscal HASAR			
	Nombre	Dec	v010724	v051122	v110423	v150423
P-614F	MODELO_614	1	SI	SI	SI	SI
P-615F	MODELO_615	2	SI	SI	SI	SI
P-PR4F	MODELO_PR4	3	SI	SI	SI	SI
P-950F	MODELO_950	4	SI	SI	SI	SI
P-951F	MODELO_951	5	SI	SI	SI	SI
PT-262F	MODELO_262	6	SI	SI	SI	SI
PJ-20F	MODELO_PJ20	7	SI	SI	SI	SI
P-320F	MODELO_P320	8	SI	SI	SI	SI
P-715F	v01.00 MODELO_715	9		SI	SI	SI
"	v02.01 MODELO_715_201	25		SI	SI	SI
"	v03.02 MODELO_715_302	29		SI	SI	SI
"	v04.03 MODELO_715_403	30		SI	SI	SI
"	V05.04 MODELO_715_504	40				SI
P-PR5F	v01.00 MODELO_PR5	10		SI	SI	SI
"	v02.01 MODELO_PR5_201	26		SI	SI	SI
"	v03.02 MODELO_PR5_302	37			SI	SI
PT-272F	v01.00 MODELO_272	11		SI	SI	SI
"	v02.01 MODELO_272_201	38			SI	SI
PL-8F	v01.00 MODELO_PPL8	12		SI	SI	SI
"	v02.01 MODELO_PPL8_201	17		SI	SI	SI
P-321F	v01.00 MODELO_P321	13		SI	SI	SI
P-322F	v01.00 MODELO_P322	14		SI	SI	SI
"	v02.01 MODELO_P322_201	18		SI	SI	SI
P-425F	v01.00 MODELO_P425	15		SI	SI	SI
"	v02.01 MODELO_P425_201	16		SI	SI	SI
P-330F	v01.00 MODELO_P330	19		SI	SI	SI
"	v02.01 MODELO_P330_201	21		SI	SI	SI
"	v02.02 MODELO_P330_202	23		SI	SI	SI
"	v02.03 MODELO_P330_203	31		SI	SI	SI
P-435F	v01.00 MODELO_P435	20		SI	SI	SI
"	v01.01 MODELO_P435_101	24		SI	SI	SI
"	v01.02 MODELO_P435_102	27		SI	SI	SI
"	v02.03 MODELO_P435_203	34		SI	SI	SI
PL-9F	v01.00 MODELO_PPL9	22		SI	SI	SI
PL-23F	v01.00 MODELO_PPL23	28		SI	SI	SI
"	v01.01 MODELO_PPL23_101	33		SI	SI	SI
"	v02.02 MODELO_PPL23_202	36			SI	SI
P-441F	v01.00 MODELO_P441	32		SI	SI	SI

Modelo SMH/	Constantes Embebidas		OCX Fiscal HASAR			
	Nombre	valor	v010724	v051122	v110423	v150423
P-441F v02.01	MODELO_P441_201	39				SI
P-1120F v01.00	MODELO_P1120	35		SI	SI	SI
P-340F v01.00	MODELO_P340	41				SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como “NO SOPORTADO”

Que un modelo de **IFH 1G** se encuentre SOPORTADO naturalmente significa que los métodos/propiedades del **OCX Fiscal HASAR** generan los strings de comandos con la cantidad de campos requeridos por el equipo fiscal al cual se van a enviar. La configuración de la propiedad *Modelo()* (des)habilita propiedades, métodos y constantes embebidas en relación con la **IFH 1G** que se indique.

El OCX Fiscal HASAR no es responsable de la información contenida en los comandos que genera. Dicha información es suministrada por el software de gestión para puntos de venta.

Si un modelo de **IFH 1G** no estuviese soportado por la versión de **OCX Fiscal HASAR** en uso, entonces:

- El desarrollador del software de gestión para puntos de venta puede actualizar la versión del **OCX Fiscal HASAR**.
- Puede mantener la versión del **OCX Fiscal HASAR** en uso, pero:
 - Se deberá seleccionar alguno de los modelos soportados, cuyo conjunto de strings de comandos resulte lo más compatible posible con el modelo real a manejar.
 - Mayormente, cuando se procede como en el punto anterior aparecen algunas incompatibilidades que resolver. Es decir, se disparan excepciones y/o eventos que anuncian la presencia de incompatibilidades a la hora de invocar métodos/propiedades del **OCX Fiscal HASAR**.

De surgir incompatibilidades estas se resuelven de manera sencilla, mediante los siguientes pasos:

- El software de gestión para puntos de venta deberá construir directamente los strings de comandos que el **OCX Fiscal HASAR** no puede generar de manera compatible, por no soportar naturalmente a la **IFH 1G** conectada.
 - Se deberá emplear el método *Enviar()* del **OCX Fiscal HASAR** para que el string de comando construido “viaje” hacia la **IFH 1G** conectada.
-

Es importante que el desarrollador del software de gestión para puntos de venta tenga siempre a mano, para consultas, la presente documentación y el *Manual de Comandos de la IFH 1G* a soportar con su producto.

Métodos del OCX Fiscal HASAR

La tabla siguiente muestra la disponibilidad de métodos del **OCX Fiscal HASAR** en las distintas versiones liberadas hasta la fecha.

Métodos disponibles	OCX Fiscal HASAR		
	v010724	v051122	v110423 v150423
<i>Abortar()</i>	SI	SI	SI
<i>AbrirCajonDeDinero()</i>	SI	SI	SI
<i>AbrirComprobanteFiscal()</i>	SI	SI	SI
<i>AbrirComprobanteNoFiscal()</i>	SI	SI	SI
<i>AbrirComprobanteNoFiscalEnSlip()</i>	SI	SI	SI
<i>AbrirComprobanteNoFiscalHomologado()</i>	SI		
<i>AbrirDNFH()</i>		SI	SI
<i>ApagarIndicadorDeTambor()</i>		SI	SI
<i>AutodetectarControlador()</i>	SI	SI	SI
<i>AutodetectarModelo()</i>			SI
<i>AvanzarPapel()</i>	SI	SI	SI
<i>BajaFiscal()</i>	SI	SI	SI
<i>BorrarDocumentoDeReferencia()</i>			SI
<i>BorrarFantasiaEncabezadoCola()</i>			SI
<i>CambiarCodigoIngresosBrutos()</i>			SI
<i>CambiarFechaInicioActividades()</i>			SI
<i>CambiarResponsabilidadIVA()</i>	SI	SI	SI
<i>CambiarVelocidad()</i>	SI	SI	SI
<i>CancelarComprobante()</i>	SI	SI	SI
<i>CancelarComprobanteFiscal()</i>	SI	SI	SI
<i>Capacidades()</i>			SI
<i>CapacidadRestante()</i>	SI	SI	SI
<i>CargarLogo()</i>	SI	SI	SI
<i>CerrarComprobanteFiscal()</i>	SI	SI	SI
<i>CerrarComprobanteNoFiscal()</i>	SI	SI	SI
<i>CerrarComprobanteNoFiscalHomologado()</i>	SI		
<i>CerrarDNFH()</i>			SI
<i>Comenzar()</i>	SI	SI	SI
<i>CortarComprobanteNoFiscal()</i>	SI	SI	SI
<i>ConfigurarControlador()</i>	SI	SI	SI
<i>ConfigurarControladorCompleto()</i>	SI	SI	SI
<i>ConfigurarControladorPorBloque()</i>	SI	SI	SI
<i>DatosCliente()</i>	SI	SI	SI
<i>DescripcionStatusFiscal()</i>	SI	SI	SI
<i>DescripcionStatusImpresor()</i>	SI	SI	SI
<i>DescuentoGeneral()</i>	SI	SI	SI
<i>DescuentoUltimoItem()</i>	SI	SI	SI
<i>DetalleRecibo()</i>	SI	SI	SI
<i>DevolucionDescuento()</i>	SI	SI	SI
<i>DevolverPago()</i>			SI
<i>DNFHFarmacias()</i>	SI	SI	SI

Métodos disponibles	OCX Fiscal HASAR		
	v010724	v051122	v110423 v150423
<i>DNFHReparto()</i>	SI	SI	SI
<i>Enviar()</i>	SI	SI	SI
<i>EnviarStatPrn()</i>	SI	SI	SI
<i>EspecificarIVANoInscripto()</i>	SI	SI	SI
<i>EspecificarNombreDeFantasia()</i>	SI	SI	SI
<i>EspecificarPercepcionGlobal()</i>	SI	SI	SI
<i>EspecificarPercepcionPorIVA()</i>	SI	SI	SI
<i>EstadoInternolImpresor()</i>		SI	SI
<i>Finalizar()</i>	SI	SI	SI
<i>ImprimirCódigoDeBarras()</i>	SI	SI	SI
<i>ImprimirFirmaOpcionalEnCierreDNFH()</i>		SI	SI
<i>ImprimirInfoEnDNFH()</i>		SI	SI
<i>ImprimirItem()</i>	SI	SI	SI
<i>ImprimirItemEnCotizacion()</i>	SI	SI	SI
<i>ImprimirItemEnCuenta()</i>	SI	SI	SI
<i>ImprimirItemEnRemito()</i>	SI	SI	SI
<i>ImprimirPago()</i>	SI	SI	SI
<i>ImprimirTextoFiscal()</i>	SI	SI	SI
<i>ImprimirTextoNoFiscal()</i>	SI	SI	SI
<i>ImprimirVoucher()</i>	SI	SI	SI
<i>ImprimirVoucherConFechaOculta()</i>		SI	SI
<i>LeerIVA()</i>		SI	SI
<i>LeerMemoriaDeTrabajo()</i>	SI	SI	SI
<i>LeerMontoTotal()</i>		SI	SI
<i>LeerPercepcionesGenerales()</i>		SI	SI
<i>LeerPercepcionesPorDescripcion()</i>		SI	SI
<i>LeerPercepcionesPorIVA()</i>		SI	SI
<i>MostrarEnDisplay()</i>	SI	SI	SI
<i>ObtenerCAIUltimoDocumentoA()</i>		SI	SI
<i>ObtenerConfiguracion()</i>		SI	SI
<i>ObtenerConfiguracionCompleta()</i>	SI	SI	SI
<i>ObtenerDatosDeConfiguracion()</i>	SI	SI	SI
<i>ObtenerDatosDeInicializacion()</i>	SI	SI	SI
<i>ObtenerNombreDeFantasia()</i>	SI	SI	SI
<i>ObtenerVersionDelControlador()</i>	SI	SI	SI
<i>PedidoDeStatus()</i>	SI	SI	SI
<i>PedirPrimerIVA()</i>	SI	SI	SI
<i>PedirSiguienteIVA()</i>	SI	SI	SI
<i>Recibir()</i>	SI	SI	SI
<i>Recomenzar()</i>	SI	SI	SI
<i>ReimprimirComprobante()</i>	SI	SI	SI
<i>ReporteX()</i>	SI	SI	SI
<i>ReporteZ()</i>	SI	SI	SI
<i>ReporteZIndividualPorFecha()</i>	SI	SI	SI
<i>ReporteZIndividualPornumero()</i>	SI	SI	SI
<i>ReporteZPorFechas()</i>	SI	SI	SI
<i>ReporteZPorNumeros()</i>	SI	SI	SI
<i>Subtotal()</i>	SI	SI	SI

Métodos disponibles	OCX Fiscal HASAR		
	v010724	v051122	v110423 v150423
TratarDeCancelarTodo()	SI	SI	SI
VerificarModelo()		SI	SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como “NO SOPORTADO”

Los métodos del **OCX Fiscal HASAR** generan y envían los strings de comandos que soportan las **IFH 1G**, a partir de la información que el software de gestión para puntos de venta le suministra a través de sus argumentos. Son los menos, pero también hay métodos que generan y envían strings de comandos sin requerir argumento alguno. Como hay otros que no generan, ni envían strings de comandos.

Por regla general, se podría decir, que existe una relación uno a uno entre un determinado método del **OCX Fiscal HASAR** y un string de comando soportado por la **IFH 1G**.

Se hace notar que existen métodos en el **OCX Fiscal HASAR** que poseen argumentos opcionales de entrada, de salida, o de ambos tipos. Esta presencia es dependiente del string de comando a construir, y que espera recibir en él la **IFH 1G** a la que se envía el comando.

Los argumentos opcionales de entrada permiten omitir información que no es requerida en forma obligatoria en la construcción del string de comando. Los argumentos opcionales de salida permiten que el software de gestión para puntos de venta obtenga información contenida en el string de respuesta al comando enviado. Los argumentos opcionales de salida evitan tener que invocar a la propiedad *Respuesta()* tantas veces como campos de información contenga el string de respuesta al comando enviado.

Es responsabilidad del desarrollador del software de gestión para puntos de venta consultar el *Manual de comandos* de la **IFH 1G** a utilizar, además del contenido de este manual. Toda la información referida a la construcción de strings de comandos, strings de respuestas, y comportamiento de la **IFH 1G**, se detalla en su correspondiente *Manual de comandos*.

Es responsabilidad del software de gestión para puntos de venta la validez de la información que suministre para la generación de los strings de comandos, así como el tratamiento de los eventos y excepciones que pudieran generarse al invocar a un método del **OCX Fiscal HASAR**.

Método: Abortar()

Este método no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: No requiere.

El software de gestión para puntos de venta puede utilizar el método *Abortar()* para interrumpir un proceso

en curso (esperar la respuesta a un comando) cuando la propiedad *ReintentConstante()* se encuentra configurada con el valor *True*, y se produce un error por *time out* en la comunicación con la **IFH 1G**. Esta interrupción se puede realizar luego de una cantidad de reintentos de envío del comando, estipulada por el usuario.,

Ejemplo de uso:

```
Private Sub HASAR1_ImpresoraNoResponde(ByVal CantReint As Long)
...
If (HASAR1.ReintentConstante) And (CantReint = 3) Then
    HASAR1.Abortar
...
End If
...
End Sub
```

En el ejemplo anterior se muestra como invocar al método *Abortar()* desde el cuerpo de la rutina de atención del evento *ImpresoraNoResponde()* cuando, por ejemplo, se envía un comando a la **IFH 1G** estando ésta apagada, la propiedad *ReintentConstante()* se encuentra en *True*, y se han superando los tres reintentos de envío del comando en cuestión.

Método: AbrirCajonDeDinero()

Comando: *OpenDrawer*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: No requiere.

El uso de este método tiene sentido si la **IFH 1G** soporta conexión de cajón de dinero.

Ejemplo de uso:

```
...
HASAR1.AbrirCajonDeDinero
HASAR1.CerrarComprobanteFiscal
...
```

En el ejemplo anterior el cajón de dinero es abierto mientras se imprime el cierre del comprobante fiscal que se está emitiendo.

Los desarrolladores del software de gestión para puntos de venta deben tener presente que:

- Son quienes deciden el momento oportuno de apertura del cajón de dinero.
- No todos los modelos y versiones de **IFH 1G** soportan la conexión de un cajón de dinero.

Método: AbrirComprobanteFiscal()

Comando: *OpenFiscalReceipt*

Disponible: **OCX Fiscal HASAR** en todas las versiones.

Argumentos: 1

AbrirComprobanteFiscal tipo

tipo	<p>De entrada. Obligatorio. Tipo: enum <i>DocumentosFiscales</i></p> <p>Permite al software de gestión para puntos de venta especificar el tipo de comprobante fiscal que desea abrir.</p> <p>Los comprobantes fiscales posibles de ser abiertos, son los identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR:</p> <p>Impresión en rollo: TICKET_FACTURA_A, TICKET_FACTURA_B, TICKET_NOTA_DEBITO_A, TICKET_NOTA_DEBITO_B, TICKET_C.</p> <p>Impresión en formulario continuo u hojas sueltas: FACTURA_A, FACTURA_B, NOTA_DEBITO_A, NOTA_DEBITO_B, RECIBO_A, RECIBO_B.</p>
-------------	--

Ejemplo de uso:

```
...
Dim respons As TiposDeResponsabilidades
...
HASAR1.DatosCliente "Razón Social...", "999999999995", TIPO_CUIT, RESPONSABLE_INSCRIPTO, _
"Domicilio..."
HASAR1.AbrirComprobanteFiscal FACTURA_A
...
```

En el ejemplo anterior se abre una factura “A” a un contribuyente Responsable Inscripto.

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- No todo los tipos de comprobantes fiscales mencionados se encuentran disponibles en todos los modelos y versiones de firmware de **IFH 1G**.
- En el caso de comprobantes fiscales “A” es obligatorio, y previo, el envío de los datos del cliente a la **IFH 1G** (ver método *DatosCliente()*).
- En el caso de comprobantes Fiscales “B” es obligatorio, y previo, el envío de los datos del cliente a la **IFH 1G** (ver método *DatosCliente()*), sólo en el caso de tratarse de un contribuyente Consumidor Final y el monto del comprobante fuese a superar los \$1000 (pesos un mil) establecidos por la normativa fiscal vigente. En cualquier otro caso, los datos del cliente se pueden omitir.

NOTA: Mediante la RG AFIP Nº 3666/14, vigente desde el 4 de Setiembre de 2014, se suspende temporalmente la aplicación del tope de \$1000 (pesos un mil) como monto máximo para Tiques y comprobantes B a consumidores finales sin nominar.

Por suspensión temporal se entiende: hasta tanto entre en vigencia lo previsto en los párrafos segundo y tercero del Artículo 3º de la RG AFIP Nº 3561/13.

- No se requieren de datos del cliente en la emisión de Tiques a contribuyentes Consumidores Finales (comprobantes fiscales identificados como TICKET_C).

- Los comprobantes fiscales “A” se pueden abrir siempre que el receptor y el emisor sean -ambos- contribuyentes Responsables Inscriptos. En cualquier otro caso, se debe abrir un comprobante fiscal “B”.
- Cuando se abre un comprobante fiscal “B” se imprime la letra ‘B’ sólo en el caso de que el emisor sea un contribuyente Responsable inscripto. Si el emisor posee otra categoría de contribuyente, se imprime la letra ‘C’.
- En *Modo Entrenamiento*, a exigencias de la normativa fiscal vigente, la impresión de los comprobantes fiscales presentan el siguiente aspecto:
 - Se imprime la leyenda “***No Fiscal ***” cada cuatro líneas de impresión.
 - No se imprime el logo fiscal *CF DGI*.
 - En las líneas que contienen al menos un carácter imprimible, los espacios son reemplazados por el carácter de interrogación (?)
- La numeración de los comprobantes fiscales es responsabilidad de la **IFH 1G**. El software de gestión para puntos de venta debe consultar esta numeración, pero no puede modificarla.
- En el caso de impresión de comprobantes fiscales sobre formularios continuos u hojas sueltas, de requerirse más de una hoja, resulta transparente para el software de gestión para puntos de venta el transporte a la hoja siguiente.
- En el caso de realizarse transporte a la hoja siguiente, la numeración de la nueva hoja se incrementa en uno.
- El número de CAI y la fecha de vencimiento, a exigencias de la normativa fiscal siguiente, se imprime solamente en comprobantes fiscales “A”.

El número de CAI impreso en comprobantes fiscales emitidos por la IFH 1G no se verifica en el sitio oficial de AFIP. Es válido por haber sido emitido con una impresora fiscal homologada.

Método: AbrirComprobanteNoFiscal()

Comando: *OpenNonFiscalReceipt*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 1

AbrirComprobanteNoFiscal estacion

estacion	De entrada. Opcional. Tipo: enum <i>TiposDeEstacion</i> <p>Permite al software de gestión para puntos de venta abrir un comprobante no fiscal, con la opción de indicar en qué estación de impresión hacerlo. Si se omite el OCX Fiscal HASAR asume el valor por defecto: ESTACION_TICKET.</p> <p>Las estaciones de impresión posibles, son las identificadas por las siguientes constantes embebidas:</p>
-----------------	---

das en el **OCX Fiscal HASAR**:
ESTACION_TICKET (rollo), ESTACION_SLIP (formulario continuo u hoja suelta).

Ejemplo de uso:

```
....  
HASAR1.AbrirComprobanteNoFiscal ESTACION_TICKET  
....
```

Invocar al método *AbrirComprobanteNoFiscal()* indicando como argumento ESTACION_SLIP es equivalente a invocar al método *AbrirComprobanteNoFiscalEnSlip()*.

El uso destinado al comprobante no fiscal en formato de hoja suelta o formulario continuo está restringido a los indicados por la RG AFIP 259/98 (con sus modificatorias y/o complementarias).

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- La existencia de ESTACION_TICKET y/o ESTACION_SLIP depende del modelo y versión de firmware fiscal de la **IFH 1G** en uso.
- Todos los modelos y versiones de firmware fiscal de **IFH 1G** admiten la emisión de comprobantes no fiscales.
- Cada cuatro líneas de impresión, por exigencia de la normativa fiscal vigente, se imprime la leyenda “*** NO FISCAL ***”. No hay mecanismo que inhabilite la impresión de la leyenda mencionada.
- Los comprobantes no fiscales no se numeran.

Método: AbrirComprobanteNoFiscalEnSlip()

Comando: *OpenNonFiscalSlip*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: No requiere.

Ejemplo de uso:

```
...  
HASAR1.AbrirComprobanteNoFiscalEnSlip  
...
```

El uso destinado al comprobante no fiscal en formato de hoja suelta o formulario continuo está restringido a los indicados por la RG AFIP 259/98 (con sus modificatorias y/o complementarias).

Invocar al método *AbrirComprobanteNoFiscalEnSlip()* es equivalente a invocar al método *AbrirComprobanteNoFiscal()* indicando como argumento ESTACION_SLIP.

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- La existencia de ESTACION_SLIP depende del modelo y versión de firmware fiscal de la **IFH 1G** en uso.
- Cada cuatro líneas de impresión, por exigencia de la normativa fiscal vigente, se imprime la leyenda “*** NO FISCAL ***”. No hay mecanismo que inhabilite la impresión de la leyenda mencionada.
- Los comprobantes no fiscales no se numeran.

Método: AbrirComprobanteNoFiscalHomologado()

Comando: *OpenDNFH*

Disponible: **OCX Fiscal HASAR v010724**

Argumentos: 2

Los métodos *AbrirComprobanteNoFiscalHomologado()* y *AbrirDNFH()* poseen los mismos argumentos y se emplean con el mismo fin.

Ver método *AbrirDNFH()*.

Método: AbrirDNFH()

Comando: *OpenDNFH*

Disponible: **OCX Fiscal HASAR v051122, y posteriores**

Argumentos: 2

La explicación de este método también es válida para el método *AbrirComprobanteNoFiscalHomologado()*.

AbrirDNFH tipo, nro

tipo	De entrada. Obligatorio. Tipo: enum <i>DocumentosNoFiscales</i> Permite al software de gestión para puntos de venta especificar el tipo de comprobante no fiscal homologado que se desea abrir. Los comprobantes no fiscales homologados posibles de ser abiertos, se identifican mediante las siguientes constantes embebidas en el OCX Fiscal HASAR : Impresión en rollo: TICKET_NOTA_CREDITO_A, TICKET_NOTA_CREDITO_B, TICKET_RECIBO_X, TICKET_PAGARE, TICKET_PRESUPUESTO, TICKET_ENTREGA, TICKET_ESTACIONAMIENTO, TICKET_COBRO_SERVICIO, TICKET_INGRESO_DINERO, TICKET_RETIRO_DINERO, TICKET_TALON_CAMBIO, TICKET_TALON_REPARTO, TICKET_TALON_REGALO, TICKET_TALON CUENTA_CORRIENTE, TICKET_AVISO_OPERACION_CREDITO, TICKET_CUPON_PROMOCION, TICKET_USO_INTERNO_FARMACIA.
-------------	--

	<p>Impresión en formulario continuo u hojas sueltas:</p> <p>NOTA_CREDITO_A, NOTA_CREDITO_B, REMITO, RECIBO_X, ORDEN_SALIDA, RESUMEN CUENTA, CARGO_HABITACION, COTIZACION, CLAUSULA_CREDITO, CLAUSULA_SEGURO, PAGARE, POLIZA_SEGURO, RECORDATORIO, SOLICITUD_CREDITO, COMUNICACION_CLIENTE, OFRECIMIENTO_CREDITO, OFRECIMIENTO_TARJETA, MINUTA_CREDITO, OFRECIMIENTO_PASAPORTE, RENOVACION_CREDITO, ADELANTO_Remuneracion, SOLICITUD_TARJETA_DEBITO, SOLICITUD_CLAVE_TARJETA, RESCATE_MERCADERIA, INGRESO_EGRESO_SUCURSAL.</p>
nro	<p>De entrada. Opcional. Tipo: <i>Variant</i></p> <p>La numeración de Tiques Nota de Crédito, Notas de Crédito, y Remitos es responsabilidad de la IFH 1G (por exigencia de la normativa fiscal vigente); por lo tanto, este argumento se puede omitir en estos casos. Para el resto de los comprobantes no fiscales homologados, posibles de ser abiertos, quien debe controlar su numeración es el software de gestión para puntos de venta. En estos casos el argumento es obligatorio.</p>

Ejemplo de uso:

```
...
HASAR1.DocumentoDeReferencia(1) = "FA 0001-00000876"
HASAR1.DatosCliente "Razón Social...", "99999999995", TIPO_CUIT, RESPONSABLE_INSCRIPTO, "Domicilio..."
HASAR1.AbrirDNFH NOTA_CREDITO_A
...
```

En el ejemplo anterior se abre una Nota de Crédito 'A' a un contribuyente Responsable Inscripto.

El desarrollador del software de gestión para puntos de venta de tener presente que:

- No todo los tipos de comprobantes no fiscales homologados mencionados se encuentran disponibles en todos los modelos y versiones de firmware fiscal de **IFH 1G**.
- En el caso de Notas de crédito es obligatorio, y previo, el envío de los datos del cliente a la **IFH 1G** (ver método *DatosCliente()*), y el envío de la referencia al comprobante que dio origen al actual documento (ver propiedad *DocumentoDeReferencia()*).
- Las Notas de Crédito "A" se pueden abrir siempre que el receptor y el emisor sean -ambos- contribuyentes Responsables Inscriptos. En cualquier otro caso, se debe abrir una Nota de Crédito "B".
- Cuando se abre una Nota de crédito "B" se imprime la letra 'B' sólo en el caso de que el emisor sea un contribuyente Responsable inscripto. Si el emisor posee otra categoría de contribuyente, se imprime la letra 'C'.
- En *Modo Entrenamiento*, a exigencias de la normativa fiscal vigente, la impresión de los comprobantes no fiscales homologados presentan el siguiente aspecto:
 - Se imprime la leyenda "***No Fiscal ***" cada cuatro líneas de impresión.
 - No se imprime el logo fiscal *CF DGI*.

- En las líneas que contienen al menos un carácter imprimible, los espacios son reemplazados por el carácter de interrogación ('?').
- En el caso de impresión de Notas de Crédito sobre formularios continuos u hojas sueltas, de requerirse más de una hoja, resulta transparente para el software de gestión para puntos de venta el transporte a la hoja siguiente.
- En el caso de realizarse transporte a la hoja siguiente, la numeración de la nueva hoja se incrementa en uno.
- El número de CAI y la fecha de vencimiento, a exigencias de la normativa fiscal vigente, se imprime solamente en Notas de Crédito "A".

*El número de CAI impreso en las Notas de Crédito "A" emitidos por la **IFH 1G** no se verifica en el sitio oficial de AFIP. Es válido por haber sido emitido con una impresora fiscal homologada.*

Los Remitos llevan la letra 'X' (no la 'R') porque así lo exige la RG AFIP Nº 259/98, Anexo II; y también según se aclara en la RG AFIP Nº 1415/03, Artículo 28.

Método: ApagarIndicadorDeTambor()

Comando: *SetConfigPage*

Disponible: **OCX Fiscal HASAR**, v051122 y posteriores.

Argumentos: No requiere.

Este método permite al software de gestión para puntos de venta apagar el LED de aviso de recambio de tambor. Su uso es válido en **IFH 1G** láser (SMH/PL-23F solamente).

Ejemplo de uso:

```
...
HASAR1.ApagarIndicadorDeTambor
...
```

Método: AutodetectarControlador()

Comando: *StatusRequest*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 1

AbrirDNFH puerto

puerto	De entrada. Optativo OCX Fiscal HASAR v051122 y posteriores. Obligatorio en v010724 . Tipo: <i>Variant</i>
---------------	--

Este método permite al software de gestión para puntos de venta determinar la velocidad (en baudios) a la que se encuentra trabajando el puerto serie de la **IFH 1G**, para así poder dialogar por ella; ya sea que esté conectada al puerto serie del host, o a un puerto serie de un conversor Ethernet/RS-232.

- RS-232
Identifica al puerto serie (COM1, COM2, etc.).
- Ethernet
Identifica al socket correspondiente al puerto serie (COM n) incluido en un conversor Ethernet/RS-232, al cual se conecta la **IFH 1G**.

En el **OCX Fiscal HASAR v010724** solamente es posible indicar un puerto RS-232.

En la versión **v051122** (y posteriores) la interpretación del valor del argumento **puerto** está asociada a la propiedad *Transporte()* del **OCX Fiscal HASAR**.

Si el argumento **puerto** se omite -sólo **OCX Fiscal HASAR v051122** (y posteriores)-, el método *AutodetectarControlador()* asume el valor configurado en la propiedad *Puerto()*.

Ejemplo de uso:

```
...
HASAR1.Puerto = 2
HASAR1.Transport = PUERTO_SERIE
...
HASAR1.AutodetectarControlador
...
```

En el ejemplo anterior se verifica si hay conexión con la **IFH 1G** a través del COM2, y a qué velocidad se estableció el diálogo.

Por defecto, las **IFH 1G** salen de fábrica a 9600 baudios.

No todos los modelos de **IFH 1G** soportan comunicación a más de una velocidad.

La verificación de conexión se realiza a todas las velocidades posibles sobre el **puerto** indicado por el software de gestión para puntos de venta.

Este método del **OCX Fiscal HASAR** genera y envía a la **IFH 1G** el string del comando *StatusRequest*, el cual es inocuo desde el punto de vista de impresión de comprobantes, acumulaciones de montos, y/o control de numeradores; pero permite determinar la existencia de comunicación entre el software de gestión para puntos de venta, y la **IFH 1G**.

El envío del comando *StatusRequest* es transparente para el software de gestión para puntos de venta, salvo por la excepción que se genera si la **IFH 1G** no es detectada.

Cada intento del método *AutodetectarControlador()* por establecer conexión con la **IFH 1G** a una nueva velocidad es equivalente a invocar al método *Comenzar()*. En cada uno de dichos intentos el **OCX Fiscal HASAR** dispara el evento *ProgresoDetección()*.

Si el método *AutodetectarControlador()* no fracasa, la propiedad *Baudios()* quedará configurada con la velo-

ciudad de puerto serie a la cual fue detectada. En caso contrario se genera una excepción.

Si se invoca al método *AutodetectarControlador()* no se requiere invocar al método *Comenzar()*.

Método: AutodetectarModelo()

Comando: *GetPrinterVersion*

Disponible: **OCX Fiscal HASAR, v051122** y posteriores

Argumentos: No requiere

Este método permite al software de gestión para puntos de venta determinar -de ser ello posible- el modelo y versión de firmware fiscal de la **IFH 1G** con la que se ha establecido conexión. Tener presente que no todos los modelos de **IFH 1G** soportan el comando *GetPrinterVersion*.

Ejemplo de uso:

```
...
HASAR1.Puerto = 2
HASAR1.Transporte = PUERTO_SERIE
...
HASAR1.AutodetectarControlador
HASAR1.AutodetectarModelo
...
```

El método *AutodetectarModelo()* debe ser invocado con posterioridad a la finalización exitosa del método *AutodetectarControlador()*.

Determinado el modelo y versión de firmware fiscal de la **IFH 1G**, el **OCX Fiscal HASAR** configura automáticamente la propiedad *Modelo()* con la constante embebida adecuada.

Si el modelo de **IFH 1G** no pudo ser identificado por el método *AutodetectarModelo()* -se genera una excepción-, el valor de la propiedad *Modelo()* será indeterminado, y deberá ser modificado por el software de gestión para puntos de venta.

Método: AvanzarPapel()

Comando: *FeedReceipt, FeedJournal, FeedReceiptJournal*

El comando a generar depende del argumento que especifica el papel a avanzar.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 2

Este método permite al software de gestión para puntos de venta realizar avances de papel cuando necesite dejar líneas en blanco.

AvanzarPapel papel, líneas

papel | De entrada. Obligatorio.

	<p>Tipo: enum <i>TiposDePapel</i></p> <p>Permite al software de gestión para puntos de venta indicar si se avanza papel en la estación de impresión de comprobantes para el cliente, en la estación de impresión de la cinta testigo, o sobre ambas estaciones de impresión.</p> <p>Los valores posibles para este argumento son los identificados por las siguientes constantes embedidas en el OCX Fiscal HASAR:</p> <p>PAPEL_DIARIO, PAPEL_TICKET, PAPEL_TICKET_Y_DIARIO</p>
Líneas	<p>De entrada. Obligatorio.</p> <p>Tipo: <i>Long</i></p> <p>Permite al software de gestión para punto de venta indicar la cantidad de líneas de avance de papel.</p>

Ejemplo de uso:

```
...
HASAR1.AvanzarPapel PAPEL_TICKET, 3
...
```

El método *AvanzarPapel()* no es aplicable sobre todos los comprobantes posibles de emitir, ni está disponible en todos los modelos de **IFH 1G** (por ejemplo, no disponible en los que solamente emiten comprobantes sobre formularios continuos u hojas sueltas -modelos de página completa-).

Método: *BajaFiscal()*

Comando: *KillEepromFiscal*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 1

Permite al software de gestión para puntos de venta dar la baja de la memoria fiscal de la **IFH 1G**.

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 IFH 1G.

Puede ocurrir que en presencia y a pedido de la autoridad fiscal competente, y 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).

BajaFiscal nreg

nreg De entrada. Opcional.

Tipo: *Variant*

En el **OCX Fiscal HASAR v010724** este argumento no existe. Los modelos soportados de **IFH 1G** no requerían esta información.

Permite al software de gestión para puntos de venta indicar el número de registro con el cual fue inicializada la **IFH 1G** (por el técnico fiscal autorizado), y siempre que el string del comando *KillE-promFiscal* requiera esta información.

Ejemplo de uso:

```
...
HASAR1.BajaFiscal "HZ1234567"
...
```

Método: BorrarDocumentoDeReferencia()

Comando: *SetEmbarkNumber*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Argumentos: No requiere.

Este método permite al software de gestión para puntos de venta eliminar de la memoria de trabajo de la **IFH 1G** todas las líneas de textos memorizadas que hacen referencia a comprobantes originales. Estas líneas de texto se supone fueron cargadas mediante la propiedad *DocumentoDeReferencia()* para que se impriman en el encabezado del comprobante que se va a abrir.

Tanto la carga de los textos mediante la propiedad *DocumentoDeReferencia()*, como su borrado mediante el método *BorrarDocumentoDeReferencia()* -si fuese necesario- son operaciones que deben realizarse antes de la apertura de los comprobantes.

Si el software de gestión para puntos de venta necesitare borrar algunas, y no todas, las líneas de texto memorizadas deberá recurrir directamente al uso de la propiedad *DocumentoDeReferencia()*.

Ejemplo de uso:

```
...
HASAR1.DocumentoDeReferencia(1) = "FA 0001-00000987"
HASAR1.DocumentoDeReferencia(2) = "FA 0001-00000988"
...
HASAR1.BorrarDocumentoDeReferencia
```

La siguiente sentencia:

```
HASAR1.BorrarDocumentoDeReferencia
```

Es equivalente a las siguientes sentencias:

```
HASAR1.DocumentoDeReferencia(1) = Chr(127)
HASAR1.DocumentoDeReferencia(2) = Chr(127)
```

Los modelos y versiones de firmware fiscal de las **IFH 1G** homologadas hasta la fecha soportan sólo dos líneas (1 y 2) de texto de referencia a otro comprobante.

Método: BorrarFantasiaEncabezadoCola()

Comando: *SetFantasyName, SetHeaderTrailer*

Uno, o ambos strings de comandos se generan en función de los valores pasados en los argumentos del método, cuando es invocado por el software de gestión para puntos de venta.

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Argumentos: 3

Este método permite al software de gestión para puntos de venta eliminar de la memoria de trabajo de la **IFH 1G** todas las líneas de texto asociadas a nombres de fantasía, encabezado y pie de los comprobantes. Estas líneas de texto se supone fueron cargadas mediante la propiedad *NombreDeFantasia()* y/o *Encabezado()* para que se impriman en el comprobante que se va a abrir.

BorrarFantasiaEncabezadoCola *fantas, encab, pie*

	BorrarFantasiaEncabezadoCola <i>fantas, encab, pie</i>
fantas	<p>De entrada. Obligatorio. Tipo: <i>Boolean</i></p> <p>Permite indicar al software de gestión para puntos de venta si se eliminan de la memoria de trabajo de la IFH 1G todas las líneas de texto de fantasía almacenadas.</p> <p>Los valores posibles para este argumento son:</p> <p><i>True</i> Eliminar todas las líneas (1 y 2). <i>False</i> Ignorar.</p>
encab	<p>De entrada. Obligatorio. Tipo: <i>Boolean</i></p> <p>Permite indicar al software de gestión para puntos de venta si se eliminan de la memoria de trabajo de la IFH 1G todas las líneas de texto de encabezado almacenadas.</p> <p>Los valores posibles para este argumento son:</p> <p><i>True</i> Eliminar todas las líneas (1 a 10). <i>False</i> Ignorar.</p>
pie	<p>De entrada. Obligatorio. Tipo: <i>Boolean</i></p> <p>Permite indicar al software de gestión para puntos de venta si se eliminan de la memoria de trabajo de la IFH 1G todas las líneas de texto de pie almacenadas.</p> <p>Los valores posibles para este argumento son:</p> <p><i>True</i> Eliminar todas las líneas (11 a 20). <i>False</i> Ignorar.</p>

Ejemplo de uso:

...
HASAR1.BorrarFantasiaEncabezadoCola False, False, True
...

La siguiente sentencia:

HASAR1.BorrarFantasiaEncabezadoCola True, False, False

Es equivalente a las siguientes sentencias:

HASAR1.NombreDeFantasia(1) = Chr(127)
HASAR1.NombreDeFantasia(2) = Chr(127)

La siguiente sentencia:

HASAR1.BorrarFantasiaEncabezadoCola False, True, False

Es equivalente a las siguientes sentencias:

HASAR1.Encabezado(1) = Chr(127)
HASAR1.Encabezado(2) = Chr(127)
...
HASAR1.Encabezado(10) = Chr(127)

La siguiente sentencia:

HASAR1.BorrarFantasiaEncabezadoCola False, False, True

Es equivalente a las siguientes sentencias:

HASAR1.Encabezado(11) = Chr(127)
HASAR1.Encabezado(12) = Chr(127)
...
HASAR1.Encabezado(20) = Chr(127)

El desarrollador del software de gestión para puntos de venta debe tener en cuenta, para elegir el punto de borrado de líneas de texto almacenadas en la memoria de trabajo de la **IFH 1G**, que:

- Las líneas de texto de fantasía las imprime el comando de apertura del comprobante, por lo que deben haber sido almacenadas previamente.
- Las líneas de texto de encabezado las imprime el comando de apertura del comprobante, por lo que deben haber sido almacenadas previamente.
- Las líneas de texto de pie las imprime el comando de cierre del comprobante, por lo que deben haber sido almacenadas previamente.

Método: CambiarCódigoIngresosBrutos()

Comando: *ChangeINumber*

Disponible: **OCX Fiscal HASAR v051122** y posteriores

Argumentos: 1

Este método permite al software de gestión para puntos de venta modificar el número de inscripción en Ingresos Brutos correspondiente al contribuyente titular de la **IFH 1G**; información que -originalmente- se encuentra grabada en la memoria fiscal del equipo.

El desarrollador del software de gestión para puntos de venta debe tener presente que no todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la modificación de esta información.

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.

CambiarCodigoIngresosBrutos *codigo*

codigo	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar el nuevo código de inscripción en Ingresos Brutos correspondiente al contribuyente titular de la IFH 1G .
---------------	---

Ejemplo de uso:

```
...
HASAR1.ReporteZ
HASAR1.CambiarCodigoIngresosBrutos "IIBB-0123456789"
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Para poder invocar al método *CambiarCodigoIngresosBrutos()* se requiere, previamente, realizar un cierre diario 'Z'.
- El código de inscripción en Ingresos Brutos, por exigencia de la normativa fiscal vigente, sólo puede modificarse hasta un máximo de 5 veces en la vida útil de la memoria fiscal de la **IFH 1G**.

Método: CambiarFechaInicioActividades()

Comando: *ChangeBusinessStartupDate*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Argumentos: 1

Este método permite al software de gestión para puntos de venta modificar la fecha de inicio de actividades correspondiente al contribuyente titular de la **IFH 1G**; información que -originalmente- se encuentra grabada en la memoria fiscal del equipo.

El desarrollador del software de gestión para puntos de venta debe tener presente que no todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la modificación de esta información.

El software de gestión para puntos de venta debería ofrecer al usuario, mediante estrictos controles de acceso, la posibilidad de modificar su fecha de inicio de actividades, si tal cambio legalmente se produce.

CambiarFechaInicioActividades fecha

fecha	De entrada. Obligatorio. Tipo: <i>Date</i> Permite al software de gestión para puntos de venta indicar la nueva fecha de inicio de actividades correspondiente al contribuyente titular de la IFH 1G .
--------------	---

Ejemplo de uso:

```
...
HASAR1.Reportez
HASAR1.CambiarFechaInicioActividades "01/11/2013"
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Para poder invocar al método *CambiarFechaInicioActividades()* se requiere, previamente, realizar un cierre diario 'Z'.
- La fecha de inicio de actividades, por exigencia de la normativa fiscal vigente, sólo puede modificarse hasta un máximo de 5 veces en la vida útil de la memoria fiscal de la **IFH 1G**.

Método: CambiarResponsabilidadIVA()

Comando: *ChangeIVAResponsibility*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 1

Este método permite al software de gestión para puntos de venta modificar la categoría de contribuyente frente a IVA, correspondiente al titular de la **IFH 1G**; información que -originalmente- se encuentra grabada en la memoria fiscal del equipo.

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

CambiarResponsabilidadIVA categ

categ	De entrada. Obligatorio. Tipo: enum <i>TiposResponsabilidadImpresor</i> Permite al software de gestión para puntos de venta indicar la nueva categoría de contribuyente frente a IVA, correspondiente al titular de la IFH 1G . Los valores admitidos para este argumento son los identificados mediante las siguientes constantes embebidas en el OCX Fiscal HASAR :
--------------	--

CLASE_RESPONSABLE_INSCRIPTO, CLASE_RESPONSABLE_NO_INSCRIPTO,
CLASE_RESPONSABLE_EXENTO, CLASE_NO_RESPONSABLE, CLASE_MONOTRIBUTO,
CLASE_MONOTRIBUTO_SOCIAL.

CLASE_RESPONSABLE_NO_INSCRIPTO: Categoría de contribuyente frente a IVA que ha sido eliminada de la normativa fiscal vigente.

Ejemplo de uso:

```
...
HASAR1.ReporteZ
HASAR1.CambiarResponsabilidadIVA CLASE_MONTRIBUTO
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Para poder invocar al método *CambiarResponsabilidadIVA()* se requiere, previamente, realizar un cierre diario 'Z'.
- La categoría de contribuyente frente a IVA, por exigencia de la normativa fiscal vigente, sólo puede modificarse hasta un máximo de 5 veces en la vida útil de la **IFH 1G**.

Método: CambiarVelocidad()

Comando: *SetComSpeed*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos. 1

Permite al software de gestión para puntos de venta modificar la velocidad (en baudios) a la que se encuentra configurado el puerto serie de la **IFH 1G**.

El desarrollador del software de gestión para puntos de venta debe tener presente que no todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan distintas velocidades de configuración de su puerto serie.

CambiarVelocidad veloc

veloc	De entrada. Obligatorio. Tipo: <i>Long</i> Permite al software de gestión para puntos de venta indicar la nueva velocidad (en baudios) para configuración del puerto serie de la IFH 1G . Los valores admitidos son: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.
--------------	---

Ejemplo de uso:

...
HASAR1.CambiarVelocidad 115200
HASAR1.Finalizar
HASAR1.Baudios = 115200
HASAR1.Comenzar
...

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- La configuración por defecto del puerto serie, en las **IFH 1G**, es de 9600 baudios.
- El comando generado por el método *CambiarVelocidad()* causa la reconfiguración del puerto serie de la **IFH 1G**, mientras que la propiedad *Baudios()*, también del **OCX Fiscal HASAR**, reconfigura el puerto serie del host.
- Al modificar la configuración del puerto serie de la **IFH 1G** se hace necesaria una secuencia de pasos posteriores, debido a que ya no coincidirán las velocidades configuradas en el host y en la **IFH 1G**. De no hacerlo, no habrá conexión entre ambas partes.

Se hace necesario entonces:

- Cerrar el puerto serie del host, invocando al método *Finalizar()*.
- Modificar la configuración puerto serie del host mediante la propiedad *Baudios()*, especificando el mismo valor de configuración del puerto serie de la **IFH 1G**.

Método: CancelarComprobante()

Comando: *Cancel*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: No requiere.

Este método permite al software de gestión para puntos de venta cancelar cualquier comprobante en curso de emisión.

Ejemplo de uso:

...
HASAR1.CancelarComprobante
...

Se entiende por cancelación del comprobante a la interrupción de la emisión en curso -de ser posible-, y la no acumulación de valores (de haber correspondido). Una vez que el comprobante ha sido cerrado no puede ser cancelado (deberá emitirse una Nota de Crédito si ha cerrado una factura, por ejemplo).

Si el comprobante cancelado fuese alguno de los que son numerados por la **IFH 1G** (comprobantes fiscales, Notas de Crédito, Tiques Nota de Crédito, o Remitos), será considerado por ésta que dicho número ha sido utilizado (se incrementará en uno para el próximo comprobante) -exigencia de la normativa fiscal vigente-.

El desarrollador del software de gestión para puntos de venta debe tener presente que no en todos los modelos y versiones de firmware fiscal de las **IFH 1G** es posible cancelar un comprobante fiscal, o una Nota de

Crédito, con posterioridad al envío del comando de pago -método *ImprimirPago()* del **OCX Fiscal HASAR**.

Método: CancelarComprobanteFiscal()

Comando: *Cancel, TotalTender*

El comando a generar depende del modelo de **IFH 1G** al que será enviado.

Disponibilidad: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: No requiere.

Este método permite al software de gestión para puntos de venta cancelar un comprobante fiscal en curso de emisión.

Ejemplo de uso:

```
...
HASAR1.CancelarComprobanteFiscal
...
```

Se entiende por cancelación del comprobante a la interrupción de la emisión en curso -de ser posible-, y la no acumulación de valores (de haber correspondido). Una vez que el comprobante ha sido cerrado no puede ser cancelado (deberá emitirse una Nota de Crédito).

El número del comprobante fiscal cancelado será considerado por la **IFH 1G** como utilizado (se incrementará en uno para el próximo comprobante fiscal) -exigencia de la normativa fiscal vigente-.

El desarrollador del software de gestión para puntos de venta debe tener presente que no en todos los modelos y versiones de firmware fiscal de las **IFH 1G** es posible cancelar un comprobante fiscal con posterioridad al envío del comando de pago -método *ImprimirPago()* del **OCX Fiscal HASAR**-.

Método: Capacidades()

Este método no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Argumentos: 3

Retorno: *Boolean*

Permite al software de gestión para puntos de venta conocer si determinada característica se encuentra disponible en la **IFH 1G** que se encuentra identificada mediante la propiedad *Modelo()*.

Capacidades capac, subcap, resul

capac	De entrada. Obligatorio. Tipo: enum <i>CapacidadesSoportadas</i> Permite al software de gestión para puntos de venta indicar la característica de la IFH 1G sobre la cual se desea hacer la consulta. Los valores posibles de este argumento son los identificados por las siguientes constantes embedidas en el OCX Fiscal HASAR : CAPACIDAD_DOCUMENTO_SLIP, CAPACIDAD_DOCUMENTO_TICKET, CAPACIDAD_ESTACION,
-------	---

	CAPACIDAD_CAJON, CAPACIDAD_TEXTO_FISCAL, CAPACIDAD_TEXTO_VENTA, CAPACIDAD_TEXTO_NO_FISCAL, CAPACIDAD_TEXTO_ENCABEZADO_COLA, CAPACIDAD_DF_SOPORTADO, CAPACIDAD_DNFH_SOPORTADO.
subcap	<p>De entrada: Opcional. Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta indicar, para cada capacidad -de correspon-der-, la subcapacidad asociada . El par capac/subcap define la característica de la IFH 1G sobre la cual se desea hacer la consulta.</p>
resul	<p>De salida. Opcional. Tipo: <i>Variant</i></p> <p>En los casos que el método retorna el valor <i>True</i>, este argumento permite al software de gestión para puntos de venta obtener información relativa a la capacidad (y subcapacidad, si así fuese) respecto de la cual se ha hecho la consulta.</p>

Subcapacidades asociadas

Para las capacidades ya mencionadas se indican a continuación las constantes embebidas del **OCX Fiscal HASAR** asociadas a ellas.

- CAPACIDAD_DOCUMENTO_SLIP, CAPACIDAD_DOCUMENTO_TICKET
Subcapacidades: enum *TiposDeDocumentoSoportado*

Ejemplo de uso:

```
...
If (HASAR1.Capacidades(CAPACIDAD_DOCUMENTO_TICKET, SOPORTA_FACTURA)) then
    Debug.Print "Soporta Tiques Factura..."
End If
...
```

- CAPACIDAD_CAJON
Subcapacidades: No tiene

Ejemplo de uso:

```
...
If (HASAR1.Capacidades(CAPACIDAD_CAJON)) then
    Debug.Print "Soporta Cajón de Dinero..."
End If
...
```

- CAPACIDAD_DF_SOPORTADO
Subcapacidades: enum *DocumentosFiscales*

Ejemplo de uso:

```
...
If (HASAR1.Capacidades(CAPACIDAD_DF_SOPORTADO, TICKET_FACTURA_A)) then
    Debug.Print "Soporta Tique Factura A..."
```

- End If
- ...
- CAPACIDAD_DNFH_SOPORTADO
Subcapacidades: enum *DocumentosNoFiscales*

Ejemplo de uso:

```
...
If (HASAR1.Capacidades(CAPACIDAD_DF_SOPORTADO, TICKET_NOTA_CREDITO_A)) then
    Debug.Print "Soporta Tique Nota de Crédito A..."
End If
...
```
- CAPACIDAD_ESTACION
Subcapacidades: enum *TiposDeEstacion*

Ejemplo de uso:

```
...
If (HASAR1.Capacidades(CAPACIDAD_ESTACION, ESTACION_SLIP)) then
    Debug.Print "Soporta impresión en formulario continuo..."
End If
...
```
- CAPACIDAD_TEXTO_ENCABEZADO_COLA, CAPACIDAD_TEXTO_FISCAL, CAPACIDAD_TEXTO_NO_FISCAL,
CAPACIDAD_TEXTO_VENTA
Subcapacidades: enum *TiposDeEstacion*

Ejemplo de uso:

```
...
Dim resul As variant

If (HASAR1.Capacidades(CAPACIDAD_TEXTO_VENTA, ESTACION_SLIP, resul)) then
    Debug.Print "Descripción ítem = " & resul & " caracteres"
End If
...
```

Método: CapacidadRestante()

Comando: *HistoryCapacity*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 2

En el **OCX Fiscal HASAR v010724** este método no posee argumentos.
Se debe recurrir al empleo de la propiedad *Respuesta()*.

Este método permite al software de gestión para puntos de venta determinar la cantidad máxima de cierres diarios 'Z' (registros) que es posible realizar durante la vida útil de la memoria fiscal de la **IFH 1G**; además de saber que número le corresponde al último cierre diario 'Z' realizado.

CapacidadRestante cantz, ultz

cantz | De salida. Opcional.

	<p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener la cantidad máxima de cierres diarios 'Z' que admite (como registros) la memoria fiscal de la IFH 1G en uso.</p>
ultz	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener la cantidad de cierres diarios 'Z' realizados hasta el momento. Es equivalente a decir que se obtiene el número del último cierre diario 'Z' realizado.</p>

Ejemplo de uso:

```
...
Dim cantz As variant
Dim ultz As variant

HASAR1.CapacidadRestante cantz, ultz
Debug.Print "Capacidad = " & cantz & " cierres Z" & vbCrLf & "Nro. Últ. Z = " & ultz
...
```

La diferencia entre **cantz** y **ultz** permite establecer la cantidad de cierres diarios 'Z' faltantes para que se agote la capacidad de la memoria fiscal de la **IFH 1G**.

Si bien éste parece ser un dato obvio, debe ser tenido en cuenta por el desarrollador del software de gestión para puntos de venta, debido a que:

- Agotada la capacidad máxima de la memoria fiscal, la **IFH 1G** inhabilita la emisión de comprobantes (a excepción de reportes de auditoría y permitir consultas de estado) hasta tanto el equipo sea dado de baja.
- El procedimiento de baja y recambio de la memoria fiscal (alta de la nueva memoria fiscal) es de orden técnico y legal. Durante el período de tiempo que transcurre entre la baja de la memoria fiscal actual y el alta de la nueva memoria fiscal la **IFH 1G** no puede ser utilizada, debiéndose recurrir a otro punto de venta declarado ante AFIP -que puede ser talonario preimpreso-).

Debido a lo anterior, es importante comenzar el proceso de recambio de memoria fiscal con antelación al agotamiento de la capacidad de la memoria fiscal de la **IFH 1G**.

Al margen de los controles que instrumenta el software de gestión para puntos de venta, la **IFH 1G** da aviso de memoria fiscal a punto de agotarse, treinta (30) cierres diarios 'Z' antes de que esto ocurra (y durante las jornadas comerciales posteriores).

El aviso de memoria fiscal a punto de agotarse es dado de dos formas:

- El **OCX Fiscal HASAR** genera el evento *EventoFiscal()*. Esto ocurre a partir de la información contenida en el campo de estado fiscal de la respuesta a un comando enviado a la **IFH 1G** por el software de gestión para puntos de venta, quien debe atrapar y tratar el evento de manera adecuada.
- Cada vez que se emite un cierre diario 'Z' se imprime en la zona del pie la leyenda: "Cierres remantes: nn" (donde *nn* es la cantidad todavía disponible).

El desarrollador del software de gestión para puntos de venta debe tener presente que no todos los modelos de **IFH 1G** poseen la misma capacidad en sus memorias fiscales. Existen homologados modelos que admiten hasta 1800 cierres diarios 'Z' (aproximadamente 5 años de uso), y otros hasta 3800 (aproximadamente 10 años de uso).

Método: CargarLogo()

Comando: *StoreLogo, ResetLogo* (según corresponda)

Disponible: **OCX Fiscal HASAR**, en todas las versiones

Argumentos: 1

Este método permite al software de gestión para puntos de venta cargar en la memoria de trabajo de la **IFH 1G** una imagen proveniente de un archivo en formato 'BMP'. Dicha imagen será impresa como logo del emisor en cada comprobante que se emita –de corresponder-. La carga del logo en la memoria de trabajo requiere el envío del comando *StoreLogo* a la **IFH 1G**.

Mediante este método será posible, también, eliminar dicha imagen de la memoria de trabajo de la **IFH 1G**. La eliminación del logo del emisor requiere el envío del comando *ResetLogo*.

El desarrollador del software de gestión para puntos de venta debe tener presente que no todos los modelos y versiones de firmware fiscal de las **IFH 1G** admiten la impresión de logo en los comprobantes que emiten, y que en los modelos y versiones de firmware fiscal donde ello es posible la impresión del logo del emisor puede no realizarse en todos los tipos de comprobantes que se emitan.

CargarLogo archbmp

archbmp	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar la ubicación y el nombre del archivo 'BMP' que contiene la imagen del logo del emisor a cargar en la memoria de trabajo de la IFH 1G . Si este argumento es un string vacío el OCX Fiscal HASAR asume que la operación a realizar es la eliminación de la memoria de trabajo de la imagen actualmente almacenada del logo del emisor.
----------------	---

Ejemplo de uso:

```
...
HASAR1.ReporteZ
HASAR1.CargarLogo "D:\hasar\logo.bmp"
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Se trate de la carga o eliminación del logo del emisor en/de la memoria de trabajo de la **IFH 1G**, se requiere la realización previa de un cierre diario 'Z'.
- Para que la operación de carga del logo del emisor finalice exitosamente, aun cuando los comandos generados por el método *CargarLogo()* hayan sido aceptados por la **IFH 1G**, el archivo 'BMP' que contiene la imagen debe cumplir con ciertos requisitos, a saber:

- El archivo contenedor de la imagen debe respetar el formato ‘BMP’.
- El atributo del archivo ‘BMP’ debe ser *monocromático*. No se acepta archivo *color*, con la imagen pintada en blanco y negro.
- Se deben respetar las dimensiones especificadas en el *Manual de comandos de la IFH 1G* correspondiente, y en concordancia con el tamaño de papel a utilizar.

Método: CerrarComprobanteFiscal()

Comando: *CloseFiscalReceipt*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 2

En el **OCX Fiscal HASAR v010724** este método no posee argumentos.

Este método permite al software de gestión para puntos de venta cerrar el comprobante fiscal cuya impresión se está realizando.

CerrarComprobanteFiscal copias, nrodoc

copias	<p>De entrada. Opcional. Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta indicar la cantidad de ejemplares que se deberán imprimir del comprobante que se intenta cerrar.</p> <p>Este argumento es válido en aquellos modelos y versiones de firmware fiscal de IFH 1G que admiten esta información en el string del comando <i>CloseFiscalReceipt</i>.</p>
nrodoc	<p>De salida: Opcional. Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener el número del comprobante que se está cerrando.</p>

Ejemplo de uso:

```
...
Dim nrodoc As Variant

HASAR1.CerrarComprobanteFiscal 2, nrodoc
Debug.Print "Se ha cerrado el Documento Nº: " & nrodoc
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El uso del argumento **copias** está relacionado con el parámetro de configuración de la **IFH 1G** asociado a la constante embebida COPIAS_DOCUMENTOS (cantidad de ejemplares que se permitirán imprimir), donde este último tiene prioridad sobre el primero.

Por ejemplo, si el parámetro COPIAS_DOCUMENTOS (que se establece mediante el método *ConfigurarControlador()*) está configurado como '2', y el argumento **copias** vale '3', la **IFH 1G** solo permitirá emitir hasta un duplicado.

- En los modelos y versiones de firmware fiscal de las **IFH 1G** para los cuales no es admisible el uso del argumento **copias**, el software de gestión para puntos de venta deberá recurrir al método *ReimprimirComprobante()* para solicitar los ejemplares adicionales que correspondan.
- En modelos y versiones de firmware fiscal de las **IFH 1G** de página completa (impresión sobre formularios continuos u hojas sueltas), y en modelos y versiones CIF (centros integrales de facturación), el número obtenido a través del argumento **nrodoc** no tiene porque coincidir con el número de apertura del mismo comprobante (puede darse el caso de emisión de documentos que abarquen más de una hoja -hay transportes a la hoja siguiente-).
- La normativa fiscal vigente exige:
 - A las impresoras fiscales incrementar en uno el número de comprobante en cada nueva hoja.
 - No permitir el cierre de comprobantes en \$ 0.00 (pesos cero), o con subtotales o impuestos negativos.

Método: CerrarComprobanteNoFiscal()

Comando: *CloseNonFiscalReceipt*

Disponible: **OCX Fiscal HASAR**, en todas las versiones

Argumentos: 1

En el **OCX Fiscal HASAR v010724** este método no posee argumentos.

Este método permite al software de gestión para puntos de venta cerrar el comprobante no fiscal cuya impresión se está realizando.

CerrarComprobanteNoFiscal copias, nrodoc

copias	De entrada. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta indicar la cantidad de ejemplares que se deberán imprimir del comprobante que se intenta cerrar. Este argumento es válido en aquellos modelos y versiones de firmware fiscal de IFH 1G que admiten esta información en el string del comando <i>CloseNonFiscalReceipt</i> .
---------------	--

Ejemplo de uso:

```
...
HASAR1.CerrarComprobanteNoFiscal 2
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El uso del argumento **copias** está relacionado con el parámetro de configuración de la **IFH 1G** asocia-

do a la constante embebida COPIAS_DOCUMENTOS (cantidad de ejemplares que se permitirán imprimir), donde este último tiene prioridad sobre el primero.

Por ejemplo, si el parámetro COPIAS_DOCUMENTOS (que se establece mediante el método *ConfigurarControlador()*) está configurado como '2', y el argumento **copias** vale '3', la **IFH 1G** solo permitirá emitir hasta un duplicado.

- En los modelos y versiones de firmware fiscal de **IFH 1G** para las cuales no es admisible el uso del argumento **copias**, el software de gestión para puntos de venta deberá recurrir al método *ReimprimirComprobante()* para solicitar los ejemplares adicionales que correspondan.

Método: CerrarComprobanteNoFiscalHomologado()

Comando: *CloseDNFH*

Disponible: **OCX Fiscal HASAR v010724**

Argumentos: 2

Los métodos *CerrarComprobanteNoFiscalHomologado()* y *CerrarDNFH()* poseen los mismos argumentos y se emplean con el mismo fin.

Ver método *CerrarDNFH()*.

Método: CerrarDNFH()

Comando: *CloseDNFH*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Argumentos: 2

Este método permite al software de gestión para puntos de venta cerrar el comprobante no fiscal homologado cuya impresión se está realizando.

CerrarDNFH copias, nrodoc

copias	De entrada. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta indicar la cantidad de ejemplares que se deberán imprimir del comprobante que se intenta cerrar. Este argumento es válido en aquellos modelos y versiones de firmware fiscal de IFH 1G que admiten esta información en el string del comando <i>CloseDNFH</i> .
nrodoc	De salida: Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el número del comprobante que se está cerrando.

Ejemplo de uso:

```

...
Dim nrodoc As Variant

HASAR1.CerrarDNFH 2, nrodoc
Debug.Print "Se ha cerrado el Documento Nº: " & nrodoc
...

```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El uso del argumento **copias** está relacionado con el parámetro de configuración de la **IFH 1G** asociado a la constante embebida COPIAS_DOCUMENTOS (cantidad de ejemplares que se permitirán imprimir), donde este último tiene prioridad sobre el primero.

Por ejemplo, si el parámetro COPIAS_DOCUMENTOS (que se establece mediante el método *ConfigurarControlador()*) está configurado como '2', y el argumento **copias** vale '3', la **IFH 1G** solo permitirá emitir hasta un duplicado.

- En los modelos y versiones de firmware fiscal de **IFH 1G** para las cuales no es admisible el uso del argumento **copias**, el software de gestión para puntos de venta deberá recurrir al método *ReimprimirComprobante()* para solicitar los ejemplares adicionales que correspondan.
- En modelos y versiones de firmware fiscal de **IFH 1G** de página completa (impresión sobre formularios continuos u hojas sueltas), y en modelos y versiones CIF (centros integrales de facturación), el número obtenido a través del argumento **nrodoc** no tiene porque coincidir con el número de apertura del mismo comprobante (puede darse el caso de emisión de documentos que abarquen más de una hoja -hay transportes a la hoja siguiente-). Esto vale para Notas de Crédito.
- La normativa fiscal vigente exige, en el caso de Notas de Crédito:
 - A las impresoras fiscales incrementar en uno el número de comprobante en cada nueva hoja.
 - No permitir el cierre de comprobantes en \$ 0.00 (pesos cero), o con subtotales o impuestos negativos.

Método: Comenzar()

Comando: *StatusRequest*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: No requiere.

Este método permite al software de gestión para puntos de venta iniciar una sesión de trabajo estableciendo (de ser posible) las comunicaciones a través del vínculo configurado en la propiedad *Transporte()* (versión v051122 y posteriores) del **OCX Fiscal HASAR**-, o a través del puerto serie local (todas las versiones).

Internamente el método *Comenzar()* emplea por defecto:

- *Conección por puerto serie*
Valores configurados en las propiedades *Transporte()* -de corresponder-, *Puerto()* y *Baudios()* del **OCX Fiscal HASAR**.
- *Conección por Ethernet*

Valores configurados de las propiedades *Transporte()*, *Puerto()* y *DireccionIP()* del **OCX Fiscal HASAR**.

Configuradas apropiadamente, por el software de gestión para puntos de venta, las propiedades asociadas al vínculo a emplear para dialogar con la **IFH 1G**, al ser invocado, el método *Comenzar()* envía de forma transparente dos consultas de estado (comando *StatusRequest*) para verificar que la conexión se ha establecido con normalidad.

Ejemplos de uso:

- Conexión por puerto serie:

```
...
HASAR1.Transport = PUERTO_SERIE
HASAR1.Puerto = 1
HASAR1.Baudios = 9600
HASAR1.Comenzar
...
```

- Conexión por Ethernet:

```
...
HASAR1.Transport = SOCKET_TCP
HASAR1.Puerto = 5555
HASAR1.DireccionIP = "192.0.5.120"
HASAR1.Comenzar
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- En el caso de conexión por Ethernet, la **IFH 1G** debe estar conectada al puerto serie de un conversor Ethernet/RS-232. Luego, la propiedad *Transporte()* solamente puede tomar los valores identificados por las constantes embebidas en el **OCX Fiscal HASAR**: *SOCKET_TCP* o *SOCKET_UDP*.

Además, la propiedad *DirecciónIP()* identifica al conversor Ethernet/RS-232 como nodo de la red, y la propiedad *Puerto()* es el número de socket con el cual se identifica al puerto serie del conversor Ethernet/RS-232 al cual está conectada la **IFH 1G**.

- Cualquiera sea la versión de **OCX Fiscal HASAR**, el software de gestión para puntos de venta puede optar por el uso del método *AutodetectarControlador()* -en lugar del método *Comenzar()*-.
- Cualquiera sea la versión del **OCX Fiscal HASAR**, considerar el valor de configuración de la propiedad *ReintentoConstante()* ya que si se encuentra configurada como *False*, ante una falta de comunicación se genera una excepción. En cambio, si se encuentra configurada como *True*, se generará el evento *ImpresoraNoResponde()*.
- Si se invoca nuevamente al método *Comenzar()*, sin haber invocado previamente al método *Finalizar()*, en el **OCX Fiscal HASAR v010724** se generará una excepción. En versiones posteriores, el incidente se resuelve internamente sin generar una excepción.
- Si se invoca al método *Comenzar()* con posterioridad al método *AutodetectarControlador()*, sin haber invocado previamente al método *Finalizar()*, en el **OCX Fiscal HASAR v010724** se generará una excepción. En versiones posteriores, el incidente se resuelve internamente sin generar una excepción.

- Si se invoca al método *ReComenzar()* con posterioridad al método *Comenzar()*, sin haber invocado previamente al método *Finalizar()*, en el **OCX Fiscal HASAR v010724** el incidente se resuelve internamente sin generar una excepción. En versiones posteriores del **OCX Fiscal HASAR**, el método *Recomenzar()* se encuentra deprecado.

Método: CortarComprobanteNoFiscal()

Comando: *CutNonFiscalReceipt*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: No requiere.

Este método permite al software de gestión para puntos de venta el corte de papel cuando se está emitiendo un comprobante no fiscal.

Ejemplo de uso:

```
...
HASAR1.CortarComprobanteNoFiscal
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Tiene sentido hablar de corte de papel cuando la impresión es sobre rollo.
- No todos los modelos y versiones de firmware fiscal de **IFH 1G** soportan el comando *CutNonFiscalReceipt*. Generalmente se trata de equipos discontinuados de producción, y que resultan ser de los primeros productos homologados por *Compañía HASAR SAIC*.

Método: ConfigurarControlador()

Comando: *ConfigureControllerByOne*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 2

Este método permite al software de gestión para puntos de venta modificar parámetros de comportamiento (en forma individual) de la **IFH 1G** durante la jornada fiscal.

El software de gestión para puntos de venta debería ofrecer estas opciones de configuración al usuario final, teniendo en cuenta el modelo de IFH 1G en uso.

ConfigurarControlador param, valor

param	De entrada. Obligatorio. Tipo: enum <i>ParametrosDeConfiguracion</i> Permite al software de gestión para puntos de venta indicar el parámetro de comportamiento
--------------	---

	<p>que se desea modificar.</p> <p>El valor que puede tomar este argumento debe ser alguno de los identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR:</p> <p>IMPRESION_CAMBIO, IMPRESIONLEYENDAS, CORTE_PAPEL, IMPRESION_MARCO, REIMPRESION_CANCELADOS, COPIAS_DOCUMENTOS, PAGO_SALDO, SONIDO_AVISO, ALTO_HOJA, ANCHO_HOJA, ESTACION_REPORTES_XZ, MODO_IMPRESION, CHEQUEO_DESBORDE, CHEQUEO_TAPA_ABIERTA.</p>
valor	<p>De entrada. Obligatorio.</p> <p>Tipo: <i>String</i></p> <p>Permite al software de gestión para puntos de venta indicar el nuevo valor a asignar al parámetro de comportamiento param.</p> <p>El valor que puede tomar este argumento debe pertenecer al conjunto de valores posibles y asociados al argumento param.</p>

Ejemplo de uso:

```
...
HASAR1.ConfigurarControlador COPIAS_DOCUMENTOS, "3"
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que casi la totalidad de los parámetros de comportamiento de la **IFH 1G** permiten invocar al método *ConfigurarControlador()* sin que resulte necesario efectuar un cierre diario ‘Z’ (invocar al método *ReporteZ()*). Otros requieren estar fuera de todo comprobante para proceder al cambio.

Valores asociados a los parámetros de comportamiento

IMPRESION_CAMBIO

Este parámetro de comportamiento permite, al software de gestión para puntos de venta, decidir si la leyenda “CAMBIO \$0.00” se imprime en el comprobante cuando la suma de los pagos realizados coincide exactamente con el total del comprobante que se está cerrando.

Tener presente que esta leyenda, en las condiciones dadas, no se imprime en todos los tipos de comprobante posibles de emitir con la **IFH 1G**, o en todos los modelos homologados.

‘P’	Imprimir leyenda, de corresponder.
Otro carácter	No imprimir.

IMPRESIÓNLEYENDAS

Este parámetro de comportamiento permite, al software de gestión para puntos de venta, decidir si las leyendas opcionales (número de inscripción en Ingresos Brutos, leyenda “A consumidor Final”, y la categoría de contribuyente frente a IVA del emisor) se imprimen en los encabezados de los comprobantes que se emitan.

La configuración de este parámetro de comportamiento tiene sentido en **IFH 1G** que permitan la emisión de comprobantes sobre rollos. La omisión de la impresión de las leyendas opcionales sólo debe configurarse

cuando el rollo se encuentre preimpreso al dorso y contenga preimpresos los datos que serán omitidos al frente.

'P'	Imprimir leyendas optionales.
Otro caracter	No imprimir.

CORTE_PAPEL

Este parámetro de comportamiento permite, al software de gestión para puntos de venta, siempre que la **IFH 1G** lo admita, especificar el tipo de corte de papel a realizar al finalizar la emisión de un comprobante.

Los valores posibles son los identificados por la siguientes constantes embebidas en el **OCX Fiscal HASAR**: CORTE_PARCIAL, CORTE_TOTAL, NO_CORTE.

IMPRESION_MARCO

Este parámetro de comportamiento permite, al software de gestión para puntos de venta, decidir si el marco de los comprobantes será impreso por la **IFH 1G**.

La modificación de este parámetro tiene sentido en impresiones sobre formularios continuos u hojas sueltas

'P'	Imprimir el marco.
Otro caracter	No imprimir.

REIMPRIMIR_CANCELADOS

Este parámetro de comportamiento permite, al software de gestión para puntos de venta, decidir si se reimprimen los comprobantes cancelados por corte de energía.

No todos los modelos y versiones de firmware fiscal de **IFH 1G** admiten la modificación de este parámetro de comportamiento.

'P'	Reimprimir comprobante cancelado por corte de energía.
Otro caracter	No reimprimir.

COPIAS_DOCUMENTOS

Este parámetro de comportamiento permite, al software de gestión para puntos de venta, establecer la cantidad máxima de ejemplares que se podrán imprimir de un mismo comprobante, en el momento de la emisión del original.

No todos los modelos y versiones de firmware fiscal de **IFH 1G** admiten la modificación de este parámetro de comportamiento.

Las **IFH 1G** imprimen automáticamente, a exigencias de la normativa fiscal vigente:

- Comprobantes fiscales y Notas de Crédito:
Original y Duplicado.
- Otros documentos:
Solamente el original.

Este comportamiento de la **IFH 1G** no se ve afectado por la modificación del parámetro **COPIAS_DOCUMENTOS**. El software de gestión para puntos de venta deberá solicitar, a continuación del cierre del

comprobante, la impresión de los ejemplares no impresos automáticamente (se debe invocar al método *ReimprimirComprobante()*).

Los valores posibles para este parámetro de comportamiento, son: '0', '1', '2', '3', y '4'.

El valor '0' implica la emisión de un único ejemplar. Se asume el uso de papel copiativo. Por el momento, los valores '1' y '2' son equivalentes: se permite emitir hasta duplicados.

PAGO_SALDO

Este parámetro de comportamiento permite, al software de gestión para puntos de venta, indicar el texto que se imprimirá por defecto como descripción de un medio de pago asociado a la diferencia entre la suma de los pagos realizados y el total del comprobante; si éste se cierra con una suma de pagos inferior al total del comprobante.

A la salida de fábrica el texto por defecto es "Cuenta Corriente".

SONIDO_AVISO / SONIDO

Este parámetro de comportamiento permite, al software de gestión para puntos de venta, indicar si se emitirán pitidos ante la falta de papel.

'P'	Emitir pitidos.
Otro carácter	No emitir sonido alguno.

ALTO_HOJA

Este parámetro de comportamiento permite, al software de gestión para puntos de venta, indicar (si la **IFH 1G** lo admite) el alto de la hoja de papel, o del formulario continuo.

Los valores posibles para este parámetro de comportamiento son los identificados por las siguientes constantes embebidas en el **OCX Fiscal HASAR**:

ALTO_A4, ALTO_CARTA, ALTO_OFICIO, ALTO_REDUCIDO.

ANCHO_HOJA

Este parámetro de comportamiento permite, al software de gestión para puntos de venta, indicar (si la **IFH 1G** lo admite) el ancho de la hoja de papel, o del formulario continuo.

Los valores posibles para este parámetro de comportamiento son los identificados por las siguientes constantes embebidas en el **OCX Fiscal HASAR**:

ANCHO_NORMAL, ANCHO_REDUCIDO.

ESTACION_REPORTES_XZ

Este parámetro de comportamiento permite, al software de gestión para puntos de venta, indicar la estación de impresión a utilizar por defecto (si la **IFH 1G** tuviese más de una) para la emisión de cierres diarios 'X'/'Z'.

Los valores posibles para este parámetro de comportamiento son los identificados por las siguientes constantes embebidas en el **OCX Fiscal HASAR**:

ESTACION_SLIP (formulario continuo), ESTACION_TICKET (rollo).

MODO_IMPRESION

Este parámetro de comportamiento permite, al software de gestión para puntos de venta, indicar como se efectuará la emisión de comprobantes (siempre que la **IFH 1G** admita esta configuración).

Los valores posibles para este parámetro de comportamiento son los identificados por las siguientes constantes embebidas en el **OCX Fiscal HASAR**:

NO_USO_ESTACION_TICKET (sólo formularios continuos), USO_ESTACION_TICKET (rollo y formularios continuos).

CHEQUEO_DESBORDE

Este parámetro de comportamiento permite, al software de gestión para puntos de venta indicar, si la **IFH 1G** lo admite, si el chequeo de desborde de montos límite para comprobantes se realiza contra cada comando que cause acumulación, o solamente se realiza al cerrar el documento.

'P'	Chequear en todo momento.
Otro caracter	Chequear sólo al cerrar el comprobante.

NOTA: Mediante la RG AFIP Nº 3666/14, vigente desde el 4 de Setiembre de 2014, se suspende temporalmente la aplicación del tope de \$1000 (pesos un mil) como monto máximo para Tiques y comprobantes 'B' a consumidores finales sin nominar.

Por suspensión temporal se entiende: hasta tanto entre en vigencia lo previsto en los párrafos segundo y tercero del Artículo 3º de la RG AFIP Nº 3561/13.

CHEQUEO_TAPA_ABIERTA.

Este parámetro de comportamiento permite, al software de gestión para puntos de venta, indicar si se puede imprimir aunque la tapa (si posee una) de la **IFH 1G** se encuentre abierta.

'P'	Imprimir con tapa abierta.
Otro caracter	No imprimir con tapa abierta.

Método: ConfigurarControladorCompleto()

Comando: *GeneralConfiguration*

Disponible: **OCX Fiscal HASAR**, en todas las versiones

Argumentos: 17

Este método permite al software de gestión para puntos de venta modificar parámetros de comportamiento (en bloque) de la **IFH 1G** durante la jornada fiscal.

Mediante este método es posible modificar, además, parámetros de comportamiento asociados a la normativa fiscal vigente.

El software de gestión para puntos de venta debería ofrecer estas opciones de configuración al usuario final, teniendo en cuenta el modelo de IFH 1G en uso.

ConfigurarControladorCompleto imprimir, defecto, limitecf, limitetf, ivanoi, copias, cambio, leyendas, corte, marco, reimpr, saldo, sonido, alto, ancho, estac, modo

imprimir De entrada. Obligatorio.
Tipo: Boolean

	<p>Este parámetro de comportamiento permite, al software de gestión para puntos de venta, indicar si la IFH 1G debe imprimir un reporte de configuración.</p> <p><i>True</i> Imprimir reporte de configuración. <i>False</i> No imprimirllo.</p>
defecto	<p>De entrada. Obligatorio. Tipo: <i>Boolean</i></p> <p>Este parámetro de comportamiento permite, al software de gestión para puntos de venta, indicar si los parámetros de comportamiento omitidos deberá ser configurados con los valores por defecto a la salida de fábrica.</p> <p><i>True</i> Configurar con valores por defecto. <i>False</i> Pedir configuración de todos los parámetros.</p>
limitecf	<p>De entrada. Opcional. Tipo: <i>Variant</i></p> <p>Este parámetro de comportamiento permite, al software de gestión para puntos de venta, indicar el nuevo valor establecido por la normativa fiscal vigente como monto límite para los comprobantes 'B' a contribuyentes Consumidores Finales, sin datos del comprador (o incompletos).</p> <p>Actualmente, este parámetro de comportamiento debería estar configurado con el valor '1000.00' (un mil pesos).</p> <p>NOTA: Mediante la RG AFIP Nº 3666/14, vigente desde el 4 de Setiembre de 2014, se suspende temporalmente la aplicación del tope de \$1000 (pesos un mil) como monto máximo para Tiques y comprobantes 'B' a consumidores finales sin nominar. Por suspensión temporal se entiende: hasta tanto entre en vigencia lo previsto en los párrafos segundo y tercero del Artículo 3º de la RG AFIP Nº 3561/13.</p>
limitetf	<p>De entrada. Opcional. Tipo: <i>Variant</i></p> <p>Este parámetro de comportamiento permite, al software de gestión para puntos de venta, indicar el nuevo valor establecido por la normativa fiscal vigente como monto límite para los comprobantes tipificados como Tiques Factura y/o Tiques Nota de Crédito/Débito.</p> <p>Actualmente, este parámetro de comportamiento debería estar configurado con el valor '25000.00' (veinticinco mil pesos).</p> <p>NOTA: La normativa fiscal vigente (RG AFIP Nº 3561/13) ha eliminado este tope para los comprobantes mencionados anteriormente.</p>
ivanoi	<p>De entrada. Opcional. Tipo: <i>Variant</i></p> <p>Actualmente en desuso. Siempre en '0.00'.</p>

copias	<p>De entrada. Opcional. Tipo: <i>Variant</i></p>				
	<p>Este parámetro de comportamiento permite, al software de gestión para puntos de venta, establecer la cantidad máxima de ejemplares que se podrán imprimir de un mismo comprobante, en el momento de la emisión del original.</p>				
	<p>No todos los modelos y versiones de firmware fiscal de IFH 1G admiten la modificación de este parámetro de comportamiento.</p>				
	<p>Las IFH 1G imprimen automáticamente, a exigencias de la normativa fiscal vigente:</p>				
	<ul style="list-style-type: none"> ▪ Comprobantes fiscales y Notas de Crédito: Original y Duplicado. ▪ Otros documentos: Solamente el original. 				
	<p>Este comportamiento de la IFH 1G no se ve afectado por la modificación del parámetro COPIAS_DOCUMENTOS. El software de gestión para puntos de venta deberá solicitar, a continuación del cierre del comprobante, la impresión de los ejemplares no impresos automáticamente (se debe invocar al método <i>ReimprimirComprobante()</i>).</p>				
	<p>Los valores posibles para este parámetro de comportamiento, son: '0', '1', '2', '3', y '4'. El valor '0' implica la emisión de un único ejemplar. Se asume el uso de papel copiativo. Por el momento, los valores '1' y '2' son equivalentes: se permite emitir hasta duplicados.</p>				
cambio	<p>De entrada. Opcional. Tipo: <i>Variant</i></p>				
	<p>Este parámetro de comportamiento permite, al software de gestión para puntos de venta, decidir si la leyenda “CAMBIO \$0.00” se imprime en el comprobante cuando la suma de los pagos realizados coincide exactamente con el total del comprobante que se está cerrando.</p>				
	<p>Tener presente que esta leyenda, en las condiciones dadas, no se imprime en todos los tipos de comprobante posibles de emitir con la IFH 1G, o en todos los modelos homologados.</p>				
	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%; padding-bottom: 5px;">‘P’</td> <td style="width: 70%; padding-bottom: 5px;">Imprimir leyenda, de corresponder.</td> </tr> <tr> <td>Otro carácter</td> <td>No imprimirla.</td> </tr> </table>	‘P’	Imprimir leyenda, de corresponder.	Otro carácter	No imprimirla.
‘P’	Imprimir leyenda, de corresponder.				
Otro carácter	No imprimirla.				
leyendas	<p>De entrada. Opcional. Tipo: <i>Variant</i></p>				
	<p>Este parámetro de comportamiento permite, al software de gestión para puntos de venta, decidir si las leyendas optionales (número de inscripción en Ingresos Brutos, leyenda “A consumidor Final”, y la categoría de contribuyente frente a IVA del emisor) se imprimen en los encabezados de los comprobantes que se emitan.</p>				
	<p>La configuración de este parámetro de comportamiento tiene sentido en IFH 1G que permitan la emisión de comprobantes sobre rollos. La omisión de la impresión de las leyendas optionales sólo debe configurarse cuando el rollo se encuentre preimpreso al dorso y contenga pre-</p>				

	impresos los datos que serán omitidos al frente.				
	'P'	Imprimir leyenda opcionales.			
	Otro caracter	No imprimir.			
corte	De entrada. Opcional. Tipo: <i>Variant</i>				
	Este parámetro de comportamiento permite, al software de gestión para puntos de venta, siempre que la IFH 1G lo admita, especificar el tipo de corte de papel a realizar al finalizar la emisión de un comprobante.				
	Los valores posibles son los identificados por la siguientes constantes embebidas en el OCX Fiscal HASAR : CORTE_PARCIAL, CORTE_TOTAL, NO_CORTE.				
marco	De entrada. Opcional. Tipo: <i>Variant</i>				
	Este parámetro de comportamiento permite, al software de gestión para puntos de venta, decidir si el marco de los comprobantes será impreso por la IFH 1G .				
	La modificación de este parámetro tiene sentido en impresiones sobre formularios continuos u hojas sueltas				
	'P'	Imprimir el marco.			
	Otro caracter	No imprimir.			
reimpr	De entrada. Opcional. Tipo: <i>Variant</i>				
	Este parámetro de comportamiento permite, al software de gestión para puntos de venta, decidir si se reimprimen los comprobantes cancelados por corte de energía.				
	No todos los modelos y versiones de firmware fiscal de IFH 1G admiten la modificación de este parámetro de comportamiento.				
	'P'	Reimprimir comprobante cancelado por corte de energía.			
	Otro caracter	No reimprimir.			
saldo	De entrada. Opcional. Tipo: <i>Variant</i>				
	Este parámetro de comportamiento permite, al software de gestión para puntos de venta, indicar el texto que se imprimirá por defecto como descripción de un medio de pago asociado a la diferencia entre la suma de los pagos realizados y el total del comprobante; si éste se cierra con una suma de pagos inferior al total del comprobante.				
	A la salida de fábrica el texto por defecto es "Cuenta Corriente".				
sonido	De entrada. Opcional. Tipo: <i>Variant</i>				

	<p>Este parámetro de comportamiento permite, al software de gestión para puntos de venta, indicar si se emitirán pitidos ante la falta de papel.</p> <table> <tr> <td>'P'</td><td>Emitir pitidos.</td></tr> <tr> <td>Otro carácter</td><td>No emitir sonido alguno.</td></tr> </table>	'P'	Emitir pitidos.	Otro carácter	No emitir sonido alguno.
'P'	Emitir pitidos.				
Otro carácter	No emitir sonido alguno.				
alto	<p>De entrada. Opcional. Tipo: <i>Variant</i></p> <p>Este parámetro de comportamiento permite, al software de gestión para puntos de venta, indicar (si la IFH 1G lo admite) el alto de la hoja de papel, o del formulario continuo.</p> <p>Los valores posibles para este parámetro de comportamiento son los identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR: ALTO_A4, ALTO_CARTA, ALTO_OFICIO, ALTO_REDUCIDO.</p>				
ancho	<p>De entrada. Opcional. Tipo: <i>Variant</i></p> <p>Este parámetro de comportamiento permite, al software de gestión para puntos de venta, indicar (si la IFH 1G lo admite) el ancho de la hoja de papel, o del formulario continuo.</p> <p>Los valores posibles para este parámetro de comportamiento son los identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR: ANCHO_NORMAL, ANCHO_REDUCIDO.</p>				
estac	<p>De entrada. Opcional. Tipo: <i>Variant</i></p> <p>Este parámetro de comportamiento permite, al software de gestión para puntos de venta, indicar la estación de impresión a utilizar por defecto (si la IFH 1G tuviese más de una) para la emisión de cierres diarios 'X'/'Z'.</p> <p>Los valores posibles para este parámetro de comportamiento son los identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR: ESTACION_SLIP (formulario continuo), ESTACION_TICKET (rollo).</p>				
modo	<p>De entrada. Opcional. Tipo: <i>Variant</i></p> <p>Este parámetro de comportamiento permite, al software de gestión para puntos de venta, indicar como se efectuará la emisión de comprobantes (siempre que la IFH 1G admita esta configuración).</p> <p>Los valores posibles para este parámetro de comportamiento son los identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR: NO_USO_ESTACION_TICKET (sólo formularios continuos), USO_ESTACION_TICKET (rollo y formularios continuos).</p>				

Ejemplo de uso:

...
HASAR1.ReporteZ
HASAR1.ConfigurarControladorCompleto True, True, "1000.00", "25000.00"
...

El desarrollador del software de gestión para puntos de venta debe tener presente que para poder invocar al método *ConfigurarControladorCompleto()* se requiere de la realización previa del cierre diario 'Z'.

Método: ConfigurarControladorPorBloque()

Comando: *ConfigureControllerByBlock*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 7

Este método permite al software de gestión para puntos de venta modificar parámetros de comportamiento (en bloque) de la **IFH 1G** durante la jornada fiscal. Mediante este método es posible modificar, además, parámetros de comportamiento asociados a la normativa fiscal vigente.

El software de gestión para puntos de venta debería ofrecer estas opciones de configuración al usuario final, teniendo en cuenta el modelo de IFH 1G en uso.

ConfigurarControladorPorBloque limitecf, limitetf, ivanoi, copias, cambio, leyendas, corte

limitecf	<p>De entrada. Obligatorio. Tipo: <i>Double</i></p> <p>Este parámetro de comportamiento permite, al software de gestión para puntos de venta, indicar el nuevo valor establecido por la normativa fiscal vigente como monto límite para los comprobantes 'B' a contribuyentes Consumidores Finales, sin datos del comprador (o incompletos).</p> <p>Actualmente, este parámetro de comportamiento debería estar configurado con el valor '1000.00' (un mil pesos).</p> <p>NOTA: Mediante la RG AFIP Nº 3666/14, vigente desde el 4 de Setiembre de 2014, se suspende temporalmente la aplicación del tope de \$1000 (pesos un mil) como monto máximo para Tiques y comprobantes 'B' a consumidores finales sin nominar. Por suspensión temporal se entiende: hasta tanto entre en vigencia lo previsto en los párrafos segundo y tercero del Artículo 3º de la RG AFIP Nº 3561/13.</p>
limitetf	<p>De entrada. Obligatorio. Tipo: <i>Double</i></p> <p>Este parámetro de comportamiento permite, al software de gestión para puntos de venta, indicar el nuevo valor establecido por la normativa fiscal vigente como monto límite para los comprobantes tipificados como Tiques Factura y/o Tiques Nota de Crédito/Débito.</p>

	<p>Actualmente, este parámetro de comportamiento debería estar configurado con el valor ‘25000.00’ (veinticinco mil pesos).</p> <p>NOTA: La normativa fiscal vigente (RG AFIP Nº 3561/13) ha eliminado este tope para los comprobantes mencionados anteriormente.</p>
ivanoi	<p>De entrada. Obligatorio.</p> <p>Tipo: <i>Double</i></p> <p>Actualmente en desuso. Siempre en ‘0.00’.</p>
copias	<p>De entrada. Obligatorio.</p> <p>Tipo: enum <i>NúmeroDeCopias</i></p> <p>Este parámetro de comportamiento permite, al software de gestión para puntos de venta, establecer la cantidad máxima de ejemplares que se podrán imprimir de un mismo comprobante, en el momento de la emisión del original.</p> <p>No todos los modelos y versiones de firmware fiscal de IFH 1G admiten la modificación de este parámetro de comportamiento.</p> <p>Las IFH 1G imprimen automáticamente, a exigencias de la normativa fiscal vigente:</p> <ul style="list-style-type: none"> ▪ Comprobantes fiscales y Notas de Crédito: Original y Duplicado. ▪ Otros documentos: Solamente el original. <p>Este comportamiento de la IFH 1G no se ve afectado por la modificación del parámetro COPIAS_DOCUMENTOS. El software de gestión para puntos de venta deberá solicitar, a continuación del cierre del comprobante, la impresión de los ejemplares no impresos automáticamente (se debe invocar al método <i>ReimprimirComprobante()</i>).</p> <p>Los valores posibles para este parámetro de comportamiento, son los identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR:</p> <p>NO_COPIAS, ORIGINAL, DUPLICADO, TRIPLICADO, CUADRUPPLICADO.</p> <p>La constante embebida NO_COPIAS implica la emisión de un único ejemplar. Se asume el uso de papel copiativo. Por el momento, las constantes embebidas ORIGINAL y DUPLICADO son equivalentes: se permite emitir hasta duplicados.</p>
cambio	<p>De entrada. Obligatorio.</p> <p>Tipo: <i>Boolean</i></p> <p>Este parámetro de comportamiento permite, al software de gestión para puntos de venta, decidir si la leyenda “CAMBIO \$0.00” se imprime en el comprobante cuando la suma de los pagos realizados coincide exactamente con el total del comprobante que se está cerrando.</p> <p>Tener presente que esta leyenda, en las condiciones dadas, no se imprime en todos los tipos de comprobante posibles de emitir con la IFH 1G, o en todos los modelos homologados.</p>

	<i>True</i>	Imprimir leyenda, de corresponder.
	<i>False</i>	No imprimirla.
leyendas	De entrada. Obligatorio. Tipo: <i>Boolean</i>	
		Este parámetro de comportamiento permite, al software de gestión para puntos de venta, decidir si las leyendas optionales (número de inscripción en Ingresos Brutos, leyenda “A consumidor Final”, y la categoría de contribuyente frente a IVA del emisor) se imprimen en los encabezados de los comprobantes que se emitan.
		La configuración de este parámetro de comportamiento tiene sentido en IFH 1G que permitan la emisión de comprobantes sobre rollos. La omisión de la impresión de las leyendas optionales sólo debe configurarse cuando el rollo se encuentre preimpreso al dorso y contenga preimpresos los datos que serán omitidos al frente.
	<i>True</i>	Imprimir leyenda optionales.
	<i>False</i>	No imprimir.
corte	De entrada. Obligatorio. Tipo: enum <i>TipoDeCorteDePapel</i>	
		Este parámetro de comportamiento permite, al software de gestión para puntos de venta, siempre que la IFH 1G lo admita, especificar el tipo de corte de papel a realizar al finalizar la emisión de un comprobante.
		Los valores posibles son los identificados por la siguientes constantes embebidas en el OCX Fiscal HASAR : CORTE_PARCIAL, CORTE_TOTAL, NO_CORTE .

Ejemplo de uso:

```
...
HASAR1.ReporteZ
HASAR1.ConfigurarControladorPorBloque 1000.00, 25000.00, 0.00, DUPLICADO, True, True, _
NO_CORTE
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que para poder invocar al método *ConfigurarControladorCompleto()* se requiere de la realización previa del cierre diario ‘Z’.

Método: DatosCliente()

Comando: *SetCustomerData*

Disponible: **OCX Fiscal HASAR**, en todas las versiones

Argumentos: 5

Este método debe ser invocado por el software de gestión para puntos de venta en forma previa al método de apertura de un comprobante distinto de comprobante no fiscal). Esto se debe a que toda la información del cliente se imprime en el encabezado del comprobante, tarea que lleva a cabo el comando de apertura.

DatosCliente rsocial, nrodoc, tipodoc, categ, domic

rsocial	<p>De entrada. Obligatorio. Tipo: <i>String</i></p> <p>Este argumento permite al software de gestión para puntos de venta indicar el nombre o razón social del cliente al cual se emitirá el comprobante.</p> <p>La palabra ‘Total’ (de uso reservado para la IFH 1G) puede estar embebida en el texto pasado como argumento, pero se exige que sea escrito como se muestra:</p> <table style="margin-left: 40px;"> <tr> <td>Total</td><td>Primer carácter en mayúscula.</td></tr> <tr> <td></td><td>El resto de los caracteres en minúsculas.</td></tr> </table> <p>De no hacerse así, dependerá del modelo y versión de firmware fiscal de la IFH 1G el tratamiento que recibirá: se puede rechazar el comando por “Campo de datos inválido”, o podrá ser aceptado (pero al momento de su impresión, el carácter ‘o’ u ‘O’ será reemplazado por el carácter '#').</p>	Total	Primer carácter en mayúscula.		El resto de los caracteres en minúsculas.
Total	Primer carácter en mayúscula.				
	El resto de los caracteres en minúsculas.				
nrodoc	<p>De entrada. Obligatorio. Tipo: <i>String</i></p> <p>Este argumento permite al software de gestión para puntos de venta indicar el número del documento especificado en el argumento tipodoc.</p> <p>De tratarse de un número de CUIT/CUIL, deben ser 11 dígitos (sin guiones) y constituir un número válido para AFIP (la IFH 1G verifica que se trate de un CUIT/CUIL válido).</p> <p>En el caso de pasaportes, este argumento puede ser alfanumérico siempre y cuando la IFH 1G soporte tal característica.</p> <p>Excepcionalmente, este argumento puede tomar el valor “ ” (un espacio) siempre y cuando el argumento tipodoc haya recibido la constante embebida del OCX Fiscal HASAR identificada como TIPO_NINGUNO, y el argumento categ haya recibido la constante embebida del OCX Fiscal HASAR identificada como CONSUMIDOR_FINAL. En estas condiciones, el comprobante ‘B’ quedaría limitado a un monto máximo de \$1000 (pesos un mil).</p> <p>NOTA: Mediante la RG AFIP Nº 3666/14, vigente desde el 4 de Setiembre de 2014, se suspende temporalmente la aplicación del tope de \$1000 (pesos un mil) como monto máximo para Tiques y comprobantes ‘B’ a consumidores finales sin nominar. Por suspensión temporal se entiende: hasta tanto entre en vigencia lo previsto en los párrafos segundo y tercero del Artículo 3º de la RG AFIP Nº 3561/13.</p>				
tipodoc	<p>De entrada. Obligatorio. Tipo: enum <i>TiposDeDocumento</i></p> <p>Este argumento permite al software de gestión para puntos de venta especificar el tipo de documento asociado al número indicado en el argumento nrodoc.</p> <p>Los valores posibles para este argumento son los identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR:</p> <p>TIPO_CUIT, TIPO_CUIL, TIPO_LE, TIPO_LC, TIPO_DNI, TIPO_PASAPORTE, TIPO_CI,</p>				

	<p>TIPO_NINGUNO.</p> <p>No todos los modelos y versiones de firmware fiscal de IFH 1G aceptan la lista completa de valores enumerados. Salvo en el caso de clientes categorizados frente a IVA como contribuyentes Consumidores Finales, para el resto de las categorías este argumento debe tomar el valor TIPO_CUIT.</p>
categ	<p>De entrada. Obligatorio. Tipo: enum <i>TiposDeResponsabilidades</i></p> <p>Este argumento permite al software de gestión para puntos de venta especificar la categoría del cliente como contribuyente frente a IVA.</p> <p>Los valores posibles para este argumento son los identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR:</p> <p>RESPONSABLE_INSCRIPTO, RESPONSABLE_NO_INSCRIPTO, RESPONSABLE_EXENTO, NO_RESPONSABLE, CONSUMIDOR_FINAL, BIENES_DE_USO, NO_CATEGORIZADO, MONOTRIBUTO, MONOTRIBUTO_SOCIAL, EVENTUAL, EVENTUAL_SOCIAL.</p> <p>No todos los modelos y versiones de firmware fiscal de IFH 1G aceptan la lista completa de valores enumerados.</p> <p>Cuando el emisor y el receptor del comprobante, son ambos contribuyentes Responsables Inscriptos, de corresponder, puede abrirse a continuación un comprobante fiscal ‘A’, o una Nota de Crédito ‘A’, o un Tique Nota de Crédito ‘A’.</p> <p>Cualquier otra combinación de categorías de contribuyentes entre emisor y receptor, de corresponder, permite abrir un comprobante fiscal ‘B’, o una Nota de Crédito ‘B’, o un Tique Nota de Crédito ‘B’ (la IFH 1G imprimirá la letra ‘C’ cuando el emisor no sea un contribuyente Responsable Inscripto).</p>
domic	<p>De entrada. Opcional. Tipo: <i>Variant</i></p> <p>Este argumento permite al software de gestión para puntos de venta especificar el domicilio del cliente cuando el comando <i>SetCustomerData</i> así lo requiera (dependerá del modelo y versión de firmware fiscal de la IFH 1G en uso).</p>

Ejemplo de uso:

...
HASAR1.DatosCliente “Razón Social del receptor...”, “9999999995”, TIPO_CUIT, _
 RESPONSABLE_INSCRIPTO, “Domicilio del receptor...”

...

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Invocar a *DatosCliente()* es obligatorio para emitir comprobantes fiscales ‘A’.
- Invocar a *DatosCliente()* es obligatorio para emitir Notas de Crédito.

- Invocar a *DatosCliente()* es obligatorio para emitir comprobantes fiscales ‘B’, a contribuyentes Consumidores Finales, por montos superiores a \$1000 (pesos un mil).

NOTA: Mediante la RG AFIP Nº 3666/14, vigente desde el 4 de Setiembre de 2014, se suspende temporalmente la aplicación del tope de \$1000 (pesos un mil) como monto máximo para Tiques y comprobantes ‘B’ a consumidores finales sin nominar.

Por suspensión temporal se entiende: hasta tanto entre en vigencia lo previsto en los párrafos segundo y tercero del Artículo 3º de la RG AFIP Nº 3561/13.

- Invocar a *DatosCliente()* es opcional en cualquier otro caso.

Método: DescripcionStatusFiscal()

Este método no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 1

Retorno: *String*

Este método permite al software de gestión para puntos de venta mostrar al usuario final un mensaje asociado a la generación del evento *ErrorFiscal()*, del evento *EventoFiscal()*, o porque la propiedad *HuboErrorFiscal()* retornó el valor *True*.

DescripcionStatusFiscal biterror

biterror	De entrada. Obligatorio. Tipo: <i>Long</i> Este argumento permite al software de gestión para puntos de venta indicar un valor identificado por alguna de las siguientes constantes embebidas en el OCX Fiscal HASAR : Asociadas al evento <i>ErrorFiscal()</i> : F_FISCAL_MEMORY_FAIL, F_FISCAL_MEMORY_FULL, F_INVALID_COMMAND, F_INVALID_FIELD_DATA, F_TOTAL_OVERFLOW, F_UNRECOGNIZED_COMMAND, F_WORKING_MEMORY_FAIL. Asociadas al evento <i>EventoFiscal()</i> : F_FISCAL_MEMORY_NEAR_FULL, F_FISCAL_TERMINAL_CERTIFIED, F_RECEIPT_SLIP_OPEN, F_FISCAL_TERMINAL_FISCALIZED, F_RECEIPT_OPEN.
-----------------	--

Ejemplo de uso:

```
...
Debug.Print HASAR1.DescripcionStatusFiscal(F_INVALID_COMMAND)
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que si la propiedad *EventosIndividuales()* del **OCX Fiscal HASAR** está configurada con el valor *False*, no se podrá utilizar el número crudo -que ingresa como argumento- de los eventos *ErrorFiscal()* y *EventoFiscal()* (se deben identificar previamente los bits con valor ‘1’ para procesarlos individualmente).

Nota al margen:

Cada vez que se envía un comando a la **IFH 1G**, ésta responde con un string cuyos dos primeros campos corresponden a *Status de Impresora* y *Status Fiscal*. Ambos campos de estado contienen como información un string de 4 caracteres, donde cada uno de ellos representa un dígito hexadecimal. Esta información debe convertirse a un número binario, y en esta nueva representación examinar los bits que se encuentran en ‘1’. El *Manual de comandos* de la **IFH 1G** cuenta con apéndices que describen como interpretar cada uno de estos bits.

Método: DescripcionStatusImpresor()

Este método no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 1

Retorno: *String*

Este método permite al software de gestión para puntos de venta mostrar al usuario final un mensaje asociado a la generación del evento *EventoImpresora()*, o bien porque la propiedad *HuboErrorMecanico()*, o porque la propiedad *HuboFaltaPapel()* retornó el valor *True*.

DescripcionStatusImpresor biterror

biterror	De entrada. Obligatorio. Tipo: <i>Long</i> Este argumento permite al software de gestión para puntos de venta indicar un valor identificado por alguna de las siguientes constantes embebidas en el OCX Fiscal HASAR : Asociadas al evento <i>EventoImpresora()</i> : P_OFFLINE, P_PPRINTER_ERROR, P_SPLIT_PLATEN_OPEN.
-----------------	--

Ejemplo de uso:

```
...
Debug.Print HASAR1.DescripcionStatusImpresor(P_OFFLINE)
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que si la propiedad *EventosIndividuales()* del **OCX Fiscal HASAR** está configurada con el valor *False*, no se podrá utilizar el número crudo -que ingresa como argumento- del evento *EventoImpresora()* (se deben identificar previamente los bits con valor ‘1’ para procesarlos individualmente).

Nota al margen:

Cada vez que se envía un comando a la **IFH 1G**, ésta responde con un string cuyos dos primeros campos corresponden a *Status de Impresora* y *Status Fiscal*. Ambos campos de estado contienen como información un string de 4 caracteres, donde cada uno de ellos representa un dígito hexadecimal. Esta información debe convertirse a un número binario, y en esta nueva representación examinar los bits que se encuentran en ‘1’. El *Manual de comandos* de la **IFH 1G** cuenta con apéndices que describen como interpretar cada uno de estos bits.

Método: DescuentoGeneral()

Comando: *GeneralDiscount*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 3

Este método permite al software de gestión para puntos de venta realizar un descuento/recargo sobre el monto total del comprobante. Cada vez que se realiza un descuento/recargo, el monto aplicado será prorrteado por la **IFH 1G** entre todas las alícuotas de IVA intervenientes en el comprobante fiscal (o Nota de Crédito, o Tique Nota de Crédito) que se está emitiendo.

DescuentoGeneral descrip, monto, negat

descrip	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar un texto a imprimir, y asociado al monto de descuento/recargo a realizar sobre el subtotal del comprobante en curso.
monto	De entrada. Obligatorio. Tipo: <i>Double</i> Permite al software de gestión para puntos de venta indicar el monto de descuento/recargo a aplicar sobre el subtotal del comprobante en curso.
negat	De entrada. Obligatorio. Tipo: <i>Boolean</i> Permite al software de gestión para puntos de venta indicar como afecta el monto al subtotal del comprobante en curso. Los valores posibles son: <i>True</i> El monto será interpretado por el OCX Fiscal HASAR como realización de un descuento. <i>False</i> El monto será interpretado por el OCX Fiscal HASAR como realización de un recargo.

Ejemplo de uso:

```
...
HASAR1.DescuentoGeneral "Oferta Especial...", 25.00, True
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El método *DescuentoGeneral()* se puede invocar más de una vez, consecutivamente.
- Una vez invocado el método *DescuentoGeneral()* no se pueden realizar más ventas.
- En el caso de realizar descuentos, la normativa fiscal vigente no permite cerrar comprobantes en cero pesos.

- La aplicación de descuentos/recargos sobre el total del comprobante no afecta los montos de impuestos internos fijos acumulados en el comprobante.

Método: DescuentoUltimoltem()

Comando: *LastItemDiscount*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 3

Este método permite al software de gestión para puntos de venta realizar un descuento/recargo sobre el último ítem vendido en el comprobante. Cada vez que se realiza un descuento/recargo, el monto aplicado será prorrteado por la **IFH 1G** entre todos los componentes (monto neto y montos de impuestos) de la última venta concretada.

DescuentoGeneral descrip, monto, negat

descrip	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar un texto a imprimir, y asociado al monto de descuento/recargo a aplicar sobre el ítem previamente facturado.
monto	De entrada. Obligatorio. Tipo: <i>Double</i> Permite al software de gestión para puntos de venta indicar el monto de descuento/recargo a aplicar sobre el ítem previamente facturado.
negat	De entrada. Obligatorio. Tipo: <i>Boolean</i> Permite al software de gestión para puntos de venta indicar como afecta el monto al ítem previamente facturado. Los valores posibles son: <i>True</i> El monto será interpretado por el OCX Fiscal HASAR como realización de un descuento. <i>False</i> El monto será interpretado por el OCX Fiscal HASAR como realización de un recargo.

Ejemplo de uso:

```
...
HASAR1.DescuentoUltimoltem "Oferta del día...", 25.00, True
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El método *DescuentoUltimoltem()* sólo se puede invocar una vez por cada ítem, y a continuación del mismo.

- Una vez invocado el método *DescuentoUltimoItem()* se puede continuar con la facturación de ítems.
- En el caso de realizar descuentos, la normativa fiscal vigente no permite cerrar comprobantes en cero pesos.
- La aplicación de descuentos/recargos sobre el último ítem vendido no afecta los montos de impuestos internos fijos del producto.

Método: DetalleRecibo()

Comando: *ReceiptText*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 1

Este método permite al software de gestión para puntos de venta imprimir líneas de texto en un Recibo ('A', 'B/C', o 'X') que conformarán el detalle de "*En concepto de...*", con objeto explicar el monto del mismo.

DetalleRecibo *texto*

texto	<p>De entrada. Obligatorio. Tipo: <i>String</i></p> <p>Permite al software de gestión para puntos de venta indicar el texto a imprimir en el Recibo 'A', 'B/C', o 'X'. El texto se imprime a continuación de la leyenda "En concepto de:".</p>
--------------	--

Ejemplo de uso:

```
...
HASAR1.DetalleRecibo "Cheques rechazados...."
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El método *DetalleRecibo()* separa el texto recibido como argumento en dos o más líneas en caso de superar el máximo aceptado por la **IFH 1G** para una línea.
- En el caso de Recibos 'X' el método *DetalleRecibo()* se puede invocar hasta un máximo de 9 veces (siempre que en cada invocación no se supere la longitud máxima para una línea).

Método: DevolucionDescuento()

Comando: *ReturnRecharge*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 6

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 **IFH 1G** se traduce en su aplicación sobre todos los ítems vendidos que se encuentren afectados por la alícuota de IVA indicada mediante el argumento **alícuota**, y el impuesto interno indicado mediante el argumento **impint**.

El otro uso del método *DevolucionDescuento()* permite al software de gestión para puntos de venta el manejo de la devolución de envases.

DevolucionDescuento descrip, monto, alicuota, impint, negat, oper

descrip	De entrada. Obligatorio. Tipo: <i>String</i>	Permite al software de gestión para puntos de venta indicar un texto para imprimir, asociado a la bonificación/recargo, o a la devolución de envases.	
monto	De entrada. Obligatorio. Tipo: <i>Double</i>	Permite al software de gestión para puntos de venta indicar el monto de descuento/recargo, o de la devolución de envases.	
alicuota	De entrada. Obligatorio. Tipo: <i>Double</i>	Permite al software de gestión para puntos de venta indicar la alícuota de IVA que afecta al monto de descuento/recargo, o de devolución de envases.	
impint	De entrada. Obligatorio. Tipo: <i>Double</i>	El valor que el software de gestión para puntos de venta indique en este argumento establece el impuesto interno que afecta al monto de descuento/recargo, o de devolución de envases. El valor de este argumento y la interpretación que de él hace el OCX Fiscal HASAR depende de la configuración de tres de sus propiedades: <i>KIVA()</i> , <i>ImpuestosInternosFijos()</i> , e <i>ImpuestosInternosPorMonto()</i> .	
negat	Factor K	Imp. Int. Fijo	<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>
		Imp. Int. Variable	<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>
		Porcentaje (impuesto interno variable)	<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>
	Monto Fijo (impuesto interno fijo)		<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>
	De entrada. Obligatorio. Tipo: <i>Boolean</i>		

	Permite al software de gestión para puntos de venta indicar si la operación será positiva, o negativa.
	<p><i>True</i> La operación es negativa.</p> <p><i>False</i> La operación es positiva.</p>
oper	<p>De entrada. Obligatorio.</p> <p>Tipo: enum <i>TiposDeDescuentos</i></p> <p>Permite al software de gestión para puntos de venta indicar si la operación corresponde a un descuento/recargo, o se trata de una devolución de envases.</p> <p>Los valores posibles para este argumento son los identificados por las siguientes constantes embebidas del OCX Fiscal HASAR:</p> <p>DESCUENTO_RECARGO, DEVOLUCION_DE_ENVASES.</p>

Ejemplo de uso:

```
...
HASAR1.KIVA = False
HASAR1.ImpuestoInternoFijo = False
HASAR1.ImpuestoInternoPorMonto = False
...
HASAR1.DevolucionDescuento "Bonificación...", 150.00, 21.00, 10.00, True, DESCUENTO_RECARGO
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El método *DevolucionDescuento()* se puede invocar más de una vez, consecutivamente.
- Una vez invocado el método *DevolucionDescuento()* no se pueden realizar más ventas.
- En el caso de realizar descuentos, la normativa fiscal vigente no permite cerrar comprobantes en cero pesos.
- La aplicación de descuentos/recargos sobre una alícuota de IVA determinada no afecta los montos de impuestos internos fijos acumulados en el comprobante.

Método: DevolverPago()

Comando: *TotalTender*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Argumentos: 4

El método *DevolverPago()* permite al software de gestión para puntos de venta realizar la anulación de un pago realizado previamente en el comprobante en curso. En la **IFH 1G** se traduce como la impresión de un pago negativo.

DevolverPago descrip, monto, txtadic, resto

descrip	De entrada. Obligatorio. Tipo: <i>String</i>
----------------	---

	Permite al software de gestión para puntos de venta indicar la descripción de un pago, que debe ser coincidente con uno realizado previamente, en el comprobante en curso y cuyo monto coincide con el indicado en el argumento monto .
monto	De entrada. Obligatorio. Tipo: <i>Double</i>
	Permite al software de gestión para puntos de venta indicar el monto de un pago realizado previamente, en el comprobante en curso, y cuya descripción coincide con el indicado en el argumento descrip .
txtadic	De entrada. Opcional. Tipo: <i>String</i>
	Permite al software de gestión para puntos de venta indicar el texto adicional de un pago con tarjeta de crédito/débito realizado previamente, en el comprobante en curso, donde además su descripción y monto coinciden con lo indicado en los argumentos descrip y monto .
	Este argumento, suele corresponderse con un número de tarjeta.
resto	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el monto pendiente de pago para el comprobante en curso. Se trata de la diferencia entre el total del comprobante y la suma de los pagos realizados (incluyendo devoluciones de pagos).

Ejemplo de uso:

```
...
Dim saldo As variant
...
HASAR1.ImprimirPago "PlastiCard...", 100.00, "*****8796", saldo
...
HASAR1.DevolverPago "PlastiCard...", 100.00, "*****8796", saldo
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- La devolución de pagos es una prestación que no está disponible en todos los modelos y versiones de firmware fiscal de **IFH 1G**.
- Las devoluciones de pagos son contabilizadas por la **IFH 1G** como si se tratara de comandos de pago habituales. Es decir, la suma de pagos y devoluciones de pagos no puede superar la máxima cantidad de pagos admitidos por la **IFH 1G** en uso (notar que se genera el mismo string de comando que en el caso de uso del método *ImprimirPago()*).
- En la devolución de un pago previo (dentro del mismo comprobante) deben coincidir exactamente: descripción del pago, monto, y –de existir- descripción adicional. Cualquier diferencia existente en alguno de estos datos causará que la **IFH 1G** rechace el string del comando generado.

Método: DNFHFarmacias()

Comando: *DNFHFarmacias*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 1

El método permite al software de gestión para puntos de venta emitir un comprobante, destinado a obras sociales, por las ventas realizadas a afiliados a las mismas.

DNFHFarmacias copias

copias	De entrada. Obligatorio. Tipo: <i>Long</i> Permite al software de gestión para puntos de venta indicar la cantidad de ejemplares que se emitirán del mismo comprobante.
---------------	---

Ejemplo de uso:

```
...
HASAR1.CerrarComprobanteFiscal
HASAR1.DNFHFarmacias 2
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El contenido del comprobante de farmacias es generado automáticamente por la **IFH 1G** a partir del Tique o Tique Factura emitido en forma previa, e inmediata. Son considerados, solamente, los primeros 20 ítems. De superarse esta cantidad, los ítems excedentes no aparecerán en el comprobante de farmacias.
- El comprobante que se genera mediante el método *DNFHFarmacias()* no se encuentra disponible en todos los modelos de **IFH 1G**.
- El comprobante que se genera mediante el método *DNFHFarmacias()*, en las **IFH 1G** más modernas, ha sido reemplazado por el comprobante no fiscal homologado “*Uso Interno Farmacia*” cuya confección es responsabilidad del software de gestión para puntos de venta, ofreciendo mayor flexibilidad a la hora de su emisión.

Método: DNFHReparto()

Comando: *DNFHReparto*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 1

El método permite al software de gestión para puntos de venta emitir un comprobante, destinado a reparto, por las ventas realizadas con envío a domicilio.

DNFHReparto copias

copias	De entrada. Obligatorio.
---------------	--------------------------

Tipo: *Long*

Permite al software de gestión para puntos de venta indicar la cantidad de ejemplares que se emitirán del mismo comprobante.

Ejemplo de uso:

```
...
HASAR1.CerrarComprobanteFiscal
HASAR1.DNFHReparto 2
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El contenido del comprobante de reparto es generado automáticamente por la **IFH 1G** a partir del Tique o Tique Factura emitido en forma previa, e inmediata. Son considerados, solamente, los primeros 20 ítems. De superarse esta cantidad, los ítems excedentes no aparecerán en el comprobante de reparto.
- El comprobante que se genera mediante el método *DNFHReparto()* no se encuentra disponible en todos los modelos de **IFH 1G**.
- El comprobante que se genera mediante el método *DNFHReparto()*, en las **IFH 1G** más modernas, ha sido reemplazado por el comprobante no fiscal homologado “*Talón de Reparto*” cuya confección es responsabilidad del software de gestión para puntos de venta, ofreciendo mayor flexibilidad a la hora de su emisión.

Método: Enviar()

Este método no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 1

Este método permite al software de gestión para puntos de venta enviar a la **IFH 1G** un string de comando construido por la aplicación, y siguiendo las indicaciones sintácticas dadas en el *Manual de Comandos* correspondiente.

Enviar comando

comando	De entrada. Obligatorio. Tipo: <i>String</i>
	Permite al software de gestión para puntos de venta indicar un string de comando (ASCII, CP437) destinado a la IFH 1G .

Ejemplo de uso:

```
...
Dim comando As String
...
HASAR1.Modelo = MODELO_615
```

```
...
comando = Chr(129)
HASAR1.Enviar comando
```

```
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que el uso del método *Enviar()* está previsto para aquellos casos donde la versión de **OCX Fiscal HASAR** en uso no puede generar correctamente alguno de los strings de comandos soportados por la **IFH 1G** conectada. Es decir, cuando el **OCX Fiscal HASAR** no posee una constante embebida que identifique al modelo y versión de firmware fiscal de la **IFH 1G** a manejar.

En consecuencia, la propiedad *Modelo()* se debe configurar con la constante embebida que identifique al modelo más similar -desde el punto de vista de conjunto de comandos soportados- a la **IFH 1G** en uso.

Por ejemplo, si la **IFH 1G** a manejar es el modelo SMH/P-715F v01.00 y el **OCX Fiscal HASAR** es versión v010724, la propiedad *Modelo()* debe configurarse con el valor de la constante embebida MODELO_615. En estas condiciones si se quisiera abrir un Tique Nota de Crédito, al invocar al método *AbrirDNFH()* se produciría una excepción (la **IFH 1G** SMH/P-715F emite tales comprobantes, pero la SMH/P-615F no). Y en este punto es donde el software de gestión para puntos de venta debe construir el string del comando (*OpenDNFH*) y recurrir al uso del método *Enviar()*.

El método *Enviar()* no está pensando para que todos los strings de comandos sean construidos por el software de gestión para puntos de venta (para ello existe la librería *winfis32.dll*), sino para poder resolver pequeñas incompatibilidades sin tener que actualizar la versión del **OCX Fiscal HASAR**.

Método: EnviarStatPrn()

Comando: *StatPRN*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

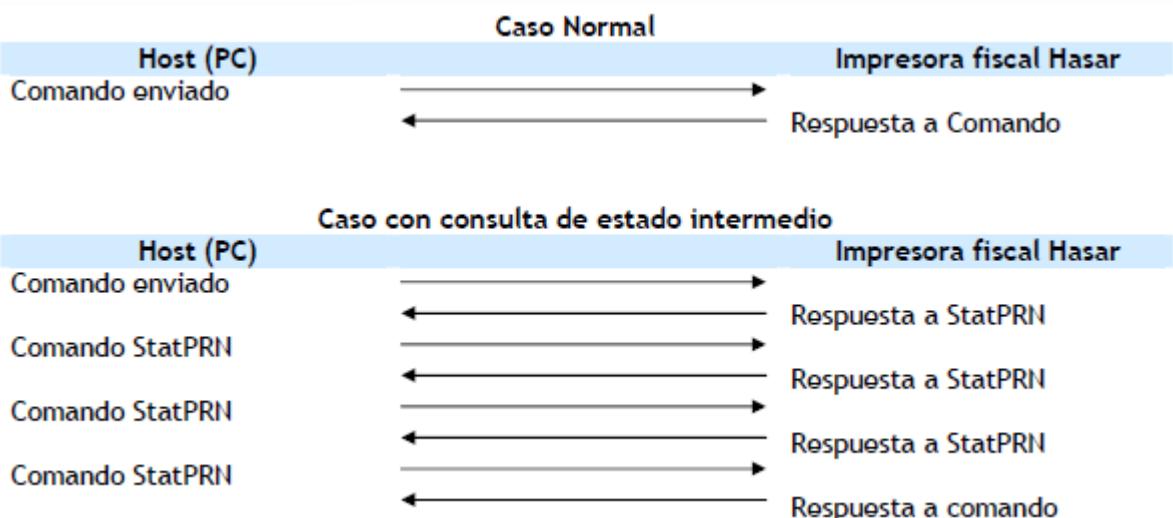
Argumentos: No requiere.

Este método permite al software de gestión para puntos de venta realizar consultas de estado intermedio a la **IFH 1G**; siempre y cuando las consultas de este tipo estén soportadas. Es decir, si la **IFH 1G** acepta el comando *StatPRN*.

Se denomina estado intermedio al período de tiempo que normalmente transcurre entre la llegada de un comando a la **IFH 1G** y la devolución de su correspondiente respuesta (sea inmediata o demorada).

Cuando se inicia una demora en la entrega de la respuesta esperada, la **IFH 1G** 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 *True*.

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, impresora *off-line*, etc. Mientras la respuesta esperada se encuentra demorada se puede consultar estado a la **IFH 1G** 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 *True*- se puede enviar otro comando *StatPRN* (nuevamente invocando al método *EnviarStatPrn()*). Si la respuesta recibida es la esperada, entonces se puede enviar el siguiente comando a la **IFH 1G**.



El manejo de la consulta de estado intermedio puede hacerlo:

- Directamente el **OCX Fiscal HASAR** (transparente para el software).
- El software de gestión para puntos de venta.

En el segundo caso deben configurarse adecuadamente las propiedades *ReintentoConstante()* y *ModoStatPrn()*; ambas disponibles en todas las versiones del **OCX Fiscal HASAR**.

Ejemplo de uso:

```

...
If (HASAR1.SoportaStatPrn) Then
    HASAR1.ReintentoConstante = False
    HASAR1.ModoStatPrn = True
End If
...
HASAR1.CerrarComprobanteFiscal

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

Método: EspecificarIVANoInscripto()

Comando: *ChargeNonRegisteredTax*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 1

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.

EspecificarIVANoInscripto monto

monto	De entrada. Obligatorio. Tipo: <i>Double</i> Permite al software de gestión para puntos de venta indicar el monto de sobretasa de IVA calculado por dicha aplicación.
--------------	---

Ejemplo de uso:

```
...
HASAR1.EspecificarIVANoInscripto 54.56
...
```

Las **IFH 1G** cuentan con un parámetro de configuración que es la sobretasa de IVA a aplicar (50 % -actualmente, los nuevos modelos salen de fábrica con el valor 0.00 % programado-) sobre cada una de las alícuotas de IVA intervenientes en el comprobante en emisión, cuando el emisor es un contribuyente Responsable Inscripto y el receptor es un contribuyente Responsable No Inscripto (categoría actualmente en desuso).

Puede ocurrir que por exigencias de la normativa fiscal vigente el software de gestión para puntos de venta deba calcular el monto que se origina por aplicación de la sobretasa de IVA a determinados ítems, y no corresponda aplicar el monto que calcula automáticamente la **IFH 1G**. Esto se debe a que la **IFH 1G** aplica la sobretasa de IVA sobre el total de la factura, y no sobre un grupo de terminado de ítems.

La invocación al método *EspecificarIVANoInscripto()* sólo corresponde cuando la aplicación de la sobretasa de IVA es parcial. Si fuese total, no se necesita del envío del string del comando *ChargeNonRegisteredTax*.

Método: EspecificarNombreDeFantasia()

Comando: *SetFantasyName*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

En las versiones *v051122* y posteriores genera una excepción por "Función discontinuada."

Argumentos: 2

Este método permite al software de gestión para puntos de venta cargar la memoria de trabajo de la **IFH 1G** con dos líneas de texto (1 y 2) 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.

EspecificarNombreDeFantasia linea1, linea2

linea1	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar el texto correspondiente a la primera línea de impresión del nombre de fantasía.
linea2	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar el texto correspondiente a la segunda línea de impresión del nombre de fantasía.

Ejemplo de uso:

```
...
HASAR1.EspecificarNombreDeFantasia "Texto línea 1...", "Texto línea 2..."
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- La ubicación de la impresión de las líneas de fantasía es siempre previa, e inmediata, a la razón social del emisor; se trate de impresión sobre rollo, formulario continuo, u hojas sueltas.
- El método *EspecificarNombreDeFantasia()* genera un string de comando *SetFantasyName* por cada línea a cargar en la memoria de trabajo de la **IFH 1G**.
- Si bien el contenido de los argumentos **linea1** o **linea2** puede ser un string vacío, no pueden ser ambos vacíos en una misma invocación al método *EspecificarNombreDeFantasia()*. Como mínimo, alguna de los dos argumentos debe contener al menos un carácter –aunque éste sea un espacio-.
- Si bien para el **OCX Fiscal HASAR v051122**, y posteriores, el método *EspecificarNombreDeFantasia()* se encuentra **DEPRECADO**, estas versiones poseen la propiedad *NombreDeFantasia()* que genera el mismo comando pero haciendo más flexible la programación.

Método: EspecificarPercepcionGlobal()

Comando: *Perceptions*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 2

Este método permite al software de gestión para puntos de venta imprimir en un comprobante fiscal (o una Nota de Crédito, o un Tique Nota de Crédito) el detalle de una percepción realizada al cliente. Se trata de percepciones aplicadas sobre el total del comprobante en curso de emisión.

EspecificarPercepcionGlobal descrip, monto

descrip	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar una descripción identificatoria de la percepción que se está incluyendo en el total del comprobante.
monto	De entrada. Obligatorio. Tipo: <i>Double</i> Permite al software de gestión para puntos de venta indicar el monto de la percepción que se está incluyendo en el total del comprobante.

Ejemplo de uso:

```
...
HASAR1.EspecificarPercepcionGlobal "Resolución...", 12.45
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que es responsabilidad de su programa la aplicación de la percepción (cuando correspondiere) y el cálculo del monto de la misma, de acuerdo a las indicaciones de la normativa fiscal en vigencia.

Método: EspecificarPercepcionPorIVA()

Comando: *Perceptions*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 3

Este método permite al software de gestión para puntos de venta imprimir en un comprobante fiscal (o una Nota de Crédito, o un Tique Nota de Crédito) el detalle de una percepción realizada al cliente. Se trata de percepciones aplicadas sobre ítems afectados por determinada alícuota de IVA y que han sido registrados en el comprobante en curso de emisión.

EspecificarPercepcionPorIVA *descrip, monto, alic*

	descrip	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar una descripción identificatoria de la percepción que se está incluyendo en el total del comprobante.
	monto	De entrada. Obligatorio. Tipo: <i>Double</i> Permite al software de gestión para puntos de venta indicar el monto de la percepción que se está incluyendo en el total del comprobante.
	alic	De entrada. Obligatorio. Tipo: <i>Double</i> Permite al software de gestión para puntos de venta indicar la alícuota de IVA sujeta a la aplicación de la percepción. Debe haberse registrado al menos un ítem con dicha alícuota en el comprobante en curso de emisión.

Ejemplo de uso:

```
...
HASAR1.EspecificarPercepcionPorIVA "Resolución...", 12.45, 10.50
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que es responsabilidad de su programa la aplicación de la percepción (cuando correspondiere) y el cálculo del monto de la misma, de acuerdo a las indicaciones de la normativa fiscal en vigencia.

Método: EstadoInternalImpresor()

Comando: *PrinterInternalStatus*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Argumentos: 2

Este método permite al software de gestión para puntos de venta consultar a la **IFH 1G** acerca del estado interno de la impresora no fiscal que la compone.

EstadoInternalImpresor *estado, descrip*

estado	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener un código numérico de hasta 5 dígitos que identifica un error, advertencia o, simplemente, el estado interno de la impresora no fiscal como tal.
descrip	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener una descripción de hasta 200 caracteres. Se trata del mensaje asociado al argumento estado .

Ejemplo de uso:

```
...
Dim estado As Variant, descrip As Variant
...
HASAR1.EstadoInternalImpresor estado, descrip
Debug.Print "Código: " & estado & vbCrLf & "Mensaje: " & descrip
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Las **IFH 1G** pueden pensarse como un equipo formado por dos partes:
 - *La impresora no fiscal.*
Estándar. Tal como la conocemos desde siempre.
 - *El módulo fiscal.*
Todo el diálogo entre el software de gestión para puntos de venta y la **IFH 1G**, se realiza a través de este módulo.
- El string de comando que genera el método *EstadoInternalImpresor()* tiene sentido de utilización, por el momento, al dialogar con **IFH 1G** de tipo láser.
El comando generado solamente es aceptado por el modelo SMH/PL-23F; en todas sus versiones homologadas hasta la fecha.

Método: Finalizar()

Este método no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: No requiere.

Este método permite al software de gestión para puntos de venta dar por terminada una sesión de trabajo, y liberar los recursos asociados a las comunicaciones con la **IFH 1G**.

Ejemplo de uso:

```
...
HASAR1.Finalizar
...
```

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

Método: ImprimirCódigoDeBarras()

Comando: *BarCode*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 4

Este método permite al software de gestión para puntos de venta imprimir (siempre que el tipo de comprobante lo permita y el modelo de **IFH 1G** lo admita) un código de barras en el comprobante en emisión.

ImprimirCodigoDeBarras tipo, codigo, connro, ahora

tipo	<p>De entrada. Obligatorio. Tipo: enum <i>TiposDeCódigosDeBarras</i></p> <p>Permite al software de gestión para puntos de venta indicar el tipo de código de barras que se va a imprimir.</p> <p>Los valores posibles para este argumento son los identificados por las siguientes constantes embebidas del OCX Fiscal HASAR:</p> <p>CODIGO TIPO EAN_13, CODIGO TIPO EAN_8, CODIGO TIPO ITS, CODIGO TIPO UPCA, CODIGO DE BORRADO.</p>
codigo	<p>De entrada. Obligatorio. Tipo: <i>String</i></p> <p>Permite al software de gestión para puntos de venta indicar el string de caracteres numéricos que permitirá generar el código de barras a imprimir.</p>
connro	<p>De entrada. Obligatorio. Tipo: <i>Boolean</i></p> <p>Permite al software de gestión para puntos de venta indicar si, bajo la impresión del código de barras, también se imprimen los caracteres indicados en el argumento codigo.</p> <p><i>True</i> Imprimir caracteres. <i>False</i> No imprimir.</p>
ahora	<p>De entrada. Obligatorio. Tipo: <i>Boolean</i></p>

Permite al software de gestión para puntos de venta indicar si el código de barras se imprime en cuanto es procesado el string de comando, o queda programado para imprimise cuando corresponda, de acuerdo al comprobante en emisión.

- | | |
|--------------|---------------------|
| <i>True</i> | Imprimir ahora. |
| <i>False</i> | Impresión diferida. |

Ejemplo de uso:

```
...
HASAR1.ImprimirCodigoDeBarras CODIGO_TIPO_EAN_13, "779123459876", True, False
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El único tipo de comprobante que admite la impresión del código de barras en el momento de la impresión, e incluso imprimir más de un código de barras por comprobante, es el comprobante no fiscal (rollo). En los demás tipos de comprobantes la ubicación de la impresión del código de barras está preestablecida.
- Al indicar el *tipo* y *codigo* del código de barras a imprimir, tener presentes las siguientes pautas:

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

(*) La cantidad máxima de pares de dígitos puede variar no sólo de un modelo a otro de **IFH 1G**, sino también entre documentos disponibles sobre un mismo modelo.

Método: ImprimirFirmaOpcionalEnCierreDNFH()

Comando: *PrintSignDNFH*

Disponible: OCX Fiscal HASAR v051122 y posteriores.

Argumentos: 1

Este método permite al software de gestión para puntos de venta (des)habilitar la impresión de las leyendas correspondientes a firma y aclaración, sólo en aquellos comprobantes que puedan requerirla opcionalmente.

ImprimirFirmaOpcionalEnCierreDNFH imprimir

imprimir	De entrada. Obligatorio. Tipo: <i>Boolean</i>
	Permite al software de gestión para puntos de venta la (des)habilitación, en determinados comprobantes no fiscales homologados, de la impresión de las leyendas correspondientes a firma y aclaración; al pie del cuerpo. Por defecto, la firma y aclaración es impresa siempre.

- | | |
|--------------|------------------------------|
| <i>True</i> | Imprimir firma y aclaración. |
| <i>False</i> | No imprimir. |

Ejemplo de uso:

```
...
HASAR1.ImprimirFirmaOpcionalEnCierreDNFH False
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que el comando generado por este método tiene efecto en la **IFH 1G** siempre y cuando ésta soporte la emisión, y se esté emitiendo –al momento de recibir el comando *PrintSignDNFH*, alguno de los siguientes comprobantes:
Talón de Cambio, Talón de Regalo, Talón de Reparto, Uso Interno Farmacias.

Si bien algunos modelos y versiones de firmware fiscal de **IFH 1G** pueden emitir otros tipos de comprobantes no fiscales homologados que incluyen impresión de firma y aclaración al pie del cuerpo (Notas de Crédito, por ejemplo), el comando generado por el método *ImprimirFirmaOpcionalEnCierre()* sólo tiene efecto sobre los indicados en la lista anterior.

Método: ImprimirInfoEnDNFH()

Comando: *PrintDNFHInfo*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Argumentos: 3

Este método permite al software de gestión para puntos de venta imprimir información en el cuerpo de un comprobante no fiscal homologado.

Es válido el uso de este método en la emisión de cualquier comprobante no fiscal homologado, excepto los siguientes, de estar soportados por la **IFH 1G**: Notas de Crédito, Tiques Nota de Crédito, Remitos, Recibos 'X', Ordenes de Salida, Resúmenes de Cuenta, Cargos a la Habitación, Cotizaciones, DNFHFarmacias, y DNFHReparto.

ImprimirInfoEnDNFH zona, texto, cant

zona	De entrada. Obligatorio. Tipo: <i>Long</i> Permite al software de gestión para puntos de venta indicar la zona de impresión de información. El valor de este argumento puede variar entre 1 y 'n'. El valor de 'n' dependerá del tipo de comprobante no fiscal homologado a emitir.
texto	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar la información a imprimir en la zona del comprobante no fiscal homologado, identificada por el argumento zona .
cant	De entrada. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta indicar una cantidad que se imprimirá asociada a la información suministrada en el argumento texto .

Ejemplo de uso:

...
HASAR1.ImprimirInfoEnDNFH 1, "Litros de pintura...", 150.0
...

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Las zonas se imprimen en forma secuencial. Se debe comenzar por la '1' y continuar en orden creciente.
- El comando generado por el método *ImprimirInfoEnDNFH()* se puede enviar una o más veces (consecutivamente) para imprimir información en la misma zona. No se admite la impresión aleatoria.
- La cantidad de zonas, y la cantidad de veces que el comando se puede enviar a una misma zona es dependiente del tipo de comprobante no fiscal homologado que se está emitiendo.
- La primera vez que se va a imprimir información en una zona, la **IFH 1G** puede imprimir previamente alguna leyenda. La misma no es modificable, ni se puede omitir.
El contenido de la leyenda puede variar de un comprobante a otro aunque el número de zona sea el mismo.
- No se puede imprimir información en la zona 'x' (siendo $x > 1$) del comprobante no fiscal homologado, si no se envió el comando generado por el método *ImprimirInfoEnDNFH()* por lo menos una vez a la zona 'x-1'.
- Se debe consultar el *Manual Comandos* de la **IFH 1G** para conocer la cantidad de zonas que componen un comprobante no fiscal homologado determinado, y la cantidad de veces que se puede enviar el comando generado por el método *ImprimirInfoEnDNFH()* a una misma zona.

Método: ImprimirItem()

Comando: *PrintLineItem*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 5

Este método permite al software de gestión para puntos de venta la impresión del ítem a facturar.

ImprimirItem descrip, cant, precio, alicuota, impt

descrip	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar un texto que corresponde a la descripción del ítem a imprimir. La longitud de la descripción dependerá del modelo de IFH 1G y si se imprime sobre rollo, formulario continuo, u hoja suelta.
cant	De entrada. Obligatorio. Tipo: <i>Double</i> Permite al software de gestión para puntos de venta indicar la cantidad de unidades, y/o fracción, correspondientes al ítem a imprimir. La cantidad de dígitos admisibles en la parte entera, y/o decimal, depende del modelo y versión de firmware fiscal de la IFH 1G en uso.

precio	<p>De entrada. Obligatorio. Tipo: <i>Double</i></p> <p>Permite al software de gestión para puntos de venta indicar el monto correspondiente al precio unitario del ítem a facturar. La cantidad de dígitos admisibles en parte entera es 6, cualquiera sea el modelo y versión de firmware fiscal de la IFH 1G. En la parte decimal se admitirán 2 ó 4 dígitos, según el modelo y versión de firmware fiscal de la IFH 1G en uso.</p> <p>El valor de este argumento se interpreta de acuerdo a la configuración de la propiedad <i>PrecioBase()</i> del OCX Fiscal HASAR.</p>
alicuota	<p>De entrada. Obligatorio. Tipo: <i>Double</i></p> <p>Permite al software de gestión para puntos de venta indicar la alícuota de IVA que afecta al ítem a imprimir.</p>
impint	<p>De entrada. Obligatorio. Tipo: <i>Double</i></p> <p>El valor que el software de gestión para puntos de venta indique en este argumento establece el impuesto interno que afecta al ítem a facturar.</p> <p>El valor de este argumento y la interpretación que de él hace el OCX Fiscal HASAR depende de la configuración de tres de sus propiedades: <i>KIVA()</i>, <i>ImpuestoInternoFijo()</i>, e <i>ImpuestosInternosPorMonto()</i>.</p>

Interpretación	Propiedad	Valor
Factor K	Imp. Int. Fijo	<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>
	Imp. Int. Variable	<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>
	Porcentaje (impuesto interno variable)	<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>
	Monto Fijo (impuesto interno fijo)	<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>
		<i>False</i>
		<i>False</i>

Ejemplo de uso:

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

...
HASAR1.**ImprimirItem** "Venta navideña...", 10.00, 13.75, 21.00, 0.00
...

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El precio unitario puede ser negativo. Sin embargo, la normativa fiscal vigente no permite la venta negativa de ítems no impresos previamente en el comprobante en curso.
Se permiten las correcciones de ventas, pero no los canjes y/o devoluciones.
- Se puede invocar al método *ImprimirItem()* con cantidad, precio unitario e impuestos en cero.
Esto permite la impresión de ítems que se regalan con la compra.
- No se aceptan ítems con precio unitario igual a cero, e impuestos distintos de cero.
- No se distingue entre ítems con alícuota 0% de IVA y productos no gravados. En ambos casos, se debe indicar alícuota de IVA 0%.
- Los montos que se generan por aplicación de impuestos internos que afecten al ítem se muestran asociados a la leyenda "CONCEPTOS NO GRAVADOS" por exigencia de la normativa fiscal vigente.

Método: **ImprimirItemEnCotizacion()**

Comando: *PrintQuotationItem*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 1

Este método permite al software de gestión para puntos de venta imprimir un línea de texto en el cuerpo del comprobante no fiscal homologado identificado como *Cotización*.

ImprimirItemEnCotizacion detalle

detalle	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta imprimir una línea de detalle en el cuerpo de la <i>Cotización</i> .
----------------	---

Ejemplo de uso:

...
HASAR1.**ImprimirItemEnCotizacion** "Texto de detalle..."
...

El desarrollador del software de gestión para puntos de venta debe tener presente que al inicio del cuerpo de la *Cotización*, la **IFH 1G** imprime automáticamente lo siguiente:

De nuestra consideración:

Adjuntamos a continuación la siguiente cotización a fin de ser presentada ante:

Esta leyendas no son modificables, ni se pueden omitir.

Método: ImprimirItemEnCuenta()

Comando: *PrintAccountItem*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 5

Este método permite al software de gestión para puntos de venta imprimir el detalle de un ítem que forma parte del comprobante no fiscal homologado identificado como *Resumen de Cuenta*, o *Cargo a la Habitación*.

ImprimirItemEnCuenta fechacomp, nrocomp, descrip, debe, haber

fechacomp	De entrada. Obligatorio. Tipo: <i>Date</i> Permite al software de gestión para puntos de venta indicar la fecha de un comprobante emitido al cliente por consumos realizados, servicios contratados, pagos anticipados, etc..
nrocomp	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar el número de un comprobante emitido al cliente por consumos realizados, servicios contratados, pagos anticipados, etc..
descrip	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar un breve detalle del consumo realizado, servicio contratado, etc..
debe	De entrada. Obligatorio. Tipo: <i>Double</i> Permite al software de gestión para puntos de venta indicar el monto del comprobante emitido al cliente por consumos realizados, o servicios contratados. Se acepta el valor '0.00' en este argumento. No se aceptan valores negativos.
haber	De entrada. Obligatorio. Tipo: <i>Double</i> Permite al software de gestión para puntos de venta indicar un monto en concepto de anticípos por consumos realizados (o a realizar), o servicios contratados (o a contratar). Se acepta el valor '0.00' en este argumento. No se aceptan valores negativos.

Ejemplo de uso:

...
HASAR1.ImprimirItemEnCuenta "15/10/2013", "0001-00000342", "Spa y Relax", 150.00, 0.00

...

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Al pie del comprobante se imprime el saldo acreedor, saldo deudor, o simplemente saldo, según corresponda.
- No todos los modelos y versiones de firmware fiscal de la **IFH 1G** muestran el saldo acreedor/deudor de la misma forma. Los homologados con anterioridad a la **IFH 1G SMH/P-322F** mostraban montos de saldo negativos. A partir del modelo mencionado, el saldo del comprobante se muestra como se indica a continuación:
 - Saldo > '0.00' (suma de la columna 'Haber' > suma de la columna 'Debe') se imprime 'SALDO ACREEDOR'.
 - Saldo < '0.00' (suma de la columna 'Debe' > suma de la columna 'Haber') se imprime 'SALDO DEUDOR'.
 - Saldo = '0.00' (suma de la columna 'Debe' = suma de la columna 'Haber') se imprime 'SALDO'.

Método: ImprimirItemEnRemito()

Comando: *PrintEmbarkItem*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 2

Este método permite al software de gestión para puntos de venta imprimir el detalle de un ítem en el cuerpo de un comprobante no fiscal homologado identificado como *Remito*, u *Orden de Salida*.

ImprimirItemEnRemito descrip, cant

descrip	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar la descripción del ítem a imprimir en el comprobante.
cant	De entrada. Obligatorio. Tipo: <i>Double</i> Permite al software de gestión indicar la cantidad a remitir del ítem a imprimir en el comprobante. Se puede indicar el valor '0.00' para utilizar la descripción del ítem como texto libre. No todo los modelos y versiones de firmware fiscal de las IFH 1G omiten imprimir la cantidad cuando ésta vale cero.

Ejemplo de uso:

...

HASAR1.ImprimirItemEnRemito "Bulto x 48 latas , 1 lts c/u", 3.00

...

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- De acuerdo a la RG AFIP Nº 259/98 (con sus complementarias y modificatorias)...
 - Anexo II, Tipos de comprobantes, Datos que deben contener...
No se exige la impresión del CAI, y no se exige la letra 'R'.
- De acuerdo a la RG AFIP Nº 1415/03 (con sus complementarias y modificatorias)...
 - Artículo 28...
Los *Remitos* emitidos con controladores llevan la letra 'X', y son válidos.
 - Artículo 4...
De haber oposición entre la RG AFIP Nº 259/98 y la RG AFIP Nº 1415/03, vale lo expresado en la RG AFIP Nº 259/98 siempre que el comprobante se hubiera emitido con una impresora fiscal homologada..

Método: ImprimirPago()

Comando: *TotalTender*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 4

Este método permite al software de gestión para puntos de venta imprimir un medio de pago en el comprobante en curso.

ImprimirPago descrip, monto, txtadic, resto

descrip	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar la descripción del medio de pago que se desea imprimir en el comprobante en curso de emisión.
monto	De entrada. Obligatorio. Tipo: <i>Double</i> Permite al software de gestión para puntos de venta indicar el monto correspondiente al medio de pago que se desea imprimir en el comprobante en curso de emisión.
txtadic	De entrada. Opcional. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar el texto adicional correspondiente al medio de pago con tarjeta de crédito/débito que se desea imprimir en el comprobante en curso de emisión. Este argumento, suele corresponderse con un número de tarjeta.
resto	De salida. Opcional.

Tipo: *Variant*

Permite al software de gestión para puntos de venta obtener el monto pendiente de pago para el comprobante en curso de emisión. Se trata de la diferencia entre el total del comprobante y la suma de los pagos realizados (incluyendo devoluciones de pagos, si los hubiera).

Ejemplo de uso:

```
...
Dim saldo As variant
...
HASAR1.ImprimirPago "PlastiCard...", 100.00, "*****8796", saldo
Debug.Print "Falta pagar/Vuelto: " & saldo
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- La suma de los pagos debe coincidir con el total del comprobante. De no ser así:
 - Si la suma de los pagos excede al total del comprobante, al cerrarlo, dependerá del tipo de comprobante si el monto de pago excedente se ve reflejado en la impresión del último pago impreso. Para la mayoría de las **IFH 1G** se imprime el monto completo del último pago en el caso de un *Tique*, mientras que en el resto de los comprobantes se imprime el monto necesario para alcanzar el total del comprobante.
 - Si la suma de los pagos es inferior al total del comprobante, al cerrarlo, se imprimirá automáticamente como medio de pago la leyenda “Cuenta Corriente” asociándola a la diferencia entre el total del comprobante y la suma de los pagos efectuados por el software de gestión para puntos de venta.
- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** aceptan la misma cantidad de pagos en un comprobante. Agotados los pagos posibles, si el monto del último comando enviado por el software de gestión para puntos de venta no cubre la diferencia entre lo pagado y el total del comprobante, el comando es rechazado por inválido.
- El uso del método *ImprimirPago()* es válido en la emisión de comprobantes fiscales, Notas de Crédito, y Tiques Nota de Crédito.
- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan al comando *TotalTender* en la emisión de Notas de Crédito/Tiques Nota de Crédito.

Método: **ImprimirTextoFiscal()**

Comando: *PrintFiscalText*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 1

Este método permite al software de gestión para puntos de venta extender la impresión de la descripción de un ítem, o bien imprimir leyendas asociadas a un ítem en particular.

ImprimirTextoFiscal texto

texto	<p>De entrada. Obligatorio. Tipo: <i>String</i></p> <p>Permite al software de gestión para punto de venta indicar el texto a imprimir en forma previa a la impresión del ítem.</p>
--------------	--

Ejemplo de uso:

```
...
HASAR1.ImprimirTextoFiscal "Guía de despacho Nº: ...."
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- La longitud del texto dependerá del modelo y versión de firmware fiscal de la **IFH 1G** y del tipo de comprobante sobre el que se realizará la impresión.
- El comando generado por este método es de aplicación en comprobantes fiscales, Notas de Crédito, y Tiques Nota de Crédito.
- La cantidad de líneas de texto fiscal a imprimir (previas a la impresión del ítem) pueden variar de acuerdo al modelo y versión de firmware fiscal de la **IFH 1G** y del tipo de comprobante sobre el que se realizará la impresión.

El **OCX Fiscal HASAR** también ofrece la propiedad *DescripcionesLargas()* para lograr una distribución automática de la descripción del ítem, empleando -de ser necesarias- líneas de texto fiscal.

Método: **ImprimirTextoNoFiscal()**

Comando: *PrintNonFiscalText*

Disponible: **OCX Fiscal HASAR**, en todas las versiones

Argumentos: 1

Este método permite al software de gestión para puntos de venta la impresión de líneas de texto en el cuerpo de un comprobante no fiscal.

ImprimirTextoNoFiscal texto

texto	<p>De entrada. Obligatorio. Tipo: <i>String</i></p> <p>Permite al software de gestión para punto de venta indicar el texto a imprimir en el cuerpo del comprobante no fiscal.</p>
--------------	---

Ejemplo de uso:

```
...
HASAR1.ImprimirTextoNoFiscal "Contenido de una línea de texto no fiscal..."
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Por exigencia de la normativa fiscal vigente, en los comprobantes no fiscales se debe imprimir automáticamente la leyenda “*** No fiscal ***” cada cuatro líneas de impresión de texto no fiscal.
- En el caso de impresión sobre formularios continuos u hojas sueltas, el uso a dar a los comprobantes no fiscales se encuentra en una lista dada a conocer en la normativa fiscal vigente. La impresión sobre rollos no tiene limitaciones.
- Si el primer carácter del texto es el ASCII 244 (decimal), será interpretado por la **IFH 1G** como un carácter de control, y el texto no fiscal se imprimirá en doble ancho.

El **OCX Fiscal HASAR** ofrece la propiedad *DobleAncho()* para concatenar al texto no fiscal a imprimir.

Por ejemplo,

HASAR1.ImprimirTextoNoFiscal HASAR1.DobleAncho & “Texto no fiscal...”

- El método *ImprimirTextoNoFiscal()* se puede invocar cuantas veces sea necesario.
- La longitud del texto dependerá del modelo y versión de firmware fiscal de la **IFH 1G**, si la impresión se realizará en rollo o sobre formulario continuo (u hoja suelta), y si se imprimirá en doble ancho (la longitud se reduce a la mitad).

Método: ImprimirVoucher()

Comando: *SetVoucherData1, SetVoucherData2, PrintVoucher*

El **OCX Fiscal HASAR** genera los tres comandos, y en el orden dado, por cada invocación al método.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 19

Este método permite al software de gestión para puntos de venta la impresión de un voucher asociado a una transacción realizada con una tarjeta de crédito/débito.

ImprimirVoucher cliente, tarjeta, tipovouch, nrotarj, fechavto, tipotarj, cuotas, codcomer, nroterm, _nrobole, nrocargo, ingreso, oper, nroaut, monto, nrocomp, copias, vendedor, estac

cliente	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar el nombre del cliente, titular de la tarjeta de crédito/débito.
tarjeta	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar la identificación del plástico involucrado en la transacción. Por ejemplo, Visa, Mastercard, American Express, etc..
tipovouch	De entrada. Obligatorio. Tipo: enum <i>TiposDeVouchers</i>

	<p>Permite al software de gestión para puntos de venta indicar el carácter de la transacción.</p> <p>Este argumento sólo puede ser alguno de los valores identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR:</p> <p>VOUCHER_DE_COMPRA, VOUCHER_DE_CANCELACION_COMPRA, VOUCHER_DE_DEVOLUCION, VOUCHER_DE_CANCELACION_DEVOLUCION.</p>
nrotarj	<p>De entrada. Obligatorio.</p> <p>Tipo: <i>String</i></p>
	<p>Permite al software de gestión para puntos de venta indicar el número de la tarjeta interveniente en la transacción.</p>
fechavto	<p>De entrada. Obligatorio.</p> <p>Tipo: <i>Date</i></p>
	<p>Permite al software de gestión para puntos de venta indicar la fecha de vencimiento de la tarjeta interveniente en la transacción.</p>
tipotarj	<p>De entrada. Obligatorio.</p> <p>Tipo: enum <i>TiposDeTarjetas</i></p>
	<p>Permite al software de gestión para puntos de venta indicar si se trata de una tarjeta de crédito o débito.</p>
	<p>Este argumento sólo puede ser alguno de los valores identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR:</p> <p>TARJETA_CREDITO, TARJETA_DEBITO.</p>
cuotas	<p>De entrada. Obligatorio.</p> <p>Tipo: <i>Long</i></p>
	<p>Permite al software de gestión para puntos de venta indicar la cantidad de cuotas de pago que se acordó con el cliente, por el monto total de la transacción.</p>
codcomer	<p>De entrada. Obligatorio.</p> <p>Tipo: <i>String</i></p>
	<p>Permite al software de gestión para puntos de venta indicar el número de identificación de comercio/establecimiento.</p>
nroterm	<p>De entrada. Obligatorio.</p> <p>Tipo: <i>Long</i></p>
	<p>Permite al software de gestión para puntos de venta indicar el número de identificación de la terminal utilizada.</p>
nrolote	<p>De entrada. Obligatorio.</p> <p>Tipo: <i>Long</i></p>
	<p>Permite al software de gestión para puntos de venta indicar el número de identificación de</p>

	un lote de transacciones.
nrocargo	<p>De entrada. Obligatorio. Tipo: <i>Long</i></p> <p>Permite al software de gestión para puntos de venta indicar el número de identificación de un cargo por operaciones de anulaciones/devoluciones.</p>
ingreso	<p>De entrada. Obligatorio. Tipo: enum <i>TiposDeIngresoDeTarjeta</i></p> <p>Permite al software de gestión para puntos de venta indicar la forma en que fue ingresado al sistema el número de la tarjeta interveniente en la transacción.</p> <p>Este argumento sólo puede tomar alguno de los valores identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR: INGRESO_DE_TARJETA_MANUAL, INGRESO_DE_TARJETA_AUTOMATIZADO.</p>
oper	<p>De entrada. Obligatorio. Tipo: <i>TiposOperacionDeTarjeta</i></p> <p>Permite al software de gestión para puntos de venta indicar si la transacción fue realizada en forma <i>offline</i>, <i>u online</i>.</p> <p>Este argumento sólo puede tomar alguno de los valores identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR: OPERACION_TARJETA_ONLINE, OPERACIÓN_TARJETA_OFFLINE.</p>
nroaut	<p>De entrada. Obligatorio. Tipo: <i>Long</i></p> <p>Permite al software de gestión para puntos de venta indicar el número de autorización otorgado a la transacción.</p>
monto	<p>De entrada. Obligatorio. Tipo: <i>String</i></p> <p>Permite al software de gestión para puntos de venta indicar el monto por el cual se realiza la transacción.</p>
nrocomp	<p>De entrada. Obligatorio. Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta indicar el número del comprobante correspondiente a las ventas que dieron origen a la transacción.</p>
copias	<p>De entrada. Obligatorio. Tipo: <i>Long</i></p> <p>Permite al software de gestión para puntos de venta indicar la cantidad de ejemplares que se desea emitir del voucher correspondiente a la transacción.</p>

vendedor	De entrada. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta indicar el nombre del vendedor involucrado en la transacción.
estac	De entrada. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta indicar la estación de impresión del voucher correspondiente a la transacción.

Ejemplo de uso:

```
...
HASAR1.ImprimirVoucher "Cliente...", "PlastiCard...", VOUCHER_DE_COMPRA, "1234567890123456", _
"18/11/2013", TARJETA_CREDITO, 3, "123", 567, 890, 112, _
INGRESO_DE_TARJETA_AUTOMATIZADO, OPERACIÓN_TARJETA_ONLINE, _
3456, "432.56", "0001-0000567", 2, "Vendedor...", "Estación..."
```

...

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Se puede invocar al método *ImprimirVoucher()* en forma consecutiva, la cantidad de veces que resulte necesario.

Método: ImprimirVoucherConFechaOculta()

Comando: *SetVoucherData1, SetVoucherData2, PrintVoucher*

El OCX Fiscal HASAR genera los tres comandos, y en el orden dado, por cada invocación al método.

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Argumentos: 18

Este método permite al software de gestión para puntos de venta la impresión de un voucher asociado a una transacción realizada con una tarjeta de crédito/débito.

ImprimirVoucher cliente, tarjeta, tipovouch, nrotarj, tipotarj, cuotas, codcomer, nroterm, nrobole, nrocargo, ingreso, oper, nroaut, monto, nrocomp, copias, vendedor, estac

cliente	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar el nombre del cliente, titular de la tarjeta de crédito/débito.
tarjeta	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar la identificación del plástico in-

	<p>teviente en la transacción. Por ejemplo, Visa, Mastercard, American Express, etc..</p>
tipovouch	<p>De entrada. Obligatorio. Tipo: enum <i>TiposDeVouchers</i></p> <p>Permite al software de gestión para puntos de venta indicar el carácter de la transacción.</p> <p>Este argumento sólo puede ser alguno de los valores identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR:</p> <p>VOUCHER_DE_COMPRA, VOUCHER_DE_CANCELACION_COMPRA, VOUCHER_DE_DEVOLUCION, VOUCHER_DE_CANCELACION_DEVOLUCION.</p>
nrotarj	<p>De entrada. Obligatorio. Tipo: <i>String</i></p> <p>Permite al software de gestión para puntos de venta indicar el número de la tarjeta interveniente en la transacción.</p>
tipotarj	<p>De entrada. Obligatorio. Tipo: enum <i>TiposDeTarjetas</i></p> <p>Permite al software de gestión para puntos de venta indicar si se trata de una tarjeta de crédito o débito.</p> <p>Este argumento sólo puede ser alguno de los valores identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR:</p> <p>TARJETA_CREDITO, TARJETA_DEBITO.</p>
cuotas	<p>De entrada. Obligatorio. Tipo: <i>Long</i></p> <p>Permite al software de gestión para puntos de venta indicar la cantidad de cuotas de pago en que se acordó con el cliente, por el monto total de la transacción.</p>
codcomer	<p>De entrada. Obligatorio. Tipo: <i>String</i></p> <p>Permite al software de gestión para puntos de venta indicar el número de identificación de comercio/establecimiento.</p>
nroterm	<p>De entrada. Obligatorio. Tipo: <i>Long</i></p> <p>Permite al software de gestión para puntos de venta indicar el número de identificación de la terminal utilizada.</p>
nrolote	<p>De entrada. Obligatorio. Tipo: <i>Long</i></p> <p>Permite al software de gestión para puntos de venta indicar el número de identificación de un lote de transacciones.</p>

nrocargo	De entrada. Obligatorio. Tipo: <i>Long</i> Permite al software de gestión para puntos de venta indicar el número de identificación de un cargo por operaciones de anulaciones/devoluciones.
ingreso	De entrada. Obligatorio. Tipo: enum <i>TiposDeIngresoDeTarjeta</i> Permite al software de gestión para puntos de venta indicar la forma en que fue ingresado al sistema el número de la tarjeta interveniente en la transacción. Este argumento sólo puede tomar alguno de los valores identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR : INGRESO_DE_TARJETA_MANUAL, INGRESO_DE_TARJETA_AUTOMATIZADO.
oper	De entrada. Obligatorio. Tipo: <i>TiposOperacionDeTarjeta</i> Permite al software de gestión para puntos de venta indicar si la transacción fue realizada en forma <i>offline</i> , u <i>online</i> . Este argumento sólo puede tomar alguno de los valores identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR : OPERACION_TARJETA_ONLINE, OPERACION_TARJETA_OFFLINE.
nroaut	De entrada. Obligatorio. Tipo: <i>Long</i> Permite al software de gestión para puntos de venta indicar el número de autorización otorgado a la transacción.
monto	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar el monto por el cual se realiza la transacción.
nrocomp	De entrada. Obligatorio. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta indicar el número del comprobante correspondiente a las ventas que dieron origen a la transacción.
copias	De entrada. Obligatorio. Tipo: <i>Long</i> Permite al software de gestión para puntos de venta indicar la cantidad de ejemplares que se desea emitir del voucher correspondiente a la transacción.
vendedor	De entrada. Opcional.

	<p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta indicar el nombre del vendedor involucrado en la transacción.</p>
estac	<p>De entrada. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta indicar la estación de impresión del voucher correspondiente a la transacción.</p>

Ejemplo de uso:

...

```
HASAR1.ImprimirVoucher "Cliente...", "PlastiCard...", VOUCHER_DE_COMPRA, "1234567890123456", _  
TARJETA_CREDITO, 3, "123", 567, 890, 112, _  
INGRESO_DE_TARJETA_AUTOMATIZADO, OPERACIÓN_TARJETA_ONLINE, _  
3456, "432.56", "0001-0000567", 2, "Vendedor...", "Estación..."
```

...

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Se puede invocar al método *ImprimirVoucherConFechaOculta()* en forma consecutiva, la cantidad de veces que resulte necesario.
- A diferencia del método *ImprimirVoucher()*, este método no recibe la fecha de vencimiento de la tarjeta como argumento. Ello se debe a que el **OCX Fiscal HASAR** genera el string de comando con la fecha ya enmascarada. En lugar de 'MMAA' envía '****'.
- Con este método no es posible evitar el enmascaramiento de la fecha de vencimiento, ni es posible modificar la forma en que lo hace.
- No todos los modelos y versiones de firmware fiscal de las **Impresoras Fiscales HASAR** soportan el enmascaramiento de la fecha de vencimiento de la tarjeta.

Método: LeerIVA()

Comando: *SendFirstIVA, NextIVATransmission*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Argumentos: 6

Retorno: *Boolean*

Este método permite al software de gestión para puntos de venta obtener el desglose de impuestos correspondiente a cada alícuota de IVA que forma parte del comprobante (hasta el momento de la consulta), o bien del comprobante completo (si la consulta se hace inmediatamente después de cerrado el comprobante), o bien el desglose correspondiente a los valores acumulados en la memoria de trabajo de la **IFH 1G**.

LeerIVA doc, alic, miva, impint, ivanoi, base

doc	De entrada. Obligatorio.
	Tipo: enum <i>DocumentosIVA</i>

	<p>Permite al software de gestión para puntos de venta indicar si la discriminación solicitada corresponde a un comprobante fiscal), o a una Nota de Crédito/Tique Nota de Crédito.</p> <p>Este argumento sólo puede tomar alguno de los valores identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR:</p> <p>DOCUMENTOS_FISCALES, NOTAS_DE_CREDITO.</p>
alic	<p>De salida. Obligatorio.</p> <p>Tipo: <i>Double</i></p> <p>Permite al software de gestión para puntos de venta obtener la alícuota de IVA a la que se encuentra asociada la información del resto de los argumentos de salida.</p>
miva	<p>De salida. Obligatorio.</p> <p>Tipo: <i>Double</i></p> <p>Permite al software de gestión para puntos de venta obtener el monto de IVA acumulado hasta el momento de la consulta, para la alícuota de IVA alic.</p>
impint	<p>De salida. Obligatorio.</p> <p>Tipo: <i>Double</i></p> <p>Permite al software de gestión para puntos de venta obtener el monto de impuestos internos acumulado hasta el momento de la consulta, para la alícuota de IVA alic.</p>
ivanoi	<p>De salida. Obligatorio.</p> <p>Tipo: <i>Double</i></p> <p>La IFH 1G entrega siempre '0.00'.</p> <p>La sobretasa de IVA No Inscripto no es de aplicación en la normativa fiscal vigente.</p>
base	<p>De salida. Obligatorio.</p> <p>Tipo: <i>Double</i></p> <p>Permite al software de gestión para puntos de venta obtener el monto neto acumulado hasta el momento de la consulta, para la alícuota de IVA alic.</p>

Ejemplo de uso:

...

```
Dim Alic As Double, MIVA As Double, ImplInt As Double, IVANol As Double, Basel As Double
Dim msg As String
```

```
HASAR1.ResumenIVA = RESUMEN_IVA_DOCUMENTO
```

```
While (HASAR1.LeerIVA(DOCUMENTOS_FISCALES, Alic, MIVA, ImplInt, IVANol, Basel))
    msg = "Alícuota IVA" & "=" & Alic & vbCrLf & _
        "Monto IVA" & "=" & MIVA & vbCrLf & _
        "Monto Imp. Int." & "=" & ImplInt & vbCrLf & _
        "Monto Neto" & "=" & Basel
```

Debug.Print msg
Wend
...

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- La propiedad *ResumenIVA()* permite al **OCX Fiscal HASAR** generar el comando de consulta indicando que se desea información correspondiente al comprobante en curso (o recién emitido), o correspondiente a la memoria de trabajo de la **IFH 1G**.
- El método *LeerIVA()* retorna *True* por cada registro de información obtenido de la **IFH 1G**.
- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la consulta de desglose de impuestos mientras haya un comprobante abierto.
- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la consulta de desglose de impuestos luego de un cierre diario 'Z'.

Método: LeerMemoriaDeTrabajo()

Comando: *GetWorkingMemory*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 23

Permite al software de gestión para puntos de venta consultar la información almacenada, hasta el momento de la consulta, en la memoria de trabajo de la **IFH 1G**.

LeerMemoriaDeTrabajo **cantdfcancel**, **cantdnfemit**, **cantdfemit**, **nroultdfbc**, **nroultdfa**, **totaldf**, _
totivadf, **totimpintdf**, **totpercdf**, **ivanoidf**, **nroultncbc**, **nroultnca**, **totalnc**, _
totivanc, **totimpointnc**, **totpercncc**, **ivanoinc**, **nroultrto**, **cantncccancel**, _
cantdfbemit, **cantdfaemit**, **cantncbemit**, **cantncaemit**

cantdfcancel	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener la cantidad de comprobantes fiscales cancelados durante la jornada fiscal, y hasta el momento de la consulta.
cantdnfemit	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener la cantidad de comprobantes no fiscales emitidos durante la jornada fiscal, y hasta el momento de la consulta.
cantdfemit	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener la cantidad de comprobantes fiscales emitidos durante la jornada fiscal, y hasta el momento de la consulta.
nroultdfbc	De salida. Opcional.

	<p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener el número del último comprobante fiscal ‘B/C’ de la jornada fiscal, haya sido emitido o cancelado.</p>
nroultdfa	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener el número del último comprobante fiscal ‘A’ de la jornada fiscal, haya sido emitido o cancelado.</p>
totaldf	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener el monto acumulado de ventas, en concepto de emisión de comprobantes fiscales.</p>
totivadf	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener el monto acumulado de IVA (suma de todas las alícuotas), en concepto de emisión de comprobantes fiscales.</p>
totimpintdf	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener el monto acumulado de impuestos internos, en concepto de emisión de comprobantes fiscales.</p>
totpercdf	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener el monto acumulado de percepciones, en concepto de emisión de comprobantes fiscales.</p>
ivanoidf	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Siempre ‘0.00’. La sobretasa de IVA No inscripto no es de aplicación en la normativa fiscal vigente.</p>
nroultncbc	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener el número de la última Nota de Crédito (Tique Nota de Crédito) ‘B/C’ de la jornada fiscal, haya sido emitida o cancelada.</p>
nroultnca	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p>

	Permite al software de gestión para puntos de venta obtener el número de la última Nota de Crédito (Tique Nota de Crédito) 'A' de la jornada fiscal, haya sido emitida o cancelada.
totalinc	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el monto total acumulado, en concepto de emisión de Notas de Crédito (Tiques Nota de Crédito).
totivanc	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el monto acumulado de IVA, en concepto de emisión de Notas de Crédito (Tiques Nota de Crédito).
totimpoinc	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el monto acumulado de impuestos internos, en concepto de emisión de Notas de Crédito (Tiques Nota de Crédito).
totpercnc	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el monto acumulado de percepciones, en concepto de emisión de Notas de Crédito (Tiques Nota de Crédito).
ivanoinc	De salida. Opcional. Tipo: <i>Variant</i>
	Siempre '0.00'. La sobretasa de IVA No Inscripto no es de aplicación en la normativa fiscal vigente.
nroultrto	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el número del último Remito de la jornada fiscal, haya sido emitido o cancelado.
cantnccancel	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener la cantidad de Notas de Crédito (Tiques Nota de Crédito) canceladas durante la jornada fiscal.
cantdfbemit	De salida. Opcional. Tipo: <i>Variant</i>

	Permite al software de gestión para puntos de venta obtener la cantidad de comprobantes fiscales 'B/C' emitidos durante la jornada fiscal.
cantdfaemit	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener la cantidad de comprobantes fiscales 'A' emitidos durante la jornada fiscal.
cantncbemit	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener la cantidad de Notas de Crédito (Tiques Nota de Crédito) 'B/C' emitidas durante la jornada fiscal.
cantncaemit	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener la cantidad de Notas de Crédito (Tiques Nota de Crédito) 'A' emitidas durante la jornada fiscal.

Ejemplo de uso:

```
...
Dim ndfcancel As Variant, ndnfemmit As Variant, ndfemmit As Variant, nrodfbc As Variant, nrodfa As Variant
Dim totdf As Variant, ivadf As Variant, iidf As Variant, percdf As Variant, nroncbc As Variant
Dim nronca As Variant, totnc As Variant, ivanc As Variant, iinc As Variant, percnc As Variant
Dim nrorto As Variant, nnccancel As Variant, ndfbcemit As Variant, ndfaemit As Variant
Dim nnbcemit As Variant, nncaemit As Variant

HASAR1.LeerMemoriaDeTrabajo ndfcancel, ndnfemmit, ndfemmit, nrodfbc, nrodfa, totdf, ivadf, iidf, percdf, ,_
nroncbc, nronca, totnc, ivanc, iinc, percnc, , nrorto, nnccancel, ndfbcemit,_
ndfaemit, nnbcemit, nncaemit
```

```
Debug.Print "DF cancelados      :" & ndfcancel
Debug.Print "DNF Emitidos       :" & ndnfemmit
Debug.Print "Ult DF B/C          :" & nrodfbc
Debug.Print "Ult DF A            :" & nrodfa
Debug.Print "Total DF             :" & totdf
Debug.Print "IVA DF              :" & ivadf
Debug.Print "Imp Int DF          :" & iidf
Debug.Print "Percep DF            :" & percdf
Debug.Print "Ult NC B/C           :" & nroncbc
Debug.Print "Ult NC A             :" & nronca
Debug.Print "Total NC              :" & totnc
Debug.Print "IVA NC               :" & ivanc
Debug.Print "Imp Int NC           :" & iinc
Debug.Print "Percep NC            :" & percnc
Debug.Print "Ult Remito           :" & nrorto
Debug.Print "Cant NC Cancel       :" & nnccancel
Debug.Print "Cant DF B/C Emit     :" & ndfbcemit
```

```

Debug.Print "Cant DF A Emit : " & ndfaemit
Debug.Print "Cant NC B/C Emit : " & nncbcemit
Debug.Print "Cant NC A Emit : " & nncaemit
...

```

El desarrollador del software de gestión para puntos debe tener presente que:

- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** acepta esta consulta cuando un comprobante se encuentra abierto.
- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** tienen disponible toda la información posible de obtener.

Método: LeerMontoTotal()

Comando: *SendFirstIVA, NextIVATransmission*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Argumentos: 2

Este método permite al software de gestión para puntos de venta obtener el monto total del comprobante (hasta el momento de la consulta), o bien del comprobante completo (si la consulta se hace inmediatamente después de cerrado el comprobante), o bien el total acumulado en la memoria de trabajo de la **IFH 1G**.

LeerMontoTotal doc, monto

doc	<p>De entrada. Obligatorio. Tipo: enum <i>DocumentosIVA</i></p> <p>Permite al software de gestión para puntos de venta indicar si el acumulado solicitado corresponde a un comprobante fiscal, o a una Nota de Crédito (Tique Nota de Crédito).</p> <p>Este argumento sólo puede tomar alguno de los valores identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR:</p> <p>DOCUMENTOS_FISCALES, NOTAS_DE_CREDITO.</p>
monto	<p>De salida. Obligatorio. Tipo: <i>Double</i></p> <p>Permite al software de gestión para puntos de venta obtener el monto acumulado (incluidos impuestos), al momento de la consulta, del comprobante consultado, o de la memoria de trabajo de la IFH 1G.</p>

Ejemplo de uso:

```

...
Dim MTotal As Double
...
HASAR1.ResumenIVA = RESUMEN_IVA_DOCUMENTO
...
HASAR1.LeerMontoTotal DOCUMENTOS_FISCALES, MTotal
Debug.Print "Monto Total" & "=" & MTotal

```

...

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- La propiedad *ResumenIVA()* permite al **OCX Fiscal HASAR** generar el comando de consulta indicando que se desea información correspondiente al comprobante en curso (o recién emitido), o correspondiente a la memoria de trabajo de la **IFH 1G**.
- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la consulta del total acumulado mientras haya un comprobante abierto.
- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la consulta de total acumulado luego de un cierre diario 'Z'.

Método: LeerPercepcionesGenerales()

Comando: *SendFirstIVA, NextIVATransmission*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Argumentos: 2

Retorno: *Boolean*

Este método permite al software de gestión para puntos de venta obtener el monto acumulado de percepciones generales del comprobante (hasta el momento de la consulta), o bien del comprobante completo (si la consulta se hace inmediatamente después de cerrado el comprobante), o bien el acumulado de percepciones generales en la memoria de trabajo de la **IFH 1G**.

LeerPercepcionesGenerales doc, monto

doc	De entrada. Obligatorio. Tipo: enum <i>DocumentosIVA</i> Permite al software de gestión para puntos de venta indicar si el acumulado solicitado corresponde a un comprobante fiscal, o a una Nota de Crédito (Tique Nota de Crédito). Este argumento sólo puede tomar alguno de los valores identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR : DOCUMENTOS_FISCALES, NOTAS_DE_CREDITO.
monto	De salida. Obligatorio. Tipo: <i>Double</i> Permite al software de gestión para puntos de venta obtener el monto acumulado de percepciones generales, al momento de la consulta, del comprobante consultado, o de la memoria de trabajo de la IFH 1G .

Ejemplo de uso:

```
...
Dim MPercep As Double, DescPercep As String, MontoPercep As Double
...
HASAR1.ResumenIVA = RESUMEN_IVA_DOCUMENTO
```

```

...
While (HASAR1.LeerPercepcionesGenerales(DOCUMENTOS_FISCALES, MPercep))
    Debug.Print "Monto Percepciones" & "=" & MPercep
Wend
...

```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- La propiedad *ResumenIVA()* permite al **OCX Fiscal HASAR** generar el comando de consulta indicando que se desea información correspondiente al comprobante en curso (o recién emitido), o correspondiente a la memoria de trabajo de la **IFH 1G**.
- El método *LeerPercepcionesGenerales()* retorna *True* por cada registro de información obtenido de la **IFH 1G**.
- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la consulta de acumulados de percepciones generales mientras haya un comprobante abierto.
- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la consulta de acumulado de percepciones generales luego de un cierre diario 'Z'.

Método: LeerPercepcionesPorDescripción()

Comando: *SendFirstIVA, NextIVATransmission*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Argumentos: 3

Retorno: *Boolean*

Este método permite al software de gestión para puntos de venta obtener el monto acumulado de percepciones por descripción del comprobante (hasta el momento de la consulta), o bien del comprobante completo (si la consulta se hace inmediatamente después de cerrado el comprobante), o bien el acumulado de percepciones por descripción en la memoria de trabajo de la **IFH 1G**.

LeerPercepcionesPorDescripción doc, descrip, monto

doc	<p>De entrada. Obligatorio. Tipo: enum <i>DocumentosIVA</i></p> <p>Permite al software de gestión para puntos de venta indicar si el acumulado solicitado corresponde a un comprobante fiscal, o a una Nota de Crédito (Tique Nota de Crédito).</p> <p>Este argumento sólo puede tomar alguno de los valores identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR: <i>DOCUMENTOS_FISCALES, NOTAS_DE_CREDITO</i>.</p>
descrip	<p>De salida. Obligatorio. Tipo: <i>String</i></p> <p>Permite al software de gestión para puntos de venta obtener la descripción de una de las percepciones acumuladas por descripción hasta el momento de la consulta (del comprobante en curso, o del recién emitido, o de la memoria de trabajo de la IFH 1G).</p>

monto	<p>De salida. Obligatorio. Tipo: <i>Double</i></p> <p>Permite al software de gestión para puntos de venta obtener el monto acumulado de una de las percepciones acumuladas por descripción hasta el momento de la consulta (del comprobante en curso, o del recién emitido, o de la memoria de trabajo de la IFH 1G).</p>
--------------	--

Ejemplo de uso:

```
...
Dim MPercep As Double, DescPercep As String, MontoPercep As Double
...
HASAR1.ResumenIVA = RESUMEN_IVA_DOCUMENTO
...
While (HASAR1.LeerPercepcionesPorDescripcion(DOCUMENTOS_FISCALES, DescPercep, MontoPercep))
    Debug.Print "Descripción Percep.= " & DescPercep & vbCrLf & "Monto Percepción = " & MontoPercep
Wend
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- La propiedad *ResumenIVA()* permite al **OCX Fiscal HASAR** generar el comando de consulta indicando que se desea información correspondiente al comprobante en curso (o recién emitido), o correspondiente a la memoria de trabajo de la **IFH 1G**.
- El método *LeerPercepcionesPorDescripcion()* retorna *True* por cada registro de información obtenido de la **IFH 1G**.
- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la consulta de acumulados de percepciones por descripción mientras haya un comprobante abierto.
- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la consulta de acumulado de percepciones por descripción luego de un cierre diario 'Z'.
- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** acumulan percepciones por descripción.

Método: LeerPercepcionesPorIVA()

Comando: *SendFirstIVA, NextIVATransmission*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Argumentos: 3

Retorno: *Boolean*

Este método permite al software de gestión para puntos de venta obtener el monto acumulado de percepciones por IVA del comprobante (hasta el momento de la consulta), o bien del comprobante completo (si la consulta se hace inmediatamente después de cerrado el comprobante), o bien el acumulado de percepciones por IVA en la memoria de trabajo de la **IFH 1G**.

LeerPercepcionesPorIVA doc, alic, monto

doc	De entrada. Obligatorio.
------------	--------------------------

	<p>Tipo: enum <i>DocumentosIVA</i></p> <p>Permite al software de gestión para puntos de venta indicar si el acumulado solicitado corresponde a un comprobante fiscal, o a una Nota de Crédito (Tique Nota de Crédito).</p> <p>Este argumento sólo puede tomar alguno de los valores identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR:</p> <p>DOCUMENTOS_FISCALES, NOTAS_DE_CREDITO.</p>
alic	<p>De salida. Obligatorio.</p> <p>Tipo: <i>Double</i></p> <p>Permite al software de gestión para puntos de venta obtener la alícuota de IVA de una de las percepciones acumuladas por IVA hasta el momento de la consulta (del comprobante en curso, o del recién emitido, o de la memoria de trabajo de la IFH 1G).</p>
monto	<p>De salida. Obligatorio.</p> <p>Tipo: <i>Double</i></p> <p>Permite al software de gestión para puntos de venta obtener el monto acumulado de una de las percepciones acumuladas por IVA hasta el momento de la consulta (del comprobante en curso, o del recién emitido, o de la memoria de trabajo de la IFH 1G).</p>

Ejemplo de uso:

```

...
Dim MPercep As Double, Alic As Double, MontoPercep As Double
...
HASAR1.ResumenIVA = RESUMEN_IVA_DOCUMENTO
...
While (HASAR1.LeerPercepcionesPorIVA(DOCUMENTOS_FISCALES, Alic, MontoPercep))
    Debug.Print "Alícuota IVA = " & Alic & vbCrLf & "Monto Percepción = " & MontoPercep
Wend
...

```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- La propiedad *ResumenIVA()* permite al **OCX Fiscal HASAR** generar el comando de consulta indicando que se desea información correspondiente al comprobante en curso (o recién emitido), o correspondiente a la memoria de trabajo de la **IFH 1G**.
- El método *LeerPercepcionesPorIVA()* retorna *True* por cada registro de información obtenido de la **IFH 1G**.
- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la consulta de acumulados de percepciones por IVA mientras haya un comprobante abierto.
- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la consulta de acumulado de percepciones por IVA luego de un cierre diario 'Z'.

Método: MostrarEnDisplay()

Comando: *WriteDisplay*

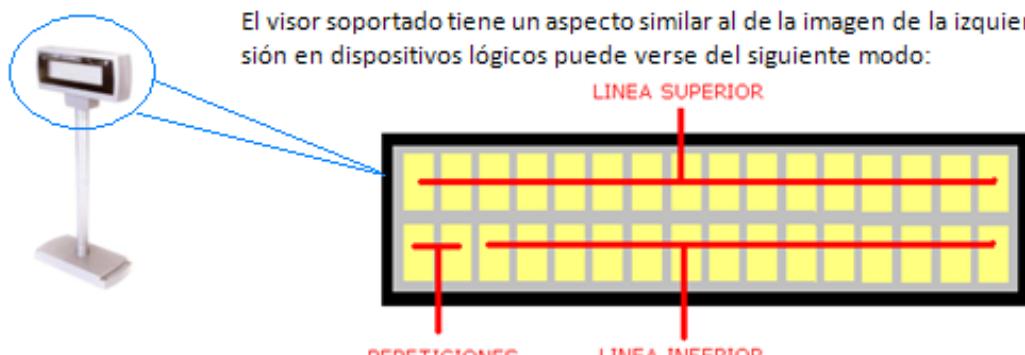
Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 2

Este método permite al software de gestión para puntos de venta mostrar información en el visor de cliente conectado a la **IFH 1G**.

MostrarEnDisplay texto, destino

texto	De entrada. Obligatorio. Tipo: <i>String</i> Permite al software de gestión para puntos de venta indicar el texto a mostrar en el visor del cliente, conectado a la IFH 1G .
destino	De entrada. Obligatorio. Tipo: enum <i>TiposDeDisplay</i> Permite al software de gestión para puntos de venta indicar en que parte del visor del cliente se muestra la información deseada. Este argumento sólo puede tomar alguno de los valores identificados por las siguientes constantes embebidas en el OCX Fiscal HASAR : LINEA_SUPERIOR , LINEA_INFERIOR , SECCION_DE_REPEATICION . Hasta el momento, los únicos modelos de IFH 1G con soporte de conexión (opcional) -y discontinuados de producción- para visor de cliente, son SMH/P-614F y SMH/P-615F. En estos casos, el visor del cliente admite dos líneas de 16 caracteres cada una, donde se participa al dispositivo físico en tres dispositivos lógicos: superior, inferior y repeticiones.



Ejemplo de uso:

```
...
HASAR1.MostrarEnDisplay "Yerba x 1 kg...", LINEA_SUPERIOR
...
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que no todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la conexión de un visor de cliente.

Método: ObtenerCAIUltimoDocumentoA()

Comando: *GetLastCAI*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Argumentos: 4

Este método permite al software de gestión para puntos de venta obtener el número de CAI, e información complementaria, del último comprobante 'A' emitido en la jornada fiscal.

ObtenerCAIUltimoDocumentoA doc, hojaini, hojafin, cai

doc	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener la identificación del último comprobante 'A' emitido. Los posibles valores de retorno coinciden con los valores de las siguientes constantes embebidas en el OCX Fiscal HASAR . D_FACTURA_A, D_NOTA_DEBITO_A, D_RECIBO_A, D_NOTA_CREDITO_A.
hojaini	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el número de comprobante correspondiente a la primera hoja del documento doc .
hojafin	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el número de comprobante correspondiente a la última hoja del documento doc .
cai	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el número de CAI impreso al pie de la última hoja del documento doc .

Ejemplo de uso:

```
...
Dim doc As Variant, nroini As Variant, nrofin As Variant, nrocai As Variant
...
HASAR1.ObtenerCAIUltimoDocumentoA doc, nroini, nrofin, nrocai
Debug.Print "Tipo Doc 'A'      : " & doc
Debug.Print "Nro primera hoja : " & nroini
Debug.Print "Nro última hoja   : " & nrofin
```

```
Debug.Print "Nro CAI           : " & nrocai  
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- La consulta tiene sólo sentido en **IFH 1G** que soporten la impresión sobre formularios continuos u hojas sueltas.
- La consulta se puede hacer inmediatamente después de cerrado el comprobante 'A', o luego de la emisión de otros documentos (siempre que no haya otro comprobante 'A' en el lote emitido).

Método: ObtenerConfiguracion()

Comando: *GetConfigurationData, GetGeneralConfigurationData*

El **OCX Fiscal HASAR** genera uno u otro string de comando de acuerdo al modelo y versión de firmware fiscal de la **IFH 1G**. No todos los modelos y versiones de firmware fiscal soportan al comando *GetGeneralConfigurationData*.

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Argumentos: 17

Este método permite al software de gestión para puntos de venta consultar por diversos parámetros de comportamiento de la **IFH 1G**.

ObtenerConfiguracion **limcf, limtf, ivanoi, copias, cambio, leyendas, corte, marco, reimpr, saldo, sonido, alto, ancho, estacxz, modo, desborde, tapa**

limcf	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el monto límite para comprobantes 'B' a contribuyentes Consumidores Finales, sin datos del cliente. El valor de este parámetro debería ser el establecido por la normativa fiscal vigente.
limtf	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el monto límite para Tiques factura, Tiques Nota de Débito, y Tiques Notas de Crédito. El valor de este parámetro debería ser el establecido por la normativa fiscal vigente.
ivanoi	De salida. Opcional. Tipo: <i>Variant</i> Siempre '0.00'. La sobretasa de IVA No Inscripto no es de aplicación en la normativa fiscal vigente.
copias	De salida. Opcional. Tipo: <i>Variant</i>

	Permite al software de gestión para puntos de venta obtener la cantidad de ejemplares de un mismo comprobante que la IFH 1G permite emitir.
cambio	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta saber si la IFH 1G se encuentra programada para imprimir la leyenda “CAMBIO \$0.00” cuando la suma de los pagos coincide exactamente con el total del comprobante.
leyendas	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta saber si la IFH 1G se encuentra programada para la impresión/omisión de leyendas optionales (por ejemplo, el número de inscripción como contribuyente en Ingresos Brutos).
corte	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta saber si la IFH 1G se encuentra programada para realizar corte de papel, y en tal caso que tipo de corte.
marco	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta saber si la IFH 1G se encuentra programada para imprimir el marco de los comprobantes.
reimpr	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta saber si la IFH 1G se encuentra programada para reimprimir los comprobantes cancelados por corte de corriente.
saldo	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el texto programado para imprimir como medio de pago por defecto, y asociarlo a la diferencia entre el total del comprobante y la suma de los pagos realizados cuando ésta última es menor, y se cierra el comprobante sin completar los pagos.
sonido	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta saber si la IFH 1G se encuentra programada para emitir pitidos ante una falta de papel.
alto	De salida. Opcional. Tipo: <i>Variant</i>

	Permite al software de gestión para puntos de venta saber el alto de la hoja programado para utilizar en la IFH 1G .
ancho	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta el ancho de la hoja programado para utilizar en la IFH 1G .
estacxz	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta saber cual es la estación de impresión programada para la emisión, por defecto, de los cierres diarios 'X'/'Z'.
modo	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta saber cual es el modo de impresión programado en la IFH 1G : sólo formulario continuo, o mixto (alternar con impresiones sobre rollo).
desborde	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta saber si la IFH 1G está programada para verificar que no se excedan los montos límite (de corresponder), ante cada comando recibido, o solamente al cerrar el comprobante.
tapa	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta saber si la IFH 1G se encuentra programada para continuar imprimiendo en caso de abrirse abierta su tapa.

Ejemplo de uso:

...

Dim limcf As Variant, limtf As Variant, copias As Variant, vuelto As Variant, leyendas As Variant, _
corte As Variant, marco As Variant, reimpr As Variant, saldo As Variant, sonido As Variant, _
alto As Variant, ancho As Variant, estacxz As Variant, modo As Variant, desborde As Variant, _
tapa As Variant

...

HASAR1.**ObtenerConfiguracion** limcf, limtf, , copias, vuelto, leyendas, corte, marco, reimpr, saldo, sonido, _
alto, ancho, estacxz, modo, desborde, tapa

```
Debug.Print "Lím Cons Final : " & limcf
Debug.Print "Lím Tiq Fac : " & limtf
Debug.Print "Copias : " & copias
Debug.Print "CAMBIO $0.00 : " & vuelto
Debug.Print "Leyendas : " & leyendas
```

```

Debug.Print "Corte Papel      : " & corte
Debug.Print "Marco         : " & marco
Debug.Print "Reimpr Cancel   : " & reimpr
Debug.Print "Texto Saldo     : " & saldo
Debug.Print "Pitido          : " & sonido
Debug.Print "Alto Hoja        : " & alto
Debug.Print "Ancho Hoja       : " & ancho
Debug.Print "Reportes XZ      : " & estacxz
Debug.Print "Modo Impr        : " & modo
Debug.Print "Desborde         : " & desborde
Debug.Print "Tapa Abierta     : " & tapa
...

```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- No todos los parámetros de comportamiento se encuentran disponibles en todos los modelos y versiones de firmware fiscal de las **IFH 1G**.

Método: ObtenerConfiguracionCompleta()

Comando: *GetGeneralConfigurationData*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: No requiere.

Este método permite al software de gestión para puntos de venta consultar por diversos parámetros de comportamiento de la **IFH 1G**.

Ejemplo de uso:

```

...
HASAR1.ObtenerConfiguracionCompleta
Debug.Print HASAR1.Respuesta(0)
...

```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Si se invoca este método utilizando el **OCX Fiscal HASAR v051122** (o posterior) se genera una excepción por “*Función discontinuada...*” .
- Al no tener argumentos de salida, la invocación de este método requiere el uso de la propiedad *Respuesta()*. En el ejemplo anterior se obtiene el string completo de respuesta al comando enviado, pero podría pedirse cada campo de información de interés.
- Para conocer la estructura de campos de información del string de respuesta, consultar el *Manual de Comandos* de la **IFH 1G**.

Método: ObtenerDatosDeConfiguracion()

Comando: *GetConfigurationData*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.
Argumentos: No requiere.

Este método permite al software de gestión para puntos de venta consultar por diversos parámetros de comportamiento de la **IFH 1G**.

Ejemplo de uso:

```
...
HASAR1.ObtenerDatosDeConfiguracion
Debug.Print HASAR1.Respuesta(0)
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Si se invoca este método utilizando el **OCX Fiscal HASAR v051122** (o posterior) se genera una excepción por “*Función discontinuada...*”.
- Al no tener argumentos de salida, la invocación de este método requiere el uso de la propiedad *Respuesta()*. En el ejemplo anterior se obtiene el string completo de respuesta al comando enviado, pero podría pedirse cada campo de información de interés.
- Para conocer la estructura de campos de información del string de respuesta, consultar el *Manual de Comandos de la IFH 1G*.

Método: ObtenerDatosDeInicializacion()

Comando: *GetInitData*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.
Argumentos: 8

Este método permite al software de gestión para puntos de venta consultar los datos del emisor, los cuales se encuentran grabados en la memoria fiscal de la **IFH 1G**.

ObtenerDatosDeInicializacion *nrocurl, rsocial, regprn, fechaini, ptovta, iniact, ingbr, categ*

nrocurl	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el número de CUIT del emisor de los comprobantes.
rsocial	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener la razón social del emisor de los comprobantes.
regprn	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el número de registro con el cual

	fue inicializada la IFH 1G .
fechaini	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener la fecha en la cual fue inicializada la IFH 1G por el servicio técnico fiscal autorizado e interviniente.
ptovta	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el número de punto de venta asignado a la IFH 1G cuando fue dada de alta ante el organismo oficial de competencia.
iniact	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener la fecha de inicio de actividades del emisor de los comprobantes.
ingbr	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el número de inscripción como contribuyente en Ingresos Brutos del emisor de los comprobantes.
categ	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener la categoría como contribuyente frente a IVA del emisor de los comprobantes.

Ejemplo de uso:

```

...
Dim cuit As Variant, rsocial As Variant, serie As Variant, fechaini As Variant, ptovta As Variant, _
iniact As Variant, ingbr As Variant, categ As Variant
...
HASAR1.ObtenerDatosDelInicializacion cuit, rsocial, serie, fechaini, ptovta, iniact, ingbr, categ

Debug.Print "Nro CUIT           : " & cuit
Debug.Print "Razón Social       : " & rsocial
Debug.Print "Nro Serie          : " & serie
Debug.Print "Fecha Inicialización : " & fechaini
Debug.Print "Nro Pto de Vt a    : " & ptovta
Debug.Print "Inicio de Actividades : " & iniact
Debug.Print "Nro Ing Brutos     : " & ingbr
Debug.Print "Categ IVA          : " & categ
...

```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El uso de este método tiene sentido si la **IFH 1G** se encuentra operando en *Modo Fiscal*. El comando generado será rechazado por inválido si la **IFH 1G** se encuentra operando en *Modo Entrenamiento* (no hay datos grabados en la memoria fiscal).
- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la consulta de datos de inicialización.

Método: ObtenerNombreDeFantasia()

Comando:

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 2

Este método permite al software de gestión para puntos de venta obtener los textos almacenados en la memoria de trabajo de la **Impresora Fiscal HASAR**, y correspondientes a líneas de nombre de fantasía.

ObtenerNombreDeFantasia linea1, linea2

linea1	<p>De salida. Obligatorio. Tipo: <i>String</i></p> <p>Permite al software de gestión para puntos de venta obtener el texto de la primera línea de fantasía almacenada en la memoria de trabajo de la IFH 1G.</p>
linea2	<p>De salida. Obligatorio. Tipo: <i>String</i></p> <p>Permite al software de gestión para puntos de venta obtener el texto de la segunda línea de fantasía almacenada en la memoria de trabajo de la IFH 1G.</p>

Ejemplo de uso:

```

...
Dim linea1 As String, linea2 As String
...
HASAR1.ObtenerNombreDeFantasia linea1, linea2

Debug.Print "Fantasia 1...:" & linea1
Debug.Print "Fantasia 2...:" & linea2
...

```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Si se invoca este método utilizando el **OCX Fiscal HASAR v051122** (o posterior) se genera una excepción por “*Función discontinuada...*”.

Método: ObtenerVersionDelControlador()

Comando: *GetPrinterVersion*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: No requiere.

Retorno: *String*

Este método permite al software de gestión para puntos de venta obtener un string con el modelo y versión de firmware fiscal de la **IFH 1G**.

Ejemplo de uso:

```
...
    Debug.Print HASAR1.ObtenerVersionDelControlador
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la consulta de modelo y versión.

Método: PedidoDeStatus()

Comando: *StatusRequest*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: No requiere.

Este método permite al software de gestión para puntos de venta obtener información general del estado en que se encuentra la **IFH 1G** al momento de la consulta.

Ejemplo de uso:

```
...
    HASAR1.PedidoDeStatus
    Debug.Print HASAR1.Respuesta(0)
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Al no tener argumentos de salida, la invocación de este método requiere el uso de la propiedad *Respuesta()*. En el ejemplo anterior se obtiene el string completo de respuesta al comando enviado, pero podría pedirse cada campo de información de interés.
- Para conocer la estructura de campos de información del string de respuesta, consultar el *Manual de Comandos de la IFH 1G*.

Método: PedirPrimerIVA()

Comando: *SendFirstIVA*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: No requiere.

Este método permite al software de gestión para puntos de venta obtener el primer registro de información

correspondiente al desglose de impuestos del comprobante en curso, del comprobante recién cerrado, o del cierre diario 'Z' recién emitido.

Ejemplo de uso:

```
...
HASAR1.PedirPrimerIVA

If (HASAR1.Respuesta(3) <> "0") Then
    Debug.Print "Alícuota IVA : " & HASAR1.Respuesta(4)
    Debug.Print "Monto IVA : " & HASAR1.Respuesta(5)
    Debug.Print "Monto Imp Int: " & HASAR1.Respuesta(6)
Else
    Debug.Print "Fin de la información disponible..."
End If
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Al no tener argumentos de salida, la invocación de este método requiere el uso de la propiedad *Respuesta()*.
- Para conocer la estructura de campos de información del string de respuesta, consultar el *Manual de Comandos de la IFH 1G*.
- Si se invoca este método utilizando el **OCX Fiscal HASAR v051122** (o posterior) se genera una excepción por “*Función discontinuada...*”.

Método: PedirSiguienteIVA()

Comando: *NextIVATransmission*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: No requiere.

Este método permite al software de gestión para puntos de venta obtener el segundo registro de información (y siguientes), correspondiente al desglose de impuestos del comprobante en curso, del comprobante recién cerrado, o del cierre diario 'Z' recién emitido.

Ejemplo de uso:

```
...
HASAR1.PedirPrimerIVA

If (HASAR1.Respuesta(3) <> "0") Then
    Debug.Print "Alícuota IVA : " & HASAR1.Respuesta(4)
    Debug.Print "Monto IVA : " & HASAR1.Respuesta(5)
    Debug.Print "Monto Imp Int : " & HASAR1.Respuesta(6)
    Debug.Print "Monto Base : " & HASAR1.Respuesta(8) & vbCrLf

    While (HASAR1.Respuesta(3) <> "0")
        HASAR1.PedirSiguienteIVA
```

```

Select Case HASAR1.Respuesta(3)
Case "1"
    Debug.Print "Alícuota IVA      :" & HASAR1.Respuesta(4)
    Debug.Print "Monto IVA       :" & HASAR1.Respuesta(5)
    Debug.Print "Monto Imp Int   :" & HASAR1.Respuesta(6)
    Debug.Print "Monto Base      :" & HASAR1.Respuesta(8) & vbCrLf
Case "2"
    Debug.Print "Alícuota IVA      :" & HASAR1.Respuesta(4)
    Debug.Print "Monto Percep     :" & HASAR1.Respuesta(5) & vbCrLf
Case "5"
    Debug.Print "TOTAL: " & HASAR1.Respuesta(4) & vbCrLf
End Select
Wend
Else
    Debug.Print "No Hay información disponible..."
End If
Debug.Print "Fin de la información disponible..."
...

```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Al no tener argumentos de salida, la invocación de este método requiere el uso de la propiedad *Respuesta()*.
- Para conocer la estructura de campos de información del string de respuesta, consultar el *Manual de Comandos de la IFH 1G*.
- Si se invoca este método utilizando el **OCX Fiscal HASAR v051122** (o posterior) se genera una excepción por “*Función discontinuada...*”.
- El ejemplo anterior corresponde a una consulta posterior a la emisión de un comprobante fiscal.

Método: Recibir()

Este método no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: No requiere.

Retorno: *String*

Este método permite al software de gestión para puntos de venta obtener el string de respuesta correspondiente al último comando enviado a la **IFH 1G**.

Ejemplo de uso:

```

...
HASAR1.PedidoDeStatus
Debug.Print HASAR1.Recibir
...

```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Invocar al método *Recibir()* es equivalente a invocar a la propiedad *Respuesta()* con argumento cero.

Método: ReComenzar()

Este método no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: No requiere.

Este método permite al software de gestión para puntos de venta resolver la excepción que se produce cuando se invoca al método *Comenzar()*, y el puerto que se desea emplear se encuentra abierto.

Ejemplo de uso:

```
...
HASAR1.Modelo = MODELO_P320
HASAR1.Puerto = 11

On Error GoTo pamplinas
    HASAR1.Comenzar
    HASAR1.Comenzar
...
Exit Sub
pamplinas:
...
    HASAR1.ReComenzar
    Resume Next
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Si se invoca este método utilizando el **OCX Fiscal HASAR v051122** (o posterior) se genera una excepción por “*Función discontinuada...*”.

Método: ReimprimirComprobante()

Comando: *Reprint*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: No requiere.

Este método permite al software de gestión para puntos de venta solicitar un ejemplar adicional del comprobante que recién se ha cerrado. Por ejemplo, un triplicado.

Ejemplo de uso:

```
...
HASAR1.CerrarDNFH
    HASAR1.ReimprimirComprobante
    HASAR1.ReimprimirComprobante
...
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la emisión de copias de comprobantes.
- El ejemplar adicional se podrá pedir siempre que la **IFH 1G** esté configurada adecuadamente (ver método *ConfigurarControlador()*, parámetro COPIAS_DOCUMENTOS).
- La **IFH 1G** imprime automáticamente el “Original” y el “Duplicado” de comprobantes fiscales, Notas de crédito y Tiques Nota de Crédito. Para el resto de los comprobantes se imprime automáticamente sólo el “Original”. Los ejemplares faltantes debe pedirlos el software de gestión para puntos de venta.

Método: ReporteX()

Comando: *DailyClose*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 25

Este método permite al software de gestión para puntos de venta imprimir un cierre diario ‘X’ y obtener la información de montos acumulados y numeradores alcanzados, correspondientes a la emisión de comprobantes realizada entre dos cierres diarios: el último cierre diario ‘X’ o ‘Z’ realizado (según corresponda), y el actual cierre diario ‘X’.

ReporteX nrox, cdfcancel, cdnfhemit, cdnfemit, cdfemit, ultdfbc, ultdfa, totaldf, ivadf, impintdf, percepdf, ivanoidf, ultncbc, ultnca, totalnc, ivanc, impintnc, percepnc, ivanoinc, ultrto, cnccancel, _cdfbcemit, cdfaemit, cncbcemit, cncaemit

nrox	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el número de cierre diario ‘X’ que se ha impreso.
cdfcancel	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener la cantidad de comprobantes fiscales cancelados durante el período reportado.
cdnfhemit	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener la cantidad de comprobantes no fiscales homologados emitidos durante el período reportado.
cdnfemit	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener la cantidad de comprobantes no fiscales emitidos durante el período reportado.

cdfemit	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener la cantidad de comprobantes fiscales emitidos durante el período reportado.
ultdfbc	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el número del último comprobante fiscal 'B/C', haya sido emitido o cancelado.
ultdfa	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el número del último comprobante fiscal 'A', haya sido emitido o cancelado.
totaldf	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el monto total acumulado, durante el período reportado, correspondiente a comprobantes fiscales emitidos.
ivadf	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el monto de IVA acumulado, durante el período reportado, correspondiente a comprobantes fiscales emitidos.
impintdf	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el monto de impuestos internos acumulado, durante el período reportado, correspondiente a comprobantes fiscales emitidos.
percepdf	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el monto de percepciones acumulado, durante el período reportado, correspondiente a comprobantes fiscales emitidos.
ivanoidf	De salida. Opcional. Tipo: <i>Variant</i> Siempre '0.00'. La sobretasa de IVA No Inscripto no es de aplicación en la normativa fiscal vigente.
ulnncbc	De salida. Opcional. Tipo: <i>Variant</i>

	Permite al software de gestión para puntos de venta obtener el número de la última Nota de Crédito (Tique Nota de Crédito) 'B/C', haya sido emitida o cancelada.
ultnca	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el número de la última Nota de Crédito (Tique Nota de Crédito) 'A', haya sido emitida o cancelada.
totalinc	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el monto total acumulado, durante el período reportado, correspondiente a Notas de Crédito (Tiques Nota de Crédito) emitidas.
ivanc	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el monto de IVA acumulado, durante el período reportado, correspondiente a Notas de Crédito (Tiques Nota de Crédito) emitidas.
impintnc	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el monto de impuestos internos acumulado, durante el período reportado, correspondiente a Notas de Crédito (Tiques Nota de Crédito) emitidas.
percepnc	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el monto de percepciones acumulado, durante el período reportado, correspondiente a Notas de Crédito (Tiques Nota de Crédito) emitidas.
ivanoinc	De salida. Opcional. Tipo: <i>Variant</i>
	Siempre '0.00'. La sobretasa de IVA No Inscripto no es de aplicación en la normativa fiscal vigente.
ultrto	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el número del último Remito, haya sido emitido o cancelado.
cnccancel	De salida. Opcional.

	<p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener la cantidad de Notas de Crédito (Tiques Nota de Crédito) canceladas durante el período reportado.</p>
cdfbcemit	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener la cantidad de comprobantes fiscales 'B/C' emitidos durante el período reportado.</p>
cdfaemit	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener la cantidad de comprobantes fiscales 'A' emitidos durante el período reportado.</p>
cncbcemit	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener la cantidad de Notas de Crédito (Tiques Nota de Crédito) 'B/C' emitidas durante el período reportado.</p>
cnaaemit	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener la cantidad de Notas de Crédito (Tiques Nota de Crédito) 'A' emitidas durante el período reportado.</p>

Ejemplo de uso:

...

Dim nrox As Variant, cdfcancel As Variant, cdnfhemit As Variant, cdnfemit As Variant, cdfemit As Variant, _
 ultdfbc As Variant, ultdfa As Variant, totaldf As Variant, ivadf As Variant, impintdf As Variant, _
 percepdf As Variant, ultncbc As Variant, ultnca As Variant, totalnc As Variant, ivanc As Variant, _
 impintnc As Variant, percepcn As Variant, UltRto As Variant, cncccancel As Variant, _
 cdfbcemit As Variant, cdfaemit As Variant, cncbcemit As Variant, cncaemit As Variant

...

HASAR1.ReporteX nrox, cdfcancel, cdnfhemit, cdnfemit, cdfemit, ultdfbc, ultdfa, totaldf, ivadf, impintdf, _
 percepdf, , ultncbc, ultnca, totalnc, ivanc, impintnc, percepcn, , UltRto, cncccancel, _
 cdfbcemit, cdfaemit, cncbcemit, cncaemit

Debug.Print "Nro 'X"	:" & nrox
Debug.Print "DF Cancel"	:" & cdfcancel
Debug.Print "DNFH Emit"	:" & cdnfhemit
Debug.Print "DNF Emit"	:" & cdnfemit
Debug.Print "Cant DF Emit"	:" & cdfemit
Debug.Print "Ult DF 'B/C'	:" & ultdfbc
Debug.Print "Ult DF 'A'"	:" & ultdfa
Debug.Print "Total DF"	:" & totaldf
Debug.Print "IVA DF"	:" & ivadf

```

Debug.Print "Imp Int DF          :" & impintdf
Debug.Print "Percep DF          :" & percepdf
Debug.Print "Ult NC 'B/C'       :" & ultncbc
Debug.Print "Ult NC 'A'"       :" & ultnca
Debug.Print "Total NC          :" & totalnc
Debug.Print "IVA NC            :" & ivanc
Debug.Print "Imp Int NC         :" & impintnc
Debug.Print "Percep NC          :" & percepnc
Debug.Print "Ult Remito         :" & UltRto
Debug.Print "Cant NC Cancel     :" & cnccancel
Debug.Print "Cant DF 'B/C' Emit  :" & cdfbcemit
Debug.Print "Cant DF 'A' Emit    :" & cdfaemit
Debug.Print "Cant NC 'B/C' Emit   :" & cncbcemit
Debug.Print "Cant NC 'A' Emit     :" & cncaemit
...

```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Si el cierre diario 'X' es el primer comprobante de la jornada fiscal, y el último comprobante de la jornada fiscal inmediata anterior fue un cierre diario 'Z', permite la verificación de acumulados en cero en la memoria de trabajo de la **IFH 1G**.
- El cierre diario 'X' es de utilidad para manejar cambios de turno entre operadores del software de gestión del punto de venta.
- Si se emiten dos cierres diarios 'X' consecutivos, el segundo mostrará sus acumulados en cero. Significa que no se han emitido comprobantes fiscales o Notas de Crédito (Tiques Nota de Crédito) entre ambos reportes.
- La emisión de un cierre diario 'X' borra los acumulados parciales de la memoria de trabajo de la **IFH 1G**, pero no altera los acumulados correspondientes a la jornada fiscal en curso.
- La emisión de un cierre diario 'X' no provoca la escritura de información en la memoria fiscal de la **IFH 1G**.

Método: ReporteZ()

Comando: *DailyClose*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 25

Este método permite al software de gestión para puntos de venta imprimir un cierre diario 'Z' y obtener la información de montos acumulados y numeradores alcanzados, correspondientes a la emisión de comprobantes realizada durante la jornada fiscal.

ReporteX nroz, cdfcancel, cdnfhemit, cdnfemิต, cdfemít, ultdfbc, ultdfa, totaldf, ivadf, impintdf, percepdf, ivanoidf, ultncbc, ultnca, totalnc, ivanc, impintnc, percepnc, ivanoinc, ultrto, cnccancel, cdfbcemit, cdfaemit, cncbcemit, cncaemit

nroz	De salida. Opcional. Tipo: Variant
-------------	---------------------------------------

	Permite al software de gestión para puntos de venta obtener el número de cierre diario 'Z' que se ha impreso.
cdfcancel	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener la cantidad de comprobantes fiscales cancelados durante la jornada fiscal.
cdnfhemit	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener la cantidad de comprobantes no fiscales homologados emitidos durante la jornada fiscal.
cdnfemit	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener la cantidad de documentos no fiscales (DNF) emitidos durante la jornada fiscal.
cdfemit	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener la cantidad de comprobantes fiscales emitidos durante la jornada fiscal.
ultdfbc	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el número del último comprobante fiscal 'B/C', haya sido emitido o cancelado.
ultdfa	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el número del último comprobante fiscal 'A', haya sido emitido o cancelado.
totaldf	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el monto total acumulado, durante la jornada fiscal, correspondiente a comprobantes fiscales emitidos.
ivadf	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el monto de IVA acumulado, durante la jornada fiscal, correspondiente a comprobantes fiscales emitidos.

impintdf	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el monto de impuestos internos acumulado, durante la jornada fiscal, correspondiente a comprobantes fiscales emitidos.
perceppdf	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el monto de percepciones acumulado, durante la jornada fiscal, correspondiente a comprobantes fiscales emitidos.
ivanoidf	De salida. Opcional. Tipo: <i>Variant</i> Siempre '0.00'. La sobretasa de IVA No Inscripto no es de aplicación en la normativa fiscal vigente.
ultncbc	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el número de la última Nota de Crédito (Tique Nota de Crédito) 'B/C', haya sido emitida o cancelada.
ultnca	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el número de la última Nota de Crédito (Tique Nota de Crédito) 'A', haya sido emitida o cancelada.
totalnc	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el monto total acumulado, durante la jornada fiscal, correspondiente a Notas de Crédito (Tiques Nota de Crédito) emitidas.
ivanc	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el monto de IVA acumulado, durante la jornada fiscal, correspondiente a Notas de Crédito (Tiques Nota de Crédito) emitidas.
impintnc	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el monto de impuestos internos acumulado, durante la jornada fiscal, correspondiente a Notas de Crédito (Tiques Nota de Crédito) emitidas.
percepnc	De salida. Opcional.

	<p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener el monto de percepciones acumulado, durante la jornada fiscal, correspondiente a Notas de Crédito (Tiques Nota de Crédito) emitidas.</p>
ivanoinc	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Siempre '0.00'. La sobretasa de IVA No Inscripto no es de aplicación en la normativa fiscal vigente.</p>
ultrto	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener el número del último Remito, haya sido emitido o cancelado.</p>
cnccancel	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener la cantidad de Notas de Crédito (Tiques Nota de Crédito) canceladas durante la jornada fiscal.</p>
cdfbcemit	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener la cantidad de comprobantes fiscales 'B/C' emitidos durante la jornada fiscal.</p>
cdfaemit	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener la cantidad de comprobantes fiscales 'A' emitidos durante la jornada fiscal.</p>
cncbccemt	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener la cantidad de Notas de Crédito (Tiques Nota de Crédito) 'B/C' emitidas durante la jornada fiscal.</p>
cnccaemit	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener la cantidad de Notas de Crédito (Tiques Nota de Crédito) 'A' emitidas durante la jornada fiscal.</p>

Ejemplo de uso:

...

Dim nroz As Variant, cdfcancel As Variant, cdnfhemit As Variant, cdnfemitt As Variant, cdfemit As Variant, _
ultdfbc As Variant, ultdfa As Variant, totaldf As Variant, ivadf As Variant, impintdf As Variant, _
percepdf As Variant, ultncbc As Variant, ultnca As Variant, totalnc As Variant, ivanc As Variant, _
impintnc As Variant, percepcn As Variant, UltRto As Variant, cncccancel As Variant, _
cdfbcemitt As Variant, cdfaemitt As Variant, cncbcemitt As Variant, cncaemitt As Variant

...

HASAR1.**ReporteZ** nroz, cdfcancel, cdnfhemit, cdnfemitt, ultdfbc, ultdfa, totaldf, ivadf, impintdf, _
percepdf, , ultncbc, ultnca, totalnc, ivanc, impintnc, percepcn, , UltRto, cncccancel, _
cdfbcemitt, cdfaemitt, cncbcemitt, cncaemitt

Debug.Print "Nro 'Z"	:" & nroz
Debug.Print "DF Cancel"	:" & cdfcancel
Debug.Print "DNFH Emit"	:" & cdnfhemit
Debug.Print "DNF Emit"	:" & cdnfemitt
Debug.Print "Cant DF Emit"	:" & cdfemit
Debug.Print "Ult DF 'B/C'	:" & ultdfbc
Debug.Print "Ult DF 'A'"	:" & ultdfa
Debug.Print "Total DF"	:" & totaldf
Debug.Print "IVA DF"	:" & ivadf
Debug.Print "Imp Int DF"	:" & impintdf
Debug.Print "Percep DF"	:" & percepdf
Debug.Print "Ult NC 'B/C'"	:" & ultncbc
Debug.Print "Ult NC 'A'"	:" & ultnca
Debug.Print "Total NC"	:" & totalnc
Debug.Print "IVA NC"	:" & ivanc
Debug.Print "Imp Int NC"	:" & impintnc
Debug.Print "Percep NC"	:" & percepcn
Debug.Print "Ult Remito"	:" & UltRto
Debug.Print "Cant NC Cancel"	:" & cncccancel
Debug.Print "Cant DF 'B/C' Emit"	:" & cdfbcemitt
Debug.Print "Cant DF 'A' Emit"	:" & cdfaemitt
Debug.Print "Cant NC 'B/C' Emit"	:" & cncbcemitt
Debug.Print "Cant NC 'A' Emit"	:" & cncaemitt

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Si se emiten dos cierres diarios 'Z' consecutivos, el segundo mostrará sus acumulados en cero. Significa que no se han emitido comprobantes fiscales o Notas de Crédito (Tiques Nota de Crédito) entre ambos reportes.
- La emisión de un cierre diario 'Z' borra todos los acumulados de la memoria de trabajo de la **IFH 1G**.
- La emisión de un cierre diario 'Z' provoca la escritura de información en la memoria fiscal de la **IFH 1G**. Se consume un registro de su capacidad de almacenamiento.

Método: ReporteZindividualPorFecha()

Comando: *GetDailyReport*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 18

Este método permite al software de gestión para puntos de venta obtener, sin generar impresión, la información de montos acumulados y numeradores alcanzados correspondientes a un único cierre diario 'Z' almacenado en la memoria fiscal de la **IFH 1G**. Debe indicarse la fecha del cierre diario 'Z' buscado.

ReporteZIndividualPorFecha **fecha, fechaz, nroz, ultdfbc, ultdfa, totaldf, ivadf, impintdf, percepdf, ivanoidf, ultncbc, ultnca, totalnc, ivanc, impintnc, percepnc, ivanoinc, ultrto**

fecha	De entrada. Obligatorio. Tipo: <i>Date</i>
	Permite al software de gestión para puntos de venta indicar la fecha correspondiente a la jornada fiscal cuya información se solicita.
fechaz	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener la fecha en que se emitió el cierre diario 'Z' cuya información se solicita.
nroz	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el número de cierre diario 'Z' cuya información se solicita.
ultdfbc	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el número del último comprobante fiscal 'B/C', haya sido emitido o cancelado, correspondiente a la jornada fiscal cuya información se solicita.
ultdfa	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el número del último comprobante fiscal 'A', haya sido emitido o cancelado, correspondiente a la jornada fiscal cuya información se solicita.
totaldf	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el monto total acumulado, durante la jornada fiscal que se solicita, y correspondiente a comprobantes fiscales emitidos.
ivadf	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el monto de IVA acumulado, durante la jornada fiscal que se solicita, y correspondiente a comprobantes fiscales emitidos.
impintdf	De salida. Opcional.

	<p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener el monto de impuestos internos acumulado, durante la jornada fiscal que se solicita, y correspondiente a comprobantes fiscales emitidos.</p>
percepdf	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener el monto de percepciones acumulado, durante la jornada fiscal que se solicita, y correspondiente a comprobantes fiscales emitidos.</p>
ivanoidf	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Siempre '0.00'. La sobretasa de IVA No Inscripto no es de aplicación en la normativa fiscal vigente.</p>
ultncbc	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener el número de la última Nota de Crédito (Tique Nota de Crédito) 'B/C', haya sido emitida o cancelada, durante la jornada fiscal que se solicita.</p>
ultnca	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener el número de la última Nota de Crédito (Tique Nota de Crédito) 'A', haya sido emitida o cancelada, durante la jornada fiscal que se solicita.</p>
totalnc	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener el monto total acumulado, durante la jornada fiscal que se solicita, correspondiente a Notas de Crédito (Tiques Nota de Crédito) emitidas.</p>
ivanc	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener el monto de IVA acumulado, durante la jornada fiscal que se solicita, correspondiente a Notas de Crédito (Tiques Nota de Crédito) emitidas.</p>
impintnc	<p>De salida. Opcional.</p> <p>Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener el monto de impuestos inter-</p>

	nos acumulado, durante la jornada fiscal que se solicita, correspondiente a Notas de Crédito (Tiques Nota de Crédito) emitidas.
percepnc	<p>De salida. Opcional. Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener el monto de percepciones acumulado, durante la jornada fiscal que se solicita, correspondiente a Notas de Crédito (Tiques Nota de Crédito) emitidas.</p>
ivanoinc	<p>De salida. Opcional. Tipo: <i>Variant</i></p> <p>Siempre '0.00'. La sobretasa de IVA No Inscripto no es de aplicación en la normativa fiscal vigente.</p>
ultrto	<p>De salida. Opcional. Tipo: <i>Variant</i></p> <p>Permite al software de gestión para puntos de venta obtener el número del último Remito, haya sido emitido o cancelado, durante la jornada fiscal que se solicita.</p>

Ejemplo de uso:

...

```
Dim fecha As Variant, fechaz As Variant, nroz As Variant, ultdfbc As Variant, ultdfa As Variant, _
totaldf As Variant, ivadf As Variant, impintdf As Variant, percepdf As Variant, ultncbc As Variant, _
ultnca As Variant, totalnc As Variant, ivanc As Variant, impintnc As Variant, percepnc As Variant, _
UltRto As Variant
```

Dim fecha As Date

...

```
HASAR1.ReporteZIndividualPorfecha fecha, fechaz, ultdfbc, ultdfa, totaldf, ivadf, impintdf, percepdf, , _
ultncbc, ultnca, totalnc, ivanc, impintnc, percepnc, , UltRto
```

```
Debug.Print "Fecha 'Z'           :" & fechaz
Debug.Print "Nro 'Z'            :" & nroz
Debug.Print "Ult DF 'B/C'      :" & ultdfbc
Debug.Print "Ult DF 'A'"       :" & ultdfa
Debug.Print "Total DF          :" & totaldf
Debug.Print "IVA DF             :" & ivadf
Debug.Print "Imp Int DF        :" & impintdf
Debug.Print "Percep DF           :" & percepdf
Debug.Print "Ult NC 'B/C'"     :" & ultncbc
Debug.Print "Ult NC 'A'"       :" & ultnca
Debug.Print "Total NC          :" & totalnc
Debug.Print "IVA NC             :" & ivanc
Debug.Print "Imp Int NC        :" & impintnc
Debug.Print "Percep NC           :" & percepnc
Debug.Print "Ult Remito         :" & UltRto
```

...

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Al solicitar la información de la jornada fiscal por fecha, si se retorna cero como número de cierre diario ‘Z’ significa que en la fecha indicada no se ha realizado un cierre diario ‘Z’.
- De haberse realizado más de un cierre diario ‘Z’ en la misma fecha, se retorna el número de la primera encontrada. Para obtener todos los cierres diarios ‘Z’ con la fecha solicitada invocar al método *ReporteZindividualPorNumero()*.

Método: ReporteZindividualPorNúmero()

Comando: *GetDailyReport*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 18

Este método permite al software de gestión para puntos de venta obtener, sin generar impresión, la información de montos acumulados y numeradores alcanzados correspondientes a un único cierre diario ‘Z’ almacenado en la memoria fiscal de la **IFH 1G**. Debe indicarse el número del cierre diario ‘Z’ buscado.

ReporteZindividualPorNúmero num, fechaz, nroz, ultdfbc, ultdfa, totaldf, ivadf, impintdf, percepdf, _
ivanoidf, ultncbc, ultnca, totalnc, ivanc, impintnc, percepcn, ivanoinc, ultrto

num	De entrada. Obligatorio. Tipo: <i>Long</i> Permite al software de gestión para puntos de venta indicar el número correspondiente a la jornada fiscal cuya información se solicita.
fechaz	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener la fecha en que se emitió el cierre diario ‘Z’ cuya información se solicita.
nroz	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el número de cierre diario ‘Z’ cuya información se solicita.
ultdfbc	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el número del último comprobante fiscal ‘B/C’, haya sido emitido o cancelado, correspondiente a la jornada fiscal cuya información se solicita.
ultdfa	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el número del último compro-

	bante fiscal ‘A’, haya sido emitido o cancelado, correspondiente a la jornada fiscal cuya información se solicita.
totaldf	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el monto total acumulado, durante la jornada fiscal que se solicita, y correspondiente a comprobantes fiscales emitidos.
ivadf	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el monto de IVA acumulado, durante la jornada fiscal que se solicita, y correspondiente a comprobantes fiscales emitidos.
impintdf	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el monto de impuestos internos acumulado, durante la jornada fiscal que se solicita, y correspondiente a comprobantes fiscales emitidos.
percepdf	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el monto de percepciones acumulado, durante la jornada fiscal que se solicita, y correspondiente a comprobantes fiscales emitidos.
ivanoidf	De salida. Opcional. Tipo: <i>Variant</i> Siempre ‘0.00’. La sobretasa de IVA No Inscripto no es de aplicación en la normativa fiscal vigente.
ultncbc	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el número de la última Nota de Crédito (Tique Nota de Crédito) ‘B/C’, haya sido emitida o cancelada, durante la jornada fiscal que se solicita.
ultnca	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el número de la última Nota de Crédito (Tique Nota de Crédito) ‘A’, haya sido emitida o cancelada, durante la jornada fiscal que se solicita.
totalnc	De salida. Opcional. Tipo: <i>Variant</i>

	Permite al software de gestión para puntos de venta obtener el monto total acumulado, durante la jornada fiscal que se solicita, correspondiente a Notas de Crédito (Tiques Nota de Crédito) emitidas.
ivanc	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el monto de IVA acumulado, durante la jornada fiscal que se solicita, correspondiente a Notas de Crédito (Tiques Nota de Crédito) emitidas.
impintnc	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el monto de impuestos internos acumulado, durante la jornada fiscal que se solicita, correspondiente a Notas de Crédito (Tiques Nota de Crédito) emitidas.
percepnc	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el monto de percepciones acumulado, durante la jornada fiscal que se solicita, correspondiente a Notas de Crédito (Tiques Nota de Crédito) emitidas.
ivanoinc	De salida. Opcional. Tipo: <i>Variant</i>
	Siempre '0.00'. La sobretasa de IVA No Inscripto no es de aplicación en la normativa fiscal vigente.
ultrto	De salida. Opcional. Tipo: <i>Variant</i>
	Permite al software de gestión para puntos de venta obtener el número del último remito, haya sido emitido o cancelado, durante la jornada fiscal que se solicita.

Ejemplo de uso:

...

Dim fecha As Variant, fechaz As Variant, nroz As Variant, ultdfbc As Variant, ultdfa As Variant, _
totaldf As Variant, ivadf As Variant, impintdf As Variant, percepdf As Variant, ultncbc As Variant, _
ultnca As Variant, totalnc As Variant, ivanc As Variant, impintnc As Variant, percepnc As Variant, _
UltRto As Variant

Dim num As Long

...

HASAR1.ReporteZIndividualPorNúmero num, fechaz, ultdfbc, ultdfa, totaldf, ivadf, impintdf, percepdf, , _
ultncbc, ultnca, totalnc, ivanc, impintnc, percepnc, , UltRto

```
Debug.Print "Fecha 'Z'"      :" & fechaz
Debug.Print "Nro 'Z'"       :" & nroz
```

```

Debug.Print "Ult DF 'B/C'      :" & ultdfbc
Debug.Print "Ult DF 'A'"     :" & ultdfa
Debug.Print "Total DF"       :" & totaldf
Debug.Print "IVA DF"        :" & ivadf
Debug.Print "Imp Int DF"    :" & impintdf
Debug.Print "Percep DF"     :" & percepdf
Debug.Print "Ult NC 'B/C'"  :" & ultncbc
Debug.Print "Ult NC 'A'"    :" & ultnca
Debug.Print "Total NC"      :" & totalnc
Debug.Print "IVA NC"        :" & ivanc
Debug.Print "Imp Int NC"   :" & impintnc
Debug.Print "Percep NC"     :" & percepnc
Debug.Print "Ult Remito"    :" & UltRto
...

```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Al solicitar la información de la jornada fiscal por número, si se retorna la fecha con sus componentes en cero significa que no se ha realizado un cierre diario 'Z' con el número solicitado.

Método: ReporteZPorFechas()

Comando: *DailyCloseByDate*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 3

Este método permite al software de gestión para puntos de venta imprimir la información de montos acumulados y numeradores alcanzados, dentro de un rango dado de fechas de jornadas fiscales. Se pueden imprimir sólo los valores generales, o imprimir también el detalle de cada jornada fiscal que se encuentra dentro del rango solicitado.

ReporteZPorFechas desde, hasta, global

desde	De entrada. Obligatorio. Tipo: <i>Date</i> Permite al software de gestión para puntos de venta indicar al fecha inicial del rango de jornadas fiscales a reportar.
hasta	De entrada.Obligatorio. Tipo: <i>Date</i> Permite al software de gestión para puntos de venta indicar la fecha final del rango de jornadas fiscales a reportar.
global	De entrada. Obligatorio. Tipo: <i>Boolean</i> Permite al software de gestión para puntos de venta indicar si el reporte de información será global, o se detallará también cada jornada fiscal dentro del rango solicitado.,

<i>True</i>	Reporte global.
<i>False</i>	Incluir detalle por jornada fiscal.

Ejemplo de uso:

```
...
HASAR1.ReporteZPorFechas "01/01/2013", "21/11/2013", True
...
```

Método: ReporteZPorNumeros()

Comando: *DailyCloseByNumber*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: 3

Este método permite al software de gestión para puntos de venta imprimir la información de montos acumulados y numeradores alcanzados, dentro de un rango dado de números de cierres diarios 'Z'. Se pueden imprimir sólo los valores generales, o imprimir también el detalle de cada jornada fiscal que se encuentra dentro del rango solicitado.

ReporteZPorNumeros desde, hasta, global

desde	De entrada. Obligatorio. Tipo: <i>Long</i> Permite al software de gestión para puntos de venta indicar el número de cierre diario 'Z' inicial del rango de jornadas fiscales a reportar.
hasta	De entrada. Obligatorio. Tipo: <i>Long</i> Permite al software de gestión para puntos de venta indicar el número de cierre diario 'Z' final del rango de jornadas fiscales a reportar.
global	De entrada. Obligatorio. Tipo: <i>Boolean</i> Permite al software de gestión para puntos de venta indicar si el reporte de información será global, o se detallará también cada jornada fiscal dentro del rango solicitado., <i>True</i> Reporte global. <i>False</i> Incluir detalle por jornada fiscal.

Ejemplo de uso:

```
...
HASAR1.ReporteZPorNumeros 32, 59, True
...
```

Método: Subtotal()

Comando: *Subtotal*

Disponible: OCX Fiscal HASAR, en todas las versiones.

Argumentos: 7

Permite al software de gestión para puntos de venta obtener los montos acumulados hasta el momento de la consulta, y correspondientes al comprobante en curso de emisión (comprobante fiscal, Nota de Crédito, o Ticket Nota de Crédito).

Subtotal impr, items, total, iva, pagado, ivanoi, impint

impr	De entrada. Obligatorio. Tipo: <i>Boolean</i> Permite al software de gestión para puntos de venta indicar si la consulta causa impresión en el comprobante en curso de emisión. <i>True</i> Imprimir. <i>False</i> Omitir impresión.
items	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener la cantidad de ítems registrados hasta el momento en el comprobante en curso de emisión.
total	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el monto total acumulado en el comprobante hasta el momento de la consulta.
iva	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener el monto de IVA acumulado en el comprobante hasta el momento de la consulta.
pagado	De salida. Opcional. Tipo: <i>Variant</i> Permite al software de gestión para puntos de venta obtener la suma de los pagos registrados en el comprobante, hasta el momento de la consulta.
ivanoi	De salida. Opcional. Tipo: <i>Variant</i> Siempre '0.00'. La sobretasa de IVA No Inscripto no es aplicable en la normativa fiscal vigente.
impint	De salida. Opcional. Tipo: <i>Variant</i>

Permite al software de gestión para puntos de venta obtener el monto de impuesto interno acumulado en el comprobante hasta el momento de la consulta.

Ejemplo de uso:

```
...
Dim items As Variant, total As Variant, iva As Variant, Pagado As Variant, Implt As Variant
...
HASAR1.Subtotal True, items, total, iva, Pagado, , Implt

Debug.Print "Cant items      :" & items
Debug.Print "Subtotal       :" & total
Debug.Print "IVA           :" & iva
Debug.Print "Pagado        :" & Pagado
Debug.Print "Imp Int       :" & Implt
....
```

El desarrollador del software para puntos de venta debe tener presente que:

- La impresión del subtotal no es admitida en todos los tipos de comprobantes para los cuales es válida la consulta. Que el argumento se coloque en *True* al invocar al método *Subtotal()* sólo significa que de estar admitida la impresión, debe concretarse.
- El monto obtenido a través del argumento **total** es el importe que se imprimiría a la derecha de la leyenda “**TOTAL \$...**” (comprobantes fiscales), o a la derecha de la leyenda “**IMPORTE TOT. \$....**” (Notas de Crédito o Tiques Nota de Crédito), si el comprobante fuese cerrado inmediatamente después de la consulta. Esto es útil para registrar pagos por el monto que dice la **IFH 1G** que ha calculado. Permite evitar la diferencia de redondeo al comparar el resultado de la aritmética de la **IFH 1G** y el resultado de la aritmética del software de gestión para puntos de venta.

Método: TratarDeCancelarTodo()

Comando: *TotalTender, Cancel, CloseFiscalReceipt, CloseNonFiscalReceipt*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Argumentos: No requiere.

Este método permite al software de gestión para puntos de venta llevar a la **IFH 1G** a un estado conocido, y así retomar la sincronización respecto de la numeración de comprobantes y montos acumulados. Por ejemplo, ir al estado de ‘*no hay comprobante abierto*’.

Ejemplo de uso:

```
...
HASAR1.TratarDeCancelarTodo
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Cuantos de los strings de comandos enunciados se generan, y el orden en que se envían, depende del modelo y versión de firmware fiscal de la **IFH 1G**. Por ejemplo, no todas soportan el comando *Cancel*.

- Al invocarse al método *TratarDeCancelarTodo()*, el **OCX Fiscal HASAR** intenta, en primer lugar, cancelar el comprobante que se encuentra abierto. De no poder hacerlo, intenta cerrarlo. Hay modelos y versiones de firmware fiscal de las **IFH 1G** que no aceptan el comando de cancelación de comprobante si ya se ha aceptado al menos un comando de impresión de pago.
- Después de invocarse este método, el software de gestión para puntos de venta debe aplicar una inteligencia que le permita saber que ocurrió con su ejecución, y actuar en consecuencia:
 - ¿ Canceló ? ¿ Qué tipo de comprobante ? ¿ Era numerado por la **IFH 1G** ?
 - ¿ Cerró ? ¿ Qué tipo de comprobante ? ¿ Era numerado por la **IFH 1G** ? Entonces, los montos fueron acumulados. ¿ Corresponde una Nota de Crédito ?
 - No había un comprobante abierto.

Método: VerificarModelo()

Comando: *GetPrinterVersion*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Argumentos: No requiere.

Retorno: *Boolean*

Este método permite al software de gestión para puntos de venta saber si el modelo y versión de firmware fiscal de la **IFH 1G** seleccionada (valor de la propiedad *Modelo()*) coincide con la conectada físicamente.

Ejemplo de uso:

```
...
If (Not(HASAR1.VerificarModelo)) Then
    Debug.Print "Impresora Fiscal seleccionada y conectada no coinciden..."
End If
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la consulta de modelo y versión.
- El método retorna *True* si modelo y versión de firmware fiscal coinciden. Nada dice acerca del número de registro, o del número de punto de venta.

Propiedades del OCX Fiscal HASAR

La tabla siguiente muestra la disponibilidad de propiedades del **OCX Fiscal HASAR** en las distintas versiones liberadas hasta la fecha.

Propiedades disponibles	OCX Fiscal HASAR		
	v010724	v051122	v110423 v150423
Baudios()	SI	SI	SI
CajonAbierto()	SI	SI	SI
ContadorImpresoraOcupada()	SI	SI	SI
DescripcionLargas()	SI	SI	SI
DescripcionEstadoControlador()	SI	SI	SI
DescripcionDocumentoEnCurso()		SI	SI
DireccionIP()		SI	SI
DobleAncho()		SI	SI
DocumentoDeReferencia()		SI	SI
DocumentoEnCurso()		SI	SI
Encabezado()	SI	SI	SI
EstadoControlador()	SI	SI	SI
EventosIndividuales()	SI	SI	SI
FechaHoraFiscal()	SI	SI	SI
HuboErrorFiscal()	SI	SI	SI
HuboErrorMecanico()	SI	SI	SI
HuboFaltaPapel()	SI	SI	SI
HuboStatPrn()	SI	SI	SI
ImpuestoInternoFijo()	SI	SI	SI
ImpuestoInternoPorMonto()	SI	SI	SI
InformacionRemito()	SI		
IndicadorFiscal()	SI	SI	SI
IndicadorImpresora()	SI	SI	SI
Interlineado()		SI	SI
KIVA()	SI	SI	SI
Modelo()	SI	SI	SI
ModoStatPrn()		SI	SI
NombreArchivoLog()			SI
NombreDeFantasia()		SI	SI
PaginasDeUltimoDocumento()		SI	SI
PrecioBase()	SI	SI	SI
PrimerNumeroDeDocumentoActual()		SI	SI
Puerto()	SI	SI	SI
ReintentoConstante()	Si	SI	SI
Reintentos()		SI	SI
Respuesta()	SI	SI	SI
ResumenIVA()		SI	SI
SoportaStatPRN()		SI	SI
TiempoDeEspera()		SI	SI
Transporte()		SI	SI
UltimaNotaCreditoA()		SI	SI

Propiedades disponibles	OCX Fiscal HASAR		
	v010724	v051122	v110423 v150423
<i>UltimaNotaCreditoBC()</i>		SI	SI
<i>UltimoDocumentoFiscalA()</i>		SI	SI
<i>UltimoDocumentoFiscalBC()</i>		SI	SI
<i>UltimoDocumentoFueCancelado()</i>		SI	SI
<i>UltimaFactura()</i>	SI		
<i>UltimoTicket()</i>	SI		
<i>UltimoRemito()</i>		SI	SI
<i>UsarASCII()</i>	SI	SI	SI
<i>UsarDisplay()</i>	SI	SI	SI
<i>VerificacionCompletaDeErrores()</i>		SI	SI
<i>Version()</i>		SI	SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como "NO SOPORTADO"

Por regla general, se puede decir que la mayoría de las propiedades afectan al comportamiento del **OCX Fiscal HASAR** y que algunas de ellas pueden generar y enviar strings de comandos a la **IFH 1G**.

Es responsabilidad del desarrollador del software de gestión para puntos de venta consultar el *Manual de Comandos* de la **IFH 1G** a utilizar, además del contenido de este manual. Toda la información referida a la construcción de strings de comandos, strings de respuestas, y comportamiento de la **IFH 1G**, se detalla en su *Manual de Comandos*.

Es responsabilidad del software de gestión para puntos de venta la validez de la información que suministre para la generación de los strings de comandos, así como el tratamiento de los eventos y excepciones que pudieran generarse al invocar a una propiedad del **OCX Fiscal HASAR**.

Propiedad: Baudios()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: *Long*

Retorno: *Long*

Esta propiedad permite al software de gestión para puntos de venta configurar la velocidad -en Baudios- del puerto serie del host, para que coincida con la utilizada por la **IFH 1G**.

Los valores admitidos son: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.

Esta propiedad también permite al software de gestión para puntos de venta consultar la configuración de velocidad -en Baudios- del puerto serie del host.

Ejemplos de uso:

```
...
HASAR1.Baudios = 9600
Debug.Print HASAR1.Baudios
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El valor de configuración de esta propiedad puede verse modificado por la invocación del *método AutodetectarControlador()*.
- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan más de un valor de velocidad en la configuración de su puerto serie.
- Valor por defecto: 9600

Propiedad: CajonAbierto()

Comando: *StatusRequest*

Disponible: **OCX Fiscal HASAR**, en todas sus versiones.

Configuración: No requiere.

Retorno: *Boolean*

Esta propiedad permite al software de gestión para puntos de venta conocer el estado del cajón de dinero conectado a la **IFH 1G**.

True El cajón de dinero se encuentra abierto.

False El cajón de dinero se encuentra cerrado, ausente, o no está soportado por la **IFH 1G**.

Ejemplo de uso:

```
...
Debug.Print HASAR1.CajonAbierto
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la conexión de un cajón de dinero.
- Valor por defecto: *True*

Propiedad: ContadorImpresoraOcupada()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: No requiere.

Retorno: *Long*

Esta propiedad permite al software de gestión para puntos de venta saber la cantidad de veces que la **IFH 1G** se reportó ocupada (se disparó el evento *ImpresoraOcupada()*), desde que fue enviado el último comando (cuya respuesta aún no se recibió).

Ejemplo de uso:

```
...
Debug.Print "PRN Fiscal Ocupada.. " & HASAR1.ContadorImpresoraOcupada & " veces (una cada 400 ms)"
...
```

El desarrollador del software de gestión para punto de venta debe tener presente que:

- Si bien la **IFH 1G** responde al comando enviado dentro de los 400 ms siguientes a su recepción, hay tareas que pueden demandarle más tiempo. Es en estos casos cuando da aviso de demora al host, enviando un carácter especial (DC2, ASCII 18 decimal) cada 400 ms, y hasta poder entregar la respuesta adeudada. Por ejemplo, cuando se imprime un cierre diario 'Z'.

Por cada carácter DC2 recibido, el **OCX Fiscal HASAR** dispara el evento *ImpresoraOcupada()* e incrementa en '1' el valor de la propiedad *ContadorImpresoraOcupada()*.

- Valor por defecto: 0 (cero)

Propiedad: DescripcionesLargas()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: *Boolean*

Retorno: *Boolean*

Esta propiedad permite al software de gestión para puntos de venta establecer el uso de la extensión de la descripción en la impresión de ítem.

True Aceptar textos de longitud mayor a la soportada como descripción al invocar al método *ImprimirItem()*.

False No extender la descripción del ítem.

El software de gestión para puntos de venta también puede consultar el valor de configuración de esta propiedad.

Ejemplos de uso:

```
...
HASAR1.DescripcionesLargas = True
Debug.Print HASAR1.DescripcionesLargas
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El valor *True* afecta al comportamiento del método *ImprimirItem()*.
 1. Acepta descripciones mayores que la soportadas en la sintaxis del comando *PrintLineItem*.
 2. Reparte la descripción del ítem entre líneas de texto fiscal y la impresión del ítem.
- La lógica de distribución del texto puede no coincidir con lo esperado, o deseado, por el usuario final. En tales casos, siempre el distribución la puede hacer el software de gestión para puntos de venta recurriendo al empleo de los métodos: *ImprimirTextoFiscal()* e *ImprimirItem()*.

Propiedad: DescripcionEstadoControlador()

Comando: *StatusRequest*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: No requiere.

Retorno: *String*

Esta propiedad permite al software de gestión para puntos de venta obtener un mensaje asociado al estado fiscal en que se encuentra la **IFH 1G** al momento de la consulta.

Ejemplo de uso:

```
...
Debug.Print HASAR1.DescripcionEstadoControlador
...
```

Por ejemplo, se podría obtener un mensaje como el siguiente:

“Comprobante fiscal abierto. Se realizó al menos un pago”

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Invocar a esta propiedad es equivalente a que el software de gestión para puntos de venta invoque al método *PedidoDeStatus()*, y luego proceda con un análisis de la respuesta. En particular del campo de información correspondiente a *Status Auxiliar*.

Propiedad: DescripcionDocumentoEnCurso()

Comando: *StatusRequest*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: No requiere.

Retorno: *String*

Esta propiedad permite al software de gestión para puntos de venta obtener un mensaje que describe al comprobante que se encuentra en curso de emisión (abierto) al momento de la consulta.

Ejemplo de uso:

```
...
Debug.Print HASAR1.DescripcionDocumentoEnCurso
...
```

Por ejemplo, se podría obtener un mensaje como el siguiente:
“Factura A”

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Invocar a esta propiedad es equivalente a que el software de gestión para puntos de venta invoque al método *PedidoDeStatus()*, y luego proceda con un análisis de la respuesta. En particular del campo de información correspondiente a *Status de Documento*.

Propiedad: DireccionIP()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: *String*

Retorno: *String*

Esta propiedad permite al software de gestión para puntos de venta configurar una dirección IP, la cual debe coincidir con la configurada en un conversor Ethernet/RS-232, a uno de cuyos puertos serie se conecta la **IFH 1G**.

El software de gestión para puntos de venta también puede consultar el valor de configuración de esta propiedad.

Ejemplos de uso:

```
...
HASAR1.DireccionIP = "192.0.5.23"
Debug.Print HASAR1.DireccionIP
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Configurar esta propiedad tiene sentido si la propiedad *Transporte()* toma alguno de los siguientes valores: SOCKET_TCP, o SOCKET_UDP.
- La idea de conectar la **IFH 1G** a un conversor Ethernet/RS-232, es para cubrir distancias que, por pérdida de señal, no se pueden alcanzar satisfactoriamente con un cable RS-232.
- Valor por defecto: “127.0.0.1”

Propiedad: DobleAncho()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: No requiere.

Retorno: *String*

Esta propiedad permite al software de gestión para puntos de venta obtener el carácter a utilizar como prefijo de textos que se deseen imprimir en doble ancho.

Ejemplo de uso:

```
...
HASAR1.ImprimirTextoNoFiscal HASAR1.DobleAncho & "Texto en doble ancho..."
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- La propiedad *DobleAncho()* retorna el carácter ASCII 244 decimal, el cual es interpretado por la **IFH 1G** como un carácter de control sólo si es el primero de un string cualquiera.
- Al imprimir en doble ancho, la longitud válida de un string, para un campo de un comando determinado, y soportado por la **IFH 1G**, se reduce a la mitad.
- Las **IFH 1G** permiten imprimir únicamente en doble ancho: líneas de nombre de fantasía, líneas de encabezado/pie del comprobante, líneas de texto fiscal, y líneas de texto no fiscal.
- La impresión en doble ancho sólo afecta al texto del comando que se está enviando. Si no se indica lo contrario, la próxima impresión será en simple ancho.

Propiedad: DocumentoDeReferencia()

Comando: *SetEmbarkNumber, GetEmbarkNumber*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: *Long* (de entrada), *String* (de entrada)

Retorno: *Long* (de entrada), *String* (de salida)

Característica: Multi-argumento

Esta propiedad permite al software de gestión para puntos de venta almacenar en la memoria de trabajo de la **IFH 1G** un texto, el cual será impreso en el encabezado de los comprobantes asociado a la leyenda: “*Comprobante Original Nº:*”, “*Remito Nº:*”, etc., según corresponda.

El software de gestión para puntos de venta también puede consultar, mediante esta propiedad, los textos ya almacenados en la memoria de trabajo de la **IFH 1G**.

Ejemplos de uso:

```
...
HASAR1.DocumentoDeReferencia(1) = "FA 0001-00000987"
Debug.Print HASAR1.DocumentoDeReferencia(1)
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- En la configuración se genera el string de comando *SetEmbarkNumber*. En la consulta se genera el string de comando *GetEmbarkNumber*.
- La invocación a la propiedad *DocumentoDeReferencia()*, para almacenar texto en la memoria de trabajo de la **IFH 1G**, debe hacerse antes de la apertura del comprobante.

- La invocación a la propiedad *DocumentoDeReferencia()* es obligatoria en el caso de Notas de Crédito, Tiques Nota de Crédito, y opcional en el resto de los comprobantes.

Propiedad: DocumentoEnCurso()

Comando: *StatusRequest*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: No requiere.

Retorno: enum *Documentos*

Esta propiedad permite al software de gestión para puntos de venta, de haber un comprobante abierto en la **IFH 1G**, saber cuál es el tipo de dicho comprobante (Factura 'A', Nota de Crédito 'B/C', etc.).

Ejemplo de uso:

```
...
Select Case HASAR1.DocumentoEnCurso
    Case D_NO_DOCUMENTO_EN_CURSO
        Debug.Print "No hay un comprobante abierto."
    Case D_FACTURA_B
        Debug.Print "Hay una Factura 'B' abierta."
End Select
...
```

Propiedad: Encabezado()

Comando: *SetHeaderTrailer, GetHeaderTrailer*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: *Long* (de entrada), *String* (de entrada)

Retorno: *Long* (de entrada), *String* (de salida)

Característica: Multi-argumento

Esta propiedad permite al software de gestión para puntos de venta almacenar en la memoria de trabajo de la **IFH 1G** un texto, el cual será impreso en alguna de las líneas de encabezado/pie de los comprobantes, según corresponda.

El software de gestión para puntos de venta también puede consultar, mediante esta propiedad, los textos ya almacenados en la memoria de trabajo de la **IFH 1G**.

Ejemplos de uso:

```
...
HASAR1.Encabezado(11) = "11 - Texto al pie del comprobante."
Debug.Print HASAR1.encabezado(11)
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- En la configuración se genera el string de comando *SetHeaderTrailer*. En la consulta se genera el string de comando *GetHeaderTrailer*.

- La invocación a la propiedad *Encabezado()*, para almacenar texto en la memoria de trabajo de la **IFH 1G**, debe hacerse antes de la apertura del comprobante si desea impresión en el encabezado, y antes del cierre del comprobante si se desea impresión en pie.
- Si los textos almacenados en la memoria de trabajo de la **IFH 1G**, para impresión de textos de encabezado/pie de comprobantes, no se elimina luego del cierre del mismo, los textos se imprimirán en todo comprobante futuro que se emita.

Por ejemplo,

```
...
HASAR1.CerrarComprobanteFiscal
HASAR1.Encabezado(11) = Chr(127)
...
```

El carácter ASCII 127 decimal (DEL) es interpretado por el **OCX Fiscal HASAR** como un carácter de control que significa eliminar de la memoria de trabajo de la **IFH 1G**.

- Las líneas 1 a 10 corresponden al encabezado del comprobante. Las líneas 11 a 20 corresponden al pie del mismo. No todas las líneas de encabezado/pie se imprimen en todos los tipos de comprobantes.
- Los textos se pueden imprimir en doble ancho, siempre que el primer carácter del texto sea el carácter ASCII 244 decimal (retornado por la propiedad *DobleAncho()* del **OCX Fiscal HASAR**).

Por ejemplo,

```
HASAR1.Encabezado(3) = HASAR1.DobleAncho & "Texto en línea 3 de encabezado."
```

Propiedad: EstadoControlador()

Comando: *StatusRequest*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: No requiere.

Retorno: *Long*

Esta propiedad permite al software de gestión para puntos de venta saber el código numérico del estado fiscal en que se encuentra la **Impresora Fiscal HASAR** al momento de la consulta.

Ejemplo de uso:

```
...
Debug.Print "Estado: " & HASAR1.EstadControlador & " => " & HASAR1.DescripcionEstadoControlador
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El valor devuelto por la propiedad *EstatControlador()* es alguno de los identificados por las constantes embebidas del **OCX Fiscal HASAR**, e identificadas bajo el enum *StateNibble*.

Propiedad: EventosIndividuales()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: *Boolean*

Retorno: *Boolean*

Esta propiedad permite al software de gestión para puntos de venta establecer la forma en que el **OCX Fiscal HASAR** generará los eventos de estado que se produzcan.

True Valor por defecto. Cada evento (bit de estado encontrado en '1') debe generarse individualmente.

False Se genera un único evento reportando todos los bits de estado encontrados en '1'.

Esta propiedad también permite al software de gestión para puntos de venta consultar el valor actual de configuración de esta propiedad.

Ejemplos de uso:

```
...
HASAR1.EventosIndividuales = True
Debug.Print HASAR1.EventosIndividuales
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Los dos primeros campos del string de respuesta a cualquier comando enviado (incluso si éste es rechazado) corresponden a *Status de Impresora* y *Status Fiscal*. Ambos campos contienen 4 caracteres ASCII que representan dígitos hexadecimales, los cuales deben convertirse a un número binario, y analizar cada bit para saber si se encuentra en '1'.
- La propiedad *EventosIndividuales()* en *True* resulta más cómoda, ya que todo el trabajo lo hace el **OCX Fiscal HASAR**, generando un evento por cada bit encontrado en '1'. El valor *False* en la propiedad significa que el software de gestión para puntos de venta deberá hacer el análisis de bits.

Propiedad: FechaHoraFiscal()

Comando: *SetDateTime, GetDateTime*

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: *Date*

Retorno: *Date*

Esta propiedad permite al software de gestión para puntos de venta actualizar la fecha y hora del reloj de tiempo real (RTC) de la **IFH 1G**.

Esta propiedad también permite al software de gestión para puntos de venta consultar la fecha y hora actual del reloj de tiempo real (RTC) de la **IFH 1G**.

Ejemplos de uso:

```

...
Debug.Print HASAR1.FechaHoraFiscal
HASAR1.ReporteZ
HASAR1.FechaHoraFiscal = "26/11/2013 14:49:34"
...

```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Para actualizar la fecha y hora de la **IFH 1G** se requiere de la realización previa de un cierre diario 'Z'. No así para el caso de la consulta.

Propiedad: HuboErrorFiscal()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: No requiere.

Retorno: *Boolean*

Esta propiedad permite al software de gestión para puntos de venta consultar si, para el último comando ejecutado previamente, se ha reportado algún error en el campo de *Status Fiscal* de la respuesta.

Ejemplo de uso:

```

...
HASAR1.CerrarComprobanteFiscal

If (HASAR1.HuboErrorFiscal) Then
    Debug.Print "Error fiscal: el comando ha sido rechazado..."
End If
...

```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- A diferencia de la captura de eventos, recurrir al uso de esta propiedad implica emplear la sentencia '*If/Then/End If*' luego de cada comando que se envía a la **IFH 1G**.
- Si la propiedad retorna el valor *True*, significa que algunos de los bits del campos de *Status Fiscal* de la respuesta al comando enviando se encontraban en '1'.

Propiedad: HuboErrorMecanico()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: No requiere.

Retorno: *Boolean*

Esta propiedad permite al software de gestión para puntos de venta consultar si, para el último comando ejecutado previamente, se ha reportado algún error en el campo de *Status de Impresora* de la respuesta, o la **IFH 1G** está enviando caracteres especiales DC4 (ASCII 20 decimal) para informar que le falta papel.

Ejemplo de uso:

```
...
HASAR1.CerrarComprobanteFiscal

If (HASAR1.HuboErrorMecánico) Then
    Debug.Print "Error mecánico: papel atascado, falta papel, etc..."
End If
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- A diferencia de la captura de eventos, recurrir al uso de esta propiedad implica emplear la sentencia '*If/Then/End If*' luego de cada comando que se envía a la **IFH 1G**.
- Si la propiedad retorna el valor *True*, significa que algunos de los bits del campos de *Status de Impresora* de la respuesta al comando enviando se encontraban en '1'.

Propiedad: HuboFaltaPapel()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: No requiere.

Retorno: *Boolean*

Esta propiedad permite al software de gestión para puntos de venta consultar si, para el último comando ejecutado previamente, la **IFH 1G** ha reportado falta de papel.

Ejemplo de uso:

```
...
HASAR1.CerrarComprobanteFiscal

If (HASAR1.HuboFaltaPapel) Then
    Debug.Print "Falta papel: realimentar por favor..."
End If
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- A diferencia de la captura de eventos, recurrir al uso de esta propiedad implica emplear la sentencia '*If/Then/End If*' luego de cada comando que se envía a la **IFH 1G**.
- Si la propiedad retorna el valor *True*, significa que algunos de los bits del campos de *Status de Impresora* de la respuesta al comando enviado se encontraban en '1'; o bien, la **IFH 1G** está enviando caracteres especiales DC4 (ASCII 20 decimal) para informar que le falta papel.

Propiedad: HuboStatPrn()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: No requiere.

Retorno: *Boolean*

Esta propiedad permite al software de gestión para puntos de venta consultar si, para el último comando ejecutado previamente, la **IFH 1G** ha reportado que la respuesta al mismo se encuentra demorada porque ha ocurrido un incidente (por ejemplo, la **IFH 1G** se encuentra *offline*).

Ejemplo de uso:

```
...
HASAR1.ModoStatPrn = True
...
HASAR1.ImprimirTextoNoFiscal "Una línea de texto no fiscal.."

If (HASAR1.HuboStatPrn) Then
    Do Until (Not (HASAR1.HuboStatPrn))
        HASAR1EnviarStatPrn
    Loop
End If
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- A diferencia de la captura de eventos, recurrir al uso de esta propiedad implica emplear la sentencia '*If/Then/End If*' luego de cada comando que se envía a la **IFH 1G**.
- Si la propiedad retorna el valor *True*, significa que la respuesta recibida no es la correspondiente al comando enviado, sino que identifica como una respuesta al comando *StatPRN*; como si el software de gestión para puntos de venta hubiese invocado al método *EnviarStatPrn()*.

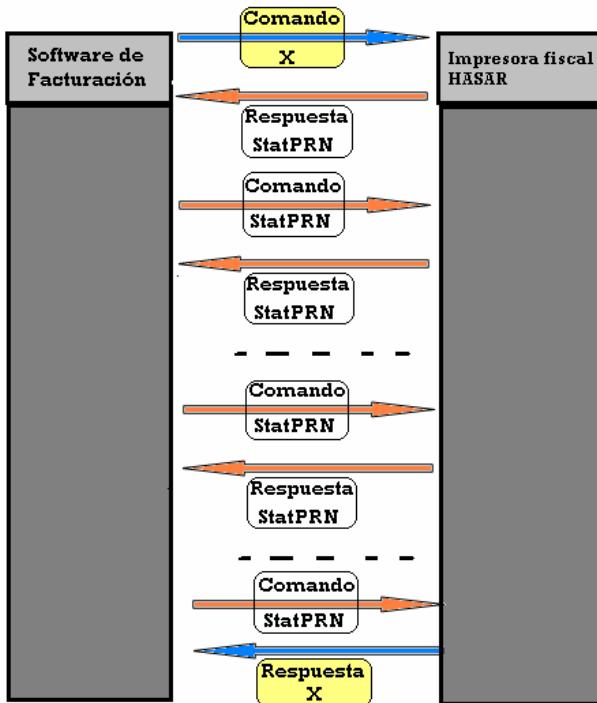
En condiciones normales, cuando el software de gestión para puntos de venta envía un string de comando a la **IFH 1G**, ésta responde con un string contenido información acorde al comando recibido.



Cuando ocurre un error en la **IFH 1G** durante la ejecución del comando enviado por el software de gestión para puntos de venta, se demora la respuesta al comando en cuestión hasta que el inconveniente sea solucionado. Luego, la **IFH 1G** responde como si el comando recibido hubiese sido el de consulta intermedia (*StatPRN*). Ante este hecho, el software de gestión para puntos de venta debe enviar el comando *StatPRN*. Proceso que deberá repetirse hasta que la respuesta recibida sea la correspondiente al comando originalmente enviado. Por ejemplo, cuando se envía un string de comando que causa impresión, y la **IFH 1G** se encuentra *offline*.

Con el **OCX Fiscal HASAR** estas situaciones pueden manejarse:

- *De forma transparente*.
Es el propio **OCX Fiscal HASAR** quien se encarga de manejar el diálogo con la **IFH 1G** hasta que el incidente se haya normalizado. Esto ocurre sólo si la propiedad *ModoStatPrn()* se encuentra configurada con el valor *False*.



- **Manualmente.**

Se permite que el software de gestión para puntos de venta maneje el diálogo con la **IFH 1G** hasta que el incidente se haya normalizado. La forma de resolverlo es como se muestra en el ejemplo de uso de la página anterior. En esta modalidad se requiere que la propiedad *ModoStatPrn()* se encuentre configurada con el valor *True*.

Cualquiera sea la modalidad que se adopte, es importante que el software de gestión para puntos de venta se asegure que el protocolo fiscal admite consultas de estado intermedio. Es decir, realizar consultas de estado entre el envío de un comando y su respuesta demorada. Para ello, puede consultar el valor de configuración de la propiedad *SoportaStatPRN()*; debe ser *True*.

Propiedad: ImpuestointernoFijo()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: *Boolean*

Retorno: *Boolean*

Esta propiedad permite al software de gestión para puntos de venta establecer una de las variantes que afectan a la interpretación, por parte del **OCX Fiscal HASAR**, de un valor relacionado con los impuestos internos que afectan a la venta de un ítem, o a la aplicación de un descuento/recargo.

True Establece que se trata de un impuestos interno fijo.

False Valor por defecto. No se trata de un impuesto interno fijo.

Esta propiedad también permite al software de gestión para puntos de venta consultar su valor de configuración actual.

Ejemplos de uso:

```

...
HASAR1.ImpuestointernoFijo = True
Debug.Print HASAR1.ImpuestointernoFijo
...

```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Todos los modelos y/o versiones de **IFH 1G** aceptan la aplicación de impuestos internos.
- Hay tres formas de especificar un impuesto interno: mediante un factor K (impuesto interno fijo, o variable), mediante un porcentaje (impuesto interno variable), o por monto (impuesto interno fijo).

No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan todas las formas de especificar el impuesto interno. Sin embargo, sí es posible especificar cualquiera de las formas a nivel del **OCX Fiscal HASAR**, ya que éste internamente se encarga de generar el string de comando con la especificación soportada.

- La interpretación del impuesto interno depende de tres propiedades: *ImpuestointernoFijo()* (la que se está describiendo), *ImpuestointernoPorMonto()*, y *KIVA()*. Como el **OCX Fiscal HASAR** interpreta la especificación del impuesto interno a partir de estas propiedades, se resume dicha interpretación en la tabla siguiente:

	Interpretación	Propiedad	Valor
Factor K	Imp. Int. Fijo	<i>KIVA()</i> <i>ImpuestointernoFijo()</i> <i>ImpuestointernoPorMonto()</i>	<i>True</i> <i>True</i> <i>False</i>
	Imp. Int. Variable	<i>KIVA()</i> <i>ImpuestointernoFijo()</i> <i>ImpuestointernoPorMonto()</i>	<i>True</i> <i>False</i> <i>False</i>
	Porcentaje (impuesto interno variable)	<i>KIVA()</i> <i>ImpuestointernoFijo()</i> <i>ImpuestointernoPorMonto()</i>	<i>False</i> <i>False</i> <i>False</i>
Monto Fijo (impuesto interno fijo)		<i>KIVA()</i> <i>ImpuestointernoFijo()</i> <i>ImpuestointernoPorMonto()</i>	<i>False</i> <i>True</i> <i>True</i>

Propiedad: **ImpuestointernoPorMonto()**

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones

Configuración: *Boolean*

Retorno: *Boolean*

Esta propiedad permite al software de gestión para puntos de venta establecer una de las variantes que afectan a la interpretación, por parte del **OCX Fiscal HASAR**, de un valor relacionado con los impuestos internos que afectan a la venta de un ítem, o a la aplicación de un descuento/recargo.

True Establece que se trata de un monto de impuesto interno.

False Valor por defecto. No se trata de un monto de impuesto interno.

Esta propiedad también permite al software de gestión para puntos de venta consultar su valor de configuración actual.

Ejemplos de uso:

```
...
HASAR1.ImpuestointernoPorMonto = True
Debug.Print HASAR1.ImpuestointernoPorMonto
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Todos los modelos y versiones de firmware fiscal de las **IFH 1G** aceptan la aplicación de impuestos internos.
- Hay tres formas de especificar un impuesto interno: mediante un factor K (impuesto interno fijo, o variable), mediante un porcentaje (impuesto interno variable), o por monto (impuesto interno fijo).

No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan todas las formas de especificar el impuesto interno. Sin embargo, sí es posible especificar cualquiera de las formas a nivel del **OCX Fiscal HASAR**, ya que éste internamente se encarga de generar el string de comando con la especificación soportada.

- La interpretación del impuesto interno depende de tres propiedades: *ImpuestoInternoPorMonto()* (la que se está describiendo), *ImpuestoInternoFijo()*, y *KIVA()*. Como el **OCX Fiscal HASAR** interpreta la especificación del impuesto interno a partir de estas propiedades, se resume dicha interpretación en la tabla siguiente:

	Interpretación	Propiedad	Valor
Factor K	Imp. Int. Fijo	<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>	<i>True</i> <i>True</i> <i>False</i>
		<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>	<i>True</i> <i>False</i> <i>False</i>
	Imp. Int. Variable	<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>	<i>False</i> <i>False</i> <i>False</i>
		<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>	<i>False</i> <i>True</i> <i>True</i>
Porcentaje (impuesto interno variable)			
Monto Fijo (impuesto interno fijo)			

Propiedad: InformacionRemito()

Comando: *SetEmbarkNumber*, *GetEmbarkNumber*

Disponible: **OCX Fiscal HASAR v010724**.

DEPRECADA en *v051122* y posteriores.

Configuración: *Long* (de entrada), *String* (de entrada)

Retorno: *Long* (de entrada), *String* (de salida)

Característica: Multi-argumento

Esta propiedad permite al software de gestión para puntos de venta almacenar en la memoria de trabajo de la **IFH 1G** un texto, el cual será impreso en el encabezado de los comprobantes, asociado a la leyenda: “*Comprobante Original Nº:*”, “*Remito Nº:*”, etc., según corresponda.

El software de gestión para puntos de venta también puede consultar, mediante esta propiedad, los textos ya almacenados en la memoria de trabajo de la **IFH 1G**.

Ejemplos de uso:

```
...
HASAR1.InformacionRemito(1) = "FA 0001-00000987"
Debug.Print HASAR1.InformacionRemito(1)
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- En la configuración se genera el string de comando *SetEmbarkNumber*. En la consulta se genera el string de comando *GetEmbarkNumber*.
- La invocación a la propiedad *InformacionRemito()*, para almacenar texto en la memoria de trabajo de la **IFH 1G**, debe hacerse antes de la apertura del comprobante.
- La invocación a la propiedad *InformacionRemito()* es obligatoria en el caso de Notas de Crédito, Tickets Nota de Crédito, y opcional en el resto de los comprobantes.

Propiedad: IndicadorFiscal()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponibilidad: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: *Long*

Retorno: *Boolean*

Esta propiedad permite al software de gestión para puntos de venta saber si determinado bit se encuentra en '1', examinando el campo de *Status Fiscal* de la respuesta al último string de comando enviado a la **IFH 1G**.

Ejemplo de uso:

```
...
HASAR1.CerrarComprobanteFiscal

If (HASAR1.HuboErrorFiscal) Then
    Debug.Print HASAR1.IndicadorFiscal(F_INVALID_COMMAND)
End If
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente:

- A diferencia de la captura de eventos, recurrir al uso de esta propiedad implica emplear la sentencia '*If/Then/End If*' luego de cada string de comando que se envía a la **IFH 1G**.
- El argumento de invocación de la propiedad debe ser alguno de los valores identificados por las constantes embebidas en el **OCX Fiscal HASAR**, y correspondientes al enum *FiscalBits*.

Propiedad: IndicadorImpresora()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponibilidad: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: *Long*

Retorno: *Boolean*

Esta propiedad permite al software de gestión para puntos de venta saber si determinado bit se encuentra en '1', examinando el campo de *Status de Impresora* de la respuesta al último string de comando enviado a la **IFH 1G**.

Ejemplo de uso:

```
...
HASAR1.CerrarComprobanteFiscal

If (HASAR1.HuboErrorMecanico) Then
    Debug.Print HASAR1.IndicadorImpresora(P_PRINTER_ERROR)
End If
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente:

- A diferencia de la captura de eventos, recurrir al uso de esta propiedad implica emplear la sentencia '*If/Then/End If*' luego de cada string de comando que se envía a la **IFH 1G**.
- El argumento de invocación de la propiedad debe ser alguno de los valores identificados por las constantes embebidas en el **OCX Fiscal HASAR**, y correspondientes al enum *PrinterBits*.

Propiedad: Interlineado()

Comando: *SetConfigPage, GetConfigPage*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: *Long*

Retorno: *Long*

Esta propiedad permite al software de gestión para puntos de venta establecer el interlineado que deberá respetar la **IFH 1G** en impresiones sobre hojas sueltas (válido sólo en modelos y versiones de firmware fiscal que emplean tecnología de impresión láser).

El software de gestión para puntos de venta también puede consultar, mediante esta propiedad, el valor de interlineado que actualmente tiene configurado la **IFH 1G**.

Ejemplos de uso:

```
...
HASAR1.Interlineado = 8
Debug.Print HASAR1.Interlineado
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Cuando se invoca a la propiedad *Interlineado()* para configuración, el **OCX Fiscal HASAR** genera el string de comando *SetConfigPage*. Cuando se la invoca para consulta, genera el string de comando *GetConfigPage*.
- Los valores posibles de configuración son: 6 (valor por defecto) u 8.
- El único modelo de **IFH 1G** (en todas sus versiones de firmware fiscal) que acepta configuración de interlineado es el SMH/PL-23F.

Propiedad: KIVA()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones

Configuración: *Boolean*

Retorno: *Boolean*

Esta propiedad permite al software de gestión para puntos de venta establecer una de las variantes que afectan a la interpretación, por parte del **OCX Fiscal HASAR**, de un valor relacionado con los impuestos internos que afectan a la venta de un ítem, o a la aplicación de un descuento/recargo.

True Establece que se trata de un valor válido de factor K.

False Valor por defecto. No se trata de un valor representativo del factor K.

Esta propiedad también permite al software de gestión para puntos de venta consultar su valor de configuración actual.

Ejemplos de uso:

```
...
HASAR1.KIVA = True
Debug.Print HASAR1.KIVA
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Todos los modelos y versiones de firmware fiscal de las **IFH 1G** aceptan la aplicación de impuestos internos.
- Hay tres formas de especificar un impuesto interno: mediante un factor K (impuesto interno fijo, o variable), mediante un porcentaje (impuesto interno variable), o por monto (impuesto interno fijo).

No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan todas las formas de especificar el impuesto interno. Sin embargo, sí es posible especificar cualquiera de las formas a nivel del **OCX Fiscal HASAR**, ya que éste internamente se encarga de generar el string de comando con la especificación soportada.

- La interpretación del impuesto interno depende de tres propiedades: *KIVA()* (la que se está describiendo), *ImpuestointernoFijo()*, e *ImpuestointernoPorMonto()*. Como el **OCX Fiscal HASAR** interpreta la especificación del impuesto interno a partir de estas propiedades, se resume dicha interpretación en la tabla siguiente:

	Interpretación	Propiedad	Valor
Factor K	Imp. Int. Fijo	<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>	<i>True</i> <i>True</i> <i>False</i>
	Imp. Int. Variable	<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>	<i>True</i> <i>False</i>
		<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>	<i>False</i> <i>False</i> <i>False</i>
	Porcentaje (impuesto interno variable)	<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>	<i>False</i> <i>False</i> <i>False</i>
Monto Fijo (impuesto interno fijo)		<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>	<i>False</i> <i>True</i>
		<i>KIVA()</i> <i>ImpuestoInternoFijo()</i> <i>ImpuestoInternoPorMonto()</i>	<i>True</i>

Propiedad: Modelo()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: enum *ModelosDeImpresoras*

Retorno: enum *ModelosDeImpresoras*

Esta propiedad permite al software de gestión para puntos de venta especificar el modelo y versión de firmware fiscal de la **IFH 1G** con la cual se va a dialogar.

Esta propiedad también permite al software de gestión para puntos de venta consultar su valor de configuración actual.

Ejemplos de uso:

```
...
HASAR1.Modelo = MODELO_P1120
Debug.Print HASAR1.Modelo
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El valor de la propiedad *Modelo()* también puede ser modificado automáticamente (en forma interna) por el **OCX Fiscal HASAR** cuando se invoca a la propiedad *AutodetectarModelo()*.
- Configurar esta propiedad con un modelo y versión de firmware fiscal determinado de **IFH 1G** causa que el **OCX Fiscal HASAR** (des)habilite propiedades, métodos y/o constantes embebidas. La finalidad es poder generar strings de comandos soportados por la **IFH 1G** indicada.
- Configurar con un valor incorrecto la propiedad *Modelo()*, o configurarla con un valor correcto pero conectando una **IFH 1G** diferente a la indicada, puede causar que el **OCX Fiscal HASAR** dispare excepciones por funciones no soportadas, y/o eventos de error fiscal por comandos con sintaxis inadecuada.

Propiedad: ModoStatPrn()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: *Boolean*

Retorno: *Boolean*

Esta propiedad permite al software de gestión para puntos de venta indicarle al **OCX Fiscal HASAR** de que forma se manejarán las consultas de estado intermedio (si la **IFH 1G** las soporta).

Esta propiedad también permite al software de gestión para puntos de venta consultar su valor de configuración actual.

True El manejo de las consultas de estado intermedio será realizado por el software de gestión para puntos de venta.

False Valor por defecto. El manejo será transparente para el software de gestión para puntos de venta. Será realizado internamente por el **OCX Fiscal HASAR**.

Ejemplo de uso:

```
...
HASAR1.ModoStatPrn = True
...
HASAR1.ImprimirTextoNoFiscal "Una línea de texto no fiscal.."
If (HASAR1.HuboStatPrn) Then
    Do Until (Not (HASAR1.HuboStatPrn))
        HASAR1EnviarStatPrn
    Loop
End If
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Se denomina consulta de estado intermedio a la consulta que se puede realizar en el período de tiempo que transcurre entre el envío de un comando y la recepción de su respuesta; sólo en los casos donde ésta última se encuentra demorada a causa de un incidente (la **IFH 1G** está esperando la normalización del incidente).
- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan las consultas de estado intermedio. El software de gestión para puntos de venta puede averiguarlo invocando a la propiedad *SoportaStatPRN()*.

Propiedad: NombreArchivoLog()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR v110423** y posteriores.

Configuración: *String*

Retorno: *String*

Esta propiedad permite al software de gestión para puntos de venta establecer un path existente y el nombre de un archivo a crear al final de ese path, el cual será utilizado por el **OCX Fiscal HASAR** para dejar registros de su actividad. Este archivo se denomina “log” del **OCX Fiscal HASAR**.

Esta propiedad también permite al software de gestión para puntos de venta consultar su valor de configuración actual.

Ejemplos de uso:

```
...
HASAR1.NombreArchivoLog = "D:\Logs\fiscal.log"
Debug.Print HASAR1.NombreArchivoLog
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El path debe existir en el host (el **OCX Fiscal HASAR** no lo crea). En cambio, en esa ubicación sí crea el archivo con el nombre y extensión que se indique.
- Como el archivo creado es abierto cada vez que es necesario escribir en él, y cerrado luego de la escritura, es posible modificar el contenido de esta propiedad y los registros de la actividad del **OCX Fiscal HASAR** se registrarán en el nuevo archivo a partir de la modificación de esta propiedad.
- El archivo “log” no crece indefinidamente, puede alcanzar un máximo de 5 Mb. Colmada su capacidad de información, es renombrado -a modo de backup- y se continua escribiendo en el archivo original desde cero nuevamente. Este proceso se repite cada vez que se llena el archivo “log”. En consecuencia, el espacio máximo que puede estar ocupándose en disco es de 10 Mb (5 Mb para el archivo en uso, y 5 Mb de backup).
- Se puede tener, si así es requerido, un archivo “log” diferente por cada instancia del **OCX Fiscal HASAR**.
- De no especificarse nunca un string que identifique a un archivo “log”, el **OCX Fiscal HASAR** inspeccionará si entre las variables de entorno (de sistema) de Window® existe la variable FILELOG, y que valor tiene. De existir, el **OCX Fiscal HASAR** dejará registro de su actividad en la ubicación y archivo que la variable FILELOG indique. Es claro que de haber más de una instancia del **OCX Fiscal HASAR**, todas registrarán su actividad sobre el mismo archivo. En tales circunstancias no se podrá distinguir que corresponde a quien.

Propiedad: NombreDeFantasia()

Comando: *SetFantasyName, GetFantasyName*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: *Long* (de entrada), *String* (de entrada)

Retorno: *Long* (de entrada), *String* (de salida)

Característica: Multi-argumento

Esta propiedad permite al software de gestión para puntos de venta almacenar en la memoria de trabajo de la **IFH 1G** un texto, el cual será impreso como una de las líneas de nombre de fantasía en el encabezado de los comprobantes.

El software de gestión para puntos de venta también puede consultar, mediante esta propiedad, los textos ya almacenados en la memoria de trabajo de la **IFH 1G**.

Ejemplos de uso:

```
...
HASAR1.NombreDeFantasia(1) = "1 - Texto de nombre de fantasía."
Debug.Print HASAR1.NombreDeFantasia(1)
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- En la configuración se genera el string de comando *SetFantasyName*. En la consulta se genera el string de comando *GetFantasyName*.
- La invocación a la propiedad *NombreDeFantasia()*, para almacenar texto en la memoria de trabajo de la **IFH 1G**, debe hacerse antes de la apertura del comprobante.
- Si los textos almacenados en la memoria de trabajo de la **IFH 1G**, para impresión de textos de nombre de fantasía, no se eliminan luego del cierre del comprobante, dichos textos se imprimirán en todo comprobante futuro que se emita.

Por ejemplo,

```
...
HASAR1.CerrarComprobanteFiscal
HASAR1.NombreDeFantasia(1) = Chr(127)
...
```

El carácter ASCII 127 decimal (DEL) es interpretado por el **OCX Fiscal HASAR** como un carácter de control que significa eliminar de la memoria de trabajo de la **IFH 1G**.

- Sólo pueden almacenarse las líneas 1 y 2.
- Los textos se pueden imprimir en doble ancho, siempre que el primer carácter del texto sea el carácter ASCII 244 decimal (retornado por la propiedad *DobleAncho()* del **OCX Fiscal HASAR**).

Por ejemplo,

```
HASAR1.NombreDeFantasia(2) = HASAR1.DobleAncho & "Texto en línea 2 de fantasía."
```

Propiedad: PáginasDeUltimoDocumento()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: No requiere.

Retorno: *Long*

Esta propiedad permite al software de gestión para puntos de venta obtener la cantidad de hojas que componen el último comprobante emitido.

Ejemplo de uso:

```
...
HASAR1.CerrarComprobanteFiscal
Debug.Print HASAR1.PaginasDeUltimoDocumento
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- En el caso de impresiones sobre rollo, esta propiedad retorna el valor '1'.

Propiedad: PrecioBase()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: *Boolean*

Retorno: *Boolean*

Esta propiedad permite al software de gestión para puntos de venta indicarle al **OCX Fiscal HASAR** como debe interpretar los montos representativos de precios unitarios, o correspondientes a descuentos/recargos.

Esta propiedad también permite al software de gestión para puntos de venta consultar su valor de configuración actual.

- True* El monto indicado por el software de gestión para puntos de venta no incluye los impuestos que pudiesen afectarlo (IVA o impuestos internos).
- False* Valor por defecto. El monto indicado por el software de gestión para puntos de venta incluye los impuestos que pudiesen afectarlo (IVA o impuestos internos).

Ejemplos de uso:

```
...
HASAR1.PrecioBase = False
Debug.Print HASAR1.PrecioBase
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- No es necesario modificar el valor de esta propiedad (y en consecuencia el monto que se indica) en función del comprobante a emitir ('A' o 'B/C'). La **IFH 1G** se encarga de (des)armar el monto en/con sus respectivas componentes para mostrar los valores correctamente en el documento a emitir, y a partir de las indicaciones recibidas.

Propiedad: PrimerNumeroDeDocumentoActual()

Esta propiedad no genera, ni envía string de comando alguno a la **Impresora Fiscal HASAR**.

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: No requiere.

Retorno: *Long*

Esta propiedad permite al software de gestión para puntos de venta obtener el número de comprobante correspondiente a la primera hoja del comprobante en curso.

Ejemplo de uso:

```
...
HASAR1.AbrirComprobanteFiscal FACTURA_B
Debug.Print HASAR1.PrimerNumeroDeDocumentoActual
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El comprobante se debe encontrar abierto.
- La consulta es válida sólo en comprobantes que son numerados por la **IFH 1G**: comprobantes fiscales, Notas de Crédito, Tiques Nota de Crédito, y Remitos.

Propiedad: Puerto()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: *Long*

Retorno: *Long*

Esta propiedad permite al software de gestión para puntos de venta indicarle al **OCX Fiscal HASAR** cual es el puerto de conexión con la **IFH 1G**.

Esta propiedad también permite al software de gestión para puntos de venta consultar su valor de configuración actual.

Ejemplo de uso:

```
...
HASAR1.Puerto = 3
Debug.Print HASAR1.Puerto
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- En el **OCX Fiscal HASAR v010724** esta propiedad identifica, únicamente, a un puerto serie RS-232. Por ejemplo, 1=COM1, 2=COM2, etc.
- En el **OCX Fiscal HASAR v051122** y posteriores, el valor de esta propiedad va de la mano con lo establecido en la propiedad *Transporte()*.

Transporte() = PUERTO_SERIE

Puerto() identifica un puerto serie RS-232.

Transporte() = SOCKET_TCP
 SOCKET_UDP

Puerto() identifica a un socket asociado a un puerto serie RS-232 perteneciente a un conversor Ethernet/RS-232 al cual se conecta la **IFH 1G**.

- El valor por defecto de esta propiedad es '1'.

Propiedad: ReintentoConstante()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: *Boolean*

Retorno: *Boolean*

Esta propiedad permite al software de gestión para puntos de venta establecer la forma de manejo del reinicio de envío de paquetes de información hacia la **IFH 1G**.

Esta propiedad también permite al software de gestión para puntos de venta consultar su valor de configuración actual.

True La cantidad de reinicios se vuelve infinita.

False Valor por defecto. La cantidad de reinicios es un número finito, dependiente de la propiedad *Reintentos()*, cuyo valor por defecto es 3.

Ejemplos de uso:

```
...
HASAR1.ReintentoConstante = True
Debug.Print HASAR1.ReintentoConstante
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Cuando esta propiedad se configura con el valor *True*, cada vez que se produzca un error por *time out*, el **OCX Fiscal HASAR** disparará el evento *ImpresoraNoResponde()*.
- Cuando esta propiedad se configura con el valor *True*, el proceso de reinicios en forma constante sólo puede ser interrumpido por el software de gestión para puntos de venta invocando al método *Abortar()*.

Propiedad: Reintentos()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: *Long*

Retorno: *Long*

Esta propiedad permite al software de gestión para puntos de venta establecer la cantidad de reinicios de envío de un comando a la **IFH 1G**, por parte del **OCX Fiscal HASAR**, cuando se produce algún inconveniente en las comunicaciones.

Esta propiedad también permite al software de gestión para puntos de venta consultar su valor de configuración actual.

Ejemplos de uso:

```
...
```

```
HASAR1.Reintentos = 3  
Debug.Print HASAR1.Reintentos  
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El valor por defecto de esta propiedad es: 3
- El valor de esta propiedad es tenido en cuenta por el **OCX Fiscal HASAR** si la propiedad *Reintento-Constante()* se encuentra configurada con el valor *False*.

Propiedad: Respuesta()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: *Long*

Retorno: *String*

Esta propiedad permite al software de gestión para puntos de venta obtener un determinado campo de información desde un string de respuesta de la **IFH 1G**, correspondiente al último string de comando enviado.

Ejemplo de uso:

```
...  
HASAR1.PedidoDeStatus  
nro_FB = HASAR1.Respuesta(3)  
...
```

En este ejemplo se solicita el tercer campo de información correspondiente a la respuesta del comando *StatusRequest*. Se trata del último número utilizado en comprobantes fiscales ‘B/C’.

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- Si la respuesta al string de comando enviado tiene ‘n’ campos, no se puede solicitar un número de campo mayor a ‘n’. De hacerlo, el **OCX Fiscal HASAR** disparará una excepción por “Campo no encontrado...”.
- Si n = 0 la propiedad *Respuesta()* retorna el string completo. Resulta equivalente a invocar al método *Recibir()*.
- Cuando un string de comando es rechazado, su string de respuesta contiene sólo los dos primeros campos: *Status de Impresora* y *Status Fiscal*.
- A partir del **OCX Fiscal HASAR v051122** se incorporan argumentos de salida en los diferentes métodos que lo componen. Esto marca la tendencia a la deprecación futura de la propiedad *Respuesta()*.

Propiedad: ResumenIVA()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: enum *TiposDeResumenIVA*

Retorno: enum *TiposDeResumenIVA*

Esta propiedad permite al software de gestión para puntos de venta establecer la forma en que el **OCX Fiscal HASAR** solicitará el desglose de impuestos: correspondientes al documento en curso (o recién emitido), o correspondiente a los acumulados hasta el momento en la memoria de trabajo de la **IFH 1G**.

Esta propiedad también permite al software de gestión para puntos de venta consultar su valor de configuración actual.

Ejemplos de uso:

```
...
HASAR1.ResumenIVA = RESUMEN_IVA_DOCUMENTO
Debug.Print HASAR1.ResumenIVA
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la consulta de desglose de impuestos mientras se encuentra abierto un comprobante.
- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la consulta de desglose de impuestos correspondiente a los acumulados de la memoria de trabajo.

Propiedad: **SoportaStatPRN()**

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: No requiere.

Retorno: Boolean

Esta propiedad permite al software de gestión para puntos de venta saber si la **IFH 1G** configurada en la propiedad *Modelo()* soporta la consulta de estado intermedio (consultas de estado posibles de realizar entre el envío del string de comando y la recepción del string de respuesta, demorada ante la ocurrencia de un incidente).

True Soporta las consultas de estado intermedio.

False Valor por defecto. No soporta este tipo de consultas.

Ejemplo de uso:

```
...
HASAR1.SoportaStatPRN = True
Debug.Print HASAR1.SoportaStatPRN
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El valor *True* significa que la **IFH 1G** soporta el comando *StatPRN* que se genera por invocación del método *EnviarStatPrn()*.

Propiedad: TiempoDeEspera()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: *Long*

Retorno: *Long*

Esta propiedad permite al software de gestión para puntos de venta establecer el tiempo, en milisegundos, que el **OCX Fiscal HASAR** deberá esperar para recibir la respuesta a un comando enviado, antes de reintentar su envío.

Esta propiedad también permite al software de gestión para puntos de venta consultar su valor de configuración actual.

Ejemplos de uso:

```
...
HASAR1.TiempoDeEspera = 3000
Debug.Print HASAR1.TiempoDeEspera
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- El valor por defecto de esta propiedad es: 3000

Propiedad: Transporte()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: enum *TiposDeTransporte*

Retorno: enum *TiposDeTransporte*

Esta propiedad permite al software de gestión para puntos de venta establecer el vínculo de conexión con la **IFH 1G**: RS-232 o Ethernet.

Esta propiedad también permite al software de gestión para puntos de venta consultar su valor de configuración actual.

Ejemplos de uso:

```
...
HASAR1.Transport = PUERTO_SERIE
Debug.Print HASAR1.Transport
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- La conexión vía Ethernet tiene sentido cuando la **IFH 1G** se conecta a un puerto serie perteneciente a un conversor Ethernet/RS-232.

Propiedad: UltimaNotaCreditoA()

Comando: *StatusRequest*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: No requiere.

Retorno: *Long*

Esta propiedad permite al software de gestión para puntos de venta obtener el último número utilizado para Notas de Crédito 'A' (haya sido ésta emitida, cancelada, o se encuentre en curso de emisión).

Ejemplo de uso:

```
...
Debug.Print HASAR1.UltimaNotaCreditoA
...
```

Propiedad: UltimaNotaCreditoBC()

Comando: *StatusRequest*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: No requiere.

Retorno: *Long*

Esta propiedad permite al software de gestión para puntos de venta obtener el último número utilizado para Notas de Crédito 'B/C' (haya sido ésta emitida, cancelada, o se encuentre en curso de emisión).

Ejemplo de uso:

```
...
Debug.Print HASAR1.UltimaNotaCreditoBC
...
```

Propiedad: UltimoDocumentoFiscalA()

Comando: *StatusRequest*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: No requiere.

Retorno: *Long*

Esta propiedad permite al software de gestión para puntos de venta obtener el último número utilizado para comprobantes fiscales 'A' (haya sido éste emitido, cancelado, o se encuentre en curso de emisión).

Ejemplo de uso:

```
...
Debug.Print HASAR1.UltimoDocumentoFiscalA
```

...

Propiedad: UltimoDocumentoFiscalBC()

Comando: *StatusRequest*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: No requiere.

Retorno: *Long*

Esta propiedad permite al software de gestión para puntos de venta obtener el último número utilizado para comprobantes fiscales 'B/C' (haya sido éste emitido, cancelado, o se encuentre en curso de emisión).

Ejemplo de uso:

```
...
Debug.Print HASAR1.UltimoDocumentoFiscalBC
...
```

Propiedad: UltimoDocumentoFueCancelado()

Comando: *StatusRequest*

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: No requiere.

Retorno: *Boolean*

Esta propiedad permite al software de gestión para puntos de venta saber si el último comprobante impreso fue cancelado o emitido correctamente.

True El último comprobante ha sido cancelado.

False El último comprobante se ha emitido correctamente.

Ejemplo de uso:

```
...
If (HASAR1.UltimoDocumentoFueCancelado) Then
    Debug.Print "CANCELADO !!"
Else
    Debug.Print "EMITIDO CORRECTAMENTE !!"
```

End If

...

Propiedad: UltimaFactura()

Comando: *StatusRequest*

Disponible: **OCX Fiscal HASAR v010724.**

DEPRECADA v051122 y posteriores.

Configuración: No requiere.

Retorno: *Long*

Esta propiedad permite al software de gestión para puntos de venta obtener el último número utilizado para Tiques Factura 'A' (haya sido éste emitido, cancelado, o se encuentre en curso de emisión).

Ejemplo de uso:

```
...
Debug.Print HASAR1.UltimaFactura
...
```

Propiedad: UltimoTicket()

Comando: *StatusRequest*

Disponible: **OCX Fiscal HASAR v010724.**
DEPRECADA v051122 y posteriores.

Configuración: No requiere.

Retorno: *Long*

Esta propiedad permite al software de gestión para puntos de venta obtener el último número utilizado para Tiques, o Tiques Factura 'B/C' (haya sido éste emitido, cancelado, o se encuentre en curso de emisión).

Ejemplo de uso:

```
...
Debug.Print HASAR1.UltimoTicket
...
```

Propiedad: UltimoRemito()

Comando: *StatusRequest*

Disponible: **OCX Fiscal HASAR v051122 y posteriores.**

Configuración: No requiere.

Retorno: *Long*

Esta propiedad permite al software de gestión para puntos de venta obtener el último número utilizado para Remitos (haya sido éste emitido, cancelado, o se encuentre en curso de emisión).

Ejemplo de uso:

```
...
Debug.Print HASAR1.UltimoRemito
...
```

Propiedad: UsarASCII()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas las versiones.

Configuración: *Boolean*

Retorno: *Boolean*

Esta propiedad permite al software de gestión para puntos de venta establecer quien es el responsable de la conversión del juego de caracteres de Windows® a caracteres ASCII (CP437).

Esta propiedad también permite al software de gestión para puntos de venta consultar su valor de configuración actual.

True El responsable de la conversión es el software de gestión para puntos de venta.

False Valor por defecto. La conversión se realizará en forma interna por parte del **OCX Fiscal HASAR**.

Ejemplos de uso:

```
...
HASAR1.UsarASCII = True
Debug.Print HASAR1.UsarASCII
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- La **IFH 1G** espera recibir (y entrega) strings de caracteres ASCII (CP437).

Propiedad: UsarDisplay()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR**, en todas sus versiones.

Configuración: *Boolean*

Retorno: *Boolean*

Esta propiedad permite al software de gestión para puntos de venta establecer si se usará el visor de cliente conectado a la **IFH 1G** –de soportarlo- en los comandos que permiten enviar automáticamente información a tal dispositivo.

Esta propiedad también permite al software de gestión para puntos de venta consultar su valor de configuración actual.

True Valor por defecto. Utilizar visor de cliente.

False No utilizarlo.

Ejemplos de uso:

```
...
HASAR1.UsarDisplay = True
Debug.Print HASAR1.UsarDisplay
...
```

El desarrollador del software de gestión para puntos de venta debe tener presente que:

- No todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan el uso de visor de cliente.

Propiedad: VerificacionCompletaDeErrores()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: *Boolean*

Retorno: *Boolean*

Esta propiedad permite al software de gestión para puntos de venta establecer si, en caso de producirse un error fiscal, el **OCX Fiscal HASAR** -además del evento correspondiente- produce una excepción asociada.

Esta propiedad también permite al software de gestión para puntos de venta consultar su valor de configuración actual.

True Además del evento correspondiente, producir excepción asociada.

False Valor por defecto. No producir la excepción. Sólo el evento correspondiente.

Ejemplos de uso:

```
...
HASAR1.VerificacionCompletaDeErrores = True
Debug.Print HASAR1.VerificacionCompletaDeErrores
...
```

Propiedad: Version()

Esta propiedad no genera, ni envía string de comando alguno a la **IFH 1G**.

Disponible: **OCX Fiscal HASAR v051122** y posteriores.

Configuración: No requiere.

Retorno: *String*

Esta propiedad permite al software de gestión para puntos de venta conocer la versión de **OCX Fiscal HASAR** que está empleando.

Ejemplo de uso:

```
...
Debug.Print HASAR1.Version
...
```

Constantes Embebidas en el OCX Fiscal HASAR

Tener presente que por problemas propios de algunos lenguajes de desarrollo, éstos no ven las constantes embebidas (definidas por enumeración) que se describen a continuación. En esos casos no se podrá referenciar a las constantes embebidas a utilizar por su nombre, sino que se deberán definir por código, o utilizar directamente el valor decimal que se muestra en las tablas siguientes.

Enum: ModelosDelImpresoras

Modelo SMH/	Constantes Embebidas		OCX Fiscal HASAR			
	Nombre	Dec	v010724	v051122	v110423	v150423
P-614F	MODELO_614	1	SI	SI	SI	SI
P-615F	MODELO_615	2	SI	SI	SI	SI
P-PR4F	MODELO_PR4	3	SI	SI	SI	SI
P-950F	MODELO_950	4	SI	SI	SI	SI
P-951F	MODELO_951	5	SI	SI	SI	SI
PT-262F	MODELO_262	6	SI	SI	SI	SI
PJ-20F	MODELO_PJ20	7	SI	SI	SI	SI
P-320F	MODELO_P320	8	SI	SI	SI	SI
P-715F	v01.00	9		SI	SI	SI
"	v02.01	25		SI	SI	SI
"	v03.02	29		SI	SI	SI
"	v04.03	30		SI	SI	SI
"	v05.04	40				SI
P-PR5F	v01.00	10		SI	SI	SI
"	v02.01	26		SI	SI	SI
"	v03.02	37			SI	SI
PT-272F	v01.00	11		SI	SI	SI
"	v02.01	38			SI	SI
PL-8F	v01.00	12		SI	SI	SI
"	v02.01	17		SI	SI	SI
P-321F	v01.00	13		SI	SI	SI
P-322F	v01.00	14		SI	SI	SI
"	v02.01	18		SI	SI	SI
P-425F	v01.00	15		SI	SI	SI
"	v02.01	16		SI	SI	SI
P-330F	v01.00	19		SI	SI	SI
"	v02.01	21		SI	SI	SI
"	v02.02	23		SI	SI	SI
"	v02.03	31		SI	SI	SI
P-435F	v01.00	20		SI	SI	SI
"	v01.01	24		SI	SI	SI
"	v01.02	27		SI	SI	SI
"	v02.03	34		SI	SI	SI
PL-9F	v01.00	22		SI	SI	SI
PL-23F	v01.00	28		SI	SI	SI
"	v01.01	33		SI	SI	SI
"	v02.02	36			SI	SI

Modelo SMH/	Constantes Embebidas		OCX Fiscal HASAR			
	Nombre	Dec	v010724	v051122	v110423	v150423
P-441F v01.00	MODELO_P441	32		SI	SI	SI
P-441F v02.01	MODELO_P441_201	39				SI
P-1120F v01.00	MODELO_P1120	35		SI	SI	SI
P-340F v01.00	MODELO_P340	41				si

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como “NO SOPORTADO”

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben asignarse a la propiedad *Modelo()* para establecer en el **OCX Fiscal HASAR** la identificación de la **IFH 1G** con la cual se va a dialogar. Generalmente, esta acción es el resultado de una selección manual por parte del usuario del software de gestión para puntos de venta.

Por ejemplo,

HASAR1.Modelo = MODELO_P441

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior son, también, las utilizadas por el método *AutodetectarModelo()* cuando la identificación de la **IFH 1G** la hace automáticamente el software de gestión para puntos de venta. La asignación del valor correspondiente a la propiedad *Modelo()* es resuelta internamente por el **OCX Fiscal HASAR**.

Por ejemplo,

HASAR1.AutodetectarModelo

Enum: TiposDeDocumento

Nombre	Dec	OCX Fiscal HASAR		
		v010724	v051122	v110423
TIPO_NINGUNO	32	SI	SI	SI
TIPO_CUIT	67	SI	SI	SI
TIPO_CUIL	76		SI	SI
TIPO_DNI	50	SI	SI	SI
TIPO_CI	52	SI	SI	SI
TIPO_LE	48	SI	SI	SI
TIPO_LC	49	SI	SI	SI
TIPO_PASAPORTE	51	SI	SI	SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como “NO SOPORTADO”

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior son las que pueden emplearse como uno de los argumentos al invocar al método *DatosCliente()*.

Por ejemplo,

HASAR1.DatosCliente “Razón Social...”, 99999999995, TIPO_CUIT, RESPONSABLE_INSCRIPTO, “Domicilio...”

Tener presente que algunas **IFH 1G** no soportan a TIPO_CUIL como tipo de documento del cliente. Consultar el *Manual de Comandos* del equipo fiscal.

Enum: TiposDeResponsabilidades

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
RESPONSABLE_INSCRIPTO	73	SI	SI	SI
RESPONSABLE_NO_INSCRIPTO	78	SI	SI	SI
RESPONSABLE_EXENTO	69	SI	SI	SI
NO_RESPONSABLE	65	SI	SI	SI
CONSUMIDOR_FINAL	67	SI	SI	SI
BIENES_DE_USO	66	SI	SI	SI
MONOTRIBUTO	77	SI	SI	SI
NO_CATEGORIZADO	84	SI	SI	SI
MONOTRIBUTO_SOCIAL	83			SI
EVENTUAL	86			SI
EVENTUAL_SOCIAL	87			SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como “NO SOPORTADO”

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior son las que pueden emplearse como uno de los argumentos al invocar al método *DatosCliente()*.

Por ejemplo,

HASAR1.DatosCliente “Razón Social...”, 99999999995, TIPO_CUIT, RESPONSABLE_INSCRIPTO, “Domicilio...”

Las categorías tabuladas corresponden a los receptores de los comprobantes que se emiten mediante la **IFH 1G**.

Tener presente que no todas las categorías están soportadas en todos los modelos de **IFH 1G**. Consultar el *Manual de Comandos* del equipo fiscal.

Enum: TiposResponsabilidadImpresor

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Valor	v010724	v051122	v110423 v150423
CLASE_RESPONSABLE_INSCRIPTO	73	SI	SI	SI
CLASE_RESPONSABLE_NO_INSCRIPTO	78	SI	SI	SI
CLASE_RESPONSABLE_EXENTO	69	SI	SI	SI
CLASE_NO_RESPONSABLE	65	SI	SI	SI
CLASE_MONOTRIBUTO	77	SI	SI	SI
CLASE_MONOTRIBUTO_SOCIAL	83			SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como “NO SOPORTADO”

Las categorías tabuladas corresponden a los emisores de los comprobantes que se emiten mediante la **IFH 1G**. Es decir, se trata de la responsabilidad frente a IVA grabada en la memoria fiscal de la **IFH 1G**. Este dato forma parte de la información posible de consultar a través del método *ObtenerDatosDeInicializacion()*.

Por ejemplo,

```
DIM reslVAemisor As Variant
HASAR1.ObtenerDatosDeInicializacion , , , , , reslVAemisor
```

```
If (reslVAemisor = CLASE_RESPONSABLE_INSCRIPTO) Then
    Debug.Print "El emisor es un Responsable Inscripto..."
End If
```

Tener presente que no todas las categorías están soportadas en todos los modelos de **IFH 1G**. Consultar el *Manual de Comandos* del equipo fiscal.

Enum: DocumentosFiscales (DF)

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
TICKET_C	84	SI	SI	SI
TICKET_FACTURA_A	65	SI	SI	SI
TICKET_FACTURA_B	66	SI	SI	SI
FACTURA_A	48	SI	SI	SI
FACTURA_B	49	SI	SI	SI
RECIBO_A	97	SI	SI	SI
RECIBO_B	98	SI	SI	SI
NOTA_DEBITO_A	68	SI	SI	SI
NOTA_DEBITO_B	69	SI	SI	SI
TICKET_NOTA_DEBITO_A	50		SI	SI
TICKET_NOTA_DEBITO_B	51		SI	SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como “NO SOPORTADO”

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar al método *AbrirComprobanteFiscal()*.

Por ejemplo,

```
HASAR1.AbrirComprobanteFiscal TICKET_FACTURA_A
```

Tener presente que no todos los comprobantes fiscales posibles de emitir están soportados en todos los modelos y versiones de firmware fiscal de las **IFH 1G**. Consultar el *Manual de comandos* del equipo fiscal.

Enum: DocumentosNoFiscales (DNFH)

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
NOTA_CREDITO_A	82	SI	SI	SI
NOTA_CREDITO_B	83	SI	SI	SI
REMITO	114	SI	SI	SI
ORDEN_SALIDA	115	SI	SI	SI
RESUMEN CUENTA	116	SI	SI	SI
CARGO_HABITACION	85	SI	SI	SI
COTIZACION	117	SI	SI	SI
RECIBO_X	120	SI	SI	SI
TICKET_NOTA_CREDITO_A	52			SI
TICKET_NOTA_CREDITO_B	53			SI
TICKET_RECIBO_X	54			SI
CLAUSULA_CREDITO	58			SI
CLAUSULA_SEGURO	59			SI
TICKET_PAGARE	55			SI
PAGARE	60			SI
POLIZA_SEGURO	61			SI
RECORDATORIO	62			SI
SOLICITUD_CREDITO	63			SI
COMUNICACION_CLIENTE	33			SI
OFRECIMIENTO_CREDITO	34			SI
OFRECIMIENTO_TARJETA	35			SI
MINUTA_CREDITO	36			SI
OFRECIMIENTO_PASAPORTE	37			SI
RENOVACION_CREDITO	38			SI
ADELANTO_REMUNERACION	39			SI
SOLICITUD_TARJETA_DEBITO	40			SI
SOLICITUD_CLAVE_TARJETA	41			SI
RESCATE_MERCADERIA	42			SI
INGRESO_EGRESO_SUCURSAL	43			SI
TICKET_PRESUPUESTO	44			SI
TICKET_ENTREGA	45			SI
TICKET_ESTACIONAMIENTO	46			SI
TICKET_COBRO_SERVICIO	47			SI
TICKET_INGRESO_DINERO	64			SI
TICKET_RETIRO_DINERO	65			SI
TICKET_TALON_CAMBIO	66			SI
TICKET_TALON_REPARTO	67			SI
TICKET_TALON_REGALO	68			SI
TICKET_TALON CUENTA CORRIENTE	69			SI
TICKET_AVISO_OPERACION_CREDITO	70			SI
TICKET_CUPON_PROMOCION	71			SI
TICKET_USO_INTERNO_FARMACIA	72			SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como "NO SOPORTADO"

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar al método *AbrirDNFH()*.

Por ejemplo,

HASAR1.AbrirDNFH TICKET_NOTA_CREDITO_A

Tener presente que no todos los comprobantes no fiscales homologados posibles de emitir están soportados en todos los modelos y versiones de firmware fiscal de las **IFH 1G**. Consultar el *Manual de Comandos* del equipo fiscal.

Enum: TiposDeDisplay

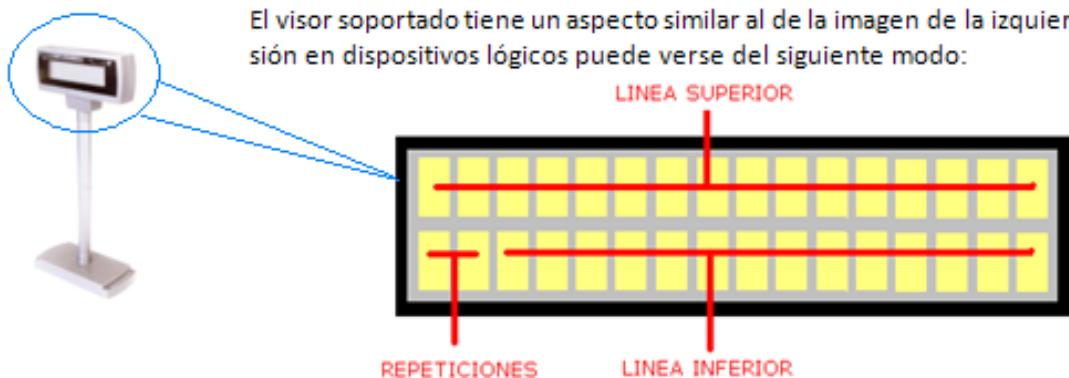
Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
LINEA_INFERIOR	75	SI	SI	SI
LINEA_SUPERIOR	76	SI	SI	SI
SECCION_DE_REPEATICION	78	SI	SI	SI

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar al método *MostrarEnDisplay()*.

Por ejemplo,

HASAR1.MostrarEnDisplay "Su compra...", LINEA_SUPERIOR

Hasta el momento, los únicos modelos de **IFH 1G** con soporte de conexión (opcional) -y discontinuados de producción- para visor de cliente, son SMH/P-614F y SMH/P-615F. En estos casos, el visor del cliente admite dos líneas de 16 caracteres cada una, donde se partitiona al dispositivo físico en tres dispositivos lógicos: superior, inferior y repeticiones.



Enum: TiposDePapel (rollo)

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
PAPEL_TICKET	0	SI	SI	SI

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
PAPEL_DIARIO	1	SI	SI	SI
PAPEL_TICKET_Y_DIARIO	2	SI	SI	SI

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar al método *AvanzarPapel()*.

Por ejemplo,

HASAR1.AvanzarPapel PAPEL_TICKET_Y_DIARIO, 10

PAPEL_TICKET	Permite avanzar, en la cantidad de líneas que se indique, el papel de la estación de impresión donde se emiten los comprobantes que se entregarán al cliente.
PAPEL_DIARIO	Permite avanzar, en la cantidad de líneas que se indique, el papel de la estación de impresión donde se emite la cinta testigo.
PAPEL_TICKET_Y_DIARIO	Permite avanzar, en la cantidad de líneas que se indique, el papel de la estación de impresión donde se emiten los comprobantes que se entregarán al cliente, y el papel de la estación de impresión donde se emite la cinta testigo. Ambos, simultáneamente.

Tener presente que no todos los modelos y versiones de firmware fiscal de las **IFH 1G** emiten el comprobante para el cliente y la cinta testigo en estaciones de impresión separadas. Consultar el *Manual de Comandos* del equipo fiscal.

Enum: TiposDeCodigoDeBarras

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
CODIGO TIPO_EAN_13	49	SI	SI	SI
CODIGO TIPO_EAN_8	50	SI	SI	SI
CODIGO TIPO_UPCA	51	SI	SI	SI
CODIGO TIPO_ITS	52	SI	SI	SI
CODIGO DE_BORRADO	48		SI	SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como "NO SOPORTADO"

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar al método *ImprimirCodigoDeBarras()*.

Por ejemplo,

HASAR1.ImprimirCodigoDeBarras CODIGO_TIPO_EAN_8, "7791234", True, True

CODIGO_TIPO_EAN_13	Identifica al código de barras EAN 13. La IFH 1G espera 12 dígitos (no indicar dígito verificador).
--------------------	--

CODIGO_TIPO_EAN_8	Identifica al código de barras EAN 8. La IFH 1G espera 7 dígitos (no indicar dígito verificador).
CODIGO_TIPO_UPCA	Identifica al código de barras UPCA. La IFH 1G espera 11 dígitos (no indicar dígito verificador).
CODIGO_TIPO_ITS	Identifica al código de barras ITS 2 de 5. La IFH 1G espera una cantidad par de dígitos (no indicar dígito verificador). La cantidad de pares de dígitos depende del modelo y versión de firmware fiscal de la IFH 1G , en función del ancho de papel donde imprimir.
CODIGO_DE_BORRADO	Le indica a la IFH 1G que elimine de su memoria de trabajo el código de barras almacenado para su posterior impresión.

Tener presente que no todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la eliminación del código de barras almacenado en la memoria de trabajo del equipo fiscal. Consultar el *Manual de Comandos* del producto.

Enum: TiposDeVouchers

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
VOUCHER_DE_COMPRA	67	SI	SI	SI
VOUCHER_DE_CANCELACION_COMPRA	86	SI	SI	SI
VOUCHER_DE_DEVOLUCION	68	SI	SI	SI
VOUCHER_DE_CANCELACION_DE_DEVOLUCION	65	SI	SI	SI

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar al método *ImprimirVoucher()*. Estas constantes definen el título del voucher, el cual está asociado al tipo de transacción realizada.

Por ejemplo,

```
HASAR1.ImprimirVoucher "Cliente...", "PlastiCard", VOUCHER_DE_COMPRA, "1234567890123456", _  
"0215", TARJETA_CREDITO, "3", "1234", "567", "8901", "12345678", _  
INGRESO_TARJETA_AUTOMATIZADO, OPERACION_TARJETA_ONLINE, _  
"654321", "236.58", "00000678", 2
```

Enum: TiposDeTarjetas

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
TARJETA_CREDITO	67	SI	SI	SI
TARJETA_DEBITO	68	SI	SI	SI

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar al método *ImprimirVoucher()*. Estas constantes definen el tipo de plástico: de débito, o de crédito.

Por ejemplo,

```
HASAR1.ImprimirVoucher "Cliente...", "PlastiCard", VOUCHER_DE_COMPRA, "1234567890123456", _  
"0215", TARJETA_CREDITO, "3", "1234", "567", "8901", "12345678", _  
INGRESO_TARJETA_AUTOMATIZADO, OPERACION_TARJETA_ONLINE, _  
"654321", "236.58", "00000678", 2
```

Enum: TiposDeIngresoDeTarjeta

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
INGRESO_DE_TARJETA_MANUAL	42	SI	SI	SI
INGRESO_DE_TARJETA_AUTOMATIZADO	32	SI	SI	SI

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar al método *ImprimirVoucher()*. Estas constantes definen como se ingresó al sistema, para realizar la transacción, del número de la tarjeta de crédito/débito.

Por ejemplo,

```
HASAR1.ImprimirVoucher "Cliente...", "PlastiCard", VOUCHER_DE_COMPRA, "1234567890123456", _  
"0215", TARJETA_CREDITO, "3", "1234", "567", "8901", "12345678", _  
INGRESO_TARJETA_AUTOMATIZADO, OPERACION_TARJETA_ONLINE, _  
"654321", "236.58", "00000678", 2
```

Enum: TiposDeOperacionDeTarjeta

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
OPERACIÓN_TARJETA_ONLINE	78	SI	SI	SI
OPERACIÓN_TARJETA_OFFLINE	70	SI	SI	SI

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar al método *ImprimirVoucher()*. Estas constantes definen como se realizó la transacción con la tarjeta de crédito/débito.

Por ejemplo,

```
HASAR1.ImprimirVoucher "Cliente...", "PlastiCard", VOUCHER_DE_COMPRA, "1234567890123456", _  
"0215", TARJETA_CREDITO, "3", "1234", "567", "8901", "12345678", _  
INGRESO_TARJETA_AUTOMATIZADO, OPERACION_TARJETA_ONLINE, _  
"654321", "236.58", "00000678", 2
```

Enum: TiposDeDescuentos

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
DESCUENTO_RECARGO	66	SI	SI	SI
DEVOLUCION_DE_ENVASES	69	SI	SI	SI

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar al método *DevolucionDescuento()*.

Por ejemplo,

HASAR1.DevolucionDescuento "Oferta del día...", 25.00, 10.50, 0.00, True, DESCUENTO_RECARGO

Tener presente que algunos modelos y versiones de firmware fiscal de las **IFH 1G** -dis continuados de producción- no soportan comandos de aplicación de descuentos/recargos. Consultar el *Manual de comandos* del equipo fiscal.

Enum: ParametrosDeConfiguracion

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
IMPRESION_CAMBIO	52	SI	SI	SI
IMPRESIONLEYENDAS	53	SI	SI	SI
CORTE_PAPEL	54	SI	SI	SI
IMPRESION_MARCO	55	SI	SI	SI
REIMPRESION_CANCELADOS	56	SI	SI	SI
COPIAS_DOCUMENTOS	57	SI	SI	SI
PAGO_SALDO	58	SI	SI	SI
SONIDO	59	SI	SI	SI
SONIDO_AVISO	59		SI	SI
ALTO_HOJA	60		SI	SI
ANCHO_HOJA	61		SI	SI
ESTACION_REPORTES_XZ	62		SI	SI
MODO_IMPRESION	63		SI	SI
CHEQUEO_DESBORDE	64		SI	SI
CHEQUEO_TAPA_ABIERTA	65		SI	SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como "NO SOPORTADO"

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar al método *ConfigurarControlador()*.

Por ejemplo,

HASAR1.ConfigurarControlador COPIAS_DOCUMENTOS, "2"

IMPRESION_CAMBIO	Establece si se imprime la leyenda “CAMBIO \$ 0.00” cuando la suma de los pagos realizados coincide con el total del comprobante.
IMPRESIONLEYENDAS	Establece si las leyendas optionales en Tiques (número de inscripción ingresos brutos, “A consumidor Final”, y Categoría IVA emisor) se imprimen. Suelen omitirse en rollos preimpresos al dorso. Dicha preimpresión (al dorso) debe incluir los datos omitidos.
CORTE_PAPEL	En los casos donde sea admitido, establece el tipo de corte de papel (parcial, total, no cortar).
IMPRESION_MARCO	En comprobantes a emitir sobre formularios continuos u hojas sueltas, establece si se imprime el marco del comprobante (si no se utiliza papel preimpreso).
REIMPRESION_CANCELADOS	Establece si los comprobantes cancelados por corte de corriente son reimpressos automáticamente al retornar la energía. La reimpresión del comprobante es hasta el punto alcanzado al momento del corte de energía.
COPIAS_DOCUMENTOS	Establece la máxima cantidad de ejemplares que se podrán emitir.
PAGO_SALDO	Establece el texto a imprimir –de ser necesario-, y asociado a la diferencia existente entre la suma de los pagos y el total del comprobante, cuando se cierra el comprobante sin haber pagado en forma completa.
SONIDO	Establece la emisión de pitidos ante falta de papel.
SONIDO_AVISO	Ver SONIDO.
ALTO_HOJA	Establece el alto del papel en IFH 1G que soportan más de un tamaño de papel.
ANCHO_HOJA	Establece el ancho del papel en IFH 1G que soportan más de un tamaño de papel.
ESTACION_REPORTES_XZ	Establece la estación de impresión (rollo o formulario continuo) de reportes ‘X’/‘Z’, en IFH 1G que admitan esta posibilidad.
MODO_IMPRESION	Establece si se puede alternar la emisión de comprobantes entre rollo y formulario continuo, o se trabaja exclusivamente sobre formularios continuos. Válido para IFH 1G que admitan esta posibilidad.
CHEQUEO_DESBORDE	Establece la forma de controlar los montos límites, establecidos por AFIP para los comprobantes. Se puede controlar en todo momento, o sólo al cerrar el comprobante.
CHEQUEO_TAPA_ABIERTA	Establece si se puede seguir imprimiendo aunque la tapa de la IFH 1G se encuentre abierta.

Tener presente que no todas las opciones de configuración están disponibles en todos los modelos y versiones de firmware fiscal de las **IFH 1G**. Consultar el *Manual de Comandos* del equipo fiscal.

Enum: NumeroDeCopias

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
NO_COPIAS	48	SI	SI	SI
ORIGINAL	49	SI	SI	SI
DUPLICADO	50	SI	SI	SI
TRIPPLICADO	51	SI	SI	SI
CUADRUPPLICADO	52	SI	SI	SI

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar a los métodos *ConfigurarControlador()*, *ConfigurarControladorCompleto()*, y *ConfigurarControladorPorBloque()*.

Por ejemplo,

HASAR1.ConfigurarControlador COPIAS_DOCUMENTOS, "2"

HASAR1.ConfigurarControladorCompleto True, True,, DUPLICADO

HASAR1.ConfigurarControladorPorBloque 1000.00, 25000.00,0.00, DUPLICADO, False, False, CORTE_PARCIAL

Tener presente que la cantidad de ejemplares que se configure en la **IFH 1G**, no implica que serán emitidos automáticamente. Se emiten automáticamente los exigidos por la normativa fiscal vigente. Los que falten emitir deberán ser solicitados por el software de gestión para puntos de venta mediante el método *ReimprimirComprobante()*.

El valor NO_COPIAS debe establecerse cuando se utiliza papel copiativo. En cualquiera de las demás configuraciones posibles, la **IFH 1G** emite automáticamente:

- Original y duplicado: De tratarse de comprobantes fiscales y/o notas de crédito.
- Original: En cualquier otro caso.

Tener presente que no todos los modelos y versiones de firmware fiscal de las **IFH 1G** admiten la emisión de ejemplares adicionales a los impresos automáticamente. Consultar el *Manual de comandos* del equipo fiscal.

Enum: TipoDeCorteDePapel

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
CORTE_TOTAL	70	SI	SI	SI
CORTE_PARCIAL	80	SI	SI	SI
NO_CORTE	78	SI	SI	SI

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar a los métodos *ConfigurarControlador()*, *ConfigurarControladorCompleto()*, y *ConfigurarControladorPorBloque()*.

Por ejemplo,

HASAR1.ConfigurarControlador CORTE_PAPEL, CORTE_PARCIAL

HASAR1.ConfigurarControladorCompleto True, True,,,CORTE_PARCIAL

HASAR1.ConfigurarControladorPorBloque 1000.00, 25000.00,0.00, DUPLICADO, False, False, CORTE_PARCIAL

Tener presente que no todos los modelos y versiones de firmware fiscal de las **IFH 1G** admiten configuración de corte de papel. Consultar el *Manual de Comandos* del equipo fiscal.

Enum: Documentos

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
D_FACTURA_A	1		SI	SI
D_FACTURA_B	2		SI	SI
D_FACTURA_C	3		SI	SI
D_NOTA_DEBITO_A	4		SI	SI
D_NOTA_DEBITO_B	5		SI	SI
D_NOTA_DEBITO_C	6		SI	SI
D_RECIBO_A	7		SI	SI
D_RECIBO_B	8		SI	SI
D_RECIBO_C	9		SI	SI
D_TICKET_BC	10		SI	SI
D_DOC_NO_FISCAL	32		SI	SI
D_NOTA_CREDITO_A	64		SI	SI
D_NOTA_CREDITO_B	65		SI	SI
D_NOTA_CREDITO_C	66		SI	SI
D_REMITO	67		SI	SI
D_RECIBO_X	68		SI	SI
D_ORDEN_SALIDA	69		SI	SI
D_RESUMEN CUENTA	70		SI	SI
D_CARGO_HABITACION	71		SI	SI
D_COTIZACION	72		SI	SI
D_VOUCHER_TARJETA	73		SI	SI
D_CLAUSULA_CREDITO	80		SI	SI
D_CLAUSULA_SEGURO	81		SI	SI
D_PAGARE	82		SI	SI
D_POLIZA_SEGURO	83		SI	SI
D_RECORDATORIO	84		SI	SI
D_SOLICITUD_CREDITO	85		SI	SI
D_COMUNICACION_CLIENTE	86		SI	SI
D_OFRECIMIENTO_CREDITO	87		SI	SI
D_OFRECIMIENTO_TARJETA	88		SI	SI
D_MINUTA_CREDITO	89		SI	SI
D_OFRECIMIENTO_PASAPORTE	90		SI	SI
D_RENOVACION_CREDITO	91		SI	SI
D_ADELANTO_REMUNERACION	92		SI	SI
D_SOLICITUD_TARJETA_DEBITO	93		SI	SI

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
D_SOLICITUD_CLAVE_TARJETA	94		SI	SI
D_RESCATE_MERCADERIA	95		SI	SI
D_INGRESO_EGRESO_SUCURSAL	96		SI	SI
D_PRESUPUESTO	97		SI	SI
D_ENTREGA	98		SI	SI
D_ESTACIONAMIENTO	99		SI	SI
D_COBRO_SERVICIO	100		SI	SI
D_INGRESO_DINERO	101		SI	SI
D_RETIRO_DINERO	102		SI	SI
D_TALON_CAMBIO	103		SI	SI
D_TALON_REPARTO	104		SI	SI
D_TALON_REGALO	105		SI	SI
D_TALON CUENTA CORRIENTE	106		SI	SI
D_AVISO_OPERACION_CREDITO	107		SI	SI
D_CUPON_PROMOCION	108		SI	SI
D_USO_INTERNO_FARMACIA	109		SI	SI
D_NO_DOCUMENTO_EN_CURSO	0		SI	SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como "NO SOPORTADO"

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al analizar la información contenida en el campo *Status Documento*, que forma parte del string de respuesta al comando *StatusRequest*. Este comando se genera con el método *PedidoDeStatus()*.

Por ejemplo,

```

DIM status_doc As String
DIM doc_actual As String

HASAR1.AbrirComprobanteFiscal FACTURA_B
HASAR1.PedidoDeStatus
status_doc = HASAR1.Respuesta(6)
doc_actual = Mid$(status_doc, 1,2)
If (Val(doc_actual) = D_FACTURA_B) Then
    .....
End If

```

Tener presente que no todos los documentos listados se encuentran disponibles en todos los modelos y versiones de firmware fiscal de las **IFH 1G**. Consultar el *Manual de Comandos* del equipo fiscal.

Enum: TiposDeAltoHoja

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
ALTO_REDUCIDO	77		SI	SI
ALTO_A4	65		SI	SI

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
ALTO_OFICIO	79		SI	SI
ALTO_CARTA	67		SI	SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como “NO SOPORTADO”

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar a los métodos *ConfigurarControlador()*, y *ConfigurarControladorCompleto()*.

Por ejemplo,

HASAR1.ConfigurarControlador ALTO_HOJA, ALTO_A4
HASAR1.ConfigurarControladorCompleto True, True,,,,,,,,,, ALTO_A4

Tener presente que no todos los modelos y versiones de firmware fiscal de las **IFH 1G** admiten distintos tamaños de papel. Consultar el *Manual de Comandos* del equipo fiscal.

Enum: TiposDeAnchoHoja

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
ANCHO_REDUCIDO	77		SI	SI
ANCHO_NORMAL	78		SI	SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como “NO SOPORTADO”

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar a los métodos *ConfigurarControlador()*, y *ConfigurarControladorCompleto()*.

Por ejemplo,

HASAR1.ConfigurarControlador ANCHO_HOJA, ANCHO_NORMAL
HASAR1.ConfigurarControladorCompleto True, True,,,,,,,,,, ANCHO_NORMAL

Tener presente que no todos los modelos de **IFH 1G** admiten distintos tamaños de papel. Consultar el *Manual de Comandos* del equipo fiscal.

Enum: TiposDeEstacion

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
ESTACION_TICKET	84		SI	SI
ESTACION_SLIP	83		SI	SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como “NO SOPORTADO”

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar a los métodos *ConfigurarControlador()*, y *ConfigurarControladorCompleto()*. Permiten identificar la estación de impresión (rollo o formulario continuo) de cierres diarios ‘X’/’Z’.

Por ejemplo,

```
HASAR1.ConfigurarControlador ESTACION_REPORTES_XZ, ESTACION_TICKET
HASAR1.ConfigurarControladorCompleto True, True,,,,,,,,,, ESTACION_TICKET
```

Tener presente que no todos los modelos y versiones de firmware fiscal de las **IFH 1G** admiten la impresión de comprobantes, de manera alternativa, sobre rollo o formularios continuos. Consultar el *Manual de Comandos* del equipo fiscal.

Enum: TiposDeModoImpresión

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
USO_ESTACION_TICKET	77		SI	SI
NO_USO_ESTACION_TICKET	65		SI	SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como “NO SOPORTADO”

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar a los métodos *ConfigurarControlador()*, y *ConfigurarControladorCompleto()*. Permiten establecer la estación de impresión (rollo o formulario continuo, de forma excluyente) para la emisión de cualquier tipo de comprobante.

Por ejemplo,

```
HASAR1.ConfigurarControlador MODO_IMPRESION, USO_ESTACION_TICKET
HASAR1.ConfigurarControladorCompleto True, True,,,,,,,,,, USO_ESTACION_TICKET
```

Tener presente que no todos los modelos y versiones de firmware fiscal de las **IFH 1G** ofrecen la posibilidad de imprimir sobre rollo y/o formularios continuos en forma alternativa. Consultar el *Manual de Comandos* del equipo fiscal.

Enum: CapacidadesSoportadas

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
CAPACIDAD_DOCUMENTO_SLIP	0		SI	SI
CAPACIDAD_DOCUMENTO_TICKET	1		SI	SI

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
CAPACIDAD_ESTACION	2		SI	SI
CAPACIDAD_CAJON	3		SI	SI
CAPACIDAD_TEXTO_FISCAL	4		SI	SI
CAPACIDAD_TEXTO_VENTA	5		SI	SI
CAPACIDAD_TEXTO_NO_FISCAL	6		SI	SI
CAPACIDAD_TEXTO_ENCABEZADO_COLA	7		SI	SI
CAPACIDAD_DF_SOPORTADO	8			SI
CAPACIDAD_DNFH_SOPORTADO	9			SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como "NO SOPORTADO"

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar al método *Capacidades()*.

Por ejemplo,

```
DIM hay_factura As Variant
...
HASAR1.Capacidades CAPACIDAD_DOCUMENTO_SLIP, SOPORTA_FACTURA, ,hay_factura
If (hay_factura) Then
    .....
End If
```

Tener presente que se está consultando si la **IFH 1G** posee determinada característica. Productos diferentes no tienen porque poseer las mismas características. Consultar el *Manual de Comandos* del equipo fiscal.

Enum: TiposDeDocumentoSoportado

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
SOPORTA_FACTURA	0		SI	SI
SOPORTA_NOTA_CREDITO	1		SI	SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como "NO SOPORTADO"

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar al método *Capacidades()*.

Por ejemplo,

```
DIM hay_factura As Variant
...
HASAR1.Capacidades CAPACIDAD_DOCUMENTO_SLIP, SOPORTA_FACTURA, ,hay_factura
```

```

If (hay_factura) Then
    ....
End If

```

Tener presente que se está consultando si la **IFH 1G** posee determinada característica. Productos diferentes no tienen porque poseer las mismas características. Consultar el *Manual de Comandos* del equipo fiscal.

Enum: TiposDeResumenIVA

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
RESUMEN_IVA_DIARIO	0		SI	SI
RESUMEN_IVA_DOCUMENTO	1		SI	SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como "NO SOPORTADO"

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar a la propiedad *ResumenIVA()*.

Por ejemplo,

```

HASAR1.CerrarComprobanteFiscal
HASAR1.ResumenIVA = RESUMEN_IVA_DOCUMENTO

Dim Alic As Double, MIVA As Double, Implt As Double, IVANoI As Double, Basel As Double
Dim msg As String

While (HASAR1.LeerIVA(DOCUMENTOS_FISCALES, Alic, MIVA, Implt, IVANoI, Basel))
    Debug.Print "Alícuota IVA" & "=" & Alic & vbCrLf & _
        "Monto IVA" & "=" & MIVA & vbCrLf & _
        "Monto Imp. Int." & "=" & Implt & vbCrLf & _
        "Monto Neto" & "=" & Basel
Wend

```

RESUMEN_IVA_DIARIO	Le indica al OCX Fiscal HASAR que a continuación se solicitará el desglose de impuestos correspondientes a los acumulados generados durante la jornada fiscal.
RESUMEN_IVA_DOCUMENTO	Le indica al OCX Fiscal HASAR que a continuación se solicitará el desglose de impuestos correspondientes al comprobante que se ha cerrado inmediatamente antes, o todavía está en curso.

Tener presente que no todos los modelos y versiones de firmware fiscal de las **IFH 1G** soportan la consulta de desglose de impuestos correspondiente a los acumulados de la jornada fiscal, o cuando hay un comprobante abierto. Consultar el *Manual de Comandos* del equipo fiscal.

Enum: DocumentosIVA

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Valor	v010724	v051122	v110423 v150423
DOCUMENTOS_FISCALES	0		SI	SI
NOTAS_DE_CREDITO	1		SI	SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como "NO SOPORTADO"

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar a los métodos *LeerIVA()*, *LeerPercepcionesPorIVA()*, *LeerPercepcionesGenerales()*, y *LeerMontoTotal()*.

Por ejemplo,

```
HASAR1.CerrarComprobanteFiscal  
HASAR1.ResumenIVA = RESUMEN_IVA_DOCUMENTO
```

```
Dim Alic As Double, MIVA As Double, Implt As Double, IVANoI As Double, Basel As Double  
Dim msg As String
```

```
While (HASAR1.LeerIVA(DOCUMENTOS_FISCALES, Alic, MIVA, Implt, IVANoI, Basel))  
    Debug.Print "Alícuota IVA" & "=" & Alic & vbCrLf & _  
    "Monto IVA" & "=" & MIVA & vbCrLf & _  
    "Monto Imp. Int." & "=" & Implt & vbCrLf & _  
    "Monto Neto" & "=" & Basel  
Wend
```

Tener presente que no todos los modelos y versiones de firmware fiscal de las **IFH 1G** emiten Notas de Crédito. Consultar el *Manual de Comandos* del equipo fiscal.

Enum: TiposDeTransporte

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
PUERTO_SERIE	0		SI	SI
SOCKET_TCP	1		SI	SI
SOCKET_UDP	2		SI	SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como "NO SOPORTADO"

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al invocar a la propiedad *Transporte()*.

Por ejemplo,

HASAR1.Transporte = PUERTO_SERIE

PUERTO_SERIE	La IFH 1G se encuentra conectada al puerto serie del host.
SOCKET_TCP	La IFH 1G se encuentra conectada al puerto serie de un conversor Ethernet/RS-232. Protocolo TCP/IP.
SOCKET_UDP	La IFH 1G se encuentra conectado al puerto serie de un conversor Ethernet/RS-232. Protocolo UDP/IP.

Cualquier **IFH 1G** puede conectarse a un conversor Ethernet/RS-232. No todas las versiones del **OCX Fiscal HASAR** manejan protocolo IP.

Estados de la Impresora Fiscal HASAR (IFH 1G)

Enum StateNibble

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
S_NONFORMATED_MEMORY	0	SI	SI	SI
S_NONINITIALIZED_MEMORY	1	SI	SI	SI
S_RECEIPT_NON_OPENED	2	SI	SI	SI
S_FISCAL_RECEIPT_OPENED	3	SI	SI	SI
S_FISCAL_TEXT_ISSUED	4	SI	SI	SI
S_NONFISCAL_RECEIPT_OPENED	5	SI	SI	SI
S_TENDER	6	SI	SI	SI
S_TENDER_CLEARED	7	SI	SI	SI
S_PERCEPTION	8	SI	SI	SI
S_KILLED	9	SI	SI	SI
S_RETURN_RECHARGE	10	SI	SI	SI
S_DISCOUNT_CHARGE	11	SI	SI	SI
S_RECEIPT_CONCEPT	12	SI	SI	SI
S_CREDIT_NOTE	13	SI	SI	SI
S_CREDIT_NOTE_DISCOUNT	14	SI	SI	SI
S_CREDIT_NOTE_RETURN	15	SI	SI	SI
S_CREDIT_NOTE_PERCEPTION	16	SI	SI	SI
S_CREDIT_NOTE_TEXT	17	SI	SI	SI
S_INTERNAL_USE_RECEIPT	18	SI	SI	SI
S_QUOTATION	19	SI	SI	SI
S_EMBARK	20	SI	SI	SI
S_ACCOUNT	21	SI	SI	SI
S_BLOCKED	22	SI	SI	SI
S_DNFH_RECEIPT_OPENED	23		SI	SI
S_CREDIT_NOTE_TENDER	24		SI	SI
S_NONE	255	SI	SI	SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como "NO SOPORTADO"

Las constantes embebidas en el **OCX Fiscal HASAR**, y definidas en la tabla anterior, son las que deben emplearse al analizar la información contenida en el campo *Status Auxiliar*, que forma parte del string de respuesta al comando *StatusRequest*. Este comando se genera con el método *PedidoDeStatus()*.

Por ejemplo,

```
DIM status_aux As String
DIM nibble_alto As String
DIM parte_baja As String
Const DATOS_CLIENTE = 2
```

```
HASAR1.AbrirComprobanteFiscal FACTURA_B
HASAR1.PedidoDeStatus
```

```

status_doc = HASAR1.Respuesta(4)
nibble_alto = Mid$(status_aux, 1,1)
parte_baja = Mid$(status_aux, 3,2)

If (Val(nibble_alto) = DATOS_CLIENTE) Then
    Debug.Print "Datos de cliente ingresados"
End IF

Select Case Val(partе_baja)
    Case S_FISCAL_RECEIPT_OPENED
        Debug.Print "Comprobante Fiscal Abierto. Venta Habilitada"
End Select

```

La sentencia *Select Case/End Select* podría ser reemplazada por la siguiente sentencia:

```
Debug.Print HASAR1.DescripcionEstadoControlador
```

Conocer los estados de la **IFH 1G** es de importancia para el software de gestión para puntos de venta. Éste no debe perder nunca el sincronismo con la **IFH 1G**. Se trate de operaciones realizadas con normalidad, o en tareas de recuperación de errores.

Diferencias en la numeración de comprobantes, en los montos acumulados, operaciones que no se pueden realizar, etc., son consecuencia de la pérdida de sincronismo (la **IFH 1G** se encuentra en determinado estado, y el software de gestión para puntos de venta en otro).

S_NONFORMATED_MEMORY	Memoria fiscal no formateada.
S_NONINITIALIZED_MEMORY	Memoria fiscal no inicializada.
S_RECEIPT_NON_OPENED	No hay ningún comprobante abierto.
S_FISCAL_RECEIPT_OPENED	Un comprobante fiscal se encuentra abierto. Venta habilitada.
S_FISCAL_TEXT_ISSUED	Comprobante fiscal abierto. Se acaba de imprimir un texto fiscal.
S_NONFISCAL_RECEIPT_OPENED	Un comprobante no fiscal se encuentra abierto.
S_TENDER	Comprobante fiscal abierto. Se realizó al menos un pago.
S_TENDER_CLEARED	Comprobante fiscal abierto. Se saldó el monto.
S_PERCEPTION	Comprobante fiscal abierto. Se registró una percepción.
S_KILLED	La memoria fiscal ha sido dada de baja.
S_RETURN_RECHARGE	Comprobante fiscal abierto. Se aplicó una bonificación/devolución de envases.
S_DISCOUNT_CHARGE	Comprobante fiscal abierto. Se aplicó un descuento/recargo general.
S_RECEIPT_CONCEPT	Recibo fiscal abierto. Se imprimió una línea de detalle.
S_CREDIT_NOTE	Nota de crédito/recibo 'X' abierto. Venta habilitada.
S_CREDIT_NOTE_DISCOUNT	Nota de crédito/recibo 'X' abierto. Se aplicó un descuento/recargo general.
S_CREDIT_NOTE_RETURN	Nota de crédito/recibo 'X' abierto. Se aplicó una bonificación/ devolución de envases.
S_CREDIT_NOTE_PERCEPTION	Nota de crédito/recibo 'X' abierto. Se registró una percepción.
S_CREDIT_NOTE_TEXT	Nota de crédito/recibo 'X' abierto. Se acaba de imprimir un texto fiscal.
S_INTERNAL_USE_RECEIPT	Recibo 'X'. Se imprimió una línea de detalle.
S_QUOTATION	Una cotización se encuentra abierta.
S_EMBARK	Un remito/orden de salida se encuentra abierto.
S_ACCOUNT	Un resumen de cuenta/cargo a la habitación se encuentra abierto.

S_BLOCKED	IFH 1G bloqueada.
S_DNFH_RECEIPT_OPENED	Comprobante no fiscal homologado abierto.
S_CREDIT_NOTE_TENDER	Nota de crédito abierta. Se realizó al menos un pago.
S_NONE	Estado fiscal no conocido.

Tener presente que no todos los estados listados se encuentran disponibles en todos los modelos y versiones de firmware fiscal de las **IFH 1G**. Consultar el *Manual de Comandos* del producto.

Eventos del OCX Fiscal HASAR

Eventos posibles de ser generados

OCX Fiscal HASAR			
Evento	v010724	v051122	v110423 v150423
<i>ErrorFiscal()</i>	SI	SI	SI
<i>ErrorImpresora()</i>	SI	SI	SI
<i>EventoCajon()</i>	SI	SI	SI
<i>EventoFiscal</i>	SI	SI	SI
<i>EventoImpresora()</i>	SI	SI	SI
<i>FaltaPapel()</i>	SI	SI	SI
<i>ImpresoraNoResponde()</i>	SI	SI	SI
<i>ImpresoraOcupada()</i>	SI	SI	SI
<i>ImpresoraOK()</i>	SI	SI	SI
<i>ProgresoDeteccion()</i>	SI	SI	SI

Los eventos que tiene capacidad de generar el **OCX Fiscal HASAR** son los listados en la tabla anterior. Estos eventos se originan a partir de strings de respuesta enviados por la **IFH 1G** al host. Estos strings pueden ser la respuesta propiamente dicha a un comando enviado, o responder a estados de la **IFH 1G** que merecen originar una advertencia al software de gestión para puntos de venta.

Es importante que el software de gestión para puntos de venta atrape y trate adecuadamente cada uno de estos eventos. No hacerlo implica el riesgo de perder el sincronismo con la **IFH 1G** respecto de montos acumulados y numeración alcanzada, así como de generar incidentes evitables.

En el tratamiento del evento debe considerarse el contexto en el cual éste se produce. Por ejemplo, antes de iniciar la impresión de un nuevo comprobante el software de gestión para puntos de venta puede enviar el comando de cancelación general a la **IFH 1G** previendo la posibilidad de encontrar un comprobante abierto (no debería suceder, pero...). Al no haber un comprobante abierto, la **IFH 1G** rechaza el comando de cancelación por inválido para su estado actual (no hay comprobante abierto). Al recibir la respuesta que informa el rechazo, el **OCX Fiscal HASAR** genera el evento *ErrorFiscal()* reportando F_INVALID_COMMAND. Esta incidencia debe interpretarse, en lugar de error, como estar en condiciones de abrir un nuevo comprobante (actualmente no hay ninguno abierto).

Una forma más elegante de resolver la misma situación es como ya se ha visto en capítulo dedicado a *Estados de la IFH 1G*: analizando la respuesta obtenida al invocar al método *PedidoDeStatus()*.

Como atrapar eventos

Cada lenguaje de programación ofrece su propio mecanismo para atrapar eventos. Los desarrolladores de software de gestión para puntos de venta deberán consultar la documentación correspondiente a su entorno de desarrollo para interiorizarse acerca de como atrapar un evento.

Se muestran a continuación dos casos, y a modo de ejemplo.

Caso: Visual Basic 6.0

Se debe crear una rutina privada que lleve por nombre el nombre del objeto (HASAR1), seguido de guión bajo, y del nombre del evento que se quiere atrapar. Cuando el **OCX Fiscal HASAR** genere el evento correspondiente (*ErrorFiscal()*), el flujo del programa será derivado a esta rutina. En el cuerpo de la misma se debe hacer el tratamiento que corresponda para el incidente reportado.

```
Private Sub HASAR1_ErrorFiscal(ByVal Flags As Long)
    Dim msg As String

    Select Case Flags
        Case F_INVALID_COMMAND
            msg = "OCX Fiscal HASAR :: Comando Inválido para el estado fiscal actual"
        Else
            msg = "OCX Fiscal HASAR :: Comando rechazado - Evento: ErrorFiscal( )"
    End Select

    Debug.Print msg
End Sub
```

Caso: Visual C#

Como puede verse en este caso, el mecanismo es muy similar al empleado en Visual Basic 6.0. La diferencia radica en que se requiere una declaración previa del evento a atrapar. En el caso de Visual Studio 2010, al dar [Enter], finalizada la sentencia de declaración, el propio entorno de desarrollo inserta la rutina en el código, con el cuerpo vacío.

```
// Creación del objeto
FiscalPrinterLib.HASAR ofh = new FiscalPrinterLib.HASAR();

// Declaración del evento a atrapar
ofh.ErrorFiscal += new FiscalPrinterLib._FiscalEvents_ErrorFiscalEventHandler(ofh_ErrorFiscal);

// Rutina de tratamiento del evento
void ofh_ErrorFiscal(int Flags)
{
    string msj = "";

    switch (Flags)
    {
        case (int)FiscalPrinterLib.FiscalBits.F_INVALID_COMMAND:
            msj = "OCX Fiscal HASAR :: Comando Inválido para el estado fiscal actual";
            break;
        default:
            msj = "OCX Fiscal HASAR :: Comando rechazado - Evento: ErrorFiscal( )";
            break;
    }

    MessageBox.Show(msj);
}
```

Los eventos del OCX Fiscal HASAR

ErrorFiscal()

Este evento es generado por el **OCX Fiscal HASAR** cuando determinado bit del campo de *Status Fiscal* del string de respuesta a un comando enviado se encuentra en '1' (ver *Manual de Comandos de la IFH 1G*). De acuerdo al bit levantado, el incidente reportado se puede identificar mediante alguna de las constantes embebidas del **OCX Fiscal HASAR** que se listan en la tabla siguiente:

Constantes Embebidas	Nombre	Dec	OCX Fiscal HASAR		
			v010724	v051122	v110423 v150423
F_FISCAL_MEMORY_FAIL		1	SI	SI	SI
F_FISCAL_MEMORY_FULL		128	SI	SI	SI
F_INVALID_COMMAND		32	SI	SI	SI
F_INVALID_FIELD_DATA		16	SI	SI	SI
F_TOTAL_OVERFLOW		64	SI	SI	SI
F_UNRECOGNIZED_COMMAND		8	SI	SI	SI
F_WORKING_MEMORY_FAIL		2	SI	SI	SI

F_FISCAL_MEMORY_FAIL	Ha ocurrido un error de chequeo de la memoria fiscal de la IFH 1G .
F_FISCAL_MEMORY_FULL	La memoria fiscal de la IFH 1G se ha llenado. No ha más capacidad para nuevos cierres diarios 'Z'. No se puede abrir un nuevo comprobante. También puede ocurrir que la memoria fiscal haya sido dada de baja, o se encuentre bloqueada.
F_INVALID_COMMAND	Comando inválido para el estado fiscal actual. Se ha rechazado el comando enviado.
F_INVALID_FIELD_DATA	Campo de datos inválido. Se ha rechazado el comando enviado.
F_TOTAL_OVERFLOW	Desborde de total. Intento de superar el monto límite para el comprobante en curso. Se rechaza el comando enviado. Los montos límites son establecidos mediante la normativa fiscal vigente.
F_UNRECOGNIZED_COMMAND	Comando desconocido. Se ha enviado un comando que no forma parte del set soportado por la IFH 1G . Se rechaza el comando enviado.
F_WORKING_MEMORY_FAIL	Ha ocurrido un error de chequeo de la memoria de trabajo de la IFH 1G .

ErrorImpresora()

Si bien este evento es miembro del **OCX Fiscal HASAR**, actualmente no existe causa alguna para su generación. Se reserva para aplicaciones futuras.

EventoCajon()

En las **IFH 1G** que soportan conexión de cajón de dinero, este evento es generado por el **OCX Fiscal HASAR** cada vez que dicho periférico cambia de estado (pasa de cerrado a abierto, o a la inversa).

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
P_DRAWER_CLOSED	16384	SI	SI	SI

P_DRAWER_CLOSED | El cajón de dinero se encuentra cerrado (o ausente).

EventoFiscal()

Este evento es generado por el **OCX Fiscal HASAR** cuando determinado bit del campo de *Status Fiscal* del string de respuesta a un comando enviado se encuentra en '1' (ver *Manual de Comandos de la IFH 1G*). De acuerdo al bit levantado, el incidente reportado se puede identificar mediante alguna de las constantes embebidas del **OCX Fiscal HASAR** que se listan en la tabla siguiente:

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
F_FISCAL_MEMORY_NEAR_FULL	256	SI	SI	SI
F_FISCAL_TERMINAL_CERTIFIED	512	SI	SI	SI
F_FISCAL_TERMINAL_FISCALIZED	1024	SI	SI	SI
F_RECEIPT_OPEN	8192	SI	SI	SI
F_RECEIPT_SLIP_OPEN	4096	SI	SI	SI

F_FISCAL_MEMORY_NEAR_FULL	La memoria fiscal de la IFH 1G está a punto de llenarse. Queda capacidad disponible para 30 cierres diarios 'Z', o menos. El software de gestión para puntos de venta debe alertar al usuario para que de inicio a los trámites ante el organismo oficial de competencia y el servicio técnico fiscal autorizado para proceder al recambio de la memoria fiscal.
F_FISCAL_TERMINAL_CERTIFIED	La IFH 1G se encuentra inicializada.
F_FISCAL_TERMINAL_FISCALIZED	La IFH 1G se encuentra inicializada.
F_RECEIPT_OPEN	Hay un comprobante fiscal abierto (rollo).
F_RECEIPT_SLIP_OPEN	Hay un comprobante fiscal abierto (formulario continuo u hoja suelta).

EventoImpresora()

Este evento es generado por el **OCX Fiscal HASAR** cuando determinado bit del campo de *Status Impresora* del string de respuesta a un comando enviado se encuentra en '1' (ver *Manual de Comandos de la IFH 1G*). De acuerdo al bit levantado, el incidente reportado se puede identificar mediante alguna de las constantes embebidas del **OCX Fiscal HASAR** que se listan en la tabla siguiente:

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
P_OFFLINE	8	SI	SI	SI
P_PRINTER_ERROR	4	SI	SI	SI
P_SPLIT_PLATEN_OPEN	256	SI	SI	SI

P_OFFLINE	La impresora propiamente dicha que forma parte de la IFH 1G se encuentra fuera de línea. El software de gestión para puntos de venta debe alertar al usuario para colocar al equipo nuevamente en línea.
P_PRINTER_ERROR	Se ha perdido la conexión entre la placa de control fiscal y la placa de control de impresión. Ambas placas son internas a la IFH 1G .
P_SPLIT_PLATEN_OPEN	De corresponder, la tapa de la IFH 1G se encuentra abierta.

FaltaPapel()

Este evento es generado por el **OCX Fiscal HASAR** cuando determinado bit del campo de *Status Impresora* del string de respuesta a un comando enviado se encuentra en '1' (ver *Manual de Comandos de la IFH 1G*). De acuerdo al bit levantado, el incidente reportado se puede identificar mediante alguna de las constantes embebidas del **OCX Fiscal HASAR** que se listan en la tabla siguiente:

Constantes Embebidas		OCX Fiscal HASAR		
Nombre	Dec	v010724	v051122	v110423 v150423
P_JOURNAL_PAPER_LOW	16	SI	SI	SI
P_RECEIPT_PAPER_LOW	32	SI	SI	SI

P_JOURNAL_PAPER_LOW	El sensor de papel de la IFH 1G ha detectado que falta papel en la estación de impresión de comprobantes a entregar al cliente.
P_RECEIPT_PAPER_LOW	El sensor de papel de la IFH 1G ha detectado que falta papel en la estación de impresión de la cinta testigo.

Este evento también puede ser generado por el **OCX Fiscal HASAR** cada vez que recibe de la **IFH 1G** el carácter especial DC4 (20 decimal), el cual se recibe cada 400 ms hasta que la situación se normaliza.

ImpresoraNoResponde()

El **OCX Fiscal HASAR** genera este evento sólo cuando la propiedad *ReintentoConstante()* se encuentra configurada con el valor *True* y se produce un error de *time out* en la comunicación con la **IFH 1G**. El valor que llega como argumento a la rutina que atrapa este evento indica cuantas veces la comunicación ha fracasado hasta el momento en que se generó el evento.

Por ejemplo,

```
Private Sub HASAR1_ImpresoraNoResponde(ByVal CantidadReintentos As Long)
    Debug.Print "Impresora No Responde: Reintentos = " & CantidadReintentos
End Sub
```

ImpresoraOcupada()

En general, la **IFH 1G** responde dentro de los 400 ms siguientes a la recepción de un string de comando. Cuando dicha respuesta debe ser demorada, porque el procesamiento y ejecución del comando demanda una ventana de tiempo mayor, entonces la **IFH 1G** envía a host un carácter especial (DC2, 18 decimal) cada

400 ms.; es para avisar al software de gestión para puntos de venta que no envíe un nuevo comando hasta que la respuesta pueda ser entregada. Por ejemplo, cuando se invoca al método *ReporteZ()*.

La propiedad *ContadorImpresoraOcupada()* informa la cantidad de veces que la **IFH 1G** se reportó ocupada hasta el momento de generarse este evento. Es importante que el software de gestión para puntos de venta controle este número ya que si por alguna razón la **IFH 1G** no puede devolverle el control (es decir, se reporta ocupada indefinidamente), se deberá alertar al usuario de dicha situación.

Por ejemplo,

```
DIM MAXIMA_CANT_DC2 = 500

Private Sub HASAR1_ImpresoraOcupada()
    If (HASAR1.ContadorImpresoraOcupada >= MAXIMA_CANT_DC2) Then
        Debug.Print "El equipo fiscal se reporta ocupado más de lo normal...!!"
    End If
End Sub
```

ImpresoraOK()

Se vio anteriormente que el **OCX Fiscal HASAR** genera el evento *ImpresoraOcupada()* -para que el software de gestión para puntos de venta no envíe un nuevo comando- hasta que la **IFH 1G** entrega la respuesta al comando previamente enviado. Cuando esta respuesta es recibida, el **OCX Fiscal HASAR** genera el evento *ImpresoraOK()*. El software de gestión para puntos de venta puede, ahora, enviar un nuevo comando.

El evento *ImpresoraOK()* también se genera cuando finaliza la situación que había provocado que se genere el evento *ImpresoraNoResponde()*, o cuando se normaliza la situación que había provocado que se genere el evento *FaltaPapel()*.

Por ejemplo,

```
Private Sub HASAR1_ImpresoraOK()
    enviar_otro_comando = True
End If
```

ProgresoDeteccion()

Cuando el software de gestión para puntos de venta invoca al método *AutodetectarControlador()*, el **OCX Fiscal HASAR** genera el evento *ProgresoDeteccion()* cada vez que intenta conexión a una nueva velocidad.

Por ejemplo,

```
Private Sub HASAR1_ProgresoDeteccion(ByVal Puerto As Long, ByVal Velocidad As Long)
    Debug.Print "Probando conexión..." & vbCrLf & "COM = " & Puerto & vbCrLf & "Baudios = " & _
    Velocidad
End Sub
```

Excepciones del OCX Fiscal HASAR

Enum: ErroresFiscales

Nombre	Constantes Embebidas		OCX Fiscal HASAR		
	Valor Dec	Valor Hexa	v010724	v051122	v110423 v150423
H_ERR_GENERIC	-2147220991	80040201	SI		
H_ERR_HANDLER	-2147220990	80040202	SI		
H_ERR_ATOMIC	-2147220989	80040203	SI	SI	SI
H_ERR_TIMEOUT	-2147220988	80040204	SI	SI	SI
H_ERR_ALREADYOPEN	-2147220987	80040205	SI	SI	SI
H_ERR_NOMEM	-2147220986	80040206	SI		
H_ERR_NOTOPENYET	-2147220985	80040207	SI	SI	SI
H_ERR_INVALIDPTR	-2147220984	80040208	SI		
H_ERR_STATPRN	-2147220983	80040209	SI		
H_ERR_ABORT	-2147220982	8004020A	SI	SI	SI
H_ERR_FIELD_NOT_FOUND	-2147220942	80040232	SI	SI	SI
H_ERR_INVALID_BUFFER	-2147220941	80040233	SI		
H_ERR_INVALID_BIT	-2147220940	80040234	SI	SI	SI
H_ERR_PRINTER_NOT_FOUND	-2147220939	80040235	SI	SI	SI
H_ERR_NOT_SUPPORTED	-2147220938	80040236	SI	SI	SI
H_ERR_NOT_OPENED	-2147220937	80040237	SI		
H_ERR_INVALID_PORT	-2147220936	80040238	SI		
H_ERR_FILENAME	-2147220935	80040239	SI	SI	SI
H_ERR_FIELD_OPTIONAL	-2147220934	8004023A	SI	SI	SI
H_ERR_FIELD_INVALID	-2147220933	8004023B	SI	SI	SI
H_ERR_SERIAL_PARAMETERS	-2147220972	80040214		SI	SI
H_ERR_OPEN	-2147220971	80040215		SI	SI
H_ERR_READ	-2147220970	80040216		SI	SI
H_ERR_WRITE	-2147220969	80040217		SI	SI
H_ERR_INIT_SOCKET	-2147220968	80040218		SI	SI
H_ERR_BIND_SOCKET	-2147220967	80040219		SI	SI
H_ERR_CONNECTION	-2147220966	8004021A		SI	SI
H_ERR_SELECT	-2147220965	8004021B		SI	SI
H_ERR_VERSION_SOCKET	-2147220964	8004021C		SI	SI
H_ERR_IP_FORMAT	-2147220963	8004021D		SI	SI
H_ERR_MAX_RETRIES	-2147220962	8004021E		SI	SI
H_ERR_WRONG_PROTOCOL	-2147220961	8004021F		SI	SI
H_ERR_BAD_PROTOCOL	-2147220960	80040220		SI	SI
H_ERR_NOT_EXECUTED	-2147220932	8004023C		SI	SI
H_ERR_CF_NOT_DETECTED	-2147220931	8004023D		SI	SI
H_ERR_INVALID_DATE	-2147220930	8004023E		SI	SI
H_ERR_INVALID_TIME	-2147220929	8004023F		SI	SI
H_ERR_MANDATORY_PARAMETER	-2147220928	80040240		SI	SI
H_ERR_INVALID_STATE	-2147220927	80040241		SI	SI
H_ERR_DEPRECATED_FUNCTION	-2147220926	80040242		SI	SI
H_ERR_INVALID_PARAMETER	-2147220925	80040243		SI	SI
H_ERR_NO_MORE_ITEMS	-2147220924	80040244		SI	SI

Constantes Embebidas			OCX Fiscal HASAR		
Nombre	Valor		v010724	v051122	v110423
	Dec	Hexa			
H_ERR_COMMUNICATIONS_ANOMALY	-2147220923	80040245		SI	SI
H_ERR_COMMAND_MISMATCH	-2147220922	80040246		SI	SI

Las celdas de la tabla anterior que presentan la coloración que se ve a izquierda, se interpretan como “NO SOPORTADO”

Las excepciones que tiene capacidad de generar el **OCX Fiscal HASAR** son las listadas en la tabla anterior. Estas excepciones se originan a partir de incidentes que ocurren en el host. Es decir, la **IFH 1G** no alcanza a recibir comando alguno debido a que la generación de la excepción impide la construcción y envío de cualquier string de comando.

Es importante que el software de gestión para puntos de venta atrape y trate adecuadamente cada una de estas excepciones. No hacerlo implica el riesgo de perder el sincronismo con la **IFH 1G** respecto de montos acumulados y numeración alcanzada, así como de generar incidentes evitables.

Como atrapar excepciones

Cada lenguaje de programación ofrece su propio mecanismo para atrapar excepciones. Los desarrolladores de software de gestión para puntos de venta deberán consultar la documentación correspondiente a su entorno de desarrollo para interiorizarse acerca de como atrapar una excepción.

Se muestran a continuación dos casos, y a modo de ejemplo.

Caso: Visual Basic 6.0

Se recurre al empleo de la sentencia “On Error GoTo”.

Por ejemplo,

```
.....
On Error GoTo Tratar_On_Error
.....
HASAR1.Comenzar
.....
Exit Sub
Tratar_On_Error:
Select Case Err.Number
    Case H_ERR_CF_NOT_DETECTED
        .....
    Else
        .....
End Select
.....
```

Caso: Visual C#

Se recurre al empleo de la sentencia “Try / Catch”.

Por ejemplo,

```
.....
Try {
    .....
    Ofh.Comenzar
    .....
}

catch(SystemException exc){
    MessageBox.Show(exc.Source + " ::: " + exc.Message);
}
.....
```

Mensajes asociados a las excepciones

H_ERR_GENERIC	OCX v010724 <i>Error general en DLL fiscal.</i> Esta excepción se genera ante un error interno de operación con la librería <i>winfis32.dll</i> . Hubo una desinteligencia. También es generada por el OCX Fiscal HASAR ante un error desconocido.
H_ERR_HANDLER	OCX v010724 <i>Handler inválido.</i> Esta excepción se genera ante un error interno de operación con la librería <i>winfis32.dll</i> . Hubo una desinteligencia. También es generada por el OCX Fiscal HASAR ante un error desconocido.
H_ERR_ATOMIC	OCX v010724 <i>Intento de operar cuando se estaba procesando un comando anterior.</i> OCX v051122, v110423, v150423 <i>Error de reentrancia en la ejecución.</i> Esta excepción se genera cuando se intenta enviar a la IFH 1G un nuevo comando antes de recibir la respuesta al comando anterior.
H_ERR_TIMEOUT	OCX v010724 <i>Error de comunicaciones. El controlador no está respondiendo.</i> OCX v051122, v110423, v150423 <i>Tiempo de espera agotado.</i> Esta excepción se genera cuando se produce un error en las comunicaciones con la IFH 1G . Se trate porque no hay equipo fiscal conectado, el puerto utilizado no funciona correctamente, el cable de comunicación es defectuoso, la IFH 1G se encuentra apagada, etc..

H_ERR_ALREADYOPEN	OCX v010724 <i>El puerto seleccionado ya estaba abierto.</i>
	OCX v051122, v110423, v150423 <i>El dispositivo ya estaba abierto.</i>
	Esta excepción se genera cuando se intenta abrir el puerto de comunicaciones y éste ya se encuentra abierto; sea por el software de gestión para puntos de venta, o por otra aplicación.
H_ERR_NOMEM	OCX v010724 <i>Falta de memoria.</i>
	Esta excepción se genera cuando el OCX Fiscal HASAR no posee memoria suficiente para completar la operación solicitada mediante un evento/propiedad.
H_ERR_NOTOPENYET	OCX v010724 <i>Intento de usar un puerto que no estaba abierto.</i>
	OCX v051122, v110423, v150423 <i>No se ha inicializado un transporte.</i>
	Esta excepción se genera cuando ha enviado un comando a la IFH 1G sin haber abierto previamente el puerto de comunicaciones. Por ejemplo, no se invocó al método <i>Comenzar()</i> .
H_ERR_INVALIDPTR	OCX v010724 <i>Error de excepción: dirección inválida o lugar insuficiente.</i>
	Esta excepción se genera por un error interno del OCX Fiscal HASAR . El buffer para almacenar una respuesta es inválido.
H_ERR_ABORT	OCX v010724 <i>Aborción provocada por el usuario.</i>
	OCX v051122, v110423, v150423 <i>Operación abortada.</i>
	Esta excepción se genera cuando el proceso en curso fue interrumpido por el software de gestión para puntos de venta mediante la invocación al método <i>Abortar()</i> . La necesidad de abortar surge cuando está configurada la propiedad <i>ReintentoConstante()</i> con el valor <i>True</i> , y se produce un error de <i>time out</i> en las comunicaciones con la IFH 1G .
H_ERR_FIELD_NOT_FOUND	OCX v010724, v051122, v110423, v150423 <i>Campo de respuesta no encontrado.</i>
	Esta excepción se genera cuando el software de gestión para puntos de venta invoca a la propiedad <i>Respuesta()</i> solicitando la información de un campo inexistente en el string recibido.

	<p>Tener presente que la cantidad de campos en un string de respuesta puede variar ante distintos modelos de IFH 1G. Algunos modelos pueden entregar respuestas más cortas, tratándose del mismo comando.</p> <p>Tener presente también que las respuesta a comandos rechazados sólo contienen los dos primeros campos: <i>status de impresora</i> y <i>status auxiliar</i> (existentes en todas las respuestas).</p>
H_ERR_INVALID_BIT	<p>OCX v010724, v051122, v110423, v150423 <i>Identificador de bit no válido.</i></p> <p>Esta excepción se genera cuando la constante embebida pasada por el software de gestión para puntos de venta como argumento de las propiedades <i>DescripcionStatusImpresor()</i> y/o <i>DescripcionStatusFiscal()</i> es incorrecta.</p> <p>También se puede generar por invocación de la propiedad <i>DescripcionEstadoControlador()</i>. Pero en este caso, se trataría de un error interno del OCX Fiscal HASAR.</p>
H_ERR_PRINTER_NOT_FOUND	<p>OCX v010724 <i>El impresor solicitado no existe.</i></p> <p>Esta excepción se genera cuando el software de gestión para puntos de venta le asigna a la propiedad <i>Modelo()</i> una constante embebida que identifica a una IFH 1G no soportada.</p>
H_ERR_NOT_SUPPORTED	<p>OCX v010724, v051122, v110423, v150423 <i>Función no soportada en versión de controlador fiscal seleccionada.</i></p> <p>Esta variante de la excepción se genera cuando se invoca a un método o una propiedad del OCX Fiscal HASAR que produciría un string de comando no soportado por la IFH 1G identificada mediante la propiedad <i>Modelo()</i>.</p> <p>También se puede generar por el intento de utilizar una constante embebida incorrecta, de acuerdo al método o propiedad invocado, y a la IFH 1G identificada mediante la propiedad <i>Modelo()</i>.</p> <p>OCX v051122, v110423, v150423 <i>Autodetección sólo soportado para puerto serie.</i></p> <p>Esta variante de la excepción se genera cuando a la propiedad <i>Transporte()</i> se le asigna la constante embebida SOCKET_TCP o SOCKET_UDP, y con posterioridad se invoca al método <i>AutodetectarControlador()</i>.</p> <p>OCX v051122, v110423, v150423 <i>El modo StatPRN es dependiente del modelo.</i></p>

	<p>Esta variante de la excepción se genera cuando se intenta asignar el valor <i>True</i> a la propiedad <i>ModoStatPrn()</i>, y la IFH 1G identificada mediante la propiedad <i>Modelo()</i> no soporta consultas de estado intermedio.</p>
H_ERR_NOT_OPENED	<p>OCX v010724 <i>No ha sido abierto el puerto.</i></p>
	<p>Esta excepción se genera cuando se invoca a un método o propiedad que produce y envía un string de comando a la IFH 1G, sin que todavía se haya abierto el puerto de comunicaciones. Es decir, previamente no se invocó al método <i>Comenzar()</i>; sea en forma directa, o indirecta por falta de invocación al método <i>AutodetectarControlador()</i>.</p>
H_ERR_INVALID_PORT	<p>OCX v010724 <i>El puerto especificado es inválido.</i></p>
	<p>Esta excepción se genera al invocar al método <i>Comenzar()</i> indicando, previamente, un valor incorrecto en la propiedad <i>Puerto()</i>.</p>
H_ERR_FILENAME	<p>OCX v010724, v051122, v110423, v150423 <i>El nombre de archivo no es válido.</i></p>
	<p>Esta excepción se genera cuando el archivo especificado al invocar a un método del OCX Fiscal HASAR, dicho archivo no existe o no puede ser abierto. Por ejemplo, al invocar al método <i>CargarLogo()</i>.</p>
H_ERR_FIELD_OPTIONAL	<p>OCX v010724, v051122, v110423, v150423 <i>Un parámetro optional no es válido o está ausente cuando es necesario.</i></p>
	<p>Esta excepción se genera al invocar un método con un argumento opcional, el cual es requerido y se lo omite. Por ejemplo, omitir el domicilio del comprador al invocar al método <i>DatosCliente()</i> cuando la propiedad <i>Modelo()</i> tiene asignado el valor MODELO_P330.</p>
H_ERR_FIELD_INVALID	<p>OCX v010724, v051122, v110423, v150423 <i>Parámetro no válido.</i></p>
	<p>Esta excepción se genera cuando alguno de los argumentos requeridos para invocar a un método o propiedad del OCX Fiscal HASAR, recibe un valor inesperado.</p>
H_ERR_SERIAL_PARAMETERS	<p>OCX v051122, v110423, v150423 <i>Error de parámetros de puerto serie.</i></p>
	<p>Esta excepción se genera cuando, ya abierto el puerto serie, se intenta modificar algún parámetro de configuración del mismo.</p>
H_ERR_OPEN	<p>OCX v051122, v110423, v150423 <i>Error de apertura.</i></p>

	<p>Esta excepción se genera cuando fracasa la apertura del puerto serie, la apertura del socket TCP, o la apertura del socket UDP.</p>
H_ERR_READ	<p>OCX v051122, v110423, v150423 <i>Error de lectura.</i></p> <p>Esta excepción se genera cuando se intenta leer un string a través del puerto serie, o del socket TCP o UDP. No tiene su origen en una falta de comunicación con la IFH 1G.</p>
H_ERR_WRITE	<p>OCX v051122, v110423, v150423 <i>Error de escritura.</i></p> <p>Esta excepción se genera cuando se intenta enviar un string a través del puerto serie, o del socket TCP o UDP. No tiene su origen en una falta de comunicación con la IFH 1G.</p>
H_ERR_INIT_SOCKET	<p>OCX v051122, v110423, v150423 <i>Error de inicialización de socket.</i></p> <p>Esta excepción se genera cuando se crea el objeto que permite el manejo interno del socket TCP o UDP.</p>
H_ERR_BIND_SOCKET	<p>OCX v051122, v110423, v150423 <i>Error de bind().</i></p> <p>Esta excepción se genera cuando se intenta asociar el objeto que permite el manejo interno del socket TCP o UDP con un puerto.</p>
H_ERR_CONNECTION	<p>OCX v051122, v110423, v150423 <i>Error de conexión.</i></p> <p>Esta excepción se genera cuando se intenta establecer conexión a través del socket TCP o UDP con el conversor Ethernet/RS-232 al que se conecta la IFH 1G.</p>
H_ERR_VERSION_SOCKET	<p>OCX v051122, v110423, v150423 <i>Error de versión de socket.</i></p> <p>Esta excepción se genera cuando ocurre una incompatibilidad entre la versión del manejador de socket y la versión de Windows®.</p>
H_ERR_IP_FORMAT	<p>OCX v051122, v110423, v150423 <i>Error de formato en dirección de IP.</i></p> <p>Esta excepción se genera cuando algunos de los campos de la dirección de IP, especificada mediante la propiedad <i>DireccionIP()</i> no es numérico, o la dirección de IP especificada no tiene exactamente cuatro campos.</p>
H_ERR_MAX_RETRIES	<p>OCX v051122, v110423, v150423</p>

	<p><i>Se han agotado la cantidad máxima de reintentos.</i></p> <p>Esta excepción se genera ante un error en las comunicaciones al intentar enviar un comando a la IFH 1G, y el OCX Fiscal HASAR ha superado la cantidad máxima de veces de reinicio de envío del comando; valor máximo programado mediante la propiedad <i>Reintentos()</i>.</p>
H_ERR_WRONG_PROTOCOL	<p>OCX v051122, v110423, v150423 <i>El impresora conectado no corresponde al modelo seleccionado.</i></p> <p>Esta excepción se genera cuando la IFH 1G está trabajando con un protocolo diferente al utilizado por el OCX Fiscal HASAR. Es decir, de un lado el protocolo incluye un carácter ESC (27 decimal) precediendo al código de operación del comando en el paquete de información, y del otro lado no se lo espera.</p>
H_ERR_BAD_PROTOCOL	<p>OCX v051122, v110423, v150423 <i>Error en protocolo fiscal.</i></p> <p>Esta excepción se genera cuando el formato del paquete de información es incorrecto.</p>
H_ERR_NOT_EXECUTED	<p>OCX v051122, v110423, v150423 <i>Función no ejecutada con éxito.</i></p> <p>Esta excepción se genera cuando, paralelamente, el OCX Fiscal HASAR dispara el evento <i>ErrorFiscal()</i> y también asigna el valor <i>True</i> a la propiedad <i>HuboErrorFiscal()</i>.</p>
H_ERR_CF_NOT_DETECTED	<p>OCX v051122, v110423, v150423 <i>No se encontró una coincidencia para el controlador identificado como...</i></p> <p>Esta excepción se genera cuando se invocó al método <i>AutodetectarModelo()</i> y el OCX Fiscal HASAR verifica si posee una constante que lo identifique para asignar a la propiedad <i>Modelo()</i>.</p> <p>OCX v051122, v110423, v150423 <i>Controlador no detectado.</i></p> <p>Esta variante de la excepción se genera cuando el identificador de comando de la respuesta recibida no coincide con el identificador del comando enviado, se han agotado los reintentos de envío del comando en cuestión.</p> <p>También se puede generar cuando se invocó al método <i>AutodetectarControlador()</i> y no se obtuvo comunicación con la IFH 1G a ninguna de las velocidades posibles.</p>
H_ERR_INVALID_DATE	<p>OCX v051122, v110423, v150423 <i>Fecha no válida.</i></p>

	<p>Esta excepción se genera cuando la fecha no ha sido informada en el formato AAMMDD (exactamente 6 dígitos).</p>
H_ERR_INVALID_TIME	<p>OCX v051122, v110423, v150423 <i>Hora no válida.</i></p>
	<p>Esta excepción se genera cuando la hora no ha sido informada en el formato HHMMSS (exactamente 6 dígitos).</p>
H_ERR_MANDATORY_PARAMETER	<p>OCX v051122, v110423, v150423 <i>Un parámetro obligatorio no ha sido suministrado.</i></p>
	<p>Esta excepción se genera cuando se invoca al método <i>BajaFiscal()</i> y se omite la indicación del número de registro de la IFH 1G.</p>
H_ERR_INVALID_STATE	<p>OCX v051122, v110423, v150423 <i>Estado no válido para el requerimiento solicitado.</i></p>
	<p>Esta excepción se genera cuando se ha invocado al método <i>DesuentoUltimoItem()</i> sin haber invocado previamente al método <i>ImprimirItem()</i>.</p>
	<p>También se genera cuando se consulta la propiedad <i>PrimerNúmeroDeDocumentoActual()</i> cuando no hay un comprobante abierto.</p>
	<p>Se puede generar, además, cuando se consulta la propiedad <i>PáginasDeUltimoDocumento()</i> y dicha información no está disponible.</p>
H_ERR_DEPRECATED_FUNCTION	<p>OCX v051122, v110423, v150423 <i>Función discontinuada.</i></p>
	<p>Esta excepción se genera cuando se invoca a un método o propiedad del OCX Fiscal HASAR no disponible para la versión del objeto en uso. Por ejemplo, invocar al método <i>Recomenzar()</i> con empleo del <i>OCX v110423</i>.</p>
H_ERR_INVALID_PARAMETER	<p>OCX v051122, v110423, v150423 <i>El parámetro de puerto debe ser numérico.</i></p>
	<p>Esta variante de la excepción se genera cuando se invoca al método <i>AutodetectarControlador()</i> con un parámetro no numérico.</p>
	<p>OCX v051122, v110423, v150423 <i>El identificador de sub-capacidad es inválido.</i></p>
	<p>Esta variante de la excepción se genera cuando se invoca al método <i>Capacidades()</i> con una constante embebida de subcapacidad que es incorrecta.</p>
	<p>OCX v051122, v110423, v150423 <i>El identificador de capacidad es inválido.</i></p>

Esta variante de la excepción se genera cuando se invoca al método *Capacidades()* con una constante embebida de capacidad que es incorrecta.

OCX v051122, v110423, v150423

El parámetro de número de serie debe ser un string.

Esta variante de la excepción se genera cuando se invoca al método *BajaFiscal()* indicando incorrectamente el número de registro con el cual fue inicializada la **IFH 1G**.

OCX v051122, v110423, v150423

El parámetro de dirección debe ser un string.

Esta variante de la excepción se produce cuando se invoca al método *DatosCliente()* indicando incorrectamente el dato que corresponde al domicilio del cliente.

OCX v051122, v110423, v150423

El límite de consumidor final debe ser un número de punto flotante.

Esta variante de la excepción se genera cuando se invoca al método *ConfigurarControladorCompleto()* indicando incorrectamente el monto límite para comprobantes “B” a consumidores finales, y sin datos del comprador.

OCX v051122, v110423, v150423

El límite del ticket factura debe ser un número de punto flotante.

Esta variante de la excepción se genera cuando se invoca al método *ConfigurarControladorCompleto()* indicando incorrectamente el monto límite para tiques factura en general.

OCX v051122, v110423, v150423

El porcentaje de IVA No Inscripto debe ser un número de punto flotante.

Esta variante de la excepción se genera cuando se invoca al método *ConfigurarControladorCompleto()* indicando incorrectamente la sobretasa de IVA No Inscripto (no vigente en la normativa fiscal actual), la cual debe ser cero.

OCX v051122, v110423, v150423

El número de copias máximo es inválido.

Esta variante de la excepción se genera cuando se invoca al método *ConfigurarControladorCompleto()* indicando incorrectamente el tipo de dato correspondiente a la máxima cantidad de ejemplares que se permitirá emitir de un mismo comprobante.

OCX v051122, v110423, v150423

El número de copias máximo debe estar comprendido entre 0 y 4.

Esta variante de la excepción se genera cuando se invoca al método *ConfigurarControladorCompleto()* indicando la máxima cantidad de ejemplares que se permitirá emitir de un mismo comprobante con un valor fuera del rango admitido.

OCX v051122, v110423, v150423

El parámetro ImprimeCambio debe ser un número booleano.

Esta variante de la excepción se genera cuando se invoca al método *ConfigurarControladorCompleto()* indicando incorrectamente el dato correspondiente a la impresión, o no, de la leyenda “Cambio \$0.00”.

OCX v051122, v110423, v150423

El parámetro ImprimeLeyendasOpcionales debe ser un número booleano.

Esta variante de la excepción se genera cuando se invoca al método *ConfigurarControladorCompleto()* indicando incorrectamente el dato correspondiente a la impresión, o no, de las leyenda opcionales (por ejemplo, número de inscripción en Ingresos Brutos).

OCX v051122, v110423, v150423

El parámetro TipoDeCorte es incorrecto.

Esta variante de la excepción se genera cuando se invoca al método *ConfigurarControladorCompleto()* indicando incorrectamente el dato que configura el tipo de corte de papel.

OCX v051122, v110423, 150423

El parámetro ImprimeMarco debe ser un número booleano.

Esta variante de la excepción se genera cuando se invoca al método *ConfigurarControladorCompleto()* indicando incorrectamente el dato que configura la impresión, o no, del marco en los comprobantes.

OCX v051122, v110423, v150423

El parámetro ReImprimeDocumentos debe ser un número booleano.

Esta variante de la excepción se genera cuando se invoca al método *ConfigurarControladorCompleto()* indicando incorrectamente el dato que configura la reimpresión de comprobantes cancelados por corte de energía.

OCX v051122, v110423, v150423

La descripción del medio de pago debe ser un string.

Esta variante de la excepción se genera cuando se invoca al método

ConfigurarControladorCompleto() indicando incorrectamente el texto a imprimir asociado a un monto de pago pendiente de ser saldado al momento del cierre del comprobante.

OCX v051122, v110423, v150423

El parámetro Sonido debe ser un número booleano.

Esta variante de la excepción se genera cuando se invoca al método *ConfigurarControladorCompleto()* indicando incorrectamente el dato que configura la emisión, o no, de pitidos por falta de papel.

OCX v051122, v110423, v150423

El parámetro AltoHoja es incorrecto.

Esta variante de la excepción se genera cuando se invoca al método *ConfigurarControladorCompleto()* indicando incorrectamente el dato que configura el alto del papel a utilizar.

OCX v051122, v110423, v150423

El parámetro AnchoHoja es incorrecto.

Esta variante de la excepción se genera cuando se invoca al método *ConfigurarControladorCompleto()* indicando incorrectamente el dato que configura el ancho del papel a utilizar.

OCX v051122, v110423, v150423

El parámetro EstacionImpresionReportesXZ es incorrecto.

Esta variante de la excepción se genera cuando se invoca al método *ConfigurarControladorCompleto()* indicando incorrectamente el dato que configura la estación de impresión de reportes "X"/"Z".

OCX v051122, v110423, v150423

El parámetro ModoImpresion es incorrecto.

Esta variante de la excepción se genera cuando se invoca al método *ConfigurarControladorCompleto()* indicando incorrectamente el dato que configura si se imprimirá sólo en formulario continuo, o se alternará con impresiones en rollo.

OCX v051122, v110423, v150423

El parámetro Copias debe ser numérico.

Esta variante de la excepción se genera cuando se invoca al método *CerrarComprobante.....()* indicando incorrectamente el argumento que establece la cantidad de ejemplares a emitir.

OCX v051122, v110423, v150423

El parámetro Estación es incorrecto.

Esta variante de la excepción se genera cuando se invoca al método *AbrirComprobanteNoFiscal()* indicando incorrectamente el argu-

mento que establece la estación de impresión del comprobante.

OCX v051122, v110423, v150423

El parámetro de identificación debe ser un string.

Esta variante de la excepción se genera cuando se invoca al método *AbrirDNFH()* indicando incorrectamente el argumento que establece la numeración que llevará el comprobante.

OCX v051122, v110423, v150423

El parámetro de cantidad debe numérico.

Esta variante de la excepción se genera cuando se invoca al método *ImprimirInfoEnDNFH()* indicando incorrectamente el argumento que establece la cantidad de unidades del ítem.

OCX v051122, v110423, v150423

El parámetro de ComprobanteAsociado debe ser un string válido o un número.

Esta variante de la excepción se genera cuando se invoca al método *ImprimirVoucher()*, o *ImprimirVoucherConFechaOculta()*, indicando incorrectamente el argumento que identifica el número de comprobante a consignar en el voucher de tarjeta a imprimir.

OCX v051122, v110423, v150423

El parámetro de Vendedor debe ser un string válido.

Esta variante de la excepción se genera cuando se invoca al método *ImprimirVoucher()*, o *ImprimirVoucherConFechaOculta()*, indicando incorrectamente el argumento que identifica al vendedor a consignar en el voucher de tarjeta a imprimir.

OCX v051122, v110423, v150423

El parámetro Estación es incorrecto.

Esta variante de la excepción se genera cuando se invoca al método *ImprimirVoucher()*, o *ImprimirVoucherConFechaOculta()*, indicando incorrectamente el argumento que establece la estación de impresión del comprobante.

OCX v051122, v110423, v150423

El parámetro de descripción adicional debe ser un string válido.

Esta variante de la excepción se genera cuando se invoca al método *ImprimirPago()*, o *DevolverPago()*, indicando incorrectamente el argumento que establece el texto adicional asociado a un pago con tarjeta.

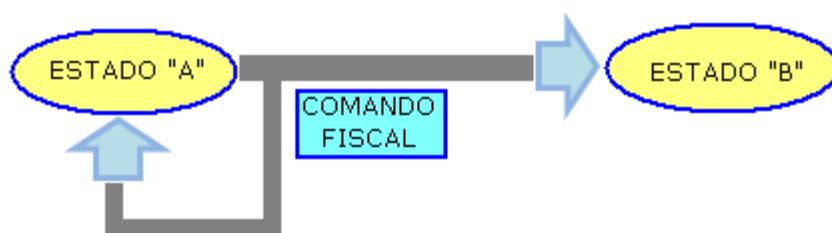
NOTA:

	<p>Es resumen, esta excepción se genera cuando alguno de los argumentos requeridos para invocar a un método o propiedad del OCX Fiscal HASAR, recibe un valor inesperado.</p>
H_ERR_NO_MORE_ITEMS	<p>OCX v051122, v110423, v150423 Esta excepción no se genera actualmente. Se encuentra reservada para futuras aplicaciones.</p>
H_ERR_COMMUNICATIONS_ANOMALY	<p>OCX v051122, v110423, v150423 Si bien esta excepción se genera internamente, no se dispara para que la atrape y trate el software de gestión para puntos de venta. Esta excepción es atrapada y manejada internamente por el OCX Fiscal HASAR.</p>
H_ERR_COMMAND_MISMATCH	<p>OCX v051122, v110423, v150423 <i>Se han agotado los reintentos por código de comando no coincidente.</i></p> <p>Esta excepción se genera cuando el identificador de comando de la respuesta recibida no coincide con el identificador del comando enviado, y se han agotado todos reintentos permitidos de reenvío del comando.</p>

Apéndice: Generación de comprobantes

Generalidades

En este apéndice se muestra la generación de comprobantes a partir de la incidencia de los posibles estados en que puede encontrarse la **IFH 1G**, y respecto de la impresión de los diferentes tipos de comprobantes que se pueden emitir.



Es importante para el desarrollador del software de gestión para puntos de venta, no sólo conocer los distintos estados en que puede encontrarse la **IFH 1G**, sino también cuales comandos puedes ser ejecutados en cada estado.

La transición de un estado 'A' a otro 'B' (y/o la permanencia en el mismo estado 'A') la provoca la ejecución del comando enviado a la **IFH 1G** por el software de gestión para puntos de venta.

Las transiciones entre estados, para la generación de comprobantes, se presentará en este apéndice de dos formas:

- *Como tabla de doble entrada*
 - Cada fila de la tabla corresponde a un string de comando soportado por la **IFH 1G**.
 - Cada columna de la tabla corresponde a un string de comando soportado por la **IFH 1G**.
- *Como diagrama de flujo*

Si la celda de intersección entre fila y columna se encuentra marcada con el carácter '•' significa que el string de comando comando mencionado en la columna se puede enviar, a la **IFH 1G**, a continuación del string de comando comando mencionado en la fila.

En el caso de strings de comandos de configuración, consulta, control fiscal, u otros de uso general, la tabla que se muestra es más simple debido a que se trata de strings de comandos que no forman parte de una secuencia (salvo aquellos que requieren un cierre diario 'Z' previo. Para este tipo de comandos no se requiere mostrar un diagrama de flujo).

La estructura de la tabla en estos casos es la siguiente:

- Cada fila de la tabla corresponde a un string de comando soportado por la **IFH 1G**.
- La tabla cuenta con tres columnas:
 - *Cierre Z Previo*

Si la celda de intersección entre la fila y esta columna se encuentra marcada con el carácter '•' significa que se requiere del cierre diario 'Z' para ejecutar con éxito el comando.
 - *Dentro de comprobante*

Si la celda de intersección entre la fila y esta columna se encuentra marcada con el carácter '•' significa que el comando se puede ejecutar aunque haya un comprobante abierto en la **IFH 1G**.
 - *Fuera de comprobante*

Si la celda de intersección entre la fila y esta columna se encuentra marcada con el carácter ‘•’ significa que el comando se puede ejecutar aunque no se haya abierto ningún comprobante en la IFH 1G.

(DF) Documentos Fiscales

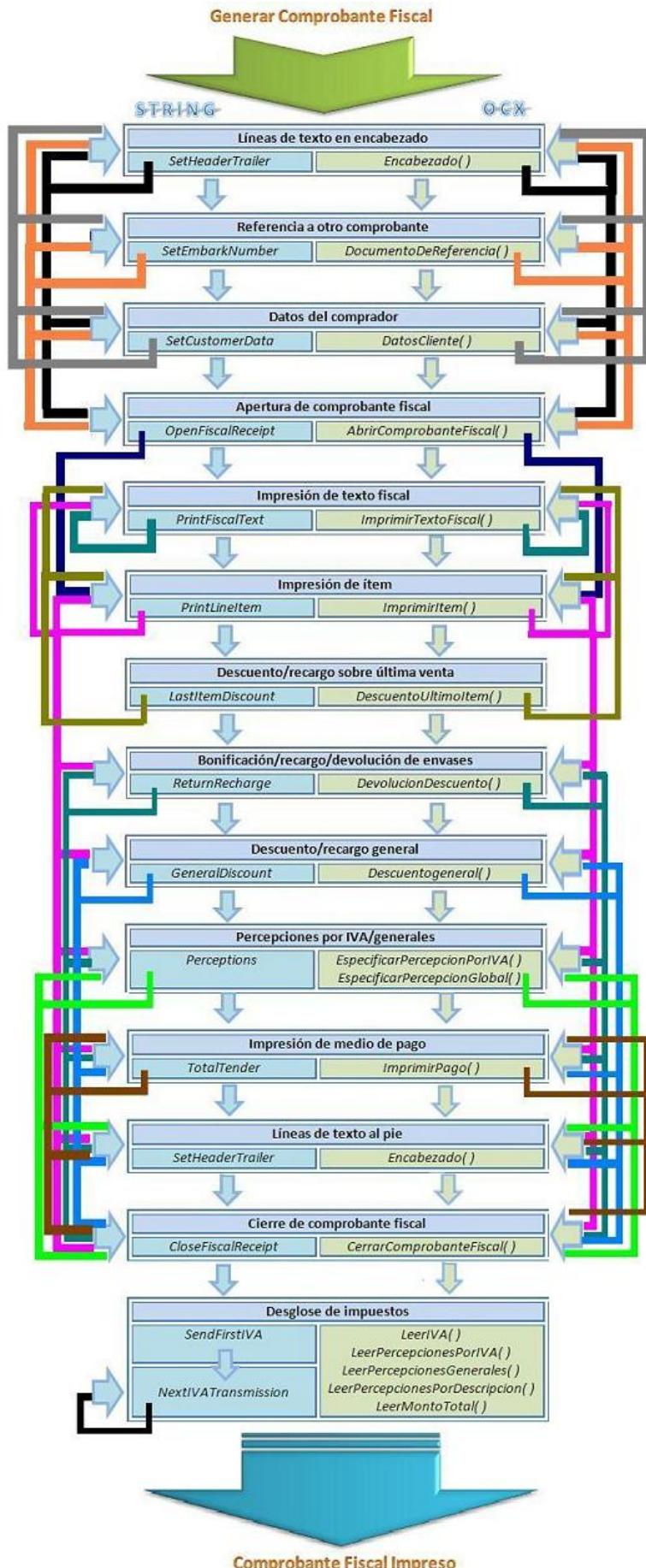
La tabla que se muestra a continuación y el diagrama de flujo de la página siguiente resultan válidos para: Tiques, Tiques Factura, Facturas, Tiques Nota de Débito y Notas de Débito.

	<i>SetHeaderTrailer</i>	<i>SetEmbarkNumber</i>	<i>SetCustomerData</i>	<i>OpenFiscalReceipt</i>	<i>PrintFiscalText</i>	<i>PrintLineItem</i>	<i>LastItemDiscount</i>	<i>ReturnRecharge</i>	<i>GeneralDiscount</i>	<i>Perceptions</i>	<i>TotalTender</i>	<i>SetHeaderTrailer</i>	<i>CloseFiscalReceipt</i>	<i>SendFirstIVA</i>	<i>NextIVATransmission</i>
<i>SetHeaderTrailer (1)</i>	•	•	•	•											
<i>SetEmbarkNumber</i>	•	•	•	•											
<i>SetCustomerData</i>	•	•	•	•											
<i>OpenFiscalReceipt</i>					•	•									
<i>PrintFiscalText</i>					•	•									
<i>PrintLineItem</i>					•	•	•	•	•	•	•	•			
<i>LastItemDiscount</i>					•	•		•	•	•	•	•			
<i>ReturnRecharge</i>								•	•	•	•	•			
<i>GeneralDiscount</i>									•	•	•	•			
<i>Perceptions</i>										•	•	•			
<i>TotalTender</i>											•	•			
<i>SetHeaderTrailer (2)</i>												•	•		
<i>CloseFiscalReceipt</i>													•		
<i>SendFirstIVA</i>														•	
<i>NextIVATransmission</i>															•

(1) Las líneas de encabezado deben cargarse en la memoria de trabajo antes de abrir el comprobante. El comando de apertura del comprobante es quien las imprime.

(2) Las líneas de pie deben cargarse en la memoria de trabajo antes de cerrar el comprobante. El comando de cierre del comprobante es quien las imprime.

Luego del string de comando *SetHeaderTrailer*, una vez abierto el comprobante fiscal, y antes de enviar el string de comando de cierre del mismo, se admite el envío de cualquier string de comando válido dentro del comprobante.



La impresión de leyendas en el encabezado de los comprobantes fiscales es opcional.

Si los textos cargados en la memoria de trabajo de la **IFH 1G** no se eliminan luego de cerrar el comprobante, serán impresos en los futuros comprobantes que se emitan.

La referencia a otro comprobante es opcional. De utilizarse, en el encabezado del comprobante se imprimirá la leyenda “*Remito Nº :*” precediendo a la información suministrada mediante este comando.

Los datos del comprador son obligatorios en comprobantes ‘A’, y opcionales en comprobantes ‘B’. No se requieren en el caso de Tiques.

Para superar los \$1000 (pesos un mil) en comprobantes ‘B’ a consumidor final se requieren en forma obligatoria los datos completos del comprador.

NOTA:

Mediante la RG AFIP Nº 3666/14, vigente desde el 4 de Setiembre de 2014, se suspende temporalmente la aplicación del tope de \$1000 (pesos un mil) como monto máximo para Tiques y comprobantes B a consumidores finales sin nominar.

Por suspensión temporal se entiende: hasta tanto entre en vigencia lo previsto en los párrafos segundo y tercero del Artículo 3º de la RG AFIP Nº 3561/13.

Si bien la **IFH 1G** no restringe el uso del comando de impresión de percepciones, éste se encuentra disponible para aquellos casos donde el emisor (contribuyente) es un agente de retención.

Los comandos de desglose de impuestos no causan impresión, son sólo de consulta. La discriminación de impuestos, de corresponder, es impresa por el comando de cierre del comprobante.

(DNFH) Tiques Notas de Crédito / Notas de Crédito

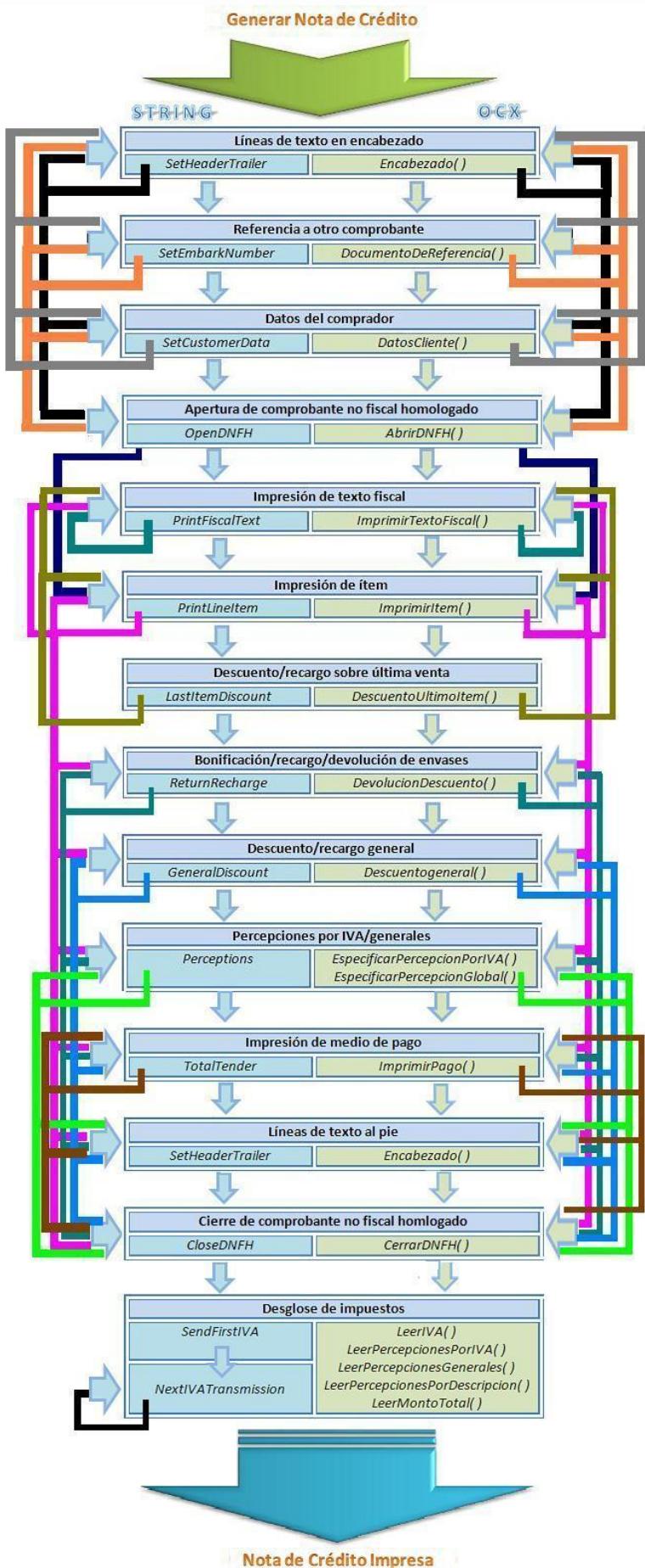
La tabla que se muestra a continuación y el diagrama de flujo de la página siguiente resultan válidos para: Tiques Nota de Crédito y Notas de Crédito.

	<i>SetHeaderTrailer</i>	<i>SetEmbarkNumber</i>	<i>SetCustomerData</i>	<i>OpenDNFH</i>	<i>PrintFiscalText</i>	<i>PrintLineItem</i>	<i>LastItemDiscount</i>	<i>ReturnRecharge</i>	<i>GeneralDiscount</i>	<i>Perceptions</i>	<i>TotalTender</i>	<i>SetHeaderTrailer</i>	<i>CloseDNFH</i>	<i>SendFirstIVA</i>	<i>NextIVATransmission</i>
<i>SetHeaderTrailer (1)</i>	●	●	●	●											
<i>SetEmbarkNumber</i>	●	●	●	●											
<i>SetCustomerData</i>	●	●	●	●											
<i>OpenDNFH</i>					●	●									
<i>PrintFiscalText</i>					●	●									
<i>PrintLineItem</i>					●	●	●	●	●	●	●	●	●	●	●
<i>LastItemDiscount</i>					●	●		●	●	●	●	●	●	●	●
<i>ReturnRecharge</i>								●	●	●	●	●	●	●	●
<i>GeneralDiscount</i>									●	●	●	●	●	●	●
<i>Perceptions</i>										●	●	●	●	●	●
<i>TotalTender</i>											●	●	●		
<i>SetHeaderTrailer (2)</i>												●	●		
<i>CloseDNFH</i>													●		
<i>SendFirstIVA</i>														●	
<i>NextIVATransmission</i>															●

(1) Las líneas de encabezado deben cargarse en la memoria de trabajo antes de abrir el comprobante.
El comando de apertura del comprobante es quien las imprime.

(2) Las líneas de pie deben cargarse en la memoria de trabajo antes de cerrar el comprobante.
El comando de cierre del comprobante es quien las imprime.

Luego del comando *SetHeaderTrailer*, una vez abierto el comprobante y antes de enviar el comando de cierre del mismo, se admite el envío de cualquier comando válido en el comprobante.



La impresión de leyendas en el encabezado de los comprobantes fiscales es opcional.

Si los textos cargados en la memoria de trabajo de la **IFH 1G** no se eliminan luego de cerrar el comprobante, serán impresos en los futuros comprobantes que se emitan.

La referencia a otro comprobante es obligatoria. En el encabezado del comprobante se imprimirá la leyenda “*Comprobante Original Nº*” precediendo a la información suministrada mediante este comando.

Si bien la **IFH 1G** no restringe el uso del comando de impresión de percepciones, éste se encuentra disponible para aquellos casos donde el emisor (contribuyente) es un agente de retención.

Los comandos de desglose de impuestos no causan impresión, son sólo de consulta. La discriminación de impuestos, de corresponder, es impresa por el comando de cierre del comprobante.

(DF) Recibos Fiscales

La tabla que se muestra a continuación y el diagrama de flujo de la página siguiente resultan válidos para: Recibos Fiscales 'A', 'B'/'C'.

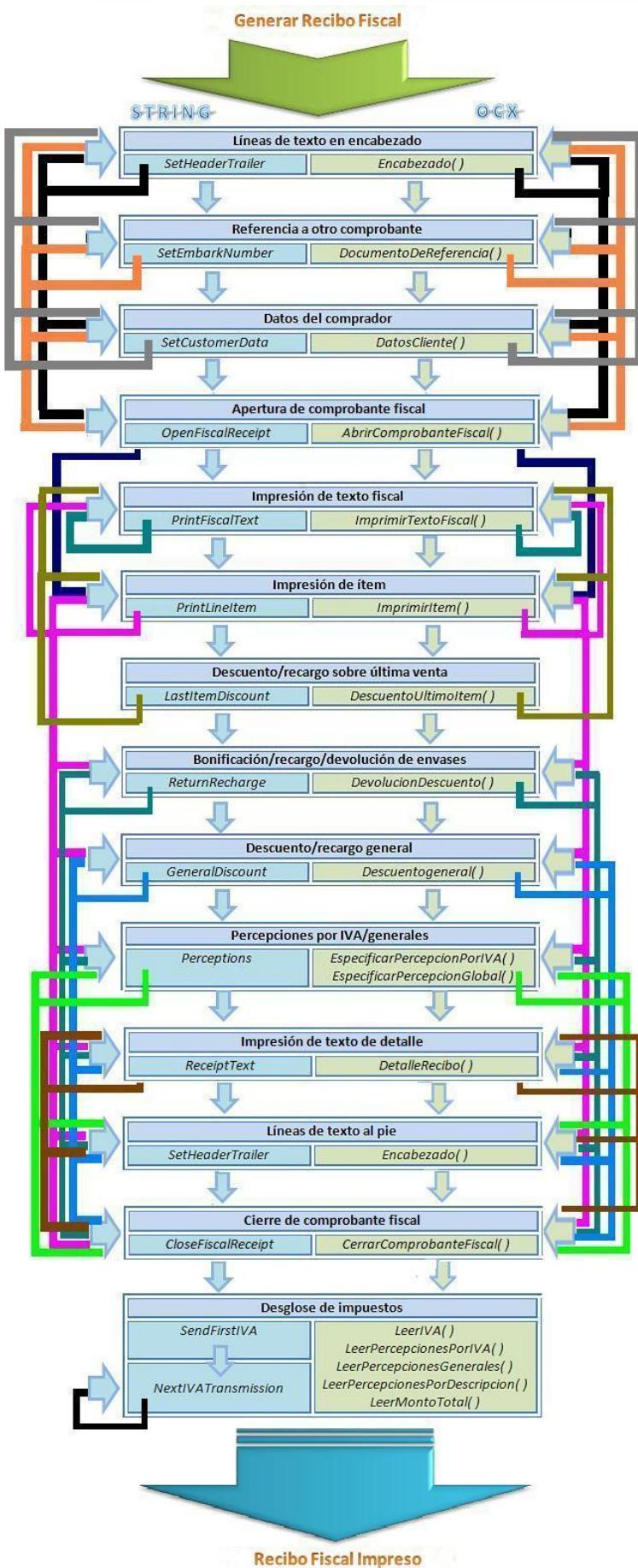
	<i>SetHeaderTrailer</i>	<i>SetEmbarkNumber</i>	<i>SetCustomerData</i>	<i>OpenFiscalReceipt</i>	<i>PrintFiscalText</i>	<i>PrintLineItem</i>	<i>LastItemDiscount</i>	<i>ReturnRecharge</i>	<i>GeneralDiscount</i>	<i>Perceptions</i>	<i>ReceiptText</i>	<i>SetHeaderTrailer</i>	<i>CloseFiscalReceipt</i>	<i>SendFirstIVA</i>	<i>NextIVATransmission</i>
<i>SetHeaderTrailer</i> (1)	•	•	•	•											
<i>SetEmbarkNumber</i>	•	•	•	•											
<i>SetCustomerData</i>	•	•	•	•											
<i>OpenFiscalReceipt</i>					•	•									
<i>PrintFiscalText</i>					•	•									
<i>PrintLineItem</i>					•	•	•	•	•	•	•	•			
<i>LastItemDiscount</i>					•	•		•	•	•	•	•	•		
<i>ReturnRecharge</i>								•	•	•	•	•	•		
<i>GeneralDiscount</i>									•	•	•	•	•		
<i>Perceptions</i>										•	•	•	•		
<i>ReceiptText</i>											•	•	•		
<i>SetHeaderTrailer</i> (2)												•	•		
<i>CloseFiscalReceipt</i>													•		
<i>SendFirstIVA</i>														•	
<i>NextIVATransmission</i>															•

(1) Las líneas de encabezado deben cargarse en la memoria de trabajo antes de abrir el comprobante. El comando de apertura del comprobante es quien las imprime.

(2) Las líneas de pie deben cargarse en la memoria de trabajo antes de cerrar el comprobante. El comando de cierre del comprobante es quien las imprime.

Luego del comando *SetHeaderTrailer*, una vez abierto el comprobante y antes de enviar el comando de cierre del mismo, se admite el envío de cualquier comando válido en el comprobante.

Una vez abierto el comprobante, todos los comandos que se envíen en forma previa al comando *ReceiptText* no provocan impresión en un recibo fiscal; tienen por finalidad generar el monto final del recibo como texto. Luego de las líneas de detalle, y al enviar el comando de cierre del comprobante, se imprimirá el cuerpo del recibo y su cierre.



La impresión de leyendas en el encabezado de los recibos fiscales es opcional.

Si los textos cargados en la memoria de trabajo de la **IFH 1G** no se eliminan luego de cerrar el comprobante, serán impresos en los futuros comprobantes que se emitan.

La referencia a otro comprobante es opcional. De utilizarse, en el encabezado del comprobante se imprimirá la leyenda “*Remito Nº :*” precediendo a la información suministrada mediante este comando.

Los datos del comprador son obligatorios en recibos ‘A’, y opcionales en recibos ‘B’/‘C’.

Para superar los \$1000 (pesos un mil) en recibos ‘B’/‘C’ a contribuyentes consumidores finales se requieren en forma obligatoria los datos del comprador.

NOTA:

Mediante la RG AFIP Nº 3666/14, vigente desde el 4 de Setiembre de 2014, se suspende temporalmente la aplicación del tope de \$1000 (pesos un mil) como monto máximo para Tiques y comprobantes B a consumidores finales sin nombrar.

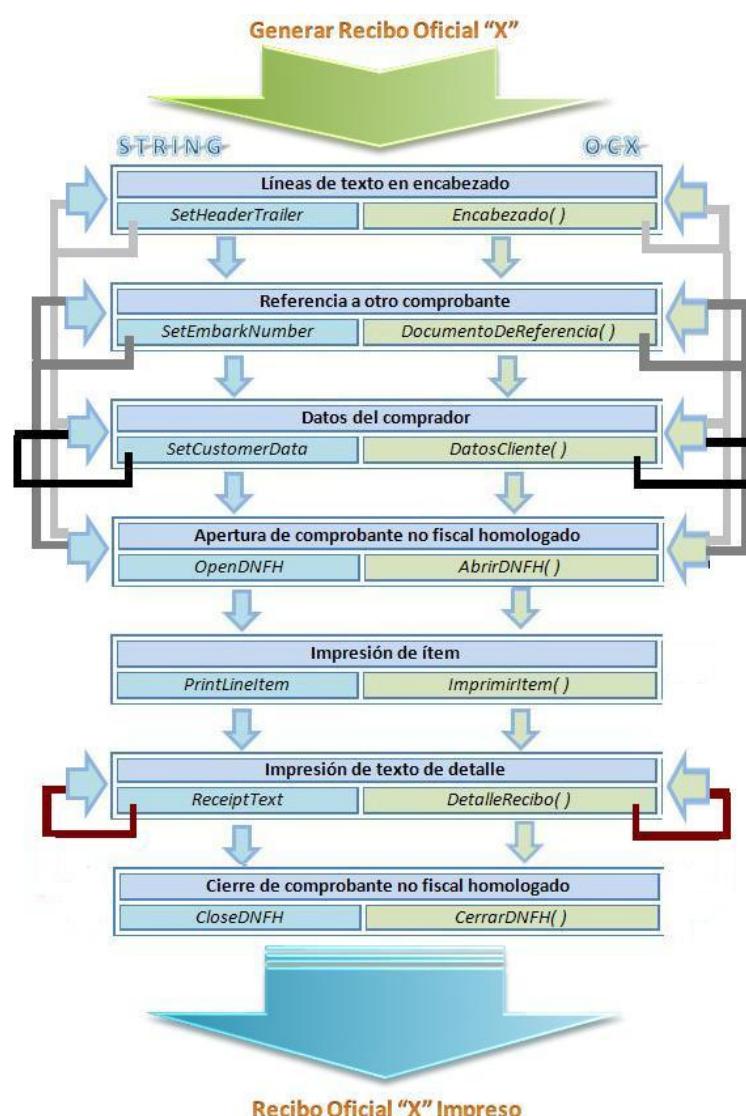
Por suspensión temporal se entiende: hasta tanto entre en vigencia lo previsto en los párrafos segundo y tercero del Artículo 3º de la RG AFIP Nº 3561/13.

Si bien la **Impresora Fiscal HASAR** no restringe el uso del comando de impresión de percepciones, éste se encuentra disponible para aquellos casos donde el emisor (contribuyente) es un agente de retención.

Los comandos de desglose de impuestos no causan impresión, son sólo de consulta. La discriminación de impuestos, de corresponder, es impresa por el comando de cierre del comprobante.

(DNFH) Recibos 'X'

	<i>SetHeaderTrailer</i>	<i>SetEmbarkNumber</i>	<i>SetCustomerData</i>	<i>OpenDNFH</i>	<i>PrintLineItem</i>	<i>ReceiptText</i>	<i>CloseDNFH</i>
<i>SetHeaderTrailer</i>	•	•	•	•			
<i>SetEmbarkNumber</i>	•	•	•	•			
<i>SetCustomerData</i>	•	•	•	•			
<i>OpenDNFH</i>					•		
<i>PrintLineItem</i>						•	
<i>ReceiptText</i>						•	•
<i>CloseDNFH</i>							



La impresión de leyendas en el encabezado de los Recibos 'X' es opcional.

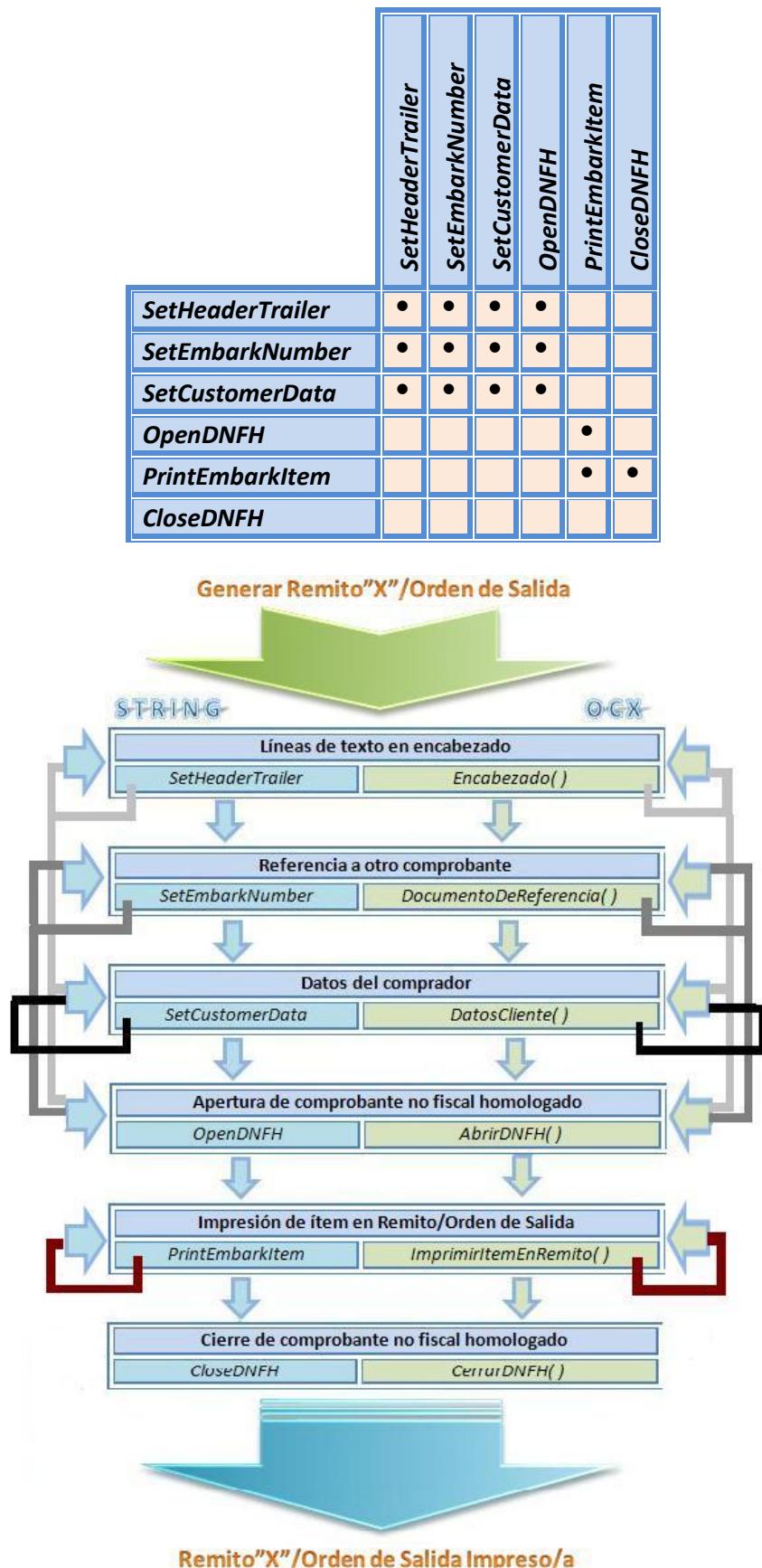
Si los textos cargados en la memoria de trabajo de la IFH 1G no se eliminan luego de cerrar el comprobante, serán impresos en los futuros comprobantes que se emitan.

La referencia a otro comprobante es opcional. De utilizarse, en el encabezado del comprobante se imprimirá la leyenda "Comprobante Original Nº :" precediendo a la información suministrada mediante este comando.

Para generar el monto total del Recibo 'X' basta con enviar una sola vez el comando de impresión de ítem indicando cantidad '1', monto total del recibo, e impuestos en cero.

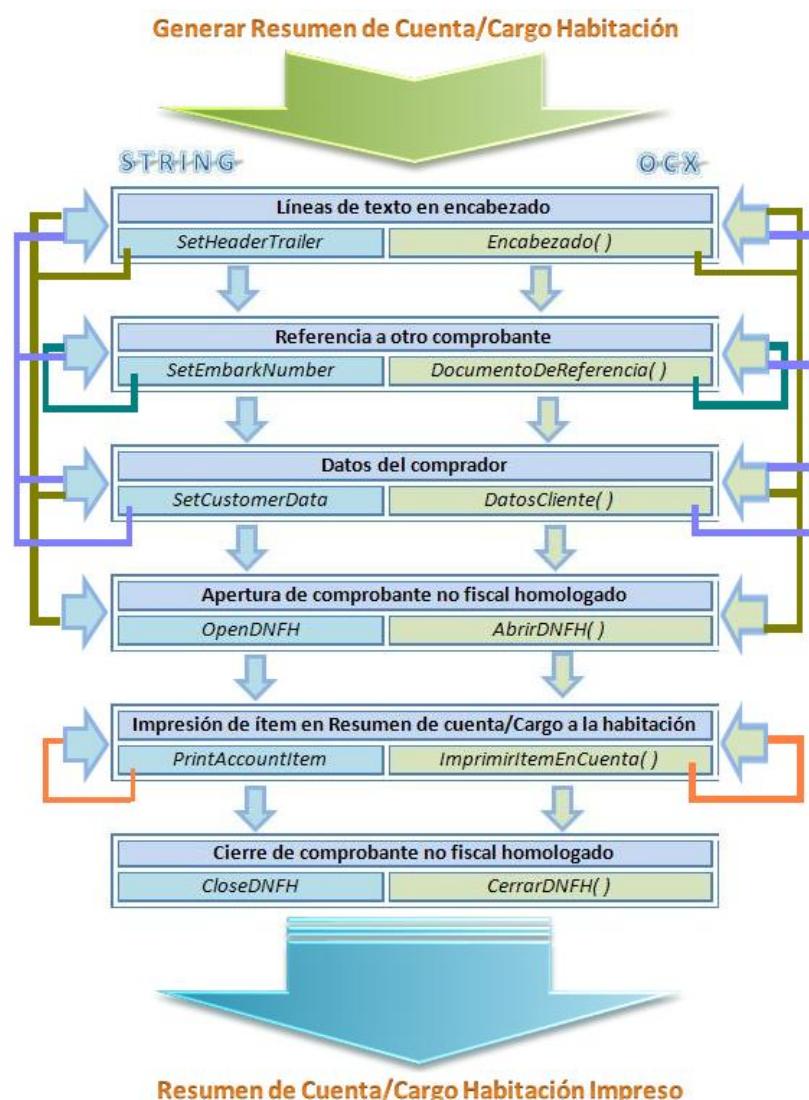
Los Recibos 'X' no provocan acumulaciones de montos.

(DNFH) Remito 'X' / Orden de Salida



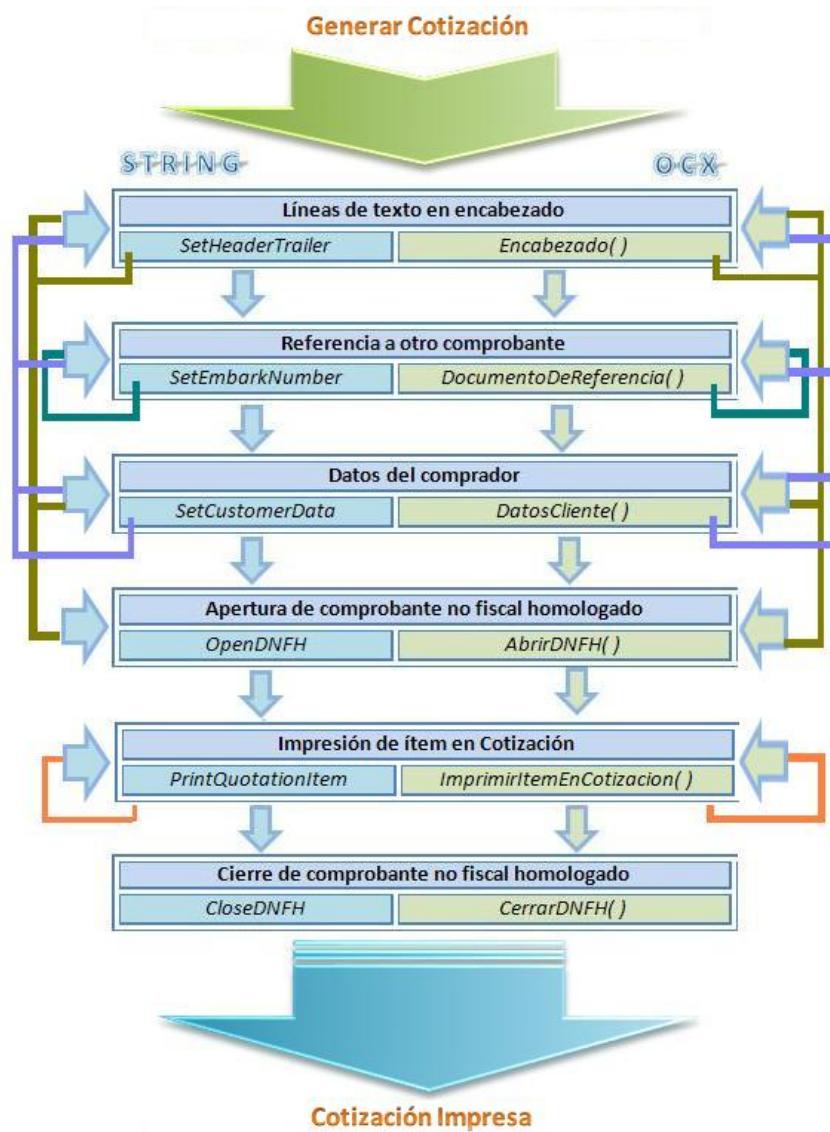
(DNFH) Resumen de Cuenta / Cargo a la Habitación

	<i>SetHeaderTrailer</i>	<i>SetEmbarkNumber</i>	<i>SetCustomerData</i>	<i>OpenDNFH</i>	<i>PrintAccountItem</i>	<i>CloseDNFH</i>
<i>SetHeaderTrailer</i>	•	•	•	•		
<i>SetEmbarkNumber</i>	•	•	•	•		
<i>SetCustomerData</i>	•	•	•	•		
<i>OpenDNFH</i>					•	
<i>PrintAccountItem</i>					•	•
<i>CloseDNFH</i>						



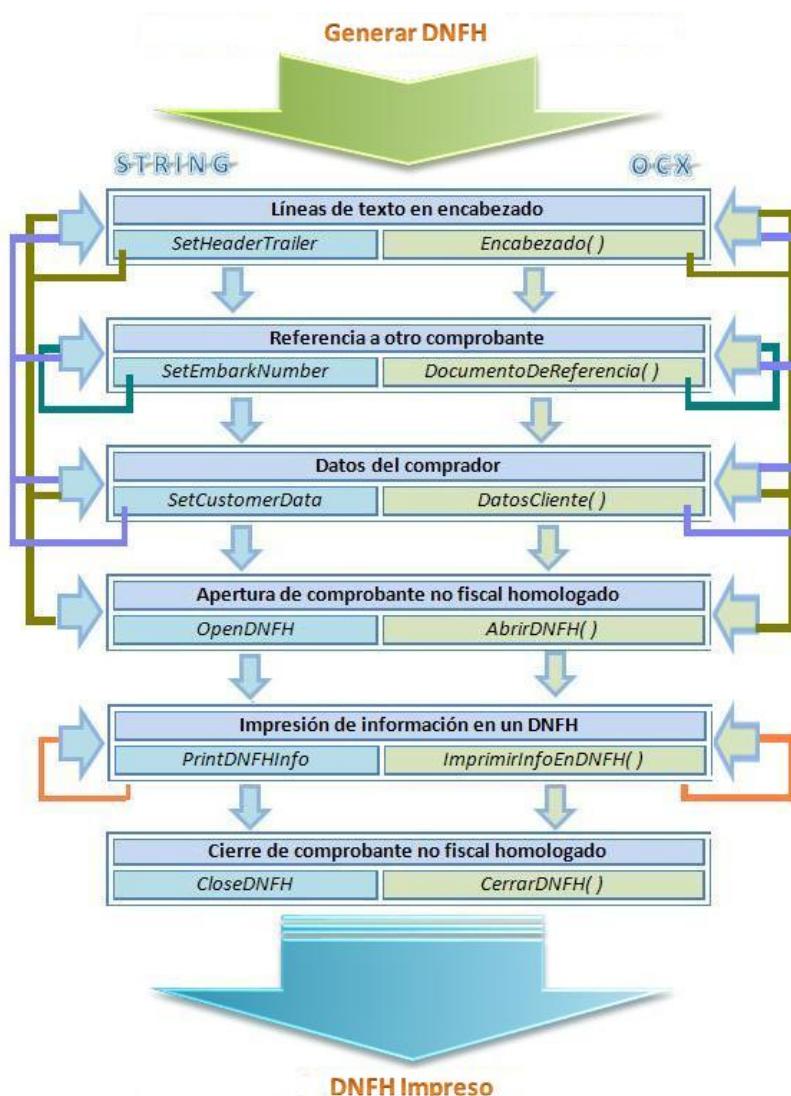
(DNFH) Cotización

	<i>SetHeaderTrailer</i>	<i>SetEmbarkNumber</i>	<i>SetCustomerData</i>	<i>OpenDNFH</i>	<i>PrintQuotationItem</i>	<i>CloseDNFH</i>
<i>SetHeaderTrailer</i>	•	•	•	•		
<i>SetEmbarkNumber</i>	•	•	•	•		
<i>SetCustomerData</i>	•	•	•	•		
<i>OpenDNFH</i>					•	
<i>PrintQuotationItem</i>					•	•
<i>CloseDNFH</i>						



(DNFH) En General

	<i>SetHeaderTrailer</i>	<i>SetEmbarkNumber</i>	<i>SetCustomerData</i>	<i>OpenDNFH</i>	<i>PrintDNFHInfo</i>	<i>CloseDNFH</i>
<i>SetHeaderTrailer</i>	•	•	•	•		
<i>SetEmbarkNumber</i>	•	•	•	•		
<i>SetCustomerData</i>	•	•	•	•		
<i>OpenDNFH</i>					•	
<i>PrintDNFHInfo</i>					•	•
<i>CloseDNFH</i>						



Válido para los siguientes comprobantes no fiscales homologados:

Cláusulas de crédito en cuotas fijas,
Cláusulas de seguro de desempleo,
Pagaré,
Póliza de seguro de garantía complementaria,
Recordatorio,
Solicitud de crédito,
Comunicación con clientes,
Ofrecimiento de crédito en efectivo,
Ofrecimiento de tarjeta de crédito,
Minuta de crédito,
Ofrecimiento de pasaporte,
Renovación de crédito,
Adelanto de remuneración,
Solicitud de tarjeta de crédito,
Solicitud de clave de tarjeta,
Rescate de mercadería,
Ingresos y egresos internos de sucursal.

Comandos de inicialización, baja y configuración

	Cierre Z Previo	Dentro de Comprobante	Fuera de Comprobante
String: InitEpromFiscal OCX:	•		•
String: KillEpromFiscal OCX: BajaFiscal()	•		•
String: ConfigureControllerByBlock OCX: ConfigurarControladorPorBloque()	•		•
String: GeneralConfiguration OCX: ConfigurarControladorCompleto()	•		•
String: ConfigureControllerByOne OCX: ConfigurarControlador()			•
String: ChangeIVAResponsability OCX: CambiarResponsabilidadIVA()	•		•
String: ChangeIBNumber OCX: CambiarCodigoIngresosBrutos()	•		•
String: ChangeStartDate OCX: CambiarFechaInicioActividades()	•		•
String: StoreLogoData OCX: CargarLogo()	•		•
String: ResetLogoData OCX: CargarLogo()	•		•
String: SetComSpeed OCX: CambiarVelocidad()			•
String: SetConfigPage OCX: Interlineado() ApagarIndicadorDeTambor()			•

Comandos de diagnóstico y consulta

	Cierre Z Previo	Dentro de Comprobante	Fuera de Comprobante
String: StatusRequest OCX: PedidoDeStatus()		•	•
String: StatPRN OCX: EnviarStatPrn()		•	•
String: GetConfigurationData OCX: ObtenerDatosDeConfiguracion() ObtenerConfiguracion()		•	•
String: GetGeneralConfigurationData OCX: ObtenerConfiguracionCompleta() ObtenerConfiguracion()		•	•
String: GetInitData OCX: ObtenerDatosDeInicializacion()		•	•
String: GetPrinterVersion OCX: ObtenerVersionDelControlador()		•	•
String: GetConfigPage OCX: Interlineado()		•	•
String: PrinterInternalStatus OCX: EstadoInternolImpresor()		•	•

Comandos de control fiscal

	Cierre Z Previo	Dentro de Comprobante	Fuera de Comprobante
String: HistoryCapacity OCX: CapacidadRestante()		•	•
String: DailyClose OCX: ReporteZ() ReporteX()			•
String: DailyCloseByDate OCX: ReporteZPorFechas()			•
String: DailyCloseByNumber OCX: ReporteZPorNumeros()			•
String: GetDailyReport OCX: ReporteZIndividualPorNumero() ReporteZIndividualPorFecha()		•	•
String: GetWorkingMemory OCX: LeerMemoriaDetrabajo()		•	•
String: SendFirstIVA OCX: LeerIVA() LeerPercepcionesPorIVA() LeerPercepcionesGenerales() LeerMontoTotal() LeerPercepcionesPorDescripcion()		•	•
String: NextIVATransmission OCX: LeerIVA() LeerPercepcionesPorIVA() LeerPercepcionesGenerales() LeerMontoTotal() LeerPercepcionesPorDescripcion()		•	•
String: GetLastCAI OCX: ObtenerCAIUltimoDocumentoA()		•	•

Comandos de uso general

	Cierre Z Previo	Dentro de Comprobante	Fuera de Comprobante
String: Subtotal OCX: Subtotal()		•	
String: Cancel OCX: TratarDeCancelarTodo() CancelarComprobanteFiscal()		•	
String: Reprint OCX: ReimprimirComprobante()			•
String: SetDateTime OCX: FechaHoraFiscal()	•		•
String: GetDateTime OCX: FechaHoraFiscal()		•	•
String: SetFantasyName OCX: NombreDeFantasia()		•	•
String: GetFantasyName OCX: NombreDeFantasia()		•	•
String: SetHeaderTrailer OCX: Encabezado()		•	•
String: GetHeaderTrailer OCX: Encabezado()		•	•
String: SetCustomerData OCX: DatosCliente()			•
String: SetEmbarkNumber OCX: DocumentoDeReferencia()			•
String: GetEmbarkNumber OCX: DocumentoDeReferencia()		•	•
String: ChangeBussinessStartupDate OCX: CambiarFechaInicioActividades()	•		•



Mayo 08 , 2015 – Rev. 002

COPYRIGHT © 1997/2015 - Compañía 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 datos 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 **Compañía HASAR SAIC**.