

Controladores fiscales 2G



CONTENIDO

HasarArgentina.ocx.....	4
Introducción.....	4
Conociendo al objeto “HasarArgentina.ocx”.....	5
Los miembros de “HasarArgentina.ocx”.....	5
Abortar().....	6
AbrirCajonDinero().....	6
AbrirDocumento().....	7
AnularItem().....	8
ArchivoRegistro().....	10
AvanzarPapelAmbasEstaciones().....	10
AvanzarPapelEstacionAuditoria().....	11
AvanzarPapelEstacionTique().....	11
CambiarCategoriaIVA().....	12
CambiarFechaInicioActividades().....	12
CambiarInscripIIBB().....	13
CambiarLogoUsuario().....	14
CambiarVelocidadPuerto().....	14
Cancelar().....	15
CargarBeneficiario().....	15
CargarCodigoBarras().....	16
CargarDatosCliente().....	17
CargarDocumentoAsociado().....	19
CargarLogoAdicional().....	20
CargarLogoEmisor().....	22
CargarTransportista().....	23
CerrarDocumento().....	24
CerrarJornadaFiscal().....	25
Conectar().....	32
ConfigurarFechaHora().....	33
ConfigurarImpresoraFiscal().....	34
ConfigurarRed().....	35
ConfigurarServidorCorreo().....	36
ConfigurarZetaBorrable().....	37
ConfigurarZona().....	37
ConsultarAcumuladosCierreZeta().....	39
ConsultarAcumuladosComprobante().....	54
ConsultarAcumuladosMemoriaDeTrabajo().....	71
ConsultarCantidadIvasOtrosTributos().....	85
ConsultarCapacidadesImpresoraFiscal().....	86
ConsultarCapacidadZetas().....	88
ConsultarConfiguracionImpresoraFiscal().....	94
ConsultarConfiguracionRed().....	96
ConsultarConfiguracionServidorCorreo().....	97
ConsultarDatosInicializacion().....	98

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

ConsultarDocumentoAsociado()	100
ConsultarEstado()	102
ConsultarEstadoECR()	108
ConsultarFechaHora()	109
ConsultarInformacionAritmetica()	110
ConsultarInformacionBloqueo()	113
ConsultarParametroHardware()	114
ConsultarSubtotal()	116
ConsultarUltimaConexion()	118
ConsultarUltimoError()	119
ConsultarVersion()	120
ConsultarVigenciaCertificados()	122
ConsultarZetaBorrable()	124
ConsultarZona()	125
ContinuarConsultaAcumulados()	127
CopiarComprobante()	147
CrearCodigoQR()	149
EliminarCodigoQR()	150
EliminarLogoAdicional()	150
EliminarLogoEmisor()	151
EnviarDocumentoCorreo()	151
ImprimirAjuste()	153
ImprimirAnticipoBonificacionEnvases()	155
ImprimirConceptoRecibo()	158
ImprimirDescuentoItem()	158
ImprimirItem()	160
ImprimirOtrosTributos()	164
ImprimirPago()	165
ImprimirTextoFiscal()	168
ImprimirTextoGenerico()	169
ObtenerPrimerBloqueDocumento()	174
ObtenerPrimerBloqueLog()	177
ObtenerPrimerBloqueReporteElectronico()	178
ObtenerRangoFechasPorZetas()	179
ObtenerRangoZetasPorFechas()	180
ObtenerRevision()	181
ObtenerSiguieteBloqueAuditoria()	181
ObtenerSiguieteBloqueDocumento()	182
ObtenerSiguieteBloqueLog()	183
ObtenerSiguieteBloqueReporteElectronico()	184
ObtenerUltimoEstadoFiscal()	185
ObtenerUltimoEstadoImpresora()	187
ObtenerVersionProtocolo()	189
OperacionCajero()	189
PedirReimpresion()	190
ReportarZetasPorFecha()	191
ReportarZetasPorNumeroZeta()	192

Introducción

La RG AFIP N° 3561/13 (emisión de comprobantes mediante el uso de controladores fiscales), con sus modificaciones y ampliatorias, da nacimiento a una nueva generación de equipos fiscales en la República Argentina. Esta nueva vida origina una renovación de las herramientas para los programadores de sistemas de facturación. Dichas herramientas satisfacen las necesidades de un correcto diálogo con esta reciente generación (2G) de impresoras fiscales HASAR, una fácil incorporación a las tecnologías de desarrollo de software existentes y, como desde el primer día de la aparición de las impresoras fiscales HASAR en el mercado, se acompaña el uso de este OCX 2G (*HasarArgentina.ocx*) con un servicio de atención de consultas especializado.

HasarArgentina.ocx es un objeto pensado para su incorporación a proyectos de desarrollo de software sobre Windows® 32/64 bits, utilizando como lenguaje Visual Basic 6.00 y/o similares. También se puede incorporar sin inconvenientes a proyectos sobre “.Net”.

A lo largo de este manual del usuario se emplearán, para los ejemplos, fragmentos de código escritos en Visual Basic, compilando en Visual Studio 2017. Se trata ejemplos orientativos y simples, para que su portación a otros lenguajes visuales no resulte traumática. Se describirán los miembros que componen el OCX 2G, y cual es el fin de su existencia. La lectura de este manual del usuario puede, y debería, complementarse con la lectura del manual de comandos de la impresora fiscal 2G (segunda generación, o nueva tecnología).

Entre las características que definen a una impresora fiscal 2G, se encuentra el protocolo de comandos fiscales y respuestas fiscales soportado, considerando además los campos que componen a tales comandos fiscales y respuestas fiscales. Este protocolo es identificado por un número (que se puede consultar), el cual debe ser igual o menor al protocolo de comandos fiscales y respuestas fiscales soportados por el OCX 2G que nos ocupa (protocolo que también es posible obtener). Si el protocolo de la impresora fiscal 2G fuese mayor al soportado por el OCX 2G, esta herramienta deberá ser actualizada. La comparación de números de protocolo le permite al programador de sistemas de facturación garantizar que el OCX 2G utilizado es compatible con la impresora fiscal 2G con la cual va a dialogar.

El intercambio de comandos fiscales y respuestas fiscales (como objetos XML) entre el OCX 2G y la impresora fiscal 2G se realiza mediante protocolo HTTP.

Muchas de las respuestas fiscales entregadas por los métodos del OCX 2G son estructuras de información, donde cada dato debe ser extraído de un campo específico de tal estructura.

Ejemplo:

```
Dim resp As HasarArgentina.RespuestaConsultarConfiguracionRed

resp = hasar.ConsultarConfiguracionRed()
Debug.Print("Dirección IP =[" & resp.DireccionIP & "]")
Debug.Print("Máscara      =[" & resp.Mascara & "]")
Debug.Print("Gateway      =[" & resp.Gateway & "]")
```

Estas estructuras de información son tipos de datos provistos por el OCX 2G. Si el lenguaje utilizado no soportara este tipo de datos, se deberá recurrir al uso del método 'ObtenerCampoRespuesta()' para obtener la información de interés.

Conociendo al objeto “HasarArgentina.ocx”

Los dos primeros elementos a identificar en el OCX 2G son el espacio de nombres (Namespace) y la clase (Class) que lo definen como un objeto:

```
Namespace    =    HasarArgentina
Class        =    ImpresoraFiscalRG3561
```

Una vez incorporado el OCX 2G al proyecto, incrustándolo en un formulario (o creándolo como un nuevo objeto), sólo resta tener en cuenta que puede lanzar eventos y/o excepciones que deben ser atrapados y tratados adecuadamente por el sistema de facturación.

Las excepciones se pueden atrapar, en el caso de Visual Basic, enmarcando el código de interés entre las sentencias Try/Catch/End Try.

Ejemplo:

```
Try
  hasar.Conectar("192.168.1.2", , 15000, 15000)
Catch
  Debug.Print(vbCrLf & "Código de error: " & Err.Number & vbCrLf & "Mensaje de error: " & _
    Err.Description)
End Try
```

Para atrapar eventos, en Visual Basic, se deben crear rutinas privadas cuyo nombre respete la sintaxis: “Clase_NombreEvento”.

Ejemplo:

```
Private WithEvents evhasar As HasarArgentina.ImpresoraFiscalRG3561

Private Sub evhasar_ComandoProcesado() Handles evhasar.ComandoProcesado
  Debug.Print("Se recibió la respuesta al comando enviado ...")
End Sub
```

Otros lenguajes de programación pueden proveer mecanismos similares para el tratamiento de las excepciones y los eventos que puede lanzar el OCX 2G.

Los miembros de “HasarArgentina.ocx”

Incluido el OCX 2G en un proyecto, y habiéndole dado un nombre (por ejemplo, “hasar”), el programador de sistemas de facturación tiene a su disposición toda una serie de miembros del objeto, para poder dialogar correcta-

mente con la impresora fiscal 2G.

Las dos primeras consideraciones de importancia al incorporar el OCX 2G al sistema de facturación, deberían ser:

- Definir con que impresora fiscal 2G se va a dialogar.
- Verificar compatibilidad de protocolo.

Ejemplo:

```
hasar.Conectar("192.168.1.2", , 15000, 15000)  
Debug.Print("CF 2G Protocolo =[" & resp.VersionProtocolo & "]")  
Debug.Print("OCX 2G Protocolo = [" & hasar.ObtenerVersionProtocolo & "]")
```

Se asume que "192.168.1.2" es la dirección IP configurada en la impresora fiscal 2G (que la identifica como un nodo más en la red). Se requiere invocar a 'Conectar()' antes de enviar a la impresora fiscal 2G el primer comando fiscal (y siguientes) de la jornada fiscal/comercial.

Si bien en el ejemplo anterior no se muestra, resuelto satisfactoriamente el tema de la conectividad, y verificada la compatibilidad, el programador del sistema de facturación puede pasar a la emisión de comprobantes, y a la descarga de información para la AFIP.

Abortar()

Permite interrumpir los reintentos constantes que efectúa el OCX 2G cuando se ha producido un error entre el envío de un comando fiscal a la impresora fiscal 2G, y la recepción de la respuesta fiscal al comando fiscal enviado. Por ejemplo, cuando la impresora fiscal 2G se encuentra offline.

Definido internamente como:

```
Abortar();
```

Ejemplo:

```
hasar.Abortar()
```

AbrirCajonDinero()

Comando fiscal disponible, únicamente, en modelos de impresoras fiscales 2G con soporte para conexión de cajón.

Permite enviar a la impresora fiscal 2G el comando fiscal de apertura del cajón de dinero conectado al equipo. El envío de este comando fiscal tiene sentido si la impresora fiscal 2G soporta conexión de cajón de dinero, y éste se encuentra físicamente conectado.

Definido internamente como:

```
AbrirCajonDinero();
```

Ejemplo:

```
hasar.AbrirCajonDinero()
```

AbrirDocumento()

Permite enviar a la impresora fiscal 2G el comando fiscal de apertura de un comprobante, el cual se identifica mediante alguna de las constantes pertenecientes al enum *TiposComprobante*.

Definido internamente como:

```
AbrirDocumento(  
    [in] TiposComprobante CodigoComprobante,  
    [out, retval] RespuestaAbrirDocumento* );
```

CodigoComprobante

Permite especificar el tipo de comprobante que se desea abrir. Debe ser alguna de las constantes pertenecientes al enum *TiposComprobante*.

***	<i>ComprobanteDonacion</i>	907	***	<i>ReciboM</i>	54
*	<i>FacturaA</i>	1	***	<i>ReciboX</i>	902
*	<i>FacturaB</i>	6	***	<i>RemitoR</i>	91
*	<i>FacturaC</i>	11	***	<i>RemitoX</i>	901
*	<i>FacturaM</i>	51	**	<i>Tique</i>	83
***	<i>Generico</i>	910	**	<i>TiqueFacturaA</i>	81
*	<i>NotaDeCreditoA</i>	3	**	<i>TiqueFacturaB</i>	82
*	<i>NotaDeCreditoB</i>	8	**	<i>TiqueFacturaC</i>	111
*	<i>NotaDeCreditoC</i>	13	**	<i>TiqueFacturaM</i>	118
*	<i>NotaDeCreditoM</i>	53	**	<i>TiqueNotaCredito</i>	110
*	<i>NotaDeDebitoA</i>	2	**	<i>TiqueNotaCreditoA</i>	112
*	<i>NotaDeDebitoB</i>	7	**	<i>TiqueNotaCreditoB</i>	113
*	<i>NotaDeDebitoC</i>	12	**	<i>TiqueNotaCreditoC</i>	114
*	<i>NotaDeDebitoM</i>	52	**	<i>TiqueNotaCreditoM</i>	119
***	<i>PresupuestoX</i>	903	**	<i>TiqueNotaDebitoA</i>	115
***	<i>ReciboA</i>	4	**	<i>TiqueNotaDebitoB</i>	116
***	<i>ReciboB</i>	9	**	<i>TiqueNotaDebitoC</i>	117
***	<i>ReciboC</i>	15	**	<i>TiqueNotaDebitoM</i>	120

*	Disponibles en impresoras fiscales 2G que soporten impresión sobre formularios continuos u hojas sueltas.
**	Disponibles en impresoras fiscales 2G que soporten impresión sobre rollos de papel continuo.
***	Disponibles en todos los modelos de impresoras fiscales 2G.

Si bien hay más constantes declaradas, el uso de las mismas carece de sentido debido a que se trata de documentos que no pueden ser abiertos por el sistema de facturación mediante la utilización del comando fiscal generado por 'AbrirDocumento()'.

Ejemplo:

```
Dim respabrir As HasarArgentina.RespuestaAbrirDocumento

respabrir = hasar.AbrirDocumento(HasarArgentina.TiposComprobante.TiqueFacturaA)
Debug.Print("Tique Factura 'A' Nº =[ " & respabrir.NumeroComprobante & "] abierto !")
```

Tener presente que algunos tipos de comprobantes pueden ser abiertos, únicamente, si en forma previa se envían a la impresora fiscal 2G determinados comandos fiscales obligatorios. Por ejemplo, el generado por el método 'CargarDatosCliente()'.

La respuesta fiscal al comando fiscal generado por 'AbrirDocumento()' es la estructura *RespuestaAbrirDocumento*, definida internamente como:

```
typedef
struct tagRespuestaAbrirDocumento {
    [helpstring("NumeroComprobante")]
}
    long NumeroComprobante;
    [helpstring("IndiceAuditoria")]
}
    long IndiceAuditoria;
} RespuestaAbrirDocumento;
```

<i>NumeroComprobante</i>	Informa el número de comprobante que ha sido abierto.
<i>IndiceAuditoria</i>	Informa el número de orden del comprobante en la CTD (cinta testigo digital, o memoria de auditoría).

AnularItem()

No disponible en los modelos de impresoras fiscales 2G: SMH/PT-1000F v01.00, y SMH/PT-250F v01.00.
 No disponible en controladores fiscales 2G con protocolo = '0'.

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

Permite enviar a la impresora fiscal 2G el comando fiscal de anulación de un ítem, previamente impreso en el comprobante en curso de emisión. No se permite la anulación de ítems correspondientes a comprobantes que ya han sido cerrados.

Definido internamente como:

```
AnularItem(  
    [in] long IndiceAuditoria,  
    [out, retval] RespuestaAnularItem* );
```

IndiceAuditoria

Permite especificar el índice al registro de la CTD (cinta testigo digital, o memoria de auditoría) donde se encuentra almacenado el ítem que se desea anular.

Cada vez que se envía al controlador fiscal 2G el comando fiscal de impresión de ítem, si éste es aceptado, la respuesta fiscal incluye un campo donde se informa el "orden" (o índice) en que la operación es registrada en la CTD (cinta testigo digital, o memoria de auditoría), dentro del comprobante en curso. El sistema de facturación puede utilizar este índice como referencia al ítem, si es que en una operación futura (dentro del mismo comprobante) se lo desea anular haciendo uso del comando fiscal generado por '*AnularItem()*'.

No permite la anulación de ítems negativos. Es decir, no se puede anular una anulación.

Ejemplo:

```
Dim resp As HasarArgentina.RespuestaImprimirItem  
Dim resp2 As HasarArgentina.RespuestaAnularItem  
  
resp = hasar.ImprimirItem("Pines ...", 3.0, 10.0, HasarArgentina.CondicionesIVA.Gravado, 10.5, _  
    HasarArgentina.ModosDeMonto.ModoSumaMonto, _  
    HasarArgentina.ModosDeImpuestosInternos.IIVariablePorcentual, 10.0, _  
    HasarArgentina.ModosDeDisplay.DisplayNo, HasarArgentina.ModosDePrecio.ModoPrecioTotal, _  
    1,, "780201", HasarArgentina.UnidadesMedida.Unidad)  
  
Debug.Print("Índice al ítem = " & resp.IndiceAuditoria)  
  
resp2 = hasar.AnularItem (resp.IndiceAuditoria )  
Debug.Print("Índice a la anulación = " & resp2.IndiceAuditoria)
```

La respuesta fiscal al comando fiscal generado por '*AnularItem()*' es la estructura *RespuestaAnularItem*, definida internamente como:

```
typedef  
struct tagRespuestaAnularItem {  
    [helpstring("ÍndiceAuditoria")  
]  
    long IndiceAuditoria;  
} RespuestaAnularItem;
```

<i>IndiceAuditoria</i>	Permite especificar el índice al registro de la CTD (cinta testigo digital, o memoria de auditoría) donde se encuentra almacenada la operación de anulación del ítem, dentro del comprobante en curso.
------------------------	--

ArchivoRegistro()

Permite especificarle al OCX 2G el nombre de un archivo (incluyendo, opcionalmente, su ubicación) que creará y utilizará para registrar sus actividades (intercambio de comandos/respuestas fiscales).

Definido internamente como:

```
ArchivoRegistro(  
    [in] BSTR NombreArchivo,  
    [in, optional, defaultvalue(0)] long MaximaLongitud);
```

<i>NombreArchivo</i>	Permite especificar el nombre del archivo donde el OCX 2G dejará registro de toda su actividad del diálogo mantenido con la impresora fiscal 2G. Se puede incluir la ubicación. Por ejemplo: [path\]nombre.ext
<i>MáximaLongitud</i>	Permite especificar el tamaño máximo en bytes que puede alcanzar el archivo. El valor cero indica que se permite crecer indefinidamente al archivo.

El OCX 2G sólo crea el archivo con el nombre que se indique. La ruta hacia él debe existir previamente a su creación. Si no se especifica una ruta, el archivo es creado por defecto en la misma carpeta que el ejecutable de la aplicación.

Si se especifica un tamaño máximo, cada vez que éste es alcanzado el archivo es renombrado y numerado. También se vuelve a crear el archivo con el nombre original, pero vacío. En consecuencia, es posible encontrar los archivos "fiscal.log", "fiscal.log.1", "fiscal.log.2", ... , "fiscal.log.n".

El sistema de facturación es responsable del borrado de estos archivos, y de la oportunidad en que deba hacerse, para liberar espacio en el disco de la PC.

Ejemplo:

```
hasar.ArchivoRegistro("C:\basura\logs\OCX2G.log", 32000)
```

AvanzarPapelAmbasEstaciones()

Permite enviar a la impresora fiscal 2G el comando fiscal de avance de papel, especificando la cantidad de líneas de avance. En impresoras fiscales 2G de una sola estación se comporta como el comando fiscal generado por 'AvanzarPapelEstacionTique()'.
'AvanzarPapelEstacionTique()'

Definido internamente como:

```
AvanzarPapelAmbasEstaciones([in] long CantidadDeLineas);
```

Ejemplo:

```
hasar.AvanzarPapelAmbasEstaciones(5)
```

NOTA:

El comando de avance de papel no tiene efecto en impresiones sobre formularios continuos u hojas sueltas.

AvanzarPapelEstacionAuditoria()

Permite enviar a la impresora fiscal 2G el comando fiscal de avance de papel, especificando la cantidad de líneas de avance. En impresoras fiscales 2G de una sola estación se comporta como el comando fiscal generado por 'AvanzarPapelEstacionTique()'.
'AvanzarPapelEstacionTique()'

Definido internamente como:

```
AvanzarPapelEstacionAuditoria([in] long CantidadDeLineas);
```

Ejemplo:

```
hasar.AvanzarPapelEstacionAuditoria(5)
```

NOTA:

El comando de avance de papel no tiene efecto en impresiones sobre formularios continuos u hojas sueltas.

AvanzarPapelEstacionTique()

Permite enviar a la impresora fiscal 2G el comando fiscal de avance de papel, especificando la cantidad de líneas de avance.

Definido internamente como:

```
AvanzarPapelEstacionTique([in] long CantidadDeLineas);
```

Ejemplo:

```
hasar.AvanzarPapelEstacionTique(5)
```

NOTA:

El comando de avance de papel no tiene efecto en impresiones sobre formularios continuos u hojas sueltas.

CambiarCategoriaIVA()

Permite enviar a la impresora fiscal 2G el comando fiscal de cambio de responsabilidad frente a IVA del emisor, especificando la nueva responsabilidad.

Definido internamente como:

```
CambiarCategoriaIVA([in] TiposDeResponsabilidadesImpresor ResponsabilidadIVA);
```

ResponsabilidadIVA Permite especificar la nueva categoría como contribuyente, del emisor de los comprobantes. Debe ser alguna de las siguientes constantes pertenecientes al enum *TiposDeResponsabilidadesImpresor*.

<i>IFMonotributo</i>	77
<i>IFMonotributoSocial</i>	83
<i>IFNoResponsable</i>	65
<i>IFResponsableExento</i>	69
<i>IFResponsableInscripto</i>	73

Para que el comando fiscal generado por '*CambiarCategoriaIVA()*' sea aceptado por la impresora fiscal 2G, previamente se debe haber enviado el comando fiscal generado por '*CerrarJornadaFiscal()*' -emitir un *Informe Diario de Cierre* (cierre 'Z')-, y no debe haber un comprobante abierto.

Ejemplo:

```
Dim cierre As HasarArgentina.RespuestaCerrarJornadaFiscal
cierre = hasar.CerrarJornadaFiscal(HasarArgentina.TipoReporte.ReporteZ)
hasar.CambiarCategoriaIVA(HasarArgentina.TiposDeResponsabilidadesImpresor.IFMonotributoSocial)
```

Ejecutado el cambio de responsabilidad frente a IVA del emisor, la impresora fiscal 2G emite automáticamente un comprobante del tipo *CambioCategorizacionIVA* (código 952).

CambiarFechaInicioActividades()

Permite enviar a la impresora fiscal 2G el comando fiscal de cambio de fecha de inicio de actividades del emisor. Para que el comando fiscal generado por '*CambiarFechaInicioActividades()*' sea aceptado por la impresora fiscal 2G, previamente se debe haber enviado el comando generado por '*CerrarJornadaFiscal()*' -emitir un *Informe Diario de Cierre* (cierre 'Z')-, y no debe haber un comprobante abierto.

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

Definido internamente como:

```
CambiarFechaInicioActividades([in] DATE Fecha);
```

Fecha | Permite especificar la nueva fecha de inicio de actividades del emisor de los comprobantes.

Ejemplo:

```
Dim cierre As HasarArgentina.RespuestaCerrarJornadaFiscal
cierre = hasar.CerrarJornadaFiscal(HasarArgentina.TipoReporte.ReporteZ)
hasar.CambiarFechaInicioActividades("01/07/2019")
```

Ejecutado el cambio de fecha de inicio de actividades del emisor, la impresora fiscal 2G emite automáticamente un comprobante del tipo *Mensaje del Sistema* (código 923).

CambiarInscripIIBB()

Permite enviar a la impresora fiscal 2G el comando fiscal de cambio de código de inscripción en Ingresos Brutos del emisor. Para que el comando fiscal generado por '*CambiarInscripIIBB()*' sea aceptado por la impresora fiscal 2G, previamente se debe haber enviado el comando fiscal generado por '*CerrarJornadaFiscal()*' -emitir un *Informe Diario de Cierre* (cierre 'Z')-, y no debe haber un comprobante abierto.

Definido internamente como:

```
CambiarInscripIIBB([in] BSTR IngBrutos);
```

IngBrutos | Permite especificar el nuevo código de inscripción en Ingresos Brutos del emisor.

Ejemplo:

```
Dim cierre As HasarArgentina.RespuestaCerrarJornadaFiscal
cierre = hasar.CerrarJornadaFiscal(HasarArgentina.TipoReporte.ReporteZ)
hasar.CambiarInscripIIBB("IIBB-1234567890")
```

Ejecutado el cambio de código de inscripción en Ingresos Brutos del emisor, la impresora fiscal 2G emite automáticamente un comprobante del tipo *CambioInscripcionIngBrutos* (código 953).

CambiarLogoUsuario()

Disponible solamente en impresoras desarrolladas para el GCABA (Gobierno de la Ciudad Autónoma de Buenos Aires).

Permite enviar a la impresora GCABA el comando de cambio de logotipo adicional, el cual se imprime al pie de los comprobantes que emite el organismo oficial que lo utiliza -de corresponder-. El logotipo a seleccionar debe ser alguno de los cargados previamente, en la memoria de trabajo de la impresora GCABA, mediante el uso del comando fiscal generado por '*CargarLogoAdicional()*'.

Definido internamente como:

```
CambiarLogoUsuario([in] long NumeroLogo);
```

<i>NumeroLogo</i>	Permite especificar el número que identifica al nuevo logotipo adicional que se desea dejar como activo. El número que identifica al logotipo adicional debe estar comprendido entre '1' y '10'. Se reserva el número '0' para indicar que no se imprimirá el logotipo adicional hasta que se envíe el comando fiscal generado por ' <i>CambiarLogoUsuario()</i> ' con un valor comprendido entre '1' y '10'.
-------------------	--

Ejemplo:

```
hasar.CambiarLogoUsuario(3)
```

CambiarVelocidadPuerto()

Permite enviar a la impresora fiscal 2G el comando fiscal de cambio de valor de baudrate para la configuración de su puerto RS-232.

Definido internamente como:

```
CambiarVelocidadPuerto([in] Baudios Baudios);
```

<i>Baudios</i>	Permite especificar el nuevo valor de baudrate. Debe ser alguna de las constantes pertenecientes al enum <i>Baudios</i> .								
	<table border="1"><tr><td><i>Baudrate1200</i></td><td>1200</td></tr><tr><td><i>Baudrate2400</i></td><td>2400</td></tr><tr><td><i>Baudrate4800</i></td><td>4800</td></tr><tr><td><i>Baudrate9600</i></td><td>9600</td></tr></table>	<i>Baudrate1200</i>	1200	<i>Baudrate2400</i>	2400	<i>Baudrate4800</i>	4800	<i>Baudrate9600</i>	9600
<i>Baudrate1200</i>	1200								
<i>Baudrate2400</i>	2400								
<i>Baudrate4800</i>	4800								
<i>Baudrate9600</i>	9600								

<i>Baudrate19200</i>	19200
<i>Baudrate38400</i>	38400
<i>Baudrate57600</i>	57600
<i>Baudrate115200</i>	115200

Ejemplo:

```
hasar.CambiarVelocidadPuerto(HasarArgentina.Baudios.Baudrate115200)
```

Cancelar()

Permite enviar a la impresora fiscal 2G el comando fiscal de cancelación de comprobante abierto. El comando fiscal es aceptado sólo si un comprobante se encuentra abierto (es decir, si todavía no se envió el comando fiscal de cierre del comprobante a la impresora fiscal 2G).

Definido internamente como:

```
Cancelar();
```

Cancelar un comprobante significa que no se acumularán los valores generados hasta el momento, pero sí se considerará utilizado el número que lo identifica.

No es posible la cancelación de comprobantes que ya hayan sido cerrados previamente.

Ejemplo:

```
hasar.Cancelar()
```

CargarBeneficiario()

Permite enviar a la impresora fiscal 2G el comando fiscal que especifica los datos del beneficiario, requeridos cuando se va a emitir un comprobante *Donación*. Este comando fiscal es obligatorio y previo al envío del comando fiscal de apertura del comprobante *Donación*.

Definido internamente como:

```
CargarBeneficiario(  
    [in] BSTR RazonSocial,  
    [in] BSTR NumeroDocumento,  
    [in] TiposDeDocumentoCliente TipoDocumento,  
    [in, optional, defaultvalue("")] BSTR Domicilio);
```

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

<i>RazonSocial</i>	Permite especificar la razón social del beneficiario de la donación.																
<i>NumeroDocumento</i>	Permite especificar el número de documento que identifica al beneficiario de la donación. De tratarse de una CUIT/CUIL, deben ser 11 dígitos y sin guiones.																
<i>TipoDocumento</i>	Permite especificar el tipo de documento que identifica al beneficiario de la donación. Debe ser alguna de las constantes pertenecientes al enum <i>TiposDeDocumentoCliente</i> . <table><tr><td><i>TipoNinguno</i></td><td>32</td></tr><tr><td><i>TipoLE</i></td><td>48</td></tr><tr><td><i>TipoLC</i></td><td>49</td></tr><tr><td><i>TipoDNI</i></td><td>50</td></tr><tr><td><i>TipoPasaporte</i></td><td>51</td></tr><tr><td><i>TipoCI</i></td><td>52</td></tr><tr><td><i>TipoCUIT</i></td><td>67</td></tr><tr><td><i>TipoCUIL</i></td><td>76</td></tr></table>	<i>TipoNinguno</i>	32	<i>TipoLE</i>	48	<i>TipoLC</i>	49	<i>TipoDNI</i>	50	<i>TipoPasaporte</i>	51	<i>TipoCI</i>	52	<i>TipoCUIT</i>	67	<i>TipoCUIL</i>	76
<i>TipoNinguno</i>	32																
<i>TipoLE</i>	48																
<i>TipoLC</i>	49																
<i>TipoDNI</i>	50																
<i>TipoPasaporte</i>	51																
<i>TipoCI</i>	52																
<i>TipoCUIT</i>	67																
<i>TipoCUIL</i>	76																
<i>Domicilio</i>	Permite especificar el domicilio del beneficiario de la donación.																

Ejemplo:

```
hasar.CargarBeneficiario("La Escuelita Uno ...", "30124567897", _  
HasarArgentina.TiposDeDocumentoCliente.TipoCUIL, _  
"Calle 100, entre 11 y 12 ...")
```

CargarCodigoBarras()

Permite enviar a la impresora fiscal 2G el comando fiscal de especificación del código de barras a imprimir en el pie del comprobante. Este comando fiscal debe enviarse, en cualquier momento previo al envío del comando fiscal de cierre del comprobante (*CerrarDocumento()*).

Definido internamente como:

```
CargarCodigoBarras(  
    [in] TiposDeCodigoDeBarras TipoCodigo,  
    [in] BSTR Numero,  
    [in, optional, defaultvalue(78)] ImpresionNumeroCodigoDeBarras ImprimeNumero,  
    [in, optional, defaultvalue(71)] ImpresionCodigoDeBarras Almacena);
```

<i>TipoCodigo</i>	Permite especificar el tipo de código de barras a imprimir. Debe ser alguna de las constantes pertenecientes al enum <i>TiposDeCodigoDeBarras</i> .
-------------------	---

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

	<table border="1"> <tr> <td><i>CodigoDeBorrado</i></td> <td>48</td> </tr> <tr> <td><i>CodigoTipoEAN13</i></td> <td>49</td> </tr> <tr> <td><i>CodigoTipoEAN8</i></td> <td>50</td> </tr> <tr> <td><i>CodigoTipoUPCA</i></td> <td>51</td> </tr> <tr> <td><i>CodigoTipoI20F5</i></td> <td>52</td> </tr> </table>	<i>CodigoDeBorrado</i>	48	<i>CodigoTipoEAN13</i>	49	<i>CodigoTipoEAN8</i>	50	<i>CodigoTipoUPCA</i>	51	<i>CodigoTipoI20F5</i>	52
<i>CodigoDeBorrado</i>	48										
<i>CodigoTipoEAN13</i>	49										
<i>CodigoTipoEAN8</i>	50										
<i>CodigoTipoUPCA</i>	51										
<i>CodigoTipoI20F5</i>	52										
<i>Numero</i>	Permite especificar el código numérico del código de barras a imprimir. Por ejemplo: 779123456789										
<i>ImprimeNumero</i>	Permite especificar si se imprimirá el argumento <i>Numero</i> bajo el código de barras. Debe ser alguna de las constantes pertenecientes al enum <i>ImpresionNumeroCodigoDeBarras</i> .										
	<table border="1"> <tr> <td><i>NoImprimeNumerosCodigo</i></td> <td>32</td> </tr> <tr> <td><i>ImprimeNumerosCodigo</i></td> <td>78</td> </tr> </table>	<i>NoImprimeNumerosCodigo</i>	32	<i>ImprimeNumerosCodigo</i>	78						
<i>NoImprimeNumerosCodigo</i>	32										
<i>ImprimeNumerosCodigo</i>	78										
<i>Almacena</i>	Cualquiera de los valores posibles que se indique no tiene incidencia sobre el comportamiento de la impresora fiscal 2G, ya que la impresión del código de barras se difiere siempre hasta que sea recibido el comando fiscal de cierre del comprobante. Debe ser alguna de las constantes pertenecientes al enum <i>ImpresionCodigoDeBarras</i> .										
	<table border="1"> <tr> <td><i>ProgramaCodigo</i></td> <td>71</td> </tr> <tr> <td><i>ImprimeCodigo</i></td> <td>80</td> </tr> </table>	<i>ProgramaCodigo</i>	71	<i>ImprimeCodigo</i>	80						
<i>ProgramaCodigo</i>	71										
<i>ImprimeCodigo</i>	80										

La ubicación para la impresión del código de barras está predeterminada, y no es modificable.

Ejemplo:

```
hasar.CargarCodigoBarras(HasarArgentina.TiposDeCodigoDeBarras.CodigoTipoEAN13, _
    "779123456789", _
    HasarArgentina.ImpresionNumeroCodigoDeBarras.ImprimeNumerosCodigo, _
    HasarArgentina.ImpresionCodigoDeBarras.ProgramaCodigo)
```

CargarDatosCliente()

Permite enviar a la impresora fiscal 2G el comando fiscal con información del receptor del comprobante que se abrirá a continuación. Sólo puede ser omitido en el caso de Tiques, Tiques Nota de Crédito, y comprobantes 'B' a consumidores finales.

Definido internamente como:

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

```

CargarDatosCliente (
    [in] BSTR RazonSocial,
    [in] BSTR NumeroDocumento,
    [in] TiposDeResponsabilidadesCliente ResponsabilidadIVA,
    [in, optional, defaultvalue(32)] TiposDeDocumentoCliente TipoDocumento,
    [in, optional, defaultvalue("")] BSTR Domicilio,
    [in, optional, defaultvalue("")] BSTR DatosAdicionales1,
    [in, optional, defaultvalue("")] BSTR DatosAdicionales2,
    [in, optional, defaultvalue("")] BSTR DatosAdicionales3);
    
```

<i>RazonSocial</i>	Permite especificar el nombre o la razón social del receptor del próximo comprobante que se emita.																		
<i>NumeroDocumento</i>	Permite especificar el número del documento presentado por el cliente, como su identificación. De tratarse de una CUIT/L, deben ser 11 dígitos (sin guiones), y satisfacer el algoritmo de verificación de la AFIP.																		
<i>ResponsabilidadIVA</i>	<p>Permite especificar la responsabilidad frente a IVA del cliente. Debe ser alguna de las constantes pertenecientes al enum <i>TiposDeResponsabilidadesCliente</i>.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td><i>NoResponsable</i></td><td>65</td></tr> <tr><td><i>ConsumidorFinal</i></td><td>67</td></tr> <tr><td><i>ResponsableExento</i></td><td>69</td></tr> <tr><td><i>ResponsableInscripto</i></td><td>73</td></tr> <tr><td><i>Monotributo</i></td><td>77</td></tr> <tr><td><i>MonotributoSocial</i></td><td>83</td></tr> <tr><td><i>NoCategorizado</i></td><td>84</td></tr> <tr><td><i>Eventual</i></td><td>86</td></tr> <tr><td><i>Eventual_Social</i></td><td>87</td></tr> </table>	<i>NoResponsable</i>	65	<i>ConsumidorFinal</i>	67	<i>ResponsableExento</i>	69	<i>ResponsableInscripto</i>	73	<i>Monotributo</i>	77	<i>MonotributoSocial</i>	83	<i>NoCategorizado</i>	84	<i>Eventual</i>	86	<i>Eventual_Social</i>	87
<i>NoResponsable</i>	65																		
<i>ConsumidorFinal</i>	67																		
<i>ResponsableExento</i>	69																		
<i>ResponsableInscripto</i>	73																		
<i>Monotributo</i>	77																		
<i>MonotributoSocial</i>	83																		
<i>NoCategorizado</i>	84																		
<i>Eventual</i>	86																		
<i>Eventual_Social</i>	87																		
<i>TipoDocumento</i>	<p>Permite especificar el tipo de documento que identifica al cliente. Debe ser alguna de las constantes pertenecientes al enum <i>TiposDeDocumentoCliente</i>.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td><i>TipoNinguno</i></td><td>32</td></tr> <tr><td><i>TipoLE</i></td><td>48</td></tr> <tr><td><i>TipoLC</i></td><td>49</td></tr> <tr><td><i>TipoDNI</i></td><td>50</td></tr> <tr><td><i>TipoPasaporte</i></td><td>51</td></tr> <tr><td><i>TipoCI</i></td><td>52</td></tr> <tr><td><i>TipoCUIT</i></td><td>67</td></tr> <tr><td><i>TipoCUIL</i></td><td>76</td></tr> </table>	<i>TipoNinguno</i>	32	<i>TipoLE</i>	48	<i>TipoLC</i>	49	<i>TipoDNI</i>	50	<i>TipoPasaporte</i>	51	<i>TipoCI</i>	52	<i>TipoCUIT</i>	67	<i>TipoCUIL</i>	76		
<i>TipoNinguno</i>	32																		
<i>TipoLE</i>	48																		
<i>TipoLC</i>	49																		
<i>TipoDNI</i>	50																		
<i>TipoPasaporte</i>	51																		
<i>TipoCI</i>	52																		
<i>TipoCUIT</i>	67																		
<i>TipoCUIL</i>	76																		
<i>Domicilio</i>	Permite especificar el domicilio del cliente.																		

<i>DomicilioAdicional1</i>	Permite especificar una extensión del domicilio del cliente.
<i>DomicilioAdicional2</i>	Permite especificar una segunda extensión del domicilio del cliente.
<i>DomicilioAdicional3</i>	Permite especificar una tercera extensión del domicilio del cliente.

Ejemplo:

```
hasar.CargarDatosCliente("Cliente Uno ...", "99999999995", _
    HasarArgentina.TiposDeResponsabilidadesCliente.ResponsableInscripto, _
    HasarArgentina.TiposDeDocumentoCliente.TipoCUIT, _
    "Calle Uno, entre Dos y Tres, Nro. 1234 ...", , , )
```

CargarDocumentoAsociado()

Permite enviar a la impresora fiscal 2G el comando fiscal que vincula al comprobante que se abrirá a continuación, con algún comprobante emitido previamente (en cualquier punto de venta declarado ante la AFIP). La información que se incluye en este comando fiscal corresponde al comprobante vinculado, y no al que se va a abrir a continuación.

Definido internamente como:

```
CargarDocumentoAsociado(
    [in] long NumeroLinea,
    [in, optional, defaultvalue(0)] TiposComprobante CodigoComprobante,
    [in, optional, defaultvalue(0)] long NumeroPos,
    [in, optional, defaultvalue(0)] long NumeroComprobante);
```

<i>NumeroLinea</i>	Permite especificar la línea (1 o 2) en la que se almacenará la información contenida en el comando fiscal. Esta información se imprimirá cuando la impresora fiscal 2G reciba el comando fiscal generado por <i>'AbrirDocumento()'</i> .					
<i>CodigoComprobante</i>	Permite especificar el tipo de comprobante que se vincula. Debe ser alguna de las constantes pertenecientes al enum <i>TiposComprobante</i> .					
	***	<i>ComprobanteDonacion</i>	907	***	<i>ReciboM</i>	54
	*	<i>FacturaA</i>	1	***	<i>ReciboX</i>	902
	*	<i>FacturaB</i>	6	***	<i>RemitoR</i>	91
	*	<i>FacturaC</i>	11	***	<i>RemitoX</i>	901
	*	<i>FacturaM</i>	51	**	<i>Tique</i>	83
	***	<i>Generico</i>	910	**	<i>TiqueFacturaA</i>	81

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

* <i>NotaDeCreditoA</i>	3	** <i>TiqueFacturaB</i>	82
* <i>NotaDeCreditoB</i>	8	** <i>TiqueFacturaC</i>	111
* <i>NotaDeCreditoC</i>	13	** <i>TiqueFacturaM</i>	118
* <i>NotaDeCreditoM</i>	53	** <i>TiqueNotaCredito</i>	110
* <i>NotaDeDebitoA</i>	2	** <i>TiqueNotaCreditoA</i>	112
* <i>NotaDeDebitoB</i>	7	** <i>TiqueNotaCreditoB</i>	113
* <i>NotaDeDebitoC</i>	12	** <i>TiqueNotaCreditoC</i>	114
* <i>NotaDeDebitoM</i>	52	** <i>TiqueNotaCreditoM</i>	119
*** <i>PresupuestoX</i>	903	** <i>TiqueNotaDebitoA</i>	115
*** <i>ReciboA</i>	4	** <i>TiqueNotaDebitoB</i>	116
*** <i>ReciboB</i>	9	** <i>TiqueNotaDebitoC</i>	117
*** <i>ReciboC</i>	15	** <i>TiqueNotaDebitoM</i>	120

- * Disponibles en impresoras fiscal 2G que soporten impresión sobre formularios continuos u hojas sueltas.
- ** Disponibles en impresoras fiscales 2G que soporten impresión sobre rollos de papel continuo.
- *** Disponibles en todos los modelos de impresoras fiscales 2G.

NumeroPOS Permite especificar el número del punto de venta donde se emitió el comprobante que se vincula.

NumeroComprobante Permite especificar el número del comprobante que se vincula.

Ejemplo:

```
hasar.CargarDocumentoAsociado(1, HasarArgentina.TiposComprobante.TiqueFacturaA, 2999, 43)
```

CargarLogoAdicional()

Permite enviar a la impresora fiscal 2G el comando fiscal de carga parcial, en su memoria de trabajo, de un logotipo adicional del emisor de los comprobantes. Dicho logotipo adicional se imprimirá al pie de los comprobantes cuando se envíe el comando fiscal generado por '*CerrarDocumento()*', y hasta que se reciba el comando fiscal generado por '*CambiarLogoUsuario()*', o '*EliminarLogoAdicional()*'.

Definido internamente como:

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

```

CargarLogoAdicional(
    [in] OperacionesCargaLogoUsuario Operacion,
    [in] long NumeroLogo,
    [in, optional, defaultvalue("")] BSTR Informacion);
    
```

Operacion Permite especificar la operación de carga del logotipo adicional del emisor. Debe ser alguna de las constantes pertenecientes al enum *OperacionesCargaLogoUsuario*.

<i>CargaLogoEnCurso</i>	67
<i>FinCargaLogo</i>	70
<i>ComienzoCargaLogo</i>	73

De aquí se desprende que para la carga completa de un logotipo adicional se requiere del envío del comando fiscal generado por '*CargarLogoAdicional()*' varias veces, y en forma consecutiva. El orden de envío debe ser el siguiente:

- Operación '*ComienzoCargaLogo*' (sólo una vez)
- Operación '*CargaLogoEncurso*' (las veces necesarias)
- Operación '*FinCargaLogo*' (sólo una vez)

NumeroLogo Permite especificar el número de logotipo adicional (1 a 10) que se está cargando.

Informacion Permite especificar el string que es una parte de la imagen contenida en el archivo BMP o PNG. Se trata de una sucesión de hasta 128 caracteres (de longitud par) que representan dígitos hexadecimales. Cada par de dígitos hexadecimales representa un byte de la imagen del logotipo adicional.

La carga de un logotipo adicional del emisor implica cargar en la memoria de trabajo de la impresora fiscal 2G una imagen almacenada en un archivo BMP o PNG, la cual debe respetar ciertos requisitos. Para poder efectuar la carga es necesario convertir el contenido del archivo BMP o PNG en una lista de comandos fiscales válidos en la impresora fiscal 2G (lista de comandos fiscales generados por '*CargarLogoAdicional()*').

El archivo BMP o PNG debe poseer como atributo ser monocromático (no se admite archivo color pintado en blanco y negro, o en escalas de grises), y su tamaño (en pixeles) no debe exceder de:

Modelo	Ancho	Alto
SMH/PT-250F	576	240
SMH/PT-1000F	552	240
SMH/PT-5100F	576	290
R-HAS-6100-FAR	378	240

Ejemplo:

<i>TipoDocumento</i>	<p>Permite especificar el tipo de documento que identifica al chofer. Debe ser alguna de las constantes pertenecientes al enum <i>TiposDeDocumentoCliente</i>.</p> <table border="1"> <tr><td><i>TipoNinguno</i></td><td>32</td></tr> <tr><td><i>TipoLE</i></td><td>48</td></tr> <tr><td><i>TipoLC</i></td><td>49</td></tr> <tr><td><i>TipoDNI</i></td><td>50</td></tr> <tr><td><i>TipoPasaporte</i></td><td>51</td></tr> <tr><td><i>TipoCI</i></td><td>52</td></tr> <tr><td><i>TipoCUIT</i></td><td>67</td></tr> <tr><td><i>TipoCUIL</i></td><td>76</td></tr> </table>	<i>TipoNinguno</i>	32	<i>TipoLE</i>	48	<i>TipoLC</i>	49	<i>TipoDNI</i>	50	<i>TipoPasaporte</i>	51	<i>TipoCI</i>	52	<i>TipoCUIT</i>	67	<i>TipoCUIL</i>	76
<i>TipoNinguno</i>	32																
<i>TipoLE</i>	48																
<i>TipoLC</i>	49																
<i>TipoDNI</i>	50																
<i>TipoPasaporte</i>	51																
<i>TipoCI</i>	52																
<i>TipoCUIT</i>	67																
<i>TipoCUIL</i>	76																
<i>NumeroDocumento</i>	Permite especificar el número del documento que identifica al chofer encargado del transporte de los bienes.																
<i>Dominio1</i>	Permite especificar la placa patente correspondiente a la unidad tractora que transportará los bienes.																
<i>Dominio2</i>	(opcional) Permite especificar la placa patente correspondiente a la unidad acoplada a la que transportará los bienes.																

Ejemplo:

```
hasar.CargarTransportista("Transportador Uno ...", "0000000000", "Calle 4 entre 5 y 6",
    "Chofer Uno ...", HasarArgentina.TiposDeDocumentoCliente.TipoDNI,
    "123345678", "ABC 123", )
```

CerrarDocumento()

Permite enviar a la impresora fiscal 2G el comando fiscal de cierre del comprobante que se encuentra abierto. Opcionalmente, se puede especificar cuantas copias deben imprimirse, y una dirección de correo electrónico. Si este último dato es especificado, el comprobante cerrado será enviado a la cuenta indicada en formato HTML (visualizable con cualquier navegador de internet).

Definido internamente como:

```
CerrarDocumento(
    [in, optional, defaultvalue(0)] long Copias,
    [in, optional, defaultvalue("")] BSTR DireccionEMail,
    [out, retval] RespuestaCerrarDocumento* );
```

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

<i>Copias</i>	Permite especificar la cantidad de ejemplares de copia que deben emitirse, del comprobante que se está cerrando.
<i>DireccionEMail</i>	Permite especificar la dirección de correo electrónico del destinatario de la copia electrónica del documento que se está cerrando. <u>NOTA:</u> El envío de la copia electrónica del comprobante, por correo electrónico, requiere que el servidor de salida de correo electrónico (SMTP) haya sido configurado correctamente en la impresora fiscal 2G; considerando, además, si el servicio requiere o no autenticación.

Ejemplo:

```
Dim respcerrar As HasarArgentina.RespuestaCerrarDocumento  
  
respcerrar = hasar.CerrarDocumento(3, "<cliente@sudominio.com>")  
Debug.Print("Remito 'R' N° =[" & respcerrar.NumeroComprobante & "]" cerrado")
```

La respuesta fiscal al comando fiscal generado por 'CerrarDocumento()' es la estructura *RespuestaCerrarDocumento*, definida internamente como:

```
typedef  
struct tagRespuestaCerrarDocumento {  
    [helpstring("NumeroComprobante")  
]  
    long NumeroComprobante;  
    [helpstring("CantidadDePaginas")  
]  
    long CantidadDePaginas;  
    [helpstring("IndiceAuditoria")  
]  
    long IndiceAuditoria;  
} RespuestaCerrarDocumento;
```

<i>NumeroComprobante</i>	Informa el número del comprobante que se ha cerrado.
<i>CantidadDePaginas</i>	Informa la cantidad de páginas del comprobante. Siempre '1' en rollos de papel continuo.
<i>IndiceAuditoria</i>	Informa el número de orden, dentro del comprobante, con que se ha registrado la operación en la CTD (cinta testigo digital, o memoria de auditoría).

CerrarJornadaFiscal()

Permite enviar a la impresora fiscal 2G el comando fiscal de impresión del *Informe Diario de Cierre* (cierre 'Z').

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

Definida internamente como:

```
CerrarJornadaFiscal(  
    [in] TipoReporte Reporte,  
    [out, retval] RespuestaCerrarJornadaFiscal* );
```

Reporte Permite especificar el tipo de reporte a imprimir. Debe ser alguna de las constantes pertenecientes al enum *TipoReporte*.

<i>ReporteX</i>	88
<i>ReporteZ</i>	90

Cuando se utiliza la constante *ReporteZ* se imprime un *Informe Diario de Cierre* (cierre 'Z'). Cuando se utiliza la constante *ReporteX* se imprime un *Detalle De Ventas*.

El contribuyente tiene la obligación de emitir un comprobante de este tipo al finalizar cada jornada fiscal/comercial. De no hacerlo, pasada la medianoche la impresora fiscal 2G no permitirá continuar con la emisión de nuevos comprobantes hasta que se realice el correspondiente *Informe Diario de Cierre* (cierre 'Z'), que se encuentra pendiente. Este comportamiento es una exigencia de la normativa fiscal vigente.

Ejemplo:

```
Dim cierre As HasarArgentina.RespuestaCerrarJornadaFiscal  
Dim zeta As HasarArgentina.CerrarJornadaFiscalZ  
  
cierre = hasar.CerrarJornadaFiscal(HasarArgentina.TipoReporte.ReporteZ)  
zeta = cierre.Z  
  
Debug.Print("Cierre 'Z' Nº           =[ " & zeta.Numero & "]" )  
Debug.Print("Fecha del Cierre      =[ " & zeta.Fecha & "]" )  
Debug.Print("")  
Debug.Print("DF Emitidos                =[ " & zeta.DF_CantidadEmitidos & "]" )  
Debug.Print("DF Cancelados                =[ " & zeta.DF_CantidadCancelados & "]" )  
Debug.Print("")  
Debug.Print("DF Total                      =[ " & zeta.DF_Total & "]" )  
Debug.Print("DF Total Gravado              =[ " & zeta.DF_TotalGravado & "]" )  
Debug.Print("DF Total No Gravado          =[ " & zeta.DF_TotalNoGravado & "]" )  
Debug.Print("DF Total Exento              =[ " & zeta.DF_TotalExento & "]" )  
Debug.Print("DF Total IVA                  =[ " & zeta.DF_TotalIVA & "]" )  
Debug.Print("DF Total Otros Tributos      =[ " & zeta.DF_TotalTributos & "]" )  
Debug.Print("")  
Debug.Print("NC Emitidas                   =[ " & zeta.NC_CantidadEmitidos & "]" )  
Debug.Print("NC Canceladas                =[ " & zeta.NC_CantidadCancelados & "]" )  
Debug.Print("")  
Debug.Print("NC Total                      =[ " & zeta.NC_Total & "]" )  
Debug.Print("NC Total Gravado              =[ " & zeta.NC_TotalGravado & "]" )  
Debug.Print("NC Total No Gravado          =[ " & zeta.NC_TotalNoGravado & "]" )  
Debug.Print("NC Total Exento              =[ " & zeta.NC_TotalExento & "]" )  
Debug.Print("NC Total IVA                  =[ " & zeta.NC_TotalIVA & "]" )  
Debug.Print("NC Total Otros Tributos      =[ " & zeta.NC_TotalTributos & "]" )
```

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

```
Debug.Print("")
Debug.Print("DNFH Emitidos          =[" & zeta.DNFH_CantidadEmitidos & "]")
Debug.Print("DNFH Total            =[" & zeta.DNFH_Total & "]")
```

La respuesta fiscal al comando fiscal generado por 'CerrarJornadaFiscal()' es la estructura *RespuestaCerrarJornadaFiscal*, definida internamente como:

```
typedef
struct tagRespuestaCerrarJornadaFiscal {
    [helpstring("Reporte")]
}
    TipoReporte Reporte;
    [helpstring("X")]
}
    CerrarJornadaFiscalX X;
    [helpstring("Z")]
}
    CerrarJornadaFiscalZ Z;
} RespuestaCerrarJornadaFiscal;
```

Reporte	Informa el tipo de reporte (<i>Informe Diario de Cierre</i> -cierre 'Z'-, o <i>Detalle de Ventas</i> -reporte 'X'-) al que corresponden los datos informados.
X	Conjunto de datos informados cuando la respuesta fiscal corresponde a la emisión de un <i>Detalle de Ventas</i> . Es una estructura interna definida como:

```
typedef
struct tagCerrarJornadaFiscalX {
    [helpstring("Numero")]
}
    long Numero;
    [helpstring("FechaInicio")]
}
    DATE FechaInicio;
    [helpstring("HoraInicio")]
}
    DATE HoraInicio;
    [helpstring("FechaCierre")]
}
    DATE FechaCierre;
    [helpstring("HoraCierre")]
}
    DATE HoraCierre;
    [helpstring("DF_Total")]
}
    double DF_Total;
    [helpstring("DF_TotalIVA")]
}
    double DF_TotalIVA;
    [helpstring("DF_TotalTributos")]
}
    double DF_TotalTributos;
```

```

    [helpstring("DF_CantidadEmitidos")
]
    long DF_CantidadEmitidos;
    [helpstring("NC_Total")
]
    double NC_Total;
    [helpstring("NC_TotalIVA")
]
    double NC_TotalIVA;
    [helpstring("NC_TotalTributos")
]
    double NC_TotalTributos;
    [helpstring("NC_CantidadEmitidos")
]
    long NC_CantidadEmitidos;
    [helpstring("DNFH_CantidadEmitidos")
]
    long DNFH_CantidadEmitidos;
} CerrarJornadaFiscalX;

```

<i>Numero</i>	Informa el número del <i>Detalle Ventas</i> cuya información se reporta.
<i>FechaInicio</i>	Informa la fecha de inicio del rango de acumulados que se reporta.
<i>HoraInicio</i>	Informa la hora de inicio del rango de acumulados que se reporta.
<i>FechaCierre</i>	Informa la fecha final del rango de acumulados que se reporta.
<i>HoraCierre</i>	Informa la hora final del rango de acumulados que se reporta.
<i>DF_Total</i>	Informa el monto total acumulado de ventas, durante el rango de fechas y horarios reportados. El acumulado involucra a todos los comprobantes fiscales, con excepción de los (<i>Tiques</i>) <i>Nota de Crédito</i> .
<i>DF_TotalIVA</i>	Informa el monto total de IVA acumulado (entre todas las alícuotas), durante el rango de fechas y horarios reportados. El acumulado involucra a todos los comprobantes fiscales, con excepción de los (<i>Tiques</i>) <i>Nota de Crédito</i> .
<i>DF_TotalTributos</i>	Informa el monto total acumulado de otros tributos, durante el rango de fechas y horarios reportados. El acumulado involucra a todos los comprobantes fiscales, con excepción de los (<i>Tiques</i>) <i>Nota de Crédito</i> .
<i>DF_CantidadEmitidos</i>	Informa el total de comprobantes fiscales emitidos durante el rango de fechas y horarios reportados. El acumulado involucra a todos los comprobantes fiscales, con excepción de los (<i>Tiques</i>) <i>Nota de Crédito</i> . La cantidad reportada incluye a los comprobantes fiscales cancelados.

<i>NC_Total</i>	Informa el monto total acumulado de ventas, durante el rango de fechas y horarios reportados, correspondiente solamente a <i>(Tiques) Nota de Crédito</i> .
<i>NC_TotalIVA</i>	Informa el monto total acumulado de IVA (entre todas las alícuotas), durante el rango de fechas y horarios reportados, correspondiente solamente a <i>(Tiques) Nota de Crédito</i> .
<i>NC_TotalTributos</i>	Informa el monto total acumulado de otros tributos, durante el rango de fechas y horarios reportados, correspondiente solamente a <i>(Tiques) Nota de Crédito</i> .
<i>NC_CantidadEmitidos</i>	Informa el total de <i>(Tiques) Nota de Crédito</i> emitidos durante el rango de fechas y horarios reportados. La cantidad reportada incluye a los <i>(Tiques) Nota de Crédito</i> cancelados.
<i>DNFH_CantidadEmitidos</i>	Informa el total de documentos no fiscales homologados (DNFH) emitidos durante el rango de fechas y horarios reportados. La cantidad reportada incluye a los DNFH cancelados.

Z Conjunto de datos informados cuando la respuesta fiscal corresponde a la emisión de un *Informe Diario de Cierre* (cierre 'Z'). Es una estructura interna definida como:

```
typedef
struct tagCerrarJornadaFiscalZ {
    [helpstring("Numero")]
}
    long Numero;
    [helpstring("Fecha")]
}
    DATE Fecha;
    [helpstring("DF_Total")]
}
    double DF_Total;
    [helpstring("DF_TotalGravado")]
}
    double DF_TotalGravado;
    [helpstring("DF_TotalNoGravado")]
}
    double DF_TotalNoGravado;
    [helpstring("DF_TotalExento")]
}
    double DF_TotalExento;
    [helpstring("DF_TotalIVA")]
}
    double DF_TotalIVA;
    [helpstring("DF_TotalTributos")]
}
    double DF_TotalTributos;
```

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

```

[helpstring("DF_CantidadEmitidos")
]
long DF_CantidadEmitidos;
[helpstring("DF_CantidadCancelados")
]
long DF_CantidadCancelados;
[helpstring("NC_Total")
]
double NC_Total;
[helpstring("NC_TotalGravado")
]
double NC_TotalGravado;
[helpstring("NC_TotalNoGravado")
]
double NC_TotalNoGravado;
[helpstring("NC_TotalExento")
]
double NC_TotalExento;
[helpstring("NC_TotalIVA")
]
double NC_TotalIVA;
[helpstring("NC_TotalTributos")
]
double NC_TotalTributos;
[helpstring("NC_CantidadEmitidos")
]
long NC_CantidadEmitidos;
[helpstring("NC_CantidadCancelados")
]
long NC_CantidadCancelados;
[helpstring("DNFH_Total")
]
double DNFH_Total;
[helpstring("DNFH_CantidadEmitidos")
]
long DNFH_CantidadEmitidos;
} CerrarJornadaFiscalZ;

```

<i>Numero</i>	Informa el número de <i>Informe Diario de Cierre</i> (cierre 'Z') que se ha emitido. Es equivalente a decir que informa el número de jornada fiscal/comercial que se ha cerrado.
<i>Fecha</i>	Informa la fecha de cierre de la jornada fiscal/comercial, que no tiene porque coincidir con la fecha de emisión del <i>Informe Diario de Cierre</i> (cierre 'Z').
<i>DF_Total</i>	Informa el monto total acumulado de ventas de la jornada fiscal/comercial. El acumulado involucra a todos los comprobantes fiscales, con excepción de los (<i>Tiques</i>) <i>Nota de Crédito</i> .
<i>DF_TotalGravado</i>	Informa el monto total acumulado de bases imponibles gravadas de la jornada fiscal/comercial. El acumulado involucra a todos los comprobantes fiscales, con excepción de los (<i>Tiques</i>) <i>Nota de Crédito</i> .

<i>DF_TotalNoGravado</i>	Informa el monto total acumulado de bases imponibles no gravadas de la jornada fiscal/comercial. El acumulado involucra a todos los comprobantes fiscales, con excepción de los <i>(Tiques) Nota de Crédito</i> .
<i>DF_TotalExento</i>	Informa el monto total acumulado de bases imponibles exentas de la jornada fiscal/comercial. El acumulado involucra a todos los comprobantes fiscales, con excepción de los <i>(Tiques) Nota de Crédito</i> .
<i>DF_TotalIVA</i>	Informa el monto total acumulado de IVA (entre todas las alícuotas) de la jornada fiscal/comercial. El acumulado involucra a todos los comprobantes fiscales, con excepción de los <i>(Tiques) Nota de Crédito</i> .
<i>DF_TotalOtrosTributos</i>	Informa el monto total acumulado de otros tributos de la jornada fiscal/comercial. El acumulado involucra a todos los comprobantes fiscales, con excepción de los <i>(Tiques) Nota de Crédito</i> .
<i>DF_CantidadEmitidos</i>	Informa la cantidad total de comprobantes fiscales emitidos durante la jornada fiscal/comercial. La cantidad total involucra a todos los comprobantes fiscales emitidos, con excepción de los <i>(Tiques) Nota de Crédito</i> .
<i>DF_CantidadCancelados</i>	Informa la cantidad total de comprobantes fiscales cancelados durante la jornada fiscal/comercial. La cantidad total involucra a todos los comprobantes fiscales emitidos, con excepción de los <i>(Tiques) Nota de Crédito</i> .
<i>NC_Total</i>	Informa el monto total acumulado de ventas de la jornada fiscal/comercial, y correspondiente solamente a los <i>(Tiques) Nota de Crédito</i> .
<i>NC_TotalGravado</i>	Informa el monto total acumulado de bases imponibles gravadas de la jornada fiscal/comercial, y correspondiente solamente a los <i>(Tiques) Nota de Crédito</i> .
<i>NC_TotalNoGravado</i>	Informa el monto total acumulado de bases imponibles no gravadas de la jornada fiscal/comercial, y correspondiente solamente a los <i>(Tiques) Nota de Crédito</i> .
<i>NC_TotalExento</i>	Informa el monto total acumulado de bases imponibles exentas de la jornada fiscal/comercial, y correspondiente solamente a los <i>(Tiques) Nota de Crédito</i> .
<i>NC_TotalIVA</i>	Informa el monto total acumulado de IVA (entre todas las alícuotas) de la jornada fiscal/comercial, y correspondiente solamente a los <i>(Tiques) Nota de Crédito</i> .

<i>NC_TotalOtrosTributos</i>	Informa el monto total acumulado de otros tributos de la jornada fiscal/comercial, y correspondiente solamente a los (<i>Tiques</i>) <i>Nota de Crédito</i> .
<i>NC_CantidadEmitidos</i>	Informa la cantidad total de (<i>Tiques</i>) <i>Nota de Crédito</i> emitidos durante la jornada fiscal/comercial.
<i>NC_CantidadCancelados</i>	Informa la cantidad total de (<i>Tiques</i>) <i>Nota de Crédito</i> cancelados durante la jornada fiscal/comercial.
<i>DNFH_Total</i>	Informa el monto total acumulado por emisión de documentos no fiscales homologados (DNFH) durante la jornada fiscal/comercial.
<i>DNFH_CantidadEmitidos</i>	Informa la cantidad total de documentos no fiscales homologados (DNFH) emitidos durante la jornada fiscal/comercial.

Conectar()

Permite especificar los datos que serán memorizados y utilizados posteriormente por el OCX 2G para intercambiar comandos/respuestas fiscales con la impresoras fiscales 2G.

Definido internamente como:

```
Conectar(
    [in] BSTR Destino,
    [in, optional, defaultvalue(80)] long Puerto,
    [in, optional, defaultvalue(2000)] VARIANT TiempoEsperaConexion,
    [in, optional, defaultvalue(2000)] VARIANT TiempoEsperaComandos);
```

<i>Destino</i>	<p>Dirección IP configurada en la impresora fiscal 2G con la que se desea establecer conexión ('192.168.1.1', a la salida de fábrica).</p> <p>Se debe especificar '127.0.0.1' (localhost) si se desea establecer conexión con un emulador de impresoras fiscales 2G, o la dirección IP de la PC donde se encuentra el emulador de impresoras fiscales 2G, si el sistema de facturación y el emulador de impresoras fiscales 2G corren en PCs diferentes.</p> <p>También se debe especificar '127.0.0.1' (localhost) si se desea establecer conexión con el "proxy fiscal Hasar", o la dirección IP de la PC donde se encuentra el "proxy fiscal Hasar", si el sistema de facturación y el "proxy fiscal Hasar" corren en PCs diferentes.</p>
<i>Puerto</i>	<p>Siempre '80' si se desea establecer conexión con una impresora fiscal 2G.</p> <p>Si se desea establecer conexión con un emulador de impresoras fiscales 2G, debe coincidir con el valor indicado en el parámetro '-h', cuando éste es especificado al</p>

lanzar la ejecución del emulador de impresoras fiscales 2G.

Si se desea establecer conexión con el “proxy fiscal Hasar”, debe coincidir con el valor especificado en el tag ‘<HttpPort>’ del archivo “proxyconfig.xml” (archivo de configuración del “proxy fiscal Hasar”).

TiempoEsperaConexión

Permite indicar la espera, en milisegundos, para que se establezca la conexión vía HTTP, antes de salir por error.

TiempoEsperaComandos

Permite indicar la espera, en milisegundos, para la recepción de la respuesta fiscal a cualquier comando fiscal que se envíe a la impresora fiscal 2G, a su emulador, o al “proxy fiscal Hasar”.

‘Conectar()’ debe invocarse antes de enviar el primer comando fiscal a la impresora fiscal 2G. De no hacerlo, se producirá una excepción al intentar enviar el primer comando fiscal.

Ejemplo:

```
hasar.Conectar("192.168.1.2", , 15000, 15000)
```

ConfigurarFechaHora()

Permite enviar a la impresora fiscal 2G el comando fiscal de actualización de fecha y hora. Esta acción modifica el reloj de tiempo real de la impresora fiscal 2G. Este comando fiscal requiere que previamente se haya enviado el comando fiscal generado por ‘CerrarJornadaFiscal()’ –Informe Diario de Cierre (cierre ‘Z’)-.

Definido internamente como:

```
ConfigurarFechaHora(  
    [in] DATE Fecha,  
    [in] DATE Hora);
```

Fecha	La fecha siempre debe coincidir con la fecha actual de la impresora fiscal 2G. Para su modificación se requiere de la intervención del servicio técnico autorizado. Este comportamiento de la impresora fiscal 2G responde a las exigencias de la normativa fiscal vigente.
Hora	Permite especificar la nueva hora en el reloj de tiempo real de la impresora fiscal 2G.

Ejemplo:

```
hasar.ConfigurarFechaHora("06/04/2017", "14:50:00")
```

ConfigurarImpresoraFiscal()

Permite enviar a la impresora fiscal 2G el comando fiscal de configuración de uno de los parámetros de comportamiento del equipo fiscal, siempre que se encuentre entre aquellos que pueden ser modificados por el sistema de facturación.

Definido internamente como:

```
ConfigurarImpresoraFiscal(
    [in] Configuracion Variable,
    [in] BSTR Valor);
```

Variable	Permite especificar el parámetro de comportamiento que se desea modificar en la impresora fiscal 2G. Debe ser alguna de las constantes pertenecientes al <i>Enum Configuracion</i> .	
	Parámetro	Valores posibles
	<i>AltoHoja</i>	50 'C' Carta. 'A' A4. 'O' Oficio.
	<i>AnchoHoja</i>	51 'M' Reducido. 'N' Normal.
	<i>BorradoAutomaticoAuditoria</i>	69 'P' Borrado automático. Otro Borrado manual.
	<i>ChequeoDesborde</i>	68 'P' Chequear en todos los comandos fiscales. Otro Chequear sólo al cerrar comprobante.
	<i>CortePapel</i>	80 'F' Corte total. 'P' Corte parcial. 'N' No cortar.
	<i>EstacionReportesXZ</i>	52 'T' Rollo. 'S' Formulario continuo u hoja suelta.
	<i>ImpresionCambio</i>	67 'P' Imprimir "CAMBIO \$0.00". Otro No imprimir.
	<i>ImpresionCodigoQR</i>	57 'P' Imprimir. Otro No imprimir.
	<i>ImpresionColorAlternativo</i>	55 'N' Negro. 'R' Rojo.
	<i>ImpresionLeyendas</i>	76 'P' Imprimir leyendas opcionales. Otro No imprimir.
	<i>ImpresionMarco</i>	49 'P' Imprimir. Otro No imprimir.
	<i>ImpresionUsarColorAlternativo</i>	54 'P' Usar color alternativo. Otro No usar.
	<i>Interlineado</i>	73 'M' Mínimo.

			'N' Normal. 'A' Máximo.
	<i>LimiteBC</i>	66	'O' Sin monto máximo en comprobantes 'B' sin nominar. 'n' Monto máximo admitido.
	<i>ModoImpresion</i>	53	'M' Mixto. 'A' Sólo rollo.
	<i>Pago_Saldo</i>	83	Texto (por defecto "Saldo").
	<i>ReimpresiónCancelados</i>	82	'P' Reimprimir. Otro No reimprimir.
	<i>SonidoAviso</i>	65	'P' Emitir pitido. Otro No emitir sonido.
	<i>TimeoutEnvioRespuestaEnEspera</i>	56	'O' a '9999' segundos para interrumpir envío DC2/DC4.
	<i>TipoHabilitacion</i>	72	'A' Comprobantes 'A'. 'L' Comprobantes 'A' con leyenda. 'M' Comprobantes 'M'.
<i>Valor</i>	Permite especificar el nuevo valor a asignar al parámetro de comportamiento indicado en el argumento <i>Variable</i> .		

Ejemplo:

```
hasar.ConfigurarImpresoraFiscal(HasarArgentina.Configuracion.Interlineado, "N")
```

ConfigurarRed()

Permite enviar a la impresora fiscal 2G el comando fiscal que modifica la identificación del equipo fiscal como un nodo en la red. Cualquiera de los datos que se modifique puede causar la pérdida de la actual conexión.

Definido internamente como:

```
ConfigurarRed(
    [in, optional, defaultvalue("")] BSTR DireccionIP,
    [in, optional, defaultvalue("")] BSTR Mascara,
    [in, optional, defaultvalue("")] BSTR Gateway,
    [in, optional, defaultvalue(78)] SiNo DHCP);
```

DirecciónIP	Permite especificar la nueva dirección IP que identificará a la impresora fiscal 2G. Por defecto, a la salida de fábrica: 192.168.1.1
Mascara	Permite especificar la nueva máscara de red que se aplica a la dirección IP de la impresora fiscal

	2G. Por defecto, a la salida de fábrica: 255.255.255.0				
Gateway	Permite especificar la dirección IP de la PC utilizada como puerta de enlace a otras redes. Por defecto, a la salida de fábrica: 192.168.1.1 NOTA: Especificar, si no se utiliza esta opción: <i>Gateway = DireccionIP</i>				
DHCP	Permite especificar si la dirección IP de la impresora fiscal 2G será estática, o dinámica. Por defecto a la salida de fábrica: Estática. Debe ser alguna de las constantes pertenecientes al enum 'SiNo'. <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>No</td> <td>78 (estática)</td> </tr> <tr> <td>Si</td> <td>83 (dinámica)</td> </tr> </table>	No	78 (estática)	Si	83 (dinámica)
No	78 (estática)				
Si	83 (dinámica)				

Ejemplo:

```
hasar.ConfigurarRed("192.168.1.5", "255.255.255.0", "192.168.1.5")
```

ConfigurarServidorCorreo()

Permite enviar a la impresora fiscal 2G el comando fiscal de configuración del servidor de salida (SMTP) de correo electrónico; utilizado para el envío de copias electrónicas de los comprobantes emitidos. Los envíos son a lazo abierto, y el servidor SMTP puede ser con/sin requerimiento de autenticación.

Definido internamente como:

```
ConfigurarServidorCorreo(
    [in, optional, defaultvalue("")] BSTR DireccionIP,
    [in, optional, defaultvalue(25)] long Puerto,
    [in, optional, defaultvalue("")] BSTR DireccionRemitente,
    [in, optional, defaultvalue("")] BSTR Usuario,
    [in, optional, defaultvalue("")] BSTR Clave);
```

<i>DirecciónIP</i>	Permite especificar la dirección IP del servidor SMTP de correo electrónico que utilizará la impresora fiscal 2G para el envío de copias electrónicas de los comprobantes que se emitan.
<i>Puerto</i>	Permite especificar el puerto utilizado por el servidor SMTP de correo electrónico.
<i>DirecciónRemitente</i>	Permite especificar la dirección de correo electrónico a la que debería responder el destinatario de los comprobantes electrónicos, en caso de tener que hacerlo.

<i>Usuario</i>	Permite especificar, en caso de ser requerida una autenticación, la identificación del usuario que intenta acceder al servidor SMTP.
<i>Clave</i>	Permite especificar, en caso de ser requerida una autenticación, la clave del usuario que intenta acceder al servidor SMTP.

Ejemplo:

```
hasar.ConfigurarServidorCorreo( , "hasarventas@hasar.com")
```

ConfigurarZetaBorrable()

Permite enviar a la impresora fiscal 2G el comando fiscal que especifica la información que puede ser borrada (hasta que número de *Informe Diario de Cierre* -cierre 'Z'-), en el supuesto caso que se requiera espacio en la CTD (cinta testigo digital, o memoria de auditoría).

Definido internamente como:

```
ConfigurarZetaBorrable([in] long NumeroZeta);
```

<i>NumeroZeta</i>	Permite indicar hasta que número de jornada fiscal/comercial (o <i>Informe Diario de Cierre</i> -cierre 'Z'-) se permite borrar información de la CTD. El borrado siempre es desde las más antiguas a las más recientes, y siempre que las jornadas fiscales/comerciales a borrar hayan sido descargadas para su presentación a la AFIP.
-------------------	--

El uso de '*ConfigurarZetaBorrable()*' tiene sentido cuando la impresora fiscal 2G está configurada para admitir el borrado manual de información. Se recomienda mantener la configuración de borrado automático.

Ejemplo:

```
hasar.ConfigurarZetaBorrable(1)
```

ConfigurarZona()

Permite enviar a la impresora fiscal 2G el comando fiscal de carga en la memoria de trabajo, de un texto asociado a alguna línea de fantasía/domicilio del emisor/encabezado/pie de los comprobantes próximos a emitir. Este comando fiscal permite especificar ubicación (número de línea dentro de una zona del comprobante), el estilo con el cual se imprimirá el texto a almacenar, y por cual estación de impresión se emitirá el comprobante (si hubiera disponible más de una).

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

Definido internamente como:

```
ConfigurarZona(
    [in] long NumeroLinea,
    [in, out] AtributosDeTexto* Atributos,
    [in] BSTR Descripcion,
    [in] TiposDeEstacion Estacion,
    [in] ZonasDeLineasDeUsuario IdentificadorZona);
```

<i>NumeroLinea</i>	<p>Permite especificar el número de línea donde se almacenará el texto, en la memoria de trabajo de la impresora fiscal 2G. La línea será impresa dentro de la zona de impresión en el comprobante, indicada por el argumento <i>IdentificadorZona</i>. No todas las zonas de impresión cuentan con la misma cantidad de líneas disponibles.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td><i>Zona1Cola</i></td> <td>Líneas 1 a 4.</td> </tr> <tr> <td><i>Zona1Encabezado</i></td> <td>Líneas 1 a 3.</td> </tr> <tr> <td><i>Zona2Cola</i></td> <td>Líneas 1 a 6.</td> </tr> <tr> <td><i>Zona2Encabezado</i></td> <td>Líneas 1 a 3.</td> </tr> <tr> <td><i>ZonaDomicilioEmisor</i></td> <td>Líneas 1 a 4.</td> </tr> <tr> <td><i>ZonaFantasia</i></td> <td>Líneas 1 y 2.</td> </tr> </table>	<i>Zona1Cola</i>	Líneas 1 a 4.	<i>Zona1Encabezado</i>	Líneas 1 a 3.	<i>Zona2Cola</i>	Líneas 1 a 6.	<i>Zona2Encabezado</i>	Líneas 1 a 3.	<i>ZonaDomicilioEmisor</i>	Líneas 1 a 4.	<i>ZonaFantasia</i>	Líneas 1 y 2.
<i>Zona1Cola</i>	Líneas 1 a 4.												
<i>Zona1Encabezado</i>	Líneas 1 a 3.												
<i>Zona2Cola</i>	Líneas 1 a 6.												
<i>Zona2Encabezado</i>	Líneas 1 a 3.												
<i>ZonaDomicilioEmisor</i>	Líneas 1 a 4.												
<i>ZonaFantasia</i>	Líneas 1 y 2.												
<i>Atributos</i>	<p>Permite especificar el estilo de impresión del texto a almacenar. Se trata de una estructura de campos booleanos, asociados cada uno de ellos a un modo de impresión (centrado, negrita, doble ancho, y borrado), definida internamente como '<i>AtributosDeTexto</i>'.</p> <pre>struct tagAtributosDeTexto { [helpstring("BorradoTexto")] } VARIANT_BOOL BorradoTexto; [helpstring("DobleAncho")] } VARIANT_BOOL DobleAncho; [helpstring("Centrado")] } VARIANT_BOOL Centrado; [helpstring("Negrita")] } VARIANT_BOOL Negrita; } AtributosDeTexto;</pre>												
<i>Descripcion</i>	<p>Permite especificar el texto que se desea almacenar en la memoria de trabajo de la impresora fiscal 2G, para su posterior impresión.</p>												
<i>Estacion</i>	<p>Permite especificar la estación de impresión del texto a almacenar. Debe ser alguna de las constantes pertenecientes al enum <i>TiposDeEstacion</i>.</p>												
	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td><i>EstacionPorDefecto</i></td> <td>68</td> </tr> </table>	<i>EstacionPorDefecto</i>	68										
<i>EstacionPorDefecto</i>	68												

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

<i>IdentificadorZona</i>	<i>EstacionSlip</i>	83
	<i>EstacionTicket</i>	84
Permite especificar la zona de impresión, donde se imprimirá el texto a almacenar. Debe ser alguna de las constantes pertenecientes al enum <i>ZonasDeLineasDeUsuario</i> .	<i>ZonaFantasia</i>	70
	<i>Zona1Encabezado</i>	72
	<i>ZonaDomicilioEmisor</i>	79
	<i>Zona1Cola</i>	84
	<i>Zona2Encabezado</i>	104
	<i>Zona2Cola</i>	116

Las líneas de fantasía, encabezado, y domicilio del emisor se deben cargar antes de enviar el comando fiscal generado por '*AbrirDocumento()*', ya que es este último comando quien las imprime.

Las líneas de cola se deben cargar antes de enviar el comando fiscal generado por '*CerrarDocumento()*', ya que es este último comando quien las imprime.

Como los textos permanecen memorizados en la impresora fiscal 2G, si no son modificados o eliminados luego de cerrar el comprobante, dichos textos serán impresos en los próximos comprobantes que se emitan.

Ejemplo:

```
Dim estilo As HasarArgentina.AtributosDeTexto

estilo.BorradoTexto = False
estilo.Centrado = True
estilo.DobleAncho = True
estilo.Negrita = True

hasar.ConfigurarZona(2, estilo, "Línea 2 fantasía ...", _
                    HasarArgentina.TiposDeEstacion.EstacionPorDefecto, _
                    HasarArgentina.ZonasDeLineasDeUsuario.ZonaFantasia)
```

ConsultarAcumuladosCierreZeta()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta de numeradores alcanzados, montos acumulados totales, montos acumulados de IVA y montos acumulados de otros tributos; todos datos correspondientes a un determinado tipo de comprobante, y para un determinado *Informe Diario de Cierre* (cierre 'Z'). Como resultado se obtiene el primer registro de información disponible.

Definido internamente como:

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

```
ConsultarAcumuladosCierreZeta (
    [in] TipoReporteZ TipoReporte,
    [in] BSTR NumeroZetaFecha,
    [in, optional, defaultvalue(0)] TiposComprobante CodigoComprobante,
    [out, retval] RespuestaConsultarAcumuladosCierreZeta* );
```

TipoReporte Permite especificar si el *Informe Diario de Cierre* (cierre 'Z') se indicará por su número, o por su fecha de jornada fiscal/comercial. Debe ser alguna de las constantes pertenecientes al enum *TipoReporteZ*.

<i>ReporteZFecha</i>	70
<i>ReporteZNumero</i>	90

NumeroZetaFecha Permite especificar el número de *Informe Diario de Cierre* (cierre 'Z'), o la fecha de la jornada fiscal/comercial de interés, de acuerdo a lo indicado en el argumento *TipoReporte*.

CodigoComprobante Permite especificar el tipo de comprobante cuya información de acumulados se desea obtener, generado durante la jornada fiscal/comercial indicada en el argumento *NumeroZetaFecha*. Si no se especifica tipo de comprobante (*NoDocumento*), se obtiene la información acumulada para cada tipo de comprobante emitido durante la jornada fiscal/comercial deseada. Debe ser alguna de las constantes pertenecientes al enum *TiposComprobante*.

*** <i>ComprobanteDonacion</i>	907	*** <i>ReciboM</i>	54
* <i>FacturaA</i>	1	*** <i>ReciboX</i>	902
* <i>FacturaB</i>	6	*** <i>RemitoR</i>	91
* <i>FacturaC</i>	11	*** <i>RemitoX</i>	901
* <i>FacturaM</i>	51	** <i>Tique</i>	83
*** <i>Generico</i>	910	** <i>TiqueFacturaA</i>	81
* <i>NotaDeCreditoA</i>	3	** <i>TiqueFacturaB</i>	82
* <i>NotaDeCreditoB</i>	8	** <i>TiqueFacturaC</i>	111
* <i>NotaDeCreditoC</i>	13	** <i>TiqueFacturaM</i>	118
* <i>NotaDeCreditoM</i>	53	** <i>TiqueNotaCredito</i>	110
* <i>NotaDeDebitoA</i>	2	** <i>TiqueNotaCreditoA</i>	112
* <i>NotaDeDebitoB</i>	7	** <i>TiqueNotaCreditoB</i>	113
* <i>NotaDeDebitoC</i>	12	** <i>TiqueNotaCrediotC</i>	114
* <i>NotaDeDebitoM</i>	52	** <i>TiqueNotaCreditoM</i>	119
*** <i>PresupuestoX</i>	903	** <i>TiqueNotaDebitoA</i>	115
*** <i>ReciboA</i>	4	** <i>TiqueNotaDebitoB</i>	116
*** <i>ReciboB</i>	9	** <i>TiqueNotaDebitoC</i>	117
*** <i>ReciboC</i>	15	** <i>TiqueNotaDebitoM</i>	120

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

*	Disponibles en impresoras fiscales 2G que soporten impresión sobre formularios continuos u hojas sueltas.
**	Disponibles en impresoras fiscales 2G que soporten impresión sobre rollos de papel continuo.
***	Disponibles en todos los modelos de impresoras fiscales 2G.

Iniciada la consulta de información con este comando fiscal, el resto de la información se obtiene mediante el comando fiscal generado por 'ContinuarConsultaAcumulados()'.

Ejemplo:

```
Dim cierre1 As HasarArgentina.RespuestaConsultarAcumuladosCierreZeta

cierre1 = hasar.ConsultarAcumuladosCierreZeta(HasarArgentina.TipoReporteZ.ReporteZNumero, _
                                             "3", HasarArgentina.TiposComprobante.NoDocumento)

If (cierre1.Registro = HasarArgentina.TiposDeRegistroInforme.RegistroFinal) Then
    Debug.Print("No hay información disponible ...")
    Exit Sub
End If

If (cierre1.Registro = HasarArgentina.TiposDeRegistroInforme.RegistroDetalladoDF) Then
    Debug.Print("(DF) Documento Fiscal ...")
    Debug.Print("-----")
    Debug.Print("Cód. Comprob.      =[ " & cierre1.RegDF.CodigoComprobante & "]" )
    Debug.Print("Desde N°           =[ " & cierre1.RegDF.NumeroInicial & "]" )
    Debug.Print("Hasta N°            =[ " & cierre1.RegDF.NumeroFinal & "]" )
    Debug.Print("Cancelados         =[ " & cierre1.RegDF.CantidadCancelados & "]" )
    Debug.Print("")
    Debug.Print("TOTAL              =[ " & cierre1.RegDF.Total & "]" )
    Debug.Print("Total Gravado      =[ " & cierre1.RegDF.TotalGravado & "]" )
    Debug.Print("Total No Gravado   =[ " & cierre1.RegDF.TotalNoGravado & "]" )
    Debug.Print("Total Exento       =[ " & cierre1.RegDF.TotalExento & "]" )
    Debug.Print("Total IVA          =[ " & cierre1.RegDF.TotalIVA & "]" )
    Debug.Print("Total Otros Trib. =[ " & cierre1.RegDF.TotalTributos & "]" )
    Debug.Print("")
    Debug.Print("Alícuota 1         =[ " & cierre1.RegDF.AlicuotaIVA_1 & "]" )
    Debug.Print("Monto IVA 1        =[ " & cierre1.RegDF.MontoIVA_1 & "]" )
    Debug.Print("Alícuota 2         =[ " & cierre1.RegDF.AlicuotaIVA_2 & "]" )
    Debug.Print("Monto IVA 2        =[ " & cierre1.RegDF.MontoIVA_2 & "]" )
    Debug.Print("Alícuota 3         =[ " & cierre1.RegDF.AlicuotaIVA_3 & "]" )
    Debug.Print("Monto IVA 3        =[ " & cierre1.RegDF.MontoIVA_3 & "]" )
    Debug.Print("Alícuota 4         =[ " & cierre1.RegDF.AlicuotaIVA_4 & "]" )
    Debug.Print("Monto IVA 4        =[ " & cierre1.RegDF.MontoIVA_4 & "]" )
    Debug.Print("Alícuota 5         =[ " & cierre1.RegDF.AlicuotaIVA_5 & "]" )
    Debug.Print("Monto IVA 5        =[ " & cierre1.RegDF.MontoIVA_5 & "]" )
    Debug.Print("Alícuota 6         =[ " & cierre1.RegDF.AlicuotaIVA_6 & "]" )
    Debug.Print("Monto IVA 6        =[ " & cierre1.RegDF.MontoIVA_6 & "]" )
    Debug.Print("Alícuota 7         =[ " & cierre1.RegDF.AlicuotaIVA_7 & "]" )
    Debug.Print("Monto IVA 7        =[ " & cierre1.RegDF.MontoIVA_7 & "]" )
    Debug.Print("Alícuota 8         =[ " & cierre1.RegDF.AlicuotaIVA_8 & "]" )
    Debug.Print("Monto IVA 8        =[ " & cierre1.RegDF.MontoIVA_8 & "]" )
    Debug.Print("Alícuota 9         =[ " & cierre1.RegDF.AlicuotaIVA_9 & "]" )
```

```
Debug.Print("Monto IVA 9      =[ " & cierre1.RegDF.MontoIVA_9 & "]" )
Debug.Print("Alícuota 10     =[ " & cierre1.RegDF.AlicuotaIVA_10 & "]" )
Debug.Print("Monto IVA 10    =[ " & cierre1.RegDF.MontoIVA_10 & "]" )
Debug.Print("")
Debug.Print("Cód. Tributo 1   =[ " & cierre1.RegDF.CodigoTributo1 & "]" )
Debug.Print("Importe 1        =[ " & cierre1.RegDF.ImporteTributo1 & "]" )
Debug.Print("Cód. Tributo 2   =[ " & cierre1.RegDF.CodigoTributo2 & "]" )
Debug.Print("Importe 2        =[ " & cierre1.RegDF.ImporteTributo2 & "]" )
Debug.Print("Cód. Tributo 3   =[ " & cierre1.RegDF.CodigoTributo3 & "]" )
Debug.Print("Importe 3        =[ " & cierre1.RegDF.ImporteTributo3 & "]" )
Debug.Print("Cód. Tributo 4   =[ " & cierre1.RegDF.CodigoTributo4 & "]" )
Debug.Print("Importe 4        =[ " & cierre1.RegDF.ImporteTributo4 & "]" )
Debug.Print("Cód. Tributo 5   =[ " & cierre1.RegDF.CodigoTributo5 & "]" )
Debug.Print("Importe 5        =[ " & cierre1.RegDF.ImporteTributo5 & "]" )
Debug.Print("Cód. Tributo 6   =[ " & cierre1.RegDF.CodigoTributo6 & "]" )
Debug.Print("Importe 6        =[ " & cierre1.RegDF.ImporteTributo6 & "]" )
Debug.Print("Cód. Tributo 7   =[ " & cierre1.RegDF.CodigoTributo7 & "]" )
Debug.Print("Importe 7        =[ " & cierre1.RegDF.ImporteTributo7 & "]" )
Debug.Print("Cód. Tributo 8   =[ " & cierre1.RegDF.CodigoTributo8 & "]" )
Debug.Print("Importe 8        =[ " & cierre1.RegDF.ImporteTributo8 & "]" )
Debug.Print("Cód. Tributo 9   =[ " & cierre1.RegDF.CodigoTributo9 & "]" )
Debug.Print("Importe 9        =[ " & cierre1.RegDF.ImporteTributo9 & "]" )
Debug.Print("Cód. Tributo 10  =[ " & cierre1.RegDF.CodigoTributo10 & "]" )
Debug.Print("Importe 10       =[ " & cierre1.RegDF.ImporteTributo10 & "]" )
Debug.Print("Cód. Tributo 11  =[ " & cierre1.RegDF.CodigoTributo11 & "]" )
Debug.Print("Importe 11       =[ " & cierre1.RegDF.ImporteTributo11 & "]" )
End If
```

La respuesta fiscal al comando fiscal generado por 'ConsultarAcumuladosCierreZeta()' es la estructura *Respuesta-ConsultarAcumuladosCierreZeta* , definida internamente como:

```
typedef
struct tagRespuestaConsultarAcumuladosCierreZeta {
    [helpstring("Registro")]
}
    TiposDeRegistroInforme Registro;
    [helpstring("RegDF")]
}
    ConsultarAcumuladosCierreZetaRegDF RegDF;
    [helpstring("RegDNFH")]
}
    ConsultarAcumuladosCierreZetaRegDNFH RegDNFH;
    [helpstring("RegDNFH_NoAcum")]
}
    ConsultarAcumuladosCierreZetaRegDNFH_NoAcum RegDNFH_NoAcum;
    [helpstring("RegGlobal")]
}
    ConsultarAcumuladosCierreZetaRegGlobal RegGlobal;
} RespuestaConsultarAcumuladosCierreZeta;
```

Registro

Informa el tipo de registro que se está reportando, que puede ser alguno de los incluidos en el enum *TiposDeRegistroInforme*.

<i>RegistroFinal</i>	48
<i>RegistroDetalladoDF</i>	49
<i>RegistroDetalladoDNFH</i>	50
<i>RegistroDetalladoDNFHNoAcum</i>	51
<i>RegistroGlobal</i>	52

RegDF

Cuando *Registro = RegistroDetalloDF* entonces la respuesta fiscal continua con la siguiente estructura de información (así definida internamente):

```
typedef
struct tagConsultarAcumuladosCierreZetaRegDF {
    [helpstring("CodigoComprobante")]
}
    TiposComprobante CodigoComprobante;
    [helpstring("NumeroInicial")]
}
    long NumeroInicial;
    [helpstring("NumeroFinal")]
}
    long NumeroFinal;
    [helpstring("CantidadCancelados")]
}
    long CantidadCancelados;
    [helpstring("Total")]
}
    double Total;
    [helpstring("TotalExento")]
}
    double TotalExento;
    [helpstring("TotalNoGravado")]
}
    double TotalNoGravado;
    [helpstring("TotalGravado")]
}
    double TotalGravado;
    [helpstring("TotalIVA")]
}
    double TotalIVA;
    [helpstring("TotalTributos")]
}
    double TotalTributos;
    [helpstring("AlicuotaIVA_1")]
}
    double AlicuotaIVA_1;
    [helpstring("MontoIVA_1")]
}
    double MontoIVA_1;
    [helpstring("AlicuotaIVA_2")]
}
    double AlicuotaIVA_2;
```

```
[helpstring("MontoIVA_2")
]
double MontoIVA_2;
[helpstring("AlicuotaIVA_3")
]
double AlicuotaIVA_3;
[helpstring("MontoIVA_3")
]
double MontoIVA_3;
[helpstring("AlicuotaIVA_4")
]
double AlicuotaIVA_4;
[helpstring("MontoIVA_4")
]
double MontoIVA_4;
[helpstring("AlicuotaIVA_5")
]
double AlicuotaIVA_5;
[helpstring("MontoIVA_5")
]
double MontoIVA_5;
[helpstring("AlicuotaIVA_6")
]
double AlicuotaIVA_6;
[helpstring("MontoIVA_6")
]
double MontoIVA_6;
[helpstring("AlicuotaIVA_7")
]
double AlicuotaIVA_7;
[helpstring("MontoIVA_7")
]
double MontoIVA_7;
[helpstring("AlicuotaIVA_8")
]
double AlicuotaIVA_8;
[helpstring("MontoIVA_8")
]
double MontoIVA_8;
[helpstring("AlicuotaIVA_9")
]
double AlicuotaIVA_9;
[helpstring("MontoIVA_9")
]
double MontoIVA_9;
[helpstring("AlicuotaIVA_10")
]
double AlicuotaIVA_10;
[helpstring("MontoIVA_10")
]
double MontoIVA_10;
[helpstring("CodigoTributo1")
]
TiposTributos CodigoTributo1;
[helpstring("ImporteTributo1")
]
double ImporteTributo1;
```

```
[helpstring("CodigoTributo2")
]
TiposTributos CodigoTributo2;
[helpstring("ImporteTributo2")
]
double ImporteTributo2;
[helpstring("CodigoTributo3")
]
TiposTributos CodigoTributo3;
[helpstring("ImporteTributo3")
]
double ImporteTributo3;
[helpstring("CodigoTributo4")
]
TiposTributos CodigoTributo4;
[helpstring("ImporteTributo4")
]
double ImporteTributo4;
[helpstring("CodigoTributo5")
]
TiposTributos CodigoTributo5;
[helpstring("ImporteTributo5")
]
double ImporteTributo5;
[helpstring("CodigoTributo6")
]
TiposTributos CodigoTributo6;
[helpstring("ImporteTributo6")
]
double ImporteTributo6;
[helpstring("CodigoTributo7")
]
TiposTributos CodigoTributo7;
[helpstring("ImporteTributo7")
]
double ImporteTributo7;
[helpstring("CodigoTributo8")
]
TiposTributos CodigoTributo8;
[helpstring("ImporteTributo8")
]
double ImporteTributo8;
[helpstring("CodigoTributo9")
]
TiposTributos CodigoTributo9;
[helpstring("ImporteTributo9")
]
double ImporteTributo9;
[helpstring("CodigoTributo10")
]
TiposTributos CodigoTributo10;
[helpstring("ImporteTributo10")
]
double ImporteTributo10;
[helpstring("CodigoTributo11")
]
TiposTributos CodigoTributo11;
[helpstring("ImporteTributo11")
]
double ImporteTributo11;
} ConsultarAcumuladosCierreZetaRegDF;
```

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

<i>CódigoComprobante</i>	Informa el tipo de comprobante que se está reportando. Es alguno de los enumerados como <i>TiposComprobante</i> .																																																																								
<table border="1"> <tr> <td><i>ComprobanteDonacion</i></td> <td>907</td> <td><i>ReciboM</i></td> <td>54</td> </tr> <tr> <td><i>FacturaA</i></td> <td>1</td> <td><i>ReciboX</i></td> <td>902</td> </tr> <tr> <td><i>FacturaB</i></td> <td>6</td> <td><i>RemitoR</i></td> <td>91</td> </tr> <tr> <td><i>FacturaC</i></td> <td>11</td> <td><i>RemitoX</i></td> <td>901</td> </tr> <tr> <td><i>FacturaM</i></td> <td>51</td> <td><i>Tique</i></td> <td>83</td> </tr> <tr> <td><i>Generico</i></td> <td>910</td> <td><i>TiqueFacturaA</i></td> <td>81</td> </tr> <tr> <td><i>NotaDeCreditoA</i></td> <td>3</td> <td><i>TiqueFacturaB</i></td> <td>82</td> </tr> <tr> <td><i>NotaDeCreditoB</i></td> <td>8</td> <td><i>TiqueFacturaC</i></td> <td>111</td> </tr> <tr> <td><i>NotaDeCreditoC</i></td> <td>13</td> <td><i>TiqueFacturaM</i></td> <td>118</td> </tr> <tr> <td><i>NotaDeCreditoM</i></td> <td>53</td> <td><i>TiqueNotaCredito</i></td> <td>110</td> </tr> <tr> <td><i>NotaDeDebitoA</i></td> <td>2</td> <td><i>TiqueNotaCreditoA</i></td> <td>112</td> </tr> <tr> <td><i>NotaDeDebitoB</i></td> <td>7</td> <td><i>TiqueNotaCreditoB</i></td> <td>113</td> </tr> <tr> <td><i>NotaDeDebitoC</i></td> <td>12</td> <td><i>TiqueNotaCreditoC</i></td> <td>114</td> </tr> <tr> <td><i>NotaDeDebitoM</i></td> <td>52</td> <td><i>TiqueNotaCreditoM</i></td> <td>119</td> </tr> <tr> <td><i>PresupuestoX</i></td> <td>903</td> <td><i>TiqueNotaDebitoA</i></td> <td>115</td> </tr> <tr> <td><i>ReciboA</i></td> <td>4</td> <td><i>TiqueNotaDebitoB</i></td> <td>116</td> </tr> <tr> <td><i>ReciboB</i></td> <td>9</td> <td><i>TiqueNotaDebitoC</i></td> <td>117</td> </tr> <tr> <td><i>ReciboC</i></td> <td>15</td> <td><i>TiqueNotaDebitoM</i></td> <td>120</td> </tr> </table>	<i>ComprobanteDonacion</i>	907	<i>ReciboM</i>	54	<i>FacturaA</i>	1	<i>ReciboX</i>	902	<i>FacturaB</i>	6	<i>RemitoR</i>	91	<i>FacturaC</i>	11	<i>RemitoX</i>	901	<i>FacturaM</i>	51	<i>Tique</i>	83	<i>Generico</i>	910	<i>TiqueFacturaA</i>	81	<i>NotaDeCreditoA</i>	3	<i>TiqueFacturaB</i>	82	<i>NotaDeCreditoB</i>	8	<i>TiqueFacturaC</i>	111	<i>NotaDeCreditoC</i>	13	<i>TiqueFacturaM</i>	118	<i>NotaDeCreditoM</i>	53	<i>TiqueNotaCredito</i>	110	<i>NotaDeDebitoA</i>	2	<i>TiqueNotaCreditoA</i>	112	<i>NotaDeDebitoB</i>	7	<i>TiqueNotaCreditoB</i>	113	<i>NotaDeDebitoC</i>	12	<i>TiqueNotaCreditoC</i>	114	<i>NotaDeDebitoM</i>	52	<i>TiqueNotaCreditoM</i>	119	<i>PresupuestoX</i>	903	<i>TiqueNotaDebitoA</i>	115	<i>ReciboA</i>	4	<i>TiqueNotaDebitoB</i>	116	<i>ReciboB</i>	9	<i>TiqueNotaDebitoC</i>	117	<i>ReciboC</i>	15	<i>TiqueNotaDebitoM</i>	120	
<i>ComprobanteDonacion</i>	907	<i>ReciboM</i>	54																																																																						
<i>FacturaA</i>	1	<i>ReciboX</i>	902																																																																						
<i>FacturaB</i>	6	<i>RemitoR</i>	91																																																																						
<i>FacturaC</i>	11	<i>RemitoX</i>	901																																																																						
<i>FacturaM</i>	51	<i>Tique</i>	83																																																																						
<i>Generico</i>	910	<i>TiqueFacturaA</i>	81																																																																						
<i>NotaDeCreditoA</i>	3	<i>TiqueFacturaB</i>	82																																																																						
<i>NotaDeCreditoB</i>	8	<i>TiqueFacturaC</i>	111																																																																						
<i>NotaDeCreditoC</i>	13	<i>TiqueFacturaM</i>	118																																																																						
<i>NotaDeCreditoM</i>	53	<i>TiqueNotaCredito</i>	110																																																																						
<i>NotaDeDebitoA</i>	2	<i>TiqueNotaCreditoA</i>	112																																																																						
<i>NotaDeDebitoB</i>	7	<i>TiqueNotaCreditoB</i>	113																																																																						
<i>NotaDeDebitoC</i>	12	<i>TiqueNotaCreditoC</i>	114																																																																						
<i>NotaDeDebitoM</i>	52	<i>TiqueNotaCreditoM</i>	119																																																																						
<i>PresupuestoX</i>	903	<i>TiqueNotaDebitoA</i>	115																																																																						
<i>ReciboA</i>	4	<i>TiqueNotaDebitoB</i>	116																																																																						
<i>ReciboB</i>	9	<i>TiqueNotaDebitoC</i>	117																																																																						
<i>ReciboC</i>	15	<i>TiqueNotaDebitoM</i>	120																																																																						
<i>NumeroInicial</i>	Informa el primer número de comprobante, para el rango de los emitidos durante la jornada fiscal/comercial.																																																																								
<i>NumeroFinal</i>	Informa el último número de comprobante, para el rango de los emitidos durante la jornada fiscal/comercial.																																																																								
<i>CantidadCancelados</i>	Informa cuantos comprobantes, dentro del rango reportado, fueron cancelados.																																																																								
<i>Total</i>	Informa el importe total y acumulado para todos los comprobantes incluidos en el rango reportado.																																																																								
<i>TotalExento</i>	Informa el importe total exento y acumulado para todos los comprobantes incluidos en el rango reportado.																																																																								
<i>TotalNoGravado</i>	Informa el importe total no gravado y acumulado para todos los comprobantes incluidos en el rango reportado.																																																																								

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

<i>TotalGravado</i>	Informa el importe total gravado y acumulado para todos los comprobantes incluidos en el rango reportado.
<i>TotalIVA</i>	Informa el importe total de IVA y acumulado para todos los comprobantes incluidos en el rango reportado.
<i>TotalTributos</i>	Informa el importe total de IVA y acumulado para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_1</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_1</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_2</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_2</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_3</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_3</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_4</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_4</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_5</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_5</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

<i>AlicuotaIVA_6</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_6</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_7</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_7</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_8</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_8</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_9</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_9</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_10</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>.MontoIVA_10</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>CodigoTributo1</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i> .

<i>IIBB</i>	5	<i>OtrasPercepciones</i>	9
<i>ImpuestoInternoltem</i>	10	<i>OtrosTributos</i>	99
<i>ImpuestosInternos</i>	4	<i>PercepcionIIBB</i>	7

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

	<table border="1"> <tr> <td><i>ImpuestosMunicipales</i></td> <td align="center">3</td> <td><i>PercepcionImpuestos Municipales</i></td> <td align="center">8</td> </tr> <tr> <td><i>ImpuestosNacionales</i></td> <td align="center">1</td> <td><i>PercepcionIVA</i></td> <td align="center">6</td> </tr> <tr> <td><i>ImpuestosProvinciales</i></td> <td align="center">2</td> <td><i>SinImpuestos</i></td> <td align="center">0</td> </tr> </table>	<i>ImpuestosMunicipales</i>	3	<i>PercepcionImpuestos Municipales</i>	8	<i>ImpuestosNacionales</i>	1	<i>PercepcionIVA</i>	6	<i>ImpuestosProvinciales</i>	2	<i>SinImpuestos</i>	0
<i>ImpuestosMunicipales</i>	3	<i>PercepcionImpuestos Municipales</i>	8										
<i>ImpuestosNacionales</i>	1	<i>PercepcionIVA</i>	6										
<i>ImpuestosProvinciales</i>	2	<i>SinImpuestos</i>	0										
<i>ImporteTributo1</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.												
<i>CodigoTributo2</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i> . (ver tabla en <i>CodigoTributo1</i>)												
<i>ImporteTributo2</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.												
<i>CodigoTributo3</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i> . (ver tabla en <i>CodigoTributo1</i>)												
<i>ImporteTributo3</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.												
<i>CodigoTributo4</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i> . (ver tabla en <i>CodigoTributo1</i>)												
<i>ImporteTributo4</i>	<i>Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.</i>												
<i>CodigoTributo5</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i> . (ver tabla en <i>CodigoTributo1</i>)												
<i>ImporteTributo5</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.												
<i>CodigoTributo6</i>	Informa un código de tributo percibido en algunos de los com-												

	<p>probantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i>.</p> <p>(ver tabla en <i>CodigoTributo1</i>)</p>
<i>ImporteTributo6</i>	<p>Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.</p>
<i>CodigoTributo7</i>	<p>Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i>.</p> <p>(ver tabla en <i>CodigoTributo1</i>)</p>
<i>ImporteTributo7</i>	<p>Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.</p>
<i>CodigoTributo8</i>	<p>Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i>.</p> <p>(ver tabla en <i>CodigoTributo1</i>)</p>
<i>ImporteTributo8</i>	<p>Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.</p>
<i>CodigoTributo9</i>	<p>Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i>.</p> <p>(ver tabla en <i>CodigoTributo1</i>)</p>
<i>ImporteTributo9</i>	<p>Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.</p>
<i>CodigoTributo10</i>	<p>Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i>.</p> <p>(ver tabla en <i>CodigoTributo1</i>)</p>
<i>ImporteTributo10</i>	<p>Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.</p>

CodigoTributo11

Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como *TiposTributos*.

(ver tabla en *CodigoTributo1*)

ImporteTributo11

Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.

RegDNFH

Quando *Registro = RegistroDetalladoDNFH* entonces la respuesta fiscal continúa con la siguiente estructura de información (así definida internamente):

```
typedef
struct tagConsultarAcumuladosCierreZetaRegDNFH {
    [helpstring("CodigoComprobante")]
}
    TiposComprobante CodigoComprobante;
    [helpstring("NumeroInicial")]
}
    long NumeroInicial;
    [helpstring("NumeroFinal")]
}
    long NumeroFinal;
    [helpstring("Total")]
}
    double Total;
} ConsultarAcumuladosCierreZetaRegDNFH;
```

CodigoComprobante

Informa el tipo de comprobante que se está reportando. Es alguno de los enumerados como *TiposComprobante*.

<i>ComprobanteDonacion</i>	907	<i>ReciboM</i>	54
<i>FacturaA</i>	1	<i>ReciboX</i>	902
<i>FacturaB</i>	6	<i>RemitoR</i>	91
<i>FacturaC</i>	11	<i>RemitoX</i>	901
<i>FacturaM</i>	51	<i>Tique</i>	83
<i>Generico</i>	910	<i>TiqueFacturaA</i>	81
<i>NotaDeCreditoA</i>	3	<i>TiqueFacturaB</i>	82
<i>NotaDeCreditoB</i>	8	<i>TiqueFacturaC</i>	111
<i>NotaDeCreditoC</i>	13	<i>TiqueFacturaM</i>	118
<i>NotaDeCreditoM</i>	53	<i>TiqueNotaCredito</i>	110
<i>NotaDeDebitoA</i>	2	<i>TiqueNotaCreditoA</i>	112
<i>NotaDeDebitoB</i>	7	<i>TiqueNotaCreditoB</i>	113
<i>NotaDeDebitoC</i>	12	<i>TiqueNotaCreditoC</i>	114

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

	<i>NotaDeDebitoM</i>	52	<i>TiqueNotaCreditoM</i>	119
	<i>PresupuestoX</i>	903	<i>TiqueNotaDebitoA</i>	115
	<i>ReciboA</i>	4	<i>TiqueNotaDebitoB</i>	116
	<i>ReciboB</i>	9	<i>TiqueNotaDebitoC</i>	117
	<i>ReciboC</i>	15	<i>TiqueNotaDebitoM</i>	120
<i>NumeroInicial</i>	Informa el primer número de comprobante, para el rango de los emitidos durante la jornada fiscal/comercial.			
<i>NumeroFinal</i>	Informa el último número de comprobante, para el rango de los emitidos durante la jornada fiscal/comercial.			
<i>Total</i>	Informa el importe total y acumulado para todos los comprobantes incluidos en el rango reportado.			

RegDNFH_NoAcum

Cuando *Registro = RegistroDetalladoDNFHNoAcum* entonces la respuesta fiscal continúa con la siguiente estructura de información (así definida internamente):

```
typedef
struct tagConsultarAcumuladosCierreZetaRegDNFH_NoAcum {
    [helpstring("CodigoComprobante")]
}
    TiposComprobante CodigoComprobante;
    [helpstring("NumeroInicial")]
}
    long NumeroInicial;
    [helpstring("NumeroFinal")]
}
    long NumeroFinal;
} ConsultarAcumuladosCierreZetaRegDNFH_NoAcum;
```

<i>CodigoComprobante</i>	Informa el tipo de comprobante que se está reportando. Es alguno de los enumerados como <i>TiposComprobante</i> .			
<i>ComprobanteDonacion</i>	907	<i>ReciboM</i>	54	
<i>FacturaA</i>	1	<i>ReciboX</i>	902	
<i>FacturaB</i>	6	<i>RemitoR</i>	91	
<i>FacturaC</i>	11	<i>RemitoX</i>	901	
<i>FacturaM</i>	51	<i>Tique</i>	83	
<i>Generico</i>	910	<i>TiqueFacturaA</i>	81	
<i>NotaDeCreditoA</i>	3	<i>TiqueFacturaB</i>	82	
<i>NotaDeCreditoB</i>	8	<i>TiqueFacturaC</i>	111	
<i>NotaDeCreditoC</i>	13	<i>TiqueFacturaM</i>	118	

	<i>NotaDeCreditoM</i>	53	<i>TiqueNotaCredito</i>	110
	<i>NotaDeDebitoA</i>	2	<i>TiqueNotaCreditoA</i>	112
	<i>NotaDeDebitoB</i>	7	<i>TiqueNotaCreditoB</i>	113
	<i>NotaDeDebitoC</i>	12	<i>TiqueNotaCreditoC</i>	114
	<i>NotaDeDebitoM</i>	52	<i>TiqueNotaCreditoM</i>	119
	<i>PresupuestoX</i>	903	<i>TiqueNotaDebitoA</i>	115
	<i>ReciboA</i>	4	<i>TiqueNotaDebitoB</i>	116
	<i>ReciboB</i>	9	<i>TiqueNotaDebitoC</i>	117
	<i>ReciboC</i>	15	<i>TiqueNotaDebitoM</i>	120
<i>NumeroInicial</i>	Informa el primer número de comprobante, para el rango de los emitidos durante la jornada fiscal/comercial.			
<i>NumeroFinal</i>	Informa el último número de comprobante, para el rango de los emitidos durante la jornada fiscal/comercial.			

RegGlobal

Cuando *Registro = RegistroGlobal* entonces la respuesta fiscal continúa con la siguiente estructura de información (así definida internamente):

```
typedef [uuid(9AED0EC7-3C9A-4693-80CF-72768CFF9178)]
struct tagConsultarAcumuladosCierreZetaRegGlobal {
    [helpstring("DF_CantidadCancelados")]
    ]
    long DF_CantidadCancelados;
    [helpstring("DF_CantidadEmitidos")]
    ]
    long DF_CantidadEmitidos;
    [helpstring("DF_Total")]
    ]
    double DF_Total;
    [helpstring("DF_TotalExento")]
    ]
    double DF_TotalExento;
    [helpstring("DF_TotalNoGravado")]
    ]
    double DF_TotalNoGravado;
    [helpstring("DF_TotalGravado")]
    ]
    double DF_TotalGravado;
    [helpstring("DF_TotalIVA")]
    ]
    double DF_TotalIVA;
    [helpstring("DF_TotalTributos")]
    ]
    double DF_TotalTributos;
}
```

```
[helpstring("DNFH_Total")
]
double DNFH_Total;
} ConsultarAcumuladosCierreZetaRegGlobal;
```

<i>DF_CantidadCancelados</i>	Informa la cantidad total de documentos fiscales (sin distinguir tipo) cancelados durante la jornada fiscal/comercial.
<i>DF_CantidadEmitidos</i>	Informa la cantidad total de documentos fiscales (sin distinguir tipo) emitidos durante la jornada fiscal/comercial.
<i>DF_Total</i>	Informa el monto total de ventas acumulado durante la jornada fiscal/comercial. No incluye <i>(Tiques) Nota de Crédito</i> .
<i>DF_TotalExento</i>	Informa el monto total de bases imponibles exentas, acumulado durante la jornada fiscal/comercial. No incluye <i>(Tiques) Nota de Crédito</i> .
<i>DF_TotalNoGravado</i>	Informa el monto total de bases imponibles no gravadas, acumulado durante la jornada fiscal/comercial. No incluye <i>(Tiques) Nota de Crédito</i> .
<i>DF_TotalGravado</i>	Informa el monto total de bases imponibles gravadas, acumulado durante la jornada fiscal/comercial. No incluye <i>(Tiques) Nota de Crédito</i> .
<i>DF_TotalIVA</i>	Informa el monto total de IVA, acumulado durante la jornada fiscal/comercial entre todas las alícuotas. No incluye <i>(Tiques) Nota de Crédito</i> .
<i>DF_TotalTributos</i>	Informa el monto total de tributos, acumulado durante la jornada fiscal/comercial entre todos los tributos percibidos. No incluye <i>(Tiques) Nota de Crédito</i> .
<i>DNFH_Total</i>	Informa el monto total acumulado durante la jornada fiscal/comercial entre todos los documentos no fiscales homologados que generan totales.

ConsultarAcumuladosComprobante()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta de montos acumulados de bases imponibles, montos acumulados de IVA y montos acumulados de otros tributos; todos datos correspondientes a un determinado comprobante, del cual se especifica tipo y número. Como resultado se obtiene el primer registro de información disponible.

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

Definido internamente como:

```
ConsultarAcumuladosComprobante (
    [in] TiposComprobante CodigoComprobante,
    [in] long NumeroComprobante,
    [out, retval] RespuestaConsultarAcumuladosComprobante* );
```

<i>CodigoComprobante</i>	Permite especificar el tipo de comprobante, respecto del cual se desean consultar sus acumulados. Debe ser alguna de las constantes pertenecientes al enum <i>TiposComprobante</i> .			
	***	907	***	54
	*	1	***	902
	*	6	***	91
	*	11	***	901
	*	51	**	83
	***	910	**	81
	*	3	**	82
	*	8	**	111
	*	13	**	118
	*	53	**	110
	*	2	**	112
	*	7	**	113
	*	12	**	114
	*	52	**	119
	***	903	**	115
	***	4	**	116
	***	9	**	117
	***	15	**	120
	*	Disponibles en impresoras fiscales 2G que soporten impresión sobre formularios continuos u hojas sueltas.		
	**	Disponibles en impresoras fiscales 2G que soporten impresión sobre rollos de papel continuo.		
	***	Disponibles en todos los modelos de impresoras fiscales 2G.		
<i>NumeroComprobante</i>	Permite especificar el número del comprobante respecto del cual se consultan sus acumulados.			

Iniciada la consulta de información con este comando fiscal, el resto de la información se obtiene mediante el comando fiscal generado por '*ContinuarConsultaAcumulados()*'.

Ejemplo:

```
Dim comprob1 As HasarArgentina.RespuestaConsultarAcumuladosComprobante

comprob1 = _
    hasar.ConsultarAcumuladosComprobante(HasarArgentina.TiposComprobante.TiqueFacturaA, 3)

If (comprob1.Registro = HasarArgentina.TiposDeRegistroInforme.RegistroFinal) Then
    Debug.Print("No hay información disponible ...")
    Exit Sub
End If

If (comprob1.Registro = HasarArgentina.TiposDeRegistroInforme.RegistroDetalladoDF) Then
    Debug.Print("(DF) Documento Fiscal ...")
    Debug.Print("-----")
    Debug.Print("Cód. Comprob.      =[" & comprob1.RegDF.CodigoComprobante & "]")
    Debug.Print("Desde Nº          =[" & comprob1.RegDF.NumeroInicial & "]")
    Debug.Print("Hasta Nº           =[" & comprob1.RegDF.NumeroFinal & "]")
    Debug.Print("Cancelados        =[" & comprob1.RegDF.CantidadCancelados & "]")
    Debug.Print("")
    Debug.Print("TOTAL              =[" & comprob1.RegDF.Total & "]")
    Debug.Print("Total Gravado      =[" & comprob1.RegDF.TotalGravado & "]")
    Debug.Print("Total No Gravado   =[" & comprob1.RegDF.TotalNoGravado & "]")
    Debug.Print("Total Exento       =[" & comprob1.RegDF.TotalExento & "]")
    Debug.Print("Total IVA          =[" & comprob1.RegDF.TotalIVA & "]")
    Debug.Print("Total Otros Trib. =[" & comprob1.RegDF.TotalTributos & "]")
    Debug.Print("")
    Debug.Print("Monto Base 1       =[" & comprob1.RegDF.MontoNetoSinIVA_1 & "]")
    Debug.Print("Alícuota 1         =[" & comprob1.RegDF.AlicuotaIVA_1 & "]")
    Debug.Print("Monto IVA 1        =[" & comprob1.RegDF.MontoIVA_1 & "]")
    Debug.Print("Monto Base 2       =[" & comprob1.RegDF.MontoNetoSinIVA_2 & "]")
    Debug.Print("Alícuota 2         =[" & comprob1.RegDF.AlicuotaIVA_2 & "]")
    Debug.Print("Monto IVA 2        =[" & comprob1.RegDF.MontoIVA_2 & "]")
    Debug.Print("Monto Base 3       =[" & comprob1.RegDF.MontoNetoSinIVA_3 & "]")
    Debug.Print("Alícuota 3         =[" & comprob1.RegDF.AlicuotaIVA_3 & "]")
    Debug.Print("Monto IVA 3        =[" & comprob1.RegDF.MontoIVA_3 & "]")
    Debug.Print("Monto Base 4       =[" & comprob1.RegDF.MontoNetoSinIVA_4 & "]")
    Debug.Print("Alícuota 4         =[" & comprob1.RegDF.AlicuotaIVA_4 & "]")
    Debug.Print("Monto IVA 4        =[" & comprob1.RegDF.MontoIVA_4 & "]")
    Debug.Print("Monto Base 5       =[" & comprob1.RegDF.MontoNetoSinIVA_5 & "]")
    Debug.Print("Alícuota 5         =[" & comprob1.RegDF.AlicuotaIVA_5 & "]")
    Debug.Print("Monto IVA 5        =[" & comprob1.RegDF.MontoIVA_5 & "]")
    Debug.Print("Monto Base 6       =[" & comprob1.RegDF.MontoNetoSinIVA_6 & "]")
    Debug.Print("Alícuota 6         =[" & comprob1.RegDF.AlicuotaIVA_6 & "]")
    Debug.Print("Monto IVA 6        =[" & comprob1.RegDF.MontoIVA_6 & "]")
    Debug.Print("Monto Base 7       =[" & comprob1.RegDF.MontoNetoSinIVA_7 & "]")
    Debug.Print("Alícuota 7         =[" & comprob1.RegDF.AlicuotaIVA_7 & "]")
    Debug.Print("Monto IVA 7        =[" & comprob1.RegDF.MontoIVA_7 & "]")
    Debug.Print("Monto Base 8       =[" & comprob1.RegDF.MontoNetoSinIVA_8 & "]")
    Debug.Print("Alícuota 8         =[" & comprob1.RegDF.AlicuotaIVA_8 & "]")
    Debug.Print("Monto IVA 8        =[" & comprob1.RegDF.MontoIVA_8 & "]")
    Debug.Print("Monto Base 9       =[" & comprob1.RegDF.MontoNetoSinIVA_9 & "]")
    Debug.Print("Alícuota 9         =[" & comprob1.RegDF.AlicuotaIVA_9 & "]")
    Debug.Print("Monto IVA 9        =[" & comprob1.RegDF.MontoIVA_9 & "]")
    Debug.Print("Monto Base 10      =[" & comprob1.RegDF.MontoNetoSinIVA_10 & "]")
    Debug.Print("Alícuota 10        =[" & comprob1.RegDF.AlicuotaIVA_10 & "]")
    Debug.Print("Monto IVA 10       =[" & comprob1.RegDF.MontoIVA_10 & "]")

```

```

Debug.Print("")
Debug.Print("Cód. Tributo 1      =[ " & comprob1.RegDF.CodigoTributo1 & "]"")
Debug.Print("Importe 1          =[ " & comprob1.RegDF.ImporteTributo1 & "]"")
Debug.Print("Cód. Tributo 2      =[ " & comprob1.RegDF.CodigoTributo2 & "]"")
Debug.Print("Importe 2          =[ " & comprob1.RegDF.ImporteTributo2 & "]"")
Debug.Print("Cód. Tributo 3      =[ " & comprob1.RegDF.CodigoTributo3 & "]"")
Debug.Print("Importe 3          =[ " & comprob1.RegDF.ImporteTributo3 & "]"")
Debug.Print("Cód. Tributo 4      =[ " & comprob1.RegDF.CodigoTributo4 & "]"")
Debug.Print("Importe 4          =[ " & comprob1.RegDF.ImporteTributo4 & "]"")
Debug.Print("Cód. Tributo 5      =[ " & comprob1.RegDF.CodigoTributo5 & "]"")
Debug.Print("Importe 5          =[ " & comprob1.RegDF.ImporteTributo5 & "]"")
Debug.Print("Cód. Tributo 6      =[ " & comprob1.RegDF.CodigoTributo6 & "]"")
Debug.Print("Importe 6          =[ " & comprob1.RegDF.ImporteTributo6 & "]"")
Debug.Print("Cód. Tributo 7      =[ " & comprob1.RegDF.CodigoTributo7 & "]"")
Debug.Print("Importe 7          =[ " & comprob1.RegDF.ImporteTributo7 & "]"")
Debug.Print("Cód. Tributo 8      =[ " & comprob1.RegDF.CodigoTributo8 & "]"")
Debug.Print("Importe 8          =[ " & comprob1.RegDF.ImporteTributo8 & "]"")
Debug.Print("Cód. Tributo 9      =[ " & comprob1.RegDF.CodigoTributo9 & "]"")
Debug.Print("Importe 9          =[ " & comprob1.RegDF.ImporteTributo9 & "]"")
Debug.Print("Cód. Tributo 10     =[ " & comprob1.RegDF.CodigoTributo10 & "]"")
Debug.Print("Importe 10         =[ " & comprob1.RegDF.ImporteTributo10 & "]"")
Debug.Print("Cód. Tributo 11     =[ " & comprob1.RegDF.CodigoTributo11 & "]"")
Debug.Print("Importe 11         =[ " & comprob1.RegDF.ImporteTributo11 & "]"")
End If

```

La respuesta fiscal al comando fiscal generado por '*ConsultarAcumuladosComprobante()*' es la estructura *RespuestaConsultarAcumuladosComprobante*, definida internamente como:

```

typedef
struct tagRespuestaConsultarAcumuladosComprobante {
    [helpstring("Registro")]
}
    TiposDeRegistroInforme Registro;
    [helpstring("RegDF")]
}
    ConsultarAcumuladosComprobanteRegDF RegDF;
    [helpstring("RegDNFH")]
}
    ConsultarAcumuladosComprobanteRegDNFH RegDNFH;
    [helpstring("RegDNFH_NoAcum")]
}
    ConsultarAcumuladosComprobanteRegDNFH_NoAcum RegDNFH_NoAcum;
    [helpstring("RegGlobal")]
}
    ConsultarAcumuladosComprobanteRegGlobal RegGlobal;
} RespuestaConsultarAcumuladosComprobante;

```

Registro

Informa el tipo de registro que se está reportando, que puede ser alguno de los incluidos en el enum *TiposDeRegistroInforme*.

<i>RegistroFinal</i>	48
<i>RegistroDetalladoDF</i>	49

RegDF

<i>RegistroDetalladoDNFH</i>	50
<i>RegistroDetalladoDNFHNoAcum</i>	51
<i>RegistroGlobal</i>	52

Cuando *Registro = RegistroDetalloDF* entonces la respuesta fiscal continua con la siguiente estructura de información (así definida internamente):

```
typedef
struct tagConsultarAcumuladosComprobanteRegD
[helpstring("CodigoComprobante")
]
TiposComprobante CodigoComprobante;
[helpstring("NumeroInicial")
]
long NumeroInicial;
[helpstring("NumeroFinal")
]
long NumeroFinal;
[helpstring("CantidadCancelados")
]
long CantidadCancelados;
[helpstring("Total")
]
double Total;
[helpstring("TotalExento")
]
double TotalExento;
[helpstring("TotalNoGravado")
]
double TotalNoGravado;
[helpstring("TotalGravado")
]
double TotalGravado;
[helpstring("TotalIVA")
]
double TotalIVA;
[helpstring("TotalTributos")
]
double TotalTributos;
[helpstring("AlicuotaIVA_1")
]
double AlicuotaIVA_1;
[helpstring("MontoIVA_1")
]
double MontoIVA_1;
[helpstring("MontoNetoSinIVA_1")
]
double MontoNetoSinIVA_1;
[helpstring("AlicuotaIVA_2")
]
double AlicuotaIVA_2;
[helpstring("MontoIVA_2")
]
double MontoIVA_2;
```

```
[helpstring("MontoNetoSinIVA_2")
]
double MontoNetoSinIVA_2;
[helpstring("AlicuotaIVA_3")
]
double AlicuotaIVA_3;
[helpstring("MontoIVA_3")
]
double MontoIVA_3;
[helpstring("MontoNetoSinIVA_3")
]
double MontoNetoSinIVA_3;
[helpstring("AlicuotaIVA_4")
]
double AlicuotaIVA_4;
[helpstring("MontoIVA_4")
]
double MontoIVA_4;
[helpstring("MontoNetoSinIVA_4")
]
double MontoNetoSinIVA_4;
[helpstring("AlicuotaIVA_5")
]
double AlicuotaIVA_5;
[helpstring("MontoIVA_5")
]
double MontoIVA_5;
[helpstring("MontoNetoSinIVA_5")
]
double MontoNetoSinIVA_5;
[helpstring("AlicuotaIVA_6")
]
double AlicuotaIVA_6;
[helpstring("MontoIVA_6")
]
double MontoIVA_6;
[helpstring("MontoNetoSinIVA_6")
]
double MontoNetoSinIVA_6;
[helpstring("AlicuotaIVA_7")
]
double AlicuotaIVA_7;
[helpstring("MontoIVA_7")
]
double MontoIVA_7;
[helpstring("MontoNetoSinIVA_7")
]
double MontoNetoSinIVA_7;
[helpstring("AlicuotaIVA_8")
]
double AlicuotaIVA_8;
[helpstring("MontoIVA_8")
]
double MontoIVA_8;
[helpstring("MontoNetoSinIVA_8")
]
double MontoNetoSinIVA_8;
```

```
[helpstring("AlicuotaIVA_9")
]
double AlicuotaIVA_9;
[helpstring("MontoIVA_9")
]
double MontoIVA_9;
[helpstring("MontoNetoSinIVA_9")
]
double MontoNetoSinIVA_9;
[helpstring("AlicuotaIVA_10")
]
double AlicuotaIVA_10;
[helpstring("MontoIVA_10")
]
double MontoIVA_10;
[helpstring("MontoNetoSinIVA_10")
]
double MontoNetoSinIVA_10;
[helpstring("CodigoTributo1")
]
TiposTributos CodigoTributo1;
[helpstring("ImporteTributo1")
]
double ImporteTributo1;
[helpstring("CodigoTributo2")
]
TiposTributos CodigoTributo2;
[helpstring("ImporteTributo2")
]
double ImporteTributo2;
[helpstring("CodigoTributo3")
]
TiposTributos CodigoTributo3;
[helpstring("ImporteTributo3")
]
double ImporteTributo3;
[helpstring("CodigoTributo4")
]
TiposTributos CodigoTributo4;
[helpstring("ImporteTributo4")
]
double ImporteTributo4;
[helpstring("CodigoTributo5")
]
TiposTributos CodigoTributo5;
[helpstring("ImporteTributo5")
]
double ImporteTributo5;
[helpstring("CodigoTributo6")
]
TiposTributos CodigoTributo6;
[helpstring("ImporteTributo6")
]
double ImporteTributo6;
[helpstring("CodigoTributo7")
]
TiposTributos CodigoTributo7;
[helpstring("ImporteTributo7")
]
```

```

double ImporteTributo7;
[helpstring("CodigoTributo8")
]
TiposTributos CodigoTributo8;
[helpstring("ImporteTributo8")
]
double ImporteTributo8;
[helpstring("CodigoTributo9")
]
TiposTributos CodigoTributo9;
[helpstring("ImporteTributo9")
]
double ImporteTributo9;
[helpstring("CodigoTributo10")
]
TiposTributos CodigoTributo10;
[helpstring("ImporteTributo10")
]
double ImporteTributo10;
[helpstring("CodigoTributo11")
]
TiposTributos CodigoTributo11;
[helpstring("ImporteTributo11")
]
double ImporteTributo11;
} ConsultarAcumuladosComprobanteRegDF;
    
```

CódigoComprobante

Informa el tipo de comprobante que se está reportando. Es alguno de los enumerados como *TiposComprobante*.

<i>ComprobanteDonacion</i>	907	<i>ReciboM</i>	54
<i>FacturaA</i>	1	<i>ReciboX</i>	902
<i>FacturaB</i>	6	<i>RemitoR</i>	91
<i>FacturaC</i>	11	<i>RemitoX</i>	901
<i>FacturaM</i>	51	<i>Tique</i>	83
<i>Generico</i>	910	<i>TiqueFacturaA</i>	81
<i>NotaDeCreditoA</i>	3	<i>TiqueFacturaB</i>	82
<i>NotaDeCreditoB</i>	8	<i>TiqueFacturaC</i>	111
<i>NotaDeCreditoC</i>	13	<i>TiqueFacturaM</i>	118
<i>NotaDeCreditoM</i>	53	<i>TiqueNotaCredito</i>	110
<i>NotaDeDebitoA</i>	2	<i>TiqueNotaCreditoA</i>	112
<i>NotaDeDebitoB</i>	7	<i>TiqueNotaCreditoB</i>	113
<i>NotaDeDebitoC</i>	12	<i>TiqueNotaCrediotC</i>	114
<i>NotaDeDebitoM</i>	52	<i>TiqueNotaCreditoM</i>	119
<i>PresupuestoX</i>	903	<i>TiqueNotaDebitoA</i>	115
<i>ReciboA</i>	4	<i>TiqueNotaDebitoB</i>	116

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

	<i>ReciboB</i>	9	<i>TiqueNotaDebitoC</i>	117
	<i>ReciboC</i>	15	<i>TiqueNotaDebitoM</i>	120
<i>NumeroInicial</i>	Informa el primer número de comprobante, para el rango reportado			
<i>NumeroFinal</i>	Informa el último número de comprobante, para el rango reportado.			
<i>CantidadCancelados</i>	Informa cuantos comprobantes, dentro del rango reportado, fueron cancelados.			
<i>Total</i>	Informa el importe total y acumulado para todos los comprobantes incluidos en el rango reportado.			
<i>TotalExento</i>	Informa el importe total exento y acumulado para todos los comprobantes incluidos en el rango reportado.			
<i>TotalNoGravado</i>	Informa el importe total no gravado y acumulado para todos los comprobantes incluidos en el rango reportado.			
<i>TotalGravado</i>	Informa el importe total gravado y acumulado para todos los comprobantes incluidos en el rango reportado.			
<i>TotalIVA</i>	Informa el importe total de IVA y acumulado para todos los comprobantes incluidos en el rango reportado.			
<i>TotalTributos</i>	Informa el importe total de IVA y acumulado para todos los comprobantes incluidos en el rango reportado.			
<i>AlicuotaIVA_1</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.			
<i>MontoIVA_1</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.			
<i>MontoNetoSinIVA_1</i>	Informa el importe total de base imponible y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.			
<i>AlicuotaIVA_2</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.			

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

<i>MontoIVA_2</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>MontoNetoSinIVA_2</i>	Informa el importe total de base imponible y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_3</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_3</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>MontoNetoSinIVA_3</i>	Informa el importe total de base imponible y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_4</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_4</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>MontoNetoSinIVA_4</i>	Informa el importe total de base imponible y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_5</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_5</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>MontoNetoSinIVA_5</i>	Informa el importe total de base imponible y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_6</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

<i>MontoIVA_6</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>MontoNetoSinIVA_6</i>	Informa el importe total de base imponible y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_7</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_7</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>MontoNetoSinIVA_7</i>	Informa el importe total de base imponible y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_8</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_8</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>MontoNetoSinIVA_8</i>	Informa el importe total de base imponible y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_9</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_9</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>MontoNetoSinIVA_9</i>	Informa el importe total de base imponible y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_10</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.

<i>.MontoIVA_10</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.																								
<i>MontoNetoSinIVA_10</i>	Informa el importe total de base imponible y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.																								
<i>CodigoTributo1</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i> .																								
	<table border="1"> <tr> <td><i>IIBB</i></td> <td>5</td> <td><i>OtrasPercepciones</i></td> <td>9</td> </tr> <tr> <td><i>ImpuestoInternoltem</i></td> <td>10</td> <td><i>OtrosTributos</i></td> <td>99</td> </tr> <tr> <td><i>ImpuestosInternos</i></td> <td>4</td> <td><i>PercepcionIIBB</i></td> <td>7</td> </tr> <tr> <td><i>ImpuestosMunicipales</i></td> <td>3</td> <td><i>PercepcionImpuestos Municipales</i></td> <td>8</td> </tr> <tr> <td><i>ImpuestosNacionales</i></td> <td>1</td> <td><i>PercepcionIVA</i></td> <td>6</td> </tr> <tr> <td><i>ImpuestosProvinciales</i></td> <td>2</td> <td><i>SinImpuestos</i></td> <td>0</td> </tr> </table>	<i>IIBB</i>	5	<i>OtrasPercepciones</i>	9	<i>ImpuestoInternoltem</i>	10	<i>OtrosTributos</i>	99	<i>ImpuestosInternos</i>	4	<i>PercepcionIIBB</i>	7	<i>ImpuestosMunicipales</i>	3	<i>PercepcionImpuestos Municipales</i>	8	<i>ImpuestosNacionales</i>	1	<i>PercepcionIVA</i>	6	<i>ImpuestosProvinciales</i>	2	<i>SinImpuestos</i>	0
<i>IIBB</i>	5	<i>OtrasPercepciones</i>	9																						
<i>ImpuestoInternoltem</i>	10	<i>OtrosTributos</i>	99																						
<i>ImpuestosInternos</i>	4	<i>PercepcionIIBB</i>	7																						
<i>ImpuestosMunicipales</i>	3	<i>PercepcionImpuestos Municipales</i>	8																						
<i>ImpuestosNacionales</i>	1	<i>PercepcionIVA</i>	6																						
<i>ImpuestosProvinciales</i>	2	<i>SinImpuestos</i>	0																						
<i>ImporteTributo1</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.																								
<i>CodigoTributo2</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i> . (ver tabla en <i>CodigoTributo1</i>)																								
<i>ImporteTributo2</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.																								
<i>CodigoTributo3</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i> . (ver tabla en <i>CodigoTributo1</i>)																								
<i>ImporteTributo3</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.																								
<i>CodigoTributo4</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los																								

	enumerados como <i>TiposTributos</i> . (ver tabla en <i>CodigoTributo1</i>)
<i>ImporteTributo4</i>	<i>Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.</i>
<i>CodigoTributo5</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i> . (ver tabla en <i>CodigoTributo1</i>)
<i>ImporteTributo5</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>CodigoTributo6</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i> . (ver tabla en <i>CodigoTributo1</i>)
<i>ImporteTributo6</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>CodigoTributo7</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i> . (ver tabla en <i>CodigoTributo1</i>)
<i>ImporteTributo7</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>CodigoTributo8</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i> . (ver tabla en <i>CodigoTributo1</i>)
<i>ImporteTributo8</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>CodigoTributo9</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los

	enumerados como <i>TiposTributos</i> .
	(ver tabla en <i>CodigoTributo1</i>)
<i>ImporteTributo9</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>CodigoTributo10</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i> .
	(ver tabla en <i>CodigoTributo1</i>)
<i>ImporteTributo10</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>CodigoTributo11</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i> .
	(ver tabla en <i>CodigoTributo1</i>)
<i>ImporteTributo11</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.

RegDNFH

Cuando *Registro = RegistroDetalladoDNFH* entonces la respuesta fiscal continúa con la siguiente estructura de información (así definida internamente):

```
typedef
struct tagConsultarAcumuladosComprobanteRegDNFH {
    [helpstring("CodigoComprobante")]
}
    TiposComprobante CodigoComprobante;
    [helpstring("NumeroInicial")]
}
    long NumeroInicial;
    [helpstring("NumeroFinal")]
}
    long NumeroFinal;
    [helpstring("Total")]
}
    double Total;
} ConsultarAcumuladosComprobanteRegDNFH;
```

<i>CodigoComprobante</i>	Informa el tipo de comprobante que se está reportando. Es alguno de los enumerados como <i>TiposComprobante</i> .
--------------------------	---

<i>ComprobanteDonacion</i>	907	<i>ReciboM</i>	54
----------------------------	-----	----------------	----

	<i>FacturaA</i>	1	<i>ReciboX</i>	902
	<i>FacturaB</i>	6	<i>RemitoR</i>	91
	<i>FacturaC</i>	11	<i>RemitoX</i>	901
	<i>FacturaM</i>	51	<i>Tique</i>	83
	<i>Generico</i>	910	<i>TiqueFacturaA</i>	81
	<i>NotaDeCreditoA</i>	3	<i>TiqueFacturaB</i>	82
	<i>NotaDeCreditoB</i>	8	<i>TiqueFacturaC</i>	111
	<i>NotaDeCreditoC</i>	13	<i>TiqueFacturaM</i>	118
	<i>NotaDeCreditoM</i>	53	<i>TiqueNotaCredito</i>	110
	<i>NotaDeDebitoA</i>	2	<i>TiqueNotaCreditoA</i>	112
	<i>NotaDeDebitoB</i>	7	<i>TiqueNotaCreditoB</i>	113
	<i>NotaDeDebitoC</i>	12	<i>TiqueNotaCreditoC</i>	114
	<i>NotaDeDebitoM</i>	52	<i>TiqueNotaCreditoM</i>	119
	<i>PresupuestoX</i>	903	<i>TiqueNotaDebitoA</i>	115
	<i>ReciboA</i>	4	<i>TiqueNotaDebitoB</i>	116
	<i>ReciboB</i>	9	<i>TiqueNotaDebitoC</i>	117
	<i>ReciboC</i>	15	<i>TiqueNotaDebitoM</i>	120
<i>NumeroInicial</i>	Informa el primer número de comprobante, para el rango de los emitidos durante la jornada fiscal/comercial.			
<i>NumeroFinal</i>	Informa el último número de comprobante, para el rango de los emitidos durante la jornada fiscal/comercial.			
<i>Total</i>	Informa el importe total y acumulado para todos los comprobantes incluidos en el rango reportado.			

RegDNFH_NoAcum

Quando *Registro = RegistroDetalladoDNFHNoAcum* entonces la respuesta fiscal continúa con la siguiente estructura de información (así definida internamente):

```
typedef
struct tagConsultarAcumuladosComprobanteRegDNFH_NoAcum {
    [helpstring("CodigoComprobante")]
}
    TiposComprobante CodigoComprobante;
    [helpstring("NumeroInicial")]
}
    long NumeroInicial;
    [helpstring("NumeroFinal")]
}
    long NumeroFinal;
} ConsultarAcumuladosComprobanteRegDNFH_NoAcum;
```

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

<i>CodigoComprobante</i>	Informa el tipo de comprobante que se está reportando. Es alguno de los enumerados como <i>TiposComprobante</i> .			
	<i>ComprobanteDonacion</i>	907	<i>ReciboM</i>	54
	<i>FacturaA</i>	1	<i>ReciboX</i>	902
	<i>FacturaB</i>	6	<i>RemitoR</i>	91
	<i>FacturaC</i>	11	<i>RemitoX</i>	901
	<i>FacturaM</i>	51	<i>Tique</i>	83
	<i>Generico</i>	910	<i>TiqueFacturaA</i>	81
	<i>NotaDeCreditoA</i>	3	<i>TiqueFacturaB</i>	82
	<i>NotaDeCreditoB</i>	8	<i>TiqueFacturaC</i>	111
	<i>NotaDeCreditoC</i>	13	<i>TiqueFacturaM</i>	118
	<i>NotaDeCreditoM</i>	53	<i>TiqueNotaCredito</i>	110
	<i>NotaDeDebitoA</i>	2	<i>TiqueNotaCreditoA</i>	112
	<i>NotaDeDebitoB</i>	7	<i>TiqueNotaCreditoB</i>	113
	<i>NotaDeDebitoC</i>	12	<i>TiqueNotaCreditoC</i>	114
	<i>NotaDeDebitoM</i>	52	<i>TiqueNotaCreditoM</i>	119
	<i>PresupuestoX</i>	903	<i>TiqueNotaDebitoA</i>	115
	<i>ReciboA</i>	4	<i>TiqueNotaDebitoB</i>	116
	<i>ReciboB</i>	9	<i>TiqueNotaDebitoC</i>	117
	<i>ReciboC</i>	15	<i>TiqueNotaDebitoM</i>	120
<i>NumeroInicial</i>	Informa el primer número de comprobante, para el rango de los emitidos durante la jornada fiscal/comercial.			
<i>NumeroFinal</i>	Informa el último número de comprobante, para el rango de los emitidos durante la jornada fiscal/comercial.			

RegGlobal

Cuando *Registro = RegistroGlobal* entonces la respuesta fiscal continúa con la siguiente estructura de información (así definida internamente):

```
typedef
struct tagConsultarAcumuladosComprobanteRegGlobal {
    [helpstring("DF_CantidadCancelados")]
    long DF_CantidadCancelados;
    [helpstring("DF_CantidadEmitidos")]
    long DF_CantidadEmitidos;
    [helpstring("DF_Total")]
    long DF_Total;
}
```

```

double DF_Total;
[helpstring("DF_TotalExento")
]
double DF_TotalExento;
[helpstring("DF_TotalNoGravado")
]
double DF_TotalNoGravado;
[helpstring("DF_TotalGravado")
]
double DF_TotalGravado;
[helpstring("DF_TotalIVA")
]
double DF_TotalIVA;
[helpstring("DF_TotalTributos")
]
double DF_TotalTributos;
[helpstring("DNFH_Total")
]
double DNFH_Total;
} ConsultarAcumuladosComprobanteRegGlobal;
    
```

<i>DF_CantidadCancelados</i>	Informa la cantidad total de documentos fiscales (sin distinguir tipo) cancelados dentro del rango reportado.
<i>DF_CantidadEmitidos</i>	Informa la cantidad total de documentos fiscales (sin distinguir tipo) emitidos dentro del rango reportado.
<i>DF_Total</i>	Informa el monto total de ventas acumulado dentro del rango reportado. No incluye <i>(Tiques) Nota de Crédito</i> .
<i>DF_TotalExento</i>	Informa el monto total de bases imponibles exentas, acumulado dentro del rango reportado. No incluye <i>(Tiques) Nota de Crédito</i> .
<i>DF_TotalNoGravado</i>	Informa el monto total de bases imponibles no gravadas, acumulado dentro del rango reportado. No incluye <i>(Tiques) Nota de Crédito</i> .
<i>DF_TotalGravado</i>	Informa el monto total de bases imponibles gravadas, acumulado dentro del rango reportado. No incluye <i>(Tiques) Nota de Crédito</i> .
<i>DF_TotalIVA</i>	Informa el monto total de IVA, acumulado durante la jornada fiscal/comercial entre todas las alícuotas. No incluye <i>(Tiques) Nota de Crédito</i> .
<i>DF_TotalTributos</i>	Informa el monto total de tributos, acumulado dentro del rango reportado entre todas los tributos percibidos. No incluye <i>(Tiques) Nota de Crédito</i> .

DNFH_Total

Informa el monto total acumulado dentro del rango reportado entre todos los documentos no fiscales homologados que generan totales.

ConsultarAcumuladosMemoriaDeTrabajo()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta de numeradores alcanzados, montos acumulados de IVA y montos acumulados de otros tributos; todos datos correspondientes a la memoria de trabajo, es decir, a la jornada fiscal/comercial en curso. Como resultado se obtiene el primer registro de información disponible.

Definido internamente como:

```
ConsultarAcumuladosMemoriaDeTrabajo(
    [in, optional, defaultvalue(0)] TiposComprobante CodigoComprobante,
    [out, retval] RespuestaConsultarAcumuladosMemoriaDeTrabajo* );
```

CodigoComprobante

Permite especificar el tipo de comprobante cuyos acumulados en la memoria de trabajo se desean consultar. Si se especifica 'NoDocumento' se obtiene información de todos los tipos de comprobantes almacenados durante la jornada fiscal/comercial, hasta el momento de la consulta. Debe ser alguna de las constantes pertenecientes al enum *TiposComprobante*.

***	<i>ComprobanteDonacion</i>	907	***	<i>ReciboM</i>	54
*	<i>FacturaA</i>	1	***	<i>ReciboX</i>	902
*	<i>FacturaB</i>	6	***	<i>RemitoR</i>	91
*	<i>FacturaC</i>	11	***	<i>RemitoX</i>	901
*	<i>FacturaM</i>	51	**	<i>Tique</i>	83
***	<i>Generico</i>	910	**	<i>TiqueFacturaA</i>	81
*	<i>NotaDeCreditoA</i>	3	**	<i>TiqueFacturaB</i>	82
*	<i>NotaDeCreditoB</i>	8	**	<i>TiqueFacturaC</i>	111
*	<i>NotaDeCreditoC</i>	13	**	<i>TiqueFacturaM</i>	118
*	<i>NotaDeCreditoM</i>	53	**	<i>TiqueNotaCredito</i>	110
*	<i>NotaDeDebitoA</i>	2	**	<i>TiqueNotaCreditoA</i>	112
*	<i>NotaDeDebitoB</i>	7	**	<i>TiqueNotaCreditoB</i>	113
*	<i>NotaDeDebitoC</i>	12	**	<i>TiqueNotaCreditoC</i>	114
*	<i>NotaDeDebitoM</i>	52	**	<i>TiqueNotaCreditoM</i>	119
***	<i>PresupuestoX</i>	903	**	<i>TiqueNotaDebitoA</i>	115
***	<i>ReciboA</i>	4	**	<i>TiqueNotaDebitoB</i>	116
***	<i>ReciboB</i>	9	**	<i>TiqueNotaDebitoC</i>	117

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

*** ReciboC	15	** TiqueNotaDebitoM	120
*	Disponibles en impresoras fiscales 2G que soporten impresión sobre formularios continuos u hojas sueltas.		
**	Disponibles en impresoras fiscales 2G que soporten impresión sobre rollos.		
***	Disponibles en todos los modelos de impresoras fiscales 2G.		

Iniciada la consulta de información con este comando fiscal, el resto de la información se obtiene mediante el comando fiscal generado por `'ContinuarConsultaAcumulados()'`.

Ejemplo:

```
Dim memo1 As HasarArgentina.RespuestaConsultarAcumuladosMemoriaDeTrabajo

memo1 = hasar.ConsultarAcumuladosMemoriaDeTrabajo(HasarArgentina.TiposComprobante.NoDocumento)

If (memo1.Registro = HasarArgentina.TiposDeRegistroInforme.RegistroFinal) Then
    Debug.Print("No hay información disponible ...")
    Exit Sub
End If

If (memo1.Registro = HasarArgentina.TiposDeRegistroInforme.RegistroDetalladoDNFH) Then
    Debug.Print("Cód. Comprob. =[ " & memo1.RegDNFH.CodigoComprobante & " ]")
    Debug.Print("Desde Nº      =[ " & memo1.RegDNFH.NumeroInicial & " ]")
    Debug.Print("Hasta Nº       =[ " & memo1.RegDNFH.NumeroFinal & " ]")
    Debug.Print("")
    Debug.Print("TOTAL          =[ " & memo1.RegDNFH.Total & " ]")
    Debug.Print("")
End If
```

La respuesta fiscal al comando fiscal generado por `'ConsultarAcumuladosMemoriaDeTrabajo()'` es la estructura `RespuestaConsultarAcumuladosmemoriaDeTrabajo`, definida internamente como:

```
typedef
struct tagRespuestaConsultarAcumuladosMemoriaDeTrabajo {
    [helpstring("Registro")]
}
    TiposDeRegistroInforme Registro;
    [helpstring("RegDF")]
}
    ConsultarAcumuladosMemoriaDeTrabajoRegDF RegDF;
    [helpstring("RegDNFH")]
}
    ConsultarAcumuladosMemoriaDeTrabajoRegDNFH RegDNFH;
    [helpstring("RegDNFH_NoAcum")]
}
```

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

```

ConsultarAcumuladosMemoriaDeTrabajoRegDNFH_NoAcum RegDNFH_NoAcum;
[helpstring("RegGlobal")
]
ConsultarAcumuladosMemoriaDeTrabajoRegGlobal RegGlobal;
} RespuestaConsultarAcumuladosMemoriaDeTrabajo;

```

Registro

Informa el tipo de registro que se está reportando, que puede ser alguno de los incluidos en el enum *TiposDeRegistroInforme*.

<i>RegistroFinal</i>	48
<i>RegistroDetalladoDF</i>	49
<i>RegistroDetalladoDNFH</i>	50
<i>RegistroDetalladoDNFHNoAcum</i>	51
<i>RegistroGlobal</i>	52

RegDF

Cuando *Registro = RegistroDetalloDF* entonces la respuesta fiscal continua con la siguiente estructura de información (así definida internamente):

```

typedef
struct tagConsultarAcumuladosMemoriaDeTrabajoRegDF {
    [helpstring("CodigoComprobante")
]
    TiposComprobante CodigoComprobante;
    [helpstring("NumeroInicial")
]
    long NumeroInicial;
    [helpstring("NumeroFinal")
]
    long NumeroFinal;
    [helpstring("CantidadCancelados")
]
    long CantidadCancelados;
    [helpstring("Total")
]
    double Total;
    [helpstring("TotalExento")
]
    double TotalExento;
    [helpstring("TotalNoGravado")
]
    double TotalNoGravado;
    [helpstring("TotalGravado")
]
    double TotalGravado;
    [helpstring("TotalIVA")
]
    double TotalIVA;
    [helpstring("TotalTributos")
]
    double TotalTributos;
    [helpstring("AlicuotaIVA_1")
]
}

```

```
double AlicuotaIVA_1;  
[helpstring("MontoIVA_1")  
]  
double MontoIVA_1;  
[helpstring("AlicuotaIVA_2")  
]  
double AlicuotaIVA_2;  
[helpstring("MontoIVA_2")  
]  
double MontoIVA_2;  
[helpstring("AlicuotaIVA_3")  
]  
double AlicuotaIVA_3;  
[helpstring("MontoIVA_3")  
]  
double MontoIVA_3;  
[helpstring("AlicuotaIVA_4")  
]  
double AlicuotaIVA_4;  
[helpstring("MontoIVA_4")  
]  
double MontoIVA_4;  
[helpstring("AlicuotaIVA_5")  
]  
double AlicuotaIVA_5;  
[helpstring("MontoIVA_5")  
]  
double MontoIVA_5;  
[helpstring("AlicuotaIVA_6")  
]  
double AlicuotaIVA_6;  
[helpstring("MontoIVA_6")  
]  
double MontoIVA_6;  
[helpstring("AlicuotaIVA_7")  
]  
double AlicuotaIVA_7;  
[helpstring("MontoIVA_7")  
]  
double MontoIVA_7;  
[helpstring("AlicuotaIVA_8")  
]  
double AlicuotaIVA_8;  
[helpstring("MontoIVA_8")  
]  
double MontoIVA_8;  
[helpstring("AlicuotaIVA_9")  
]  
double AlicuotaIVA_9;  
[helpstring("MontoIVA_9")  
]  
double MontoIVA_9;  
[helpstring("AlicuotaIVA_10")  
]  
]
```

```
double AlicuotaIVA_10;  
[helpstring("MontoIVA_10")  
]  
double MontoIVA_10;  
[helpstring("CodigoTributo1")  
]  
TiposTributos CodigoTributo1;  
[helpstring("ImporteTributo1")  
]  
double ImporteTributo1;  
[helpstring("CodigoTributo2")  
]  
TiposTributos CodigoTributo2;  
[helpstring("ImporteTributo2")  
]  
double ImporteTributo2;  
[helpstring("CodigoTributo3")  
]  
TiposTributos CodigoTributo3;  
[helpstring("ImporteTributo3")  
]  
double ImporteTributo3;  
[helpstring("CodigoTributo4")  
]  
TiposTributos CodigoTributo4;  
[helpstring("ImporteTributo4")  
]  
double ImporteTributo4;  
[helpstring("CodigoTributo5")  
]  
TiposTributos CodigoTributo5;  
[helpstring("ImporteTributo5")  
]  
double ImporteTributo5;  
[helpstring("CodigoTributo6")  
]  
TiposTributos CodigoTributo6;  
[helpstring("ImporteTributo6")  
]  
double ImporteTributo6;  
[helpstring("CodigoTributo7")  
]  
TiposTributos CodigoTributo7;  
[helpstring("ImporteTributo7")  
]  
double ImporteTributo7;  
[helpstring("CodigoTributo8")  
]  
TiposTributos CodigoTributo8;  
[helpstring("ImporteTributo8")  
]  
double ImporteTributo8;  
[helpstring("CodigoTributo9")  
]  
double ImporteTributo9;  
[helpstring("CodigoTributo10")  
]
```

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

```

TiposTributos CodigoTributo10;
[helpstring("ImporteTributo10")
]
double ImporteTributo10;
[helpstring("CodigoTributo11")
]
TiposTributos CodigoTributo11;
[helpstring("ImporteTributo11")
]
double ImporteTributo11;
} ConsultarAcumuladosMemoriaDeTrabajoRegDF;

```

CódigoComprobante Informa el tipo de comprobante que se está reportando. Es alguno de los enumerados como *TiposComprobante*.

<i>ComprobanteDonacion</i>	907	<i>ReciboM</i>	54
<i>FacturaA</i>	1	<i>ReciboX</i>	902
<i>FacturaB</i>	6	<i>RemitoR</i>	91
<i>FacturaC</i>	11	<i>RemitoX</i>	901
<i>FacturaM</i>	51	<i>Tique</i>	83
<i>Generico</i>	910	<i>TiqueFacturaA</i>	81
<i>NotaDeCreditoA</i>	3	<i>TiqueFacturaB</i>	82
<i>NotaDeCreditoB</i>	8	<i>TiqueFacturaC</i>	111
<i>NotaDeCreditoC</i>	13	<i>TiqueFacturaM</i>	118
<i>NotaDeCreditoM</i>	53	<i>TiqueNotaCredito</i>	110
<i>NotaDeDebitoA</i>	2	<i>TiqueNotaCreditoA</i>	112
<i>NotaDeDebitoB</i>	7	<i>TiqueNotaCreditoB</i>	113
<i>NotaDeDebitoC</i>	12	<i>TiqueNotaCrediotC</i>	114
<i>NotaDeDebitoM</i>	52	<i>TiqueNotaCreditoM</i>	119
<i>PresupuestoX</i>	903	<i>TiqueNotaDebitoA</i>	115
<i>ReciboA</i>	4	<i>TiqueNotaDebitoB</i>	116
<i>ReciboB</i>	9	<i>TiqueNotaDebitoC</i>	117
<i>ReciboC</i>	15	<i>TiqueNotaDebitoM</i>	120

NúmeroInicial Informa el primer número de comprobante, para el rango de los emitidos hasta el momento de la consulta durante la jornada fiscal/comercial.

NúmeroFinal Informa el último número de comprobante, para el rango de los emitidos hasta el momento de la consulta durante la jornada fiscal/comercial.

CantidadCancelados Informa cuantos comprobantes, dentro del rango reportado,

	fueron cancelados.
<i>Total</i>	Informa el importe total y acumulado para todos los comprobantes incluidos en el rango reportado.
<i>TotalExento</i>	Informa el importe total exento y acumulado para todos los comprobantes incluidos en el rango reportado.
<i>TotalNoGravado</i>	Informa el importe total no gravado y acumulado para todos los comprobantes incluidos en el rango reportado.
<i>TotalGravado</i>	Informa el importe total gravado y acumulado para todos los comprobantes incluidos en el rango reportado.
<i>TotalIVA</i>	Informa el importe total de IVA y acumulado para todos los comprobantes incluidos en el rango reportado.
<i>TotalTributos</i>	Informa el importe total de IVA y acumulado para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_1</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_1</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_2</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_2</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_3</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_3</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_4</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.

<i>MontoIVA_4</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_5</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_5</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_6</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_6</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_7</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_7</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_8</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_8</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_9</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>MontoIVA_9</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>AlicuotaIVA_10</i>	Informa una de las alícuotas de IVA que conforma algunos de los comprobantes incluidos en el rango reportado.
<i>.MontoIVA_10</i>	Informa el importe total de IVA y acumulado, para la alícuota en cuestión y para todos los comprobantes incluidos en el rango

	reportado.																								
<i>CodigoTributo1</i>	<p>Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i>.</p> <table border="1"> <tr> <td><i>IIBB</i></td> <td>5</td> <td><i>OtrasPercepciones</i></td> <td>9</td> </tr> <tr> <td><i>ImpuestoInternoltem</i></td> <td>10</td> <td><i>OtrosTributos</i></td> <td>99</td> </tr> <tr> <td><i>ImpuestosInternos</i></td> <td>4</td> <td><i>PercepcionIIBB</i></td> <td>7</td> </tr> <tr> <td><i>ImpuestosMunicipales</i></td> <td>3</td> <td><i>PercepcionImpuestos Municipales</i></td> <td>8</td> </tr> <tr> <td><i>ImpuestosNacionales</i></td> <td>1</td> <td><i>PercepcionIVA</i></td> <td>6</td> </tr> <tr> <td><i>ImpuestosProvinciales</i></td> <td>2</td> <td><i>SinImpuestos</i></td> <td>0</td> </tr> </table>	<i>IIBB</i>	5	<i>OtrasPercepciones</i>	9	<i>ImpuestoInternoltem</i>	10	<i>OtrosTributos</i>	99	<i>ImpuestosInternos</i>	4	<i>PercepcionIIBB</i>	7	<i>ImpuestosMunicipales</i>	3	<i>PercepcionImpuestos Municipales</i>	8	<i>ImpuestosNacionales</i>	1	<i>PercepcionIVA</i>	6	<i>ImpuestosProvinciales</i>	2	<i>SinImpuestos</i>	0
<i>IIBB</i>	5	<i>OtrasPercepciones</i>	9																						
<i>ImpuestoInternoltem</i>	10	<i>OtrosTributos</i>	99																						
<i>ImpuestosInternos</i>	4	<i>PercepcionIIBB</i>	7																						
<i>ImpuestosMunicipales</i>	3	<i>PercepcionImpuestos Municipales</i>	8																						
<i>ImpuestosNacionales</i>	1	<i>PercepcionIVA</i>	6																						
<i>ImpuestosProvinciales</i>	2	<i>SinImpuestos</i>	0																						
<i>ImporteTributo1</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.																								
<i>CodigoTributo2</i>	<p>Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i>.</p> <p align="center">(ver tabla en <i>CodigoTributo1</i>)</p>																								
<i>ImporteTributo2</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.																								
<i>CodigoTributo3</i>	<p>Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i>.</p> <p align="center">(ver tabla en <i>CodigoTributo1</i>)</p>																								
<i>ImporteTributo3</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.																								
<i>CodigoTributo4</i>	<p>Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i>.</p> <p align="center">(ver tabla en <i>CodigoTributo1</i>)</p>																								
<i>ImporteTributo4</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.																								

<i>CodigoTributo5</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i> . (ver tabla en <i>CodigoTributo1</i>)
<i>ImporteTributo5</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>CodigoTributo6</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i> . (ver tabla en <i>CodigoTributo1</i>)
<i>ImporteTributo6</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>CodigoTributo7</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i> . (ver tabla en <i>CodigoTributo1</i>)
<i>ImporteTributo7</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>CodigoTributo8</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i> . (ver tabla en <i>CodigoTributo1</i>)
<i>ImporteTributo8</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.
<i>CodigoTributo9</i>	Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i> . (ver tabla en <i>CodigoTributo1</i>)
<i>ImporteTributo9</i>	Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

<i>CodigoTributo10</i>	<p>Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i>.</p> <p align="center">(ver tabla en <i>CodigoTributo1</i>)</p>
<i>ImporteTributo10</i>	<p>Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.</p>
<i>CodigoTributo11</i>	<p>Informa un código de tributo percibido en algunos de los comprobantes incluidos en el rango reportado. Es alguno de los enumerados como <i>TiposTributos</i>.</p> <p align="center">(ver tabla en <i>CodigoTributo1</i>)</p>
<i>ImporteTributo11</i>	<p>Informa el importe total acumulado, para el tributo en cuestión y para todos los comprobantes incluidos en el rango reportado.</p>

RegDNFH

Quando *Registro = RegistroDetalladoDNFH* entonces la respuesta fiscal continúa con la siguiente estructura de información (así definida internamente):

```
typedef
struct tagConsultarAcumuladosCierreZetaRegDNFH {
    [helpstring("CodigoComprobante")]
}
    TiposComprobante CodigoComprobante;
    [helpstring("NumeroInicial")]
}
    long NumeroInicial;
    [helpstring("NumeroFinal")]
}
    long NumeroFinal;
    [helpstring("Total")]
}
    double Total;
} ConsultarAcumuladosCierreZetaRegDNFH;
```

<i>CodigoComprobante</i>	<p>Informa el tipo de comprobante que se está reportando. Es alguno de los enumerados como <i>TiposComprobante</i>.</p>		
<i>ComprobanteDonacion</i>	907	<i>ReciboM</i>	54
<i>FacturaA</i>	1	<i>ReciboX</i>	902
<i>FacturaB</i>	6	<i>RemitoR</i>	91
<i>FacturaC</i>	11	<i>RemitoX</i>	901
<i>FacturaM</i>	51	<i>Tique</i>	83
<i>Generico</i>	910	<i>TiqueFacturaA</i>	81

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

	<i>NotaDeCreditoA</i>	3	<i>TiqueFacturaB</i>	82
	<i>NotaDeCreditoB</i>	8	<i>TiqueFacturaC</i>	111
	<i>NotaDeCreditoC</i>	13	<i>TiqueFacturaM</i>	118
	<i>NotaDeCreditoM</i>	53	<i>TiqueNotaCredito</i>	110
	<i>NotaDeDebitoA</i>	2	<i>TiqueNotaCreditoA</i>	112
	<i>NotaDeDebitoB</i>	7	<i>TiqueNotaCreditoB</i>	113
	<i>NotaDeDebitoC</i>	12	<i>TiqueNotaCrediotC</i>	114
	<i>NotaDeDebitoM</i>	52	<i>TiqueNotaCreditoM</i>	119
	<i>PresupuestoX</i>	903	<i>TiqueNotaDebitoA</i>	115
	<i>ReciboA</i>	4	<i>TiqueNotaDebitoB</i>	116
	<i>ReciboB</i>	9	<i>TiqueNotaDebitoC</i>	117
	<i>ReciboC</i>	15	<i>TiqueNotaDebitoM</i>	120
<i>NumeroInicial</i>	Informa el primer número de comprobante, para el rango de los emitidos durante la jornada fiscal/comercial.			
<i>NumeroFinal</i>	Informa el último número de comprobante, para el rango de los emitidos durante la jornada fiscal/comercial.			
<i>Total</i>	Informa el importe total y acumulado para todos los comprobantes incluidos en el rango reportado.			
<i>RegDNFH_NoAcum</i>	Cuando <i>Registro = RegistroDetalladoDNFHNoAcum</i> entonces la respuesta fiscal continúa con la siguiente estructura de información (así definida internamente):			
	<pre>typedef struct tagConsultarAcumuladosCierreZetaRegDNFH_NoAcum { [helpstring("CodigoComprobante")] } TiposComprobante CodigoComprobante; [helpstring("NumeroInicial")] } long NumeroInicial; [helpstring("NumeroFinal")] } long NumeroFinal; } ConsultarAcumuladosCierreZetaRegDNFH_NoAcum;</pre>			
<i>CodigoComprobante</i>	Informa el tipo de comprobante que se está reportando. Es alguno de los enumerados como <i>TiposComprobante</i> .			
	<i>ComprobanteDonacion</i>	907	<i>ReciboM</i>	54
	<i>FacturaA</i>	1	<i>ReciboX</i>	902

	<i>FacturaB</i>	6	<i>RemitoR</i>	91
	<i>FacturaC</i>	11	<i>RemitoX</i>	901
	<i>FacturaM</i>	51	<i>Tique</i>	83
	<i>Generico</i>	910	<i>TiqueFacturaA</i>	81
	<i>NotaDeCreditoA</i>	3	<i>TiqueFacturaB</i>	82
	<i>NotaDeCreditoB</i>	8	<i>TiqueFacturaC</i>	111
	<i>NotaDeCreditoC</i>	13	<i>TiqueFacturaM</i>	118
	<i>NotaDeCreditoM</i>	53	<i>TiqueNotaCredito</i>	110
	<i>NotaDeDebitoA</i>	2	<i>TiqueNotaCreditoA</i>	112
	<i>NotaDeDebitoB</i>	7	<i>TiqueNotaCreditoB</i>	113
	<i>NotaDeDebitoC</i>	12	<i>TiqueNotaCreditoC</i>	114
	<i>NotaDeDebitoM</i>	52	<i>TiqueNotaCreditoM</i>	119
	<i>PresupuestoX</i>	903	<i>TiqueNotaDebitoA</i>	115
	<i>ReciboA</i>	4	<i>TiqueNotaDebitoB</i>	116
	<i>ReciboB</i>	9	<i>TiqueNotaDebitoC</i>	117
	<i>ReciboC</i>	15	<i>TiqueNotaDebitoM</i>	120
<i>NumeroInicial</i>	Informa el primer número de comprobante, para el rango de los emitidos durante la jornada fiscal/comercial.			
<i>NumeroFinal</i>	Informa el último número de comprobante, para el rango de los emitidos durante la jornada fiscal/comercial.			

RegGlobal

Cuando *Registro = RegistroGlobal* entonces la respuesta fiscal continúa con la siguiente estructura de información (así definida internamente):

```
typedef
struct tagConsultarAcumuladosCierreZetaRegGlobal {
    [helpstring("DF_CantidadCancelados")]
    ]
    long DF_CantidadCancelados;
    [helpstring("DF_CantidadEmitidos")]
    ]
    long DF_CantidadEmitidos;
    [helpstring("DF_Total")]
    ]
    double DF_Total;
    [helpstring("DF_TotalExento")]
    ]
    double DF_TotalExento;
    [helpstring("DF_TotalNoGravado")]
    ]
}
```

```

double DF_TotalNoGravado;
[helpstring("DF_TotalGravado")
]
double DF_TotalGravado;
[helpstring("DF_TotalIVA")
]
double DF_TotalIVA;
[helpstring("DF_TotalTributos")
]
double DF_TotalTributos;
    
```

<i>DF_CantidadCancelados</i>	Informa la cantidad total de documentos fiscales (sin distinguir tipo) cancelados durante la jornada fiscal/comercial, hasta el momento de la consulta.
<i>DF_CantidadEmitidos</i>	Informa la cantidad total de documentos fiscales (sin distinguir tipo) emitidos durante la jornada fiscal/comercial, hasta el momento de la consulta.
<i>DF_Total</i>	Informa el monto total de ventas acumulado durante la jornada fiscal/comercial, hasta el momento de la consulta. No incluye <i>(Tiques) Nota de Crédito</i> .
<i>DF_TotalExento</i>	Informa el monto total de bases imponibles exentas, acumulado durante la jornada fiscal/comercial, hasta el momento de la consulta. No incluye <i>(Tiques) Nota de Crédito</i> .
<i>DF_TotalNoGravado</i>	Informa el monto total de bases imponibles no gravadas, acumulado durante la jornada fiscal/comercial, hasta el momento de la consulta. No incluye <i>(Tiques) Nota de Crédito</i> .
<i>DF_TotalGravado</i>	Informa el monto total de bases imponibles gravadas, acumulado durante la jornada fiscal/comercial, hasta el momento de la consulta. No incluye <i>(Tiques) Nota de Crédito</i> .
<i>DF_TotalIVA</i>	Informa el monto total de IVA, acumulado durante la jornada fiscal/comercial entre todas las alícuotas, hasta el momento de la consulta. No incluye <i>(Tiques) Nota de Crédito</i> .
<i>DF_TotalTributos</i>	Informa el monto total de tributos, acumulado durante la jornada fiscal/comercial entre todos los tributos percibidos, hasta el momento de la consulta. No incluye <i>(Tiques) Nota de Crédito</i> .
<i>DNFH_Total</i>	Informa el monto total acumulado durante la jornada fiscal/comercial entre todos los documentos no fiscales ho-

mologados que generan totales, hasta el momento de la consulta.

ConsultarCantidadIvasOtrosTributos()

No disponible en los modelos: SMH/PT-1000F v01.00 y SMH/PT-250F v01.00.

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta de la máxima cantidad de alícuotas de IVA y de otros tributos soportados por el comprobante en curso. El comando fiscal generado por 'ConsultarCantidadIvasOtrosTributos()' será rechazado si el comprobante en curso es un *Documento Genérico*, o no hay un comprobante abierto.

Definido internamente como:

```
ConsultarCantidadIvasOtrosTributos([out, retval] RespuestaConsultarCantidadIvasOtrosTributos* );
```

Ejemplo:

```
Dim trib As HasarArgentina.RespuestaConsultarCantidadIvasOtrosTributos  
  
trib = hasar.ConsultarCantidadIvasOtrosTributos()  
Debug.Print("Soporta " & trib.CantidadAlicuotasIVA & " Alícuotas IVA")  
Debug.Print("Soporta " & trib.CantidadOtrosTributos & " Otros tributos")
```

El valor cero como cantidad de otros tributos, en la respuesta al comando fiscal generado por 'ConsultarCantidadIvasOtrosTributos()', indica que no hay límite en la cantidad de veces que puede enviarse el comando fiscal generado por 'ImprimirOtrosTributos()', en el comprobante en curso.

La respuesta fiscal al comando fiscal generado por 'ConsultarCantidadIvasOtrosTributos()' es la estructura *RespuestaConsultarCantidadIvasOtrosTributos*, definida internamente como:

```
typedef  
struct tagRespuestaConsultarCantidadIvasOtrosTributos {  
    [helpstring("CantidadAlicuotasIVA")  
]  
    long CantidadAlicuotasIVA;  
    [helpstring("CantidadOtrosTributos")  
]  
    long CantidadOtrosTributos;  
} RespuestaConsultarCantidadIvasOtrosTributos;
```

CantidadAlicuotasIVA

Informa la cantidad máxima de alícuotas IVA soportadas, correspondiente a la condición de IVA Gravado. Si el comprobante en curso es un DNFH (Documento No Fiscal Homologado) -se excluye al *Documento Genérico*- en el cual el IVA no tiene relevancia impresa -por ejemplo un *Remito 'X'/'R'*-, este campo se informará en cero.

CantidadOtrosTributos

Informa la cantidad máxima soportada de otros tributos a informar por el sistema de facturación -no incluye a los Impuestos Internos Items-. Si el comprobante en curso no tuviese límite en la cantidad de otros tributos soportados, este campo se informará en cero. Si para el comprobante en curso la carga de otros tributos fuese una operación no permitida -es el caso de *Remitos 'X'/R* y *Documento Donación*-, este campo se informará en cero.

ConsultarCapacidadesImpresoraFiscal()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta de las características soportadas; propias del modelo y/o versión encuestado.

Definido internamente como:

```
ConsultarCapacidadesImpresoraFiscal(
    [in] Capacidades Capacidad,
    [in, optional, defaultvalue(68)] TiposDeEstacion Estacion,
    [in, optional, defaultvalue(0)] TiposComprobante CodigoComprobante,
    [out, retval] RespuestaConsultarCapacidadesImpresoraFiscal* );
```

Capacidad

Permite especificar la capacidad supuestamente soportada por la impresora fiscal 2G, sobre la cual se desea consultar. Debe ser alguna de las constantes pertenecientes al enum *Capacidades*.

<i>SoportaComprobante</i>	67
<i>SoportaCajon</i>	68
<i>SoportaEstacion</i>	69
<i>AnchoTotalImpresion</i>	73
<i>AnchoLogo</i>	76
<i>AnchoTextoNoFiscal</i>	78
<i>AnchoRazonSocial</i>	82
<i>AnchoTextoFiscal</i>	84
<i>AnchoTextoLineasUsuario</i>	85
<i>AnchoTextoVenta</i>	86
<i>EstacionPrincipal</i>	101
<i>AltoLogo</i>	108

Estacion

Permite especificar la estación de impresión sobre la cual se consulta por la validez y/o valor de la capacidad de interés. Debe ser alguna de las constantes pertenecientes al enum *TiposDeEstacion*.

<i>EstacionPorDefecto</i>	68
<i>EstacionSlip</i>	83

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

EstacionTicket

84

CodigoComprobante

Permite especificar el tipo de comprobante, que puede poseer la capacidad que se consulta, y que se está emitiendo por la estación de impresión indicada en el argumento *Estacion*. Debe ser alguna de las constantes pertenecientes al enum *TiposComprobante*.

***	<i>ComprobanteDonacion</i>	907	***	<i>ReciboM</i>	54
*	<i>FacturaA</i>	1	***	<i>ReciboX</i>	902
*	<i>FacturaB</i>	6	***	<i>RemitoR</i>	91
*	<i>FacturaC</i>	11	***	<i>RemitoX</i>	901
*	<i>FacturaM</i>	51	**	<i>Tique</i>	83
***	<i>Generico</i>	910	**	<i>TiqueFacturaA</i>	81
*	<i>NotaDeCreditoA</i>	3	**	<i>TiqueFacturaB</i>	82
*	<i>NotaDeCreditoB</i>	8	**	<i>TiqueFacturaC</i>	111
*	<i>NotaDeCreditoC</i>	13	**	<i>TiqueFacturaM</i>	118
*	<i>NotaDeCreditoM</i>	53	**	<i>TiqueNotaCredito</i>	110
*	<i>NotaDeDebitoA</i>	2	**	<i>TiqueNotaCreditoA</i>	112
*	<i>NotaDeDebitoB</i>	7	**	<i>TiqueNotaCreditoB</i>	113
*	<i>NotaDeDebitoC</i>	12	**	<i>TiqueNotaCreditoC</i>	114
*	<i>NotaDeDebitoM</i>	52	**	<i>TiqueNotaCreditoM</i>	119
***	<i>PresupuestoX</i>	903	**	<i>TiqueNotaDebitoA</i>	115
***	<i>ReciboA</i>	4	**	<i>TiqueNotaDebitoB</i>	116
***	<i>ReciboB</i>	9	**	<i>TiqueNotaDebitoC</i>	117
***	<i>ReciboC</i>	15	**	<i>TiqueNotaDebitoM</i>	120

- * Disponibles en impresoras fiscal 2G que soporten impresión sobre formularios continuos u hojas sueltas.
- ** Disponibles en impresoras fiscales 2G que soporten impresión sobre rollos de papel continuo.
- *** Disponibles en todos los modelos de impresoras fiscales 2G.

Ejemplo:

```
Dim resp As HasarArgentina.RespuestaConsultarCapacidadesImpresoraFiscal
```

```
resp = _
hasar.ConsultarCapacidadesImpresoraFiscal(HasarArgentina.Capacidades.AnchoTextoVenta, _
HasarArgentina.TiposDeEstacion.EstacionPorDefecto, _
HasarArgentina.TiposComprobante.NoDocumento)
```

```
Debug.Print("Ancho Texto Venta          =[" & resp.Valor & "]")
```

La respuesta fiscal al comando fiscal generado por 'ConsultarCapacidadesImpresoraFiscal()' es la estructura *RespuestaConsultarCapacidadesImpresoraFiscal*, definida internamente como:

```
typedef  
struct tagRespuestaConsultarCapacidadesImpresoraFiscal {  
    [helpstring("Valor")  
]  
    BSTR Valor;  
} RespuestaConsultarCapacidadesImpresoraFiscal;
```

Valor | Informa el valor correspondiente a la capacidad consultada.

ConsultarCapacidadZetas()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta de información referida a la capacidad de almacenamiento, en la memoria fiscal de la impresora fiscal 2G, de *Informes Diarios de Cierre* (cierres 'Z'), y otros datos de interés.

Definido internamente como:

```
ConsultarCapacidadZetas([out, retval] RespuestaConsultarCapacidadZetas* );
```

Ejemplo:

```
Dim resp As HasarArgentina.RespuestaConsultarCapacidadZetas  
  
resp = hasar.ConsultarCapacidadZetas()  
Debug.Print("Último Cierre 'Z'      ' =[" & resp.UltimaZeta & "]")  
Debug.Print("Cierres 'Z' Remanentes =[" & resp.CantidadDeZetasRemanentes & "]")  
Debug.Print("Cierres 'Z' Remanentes =[" & resp.AlmacenamientoLibreMemoriaAuditoria & "]")  
Debug.Print("Cierres 'Z' Remanentes =[" & resp.AlmacenamientoSecundario & "]")  
Debug.Print("Cierres 'Z' Remanentes =[" & resp.AlmacenamientoTotalMemoriaAuditoria s & "]")  
Debug.Print("Cierres 'Z' Remanentes =[" & resp.BajasAnteriores & "]")  
Debug.Print("Cierres 'Z' Remanentes =[" & resp.BytesAlmacenamientoPrincipal & "]")  
Debug.Print("Cierres 'Z' Remanentes =[" & resp.BytesAlmacenamientoPrincipallibre & "]")  
Debug.Print("Cierres 'Z' Remanentes =[" & resp.BytesAlmacenamientoSecundario & "]")  
Debug.Print("Cierres 'Z' Remanentes =[" & resp.BytesAlmacenamientoSecundariolibre & "]")  
Debug.Print("Cierres 'Z' Remanentes =[" & resp.BytesMemoriaDinamicaLibre & "]")  
Debug.Print("Cierres 'Z' Remanentes =[" & resp.BytesMemoriaDinamicaUsada & "]")  
Debug.Print("Cierres 'Z' Remanentes =[" & resp.DigestosDisponibles & "]")  
Debug.Print("Cierres 'Z' Remanentes =[" & resp.DigestosGrabados & "]")  
Debug.Print("Cierres 'Z' Remanentes =[" & resp.FechaJornadaActual & "]")  
Debug.Print("Cierres 'Z' Remanentes =[" & resp.FechaUltimaBajada & "]")  
Debug.Print("Cierres 'Z' Remanentes =[" & resp.FechaUltimaZetaBajada & "]")  
Debug.Print("Cierres 'Z' Remanentes =[" & resp.PrimerazetaEnMemoriaAuditoria & "]")  
Debug.Print("Cierres 'Z' Remanentes =[" & resp.UltimaZeta & "]")
```

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

```
Debug.Print("Cierres 'Z' Remanentes =[ " & resp.UltimaZetaBajada & "]")
Debug.Print("Cierres 'Z' Remanentes =[ " & resp.UltimaZetaBorrable & "]")
Debug.Print("Cierres 'Z' Remanentes =[ " & resp.UltimaZetaEnMemoriaAuditoria & "]")
Debug.Print("Cierres 'Z' Remanentes =[ " & resp.UmbraIMemoriaAuditoriaCasiLlena & "]")
Debug.Print("Cierres 'Z' Remanentes =[ " & resp.UmbraIMemoriaAuditoriaLlena & "]")
Debug.Print("Cierres 'Z' Remanentes =[ " & resp.UmbraIMemoriaFiscalCasiLlena & "]")
Debug.Print("Cierres 'Z' Remanentes =[ " & resp.UmbraIMemoriaFiscalLlena & "]")
```

La suma entre la cantidad total de registros disponibles y la cantidad de registros utilizados da como resultado la capacidad máxima de almacenamiento de la memoria fiscal, si es que la impresora fiscal 2G se encuentra operando en modo fiscal (es decir, ha sido inicializada y declarada ante la AFIP).

La respuesta fiscal al comando fiscal generado por '*ConsultarCapacidadZetas()*' es la estructura *RespuestaConsultarCapacidadZetas* , definida internamente como:

```
typedef
struct tagRespuestaConsultarCapacidadZetas {
    [helpstring("CantidadDeZetasRemanentes")]
}
    long CantidadDeZetasRemanentes;
    [helpstring("UltimaZeta")]
}
    long UltimaZeta;
    [helpstring("UltimaZetaBajada")]
}
    long UltimaZetaBajada;
    [helpstring("UltimaZetaBorrable")]
}
    long UltimaZetaBorrable;
    [helpstring("FechaUltimaBajada")]
}
    DATE FechaUltimaBajada;
    [helpstring("FechaUltimaZetaBajada")]
}
    DATE FechaUltimaZetaBajada;
    [helpstring("DigestosGrabados")]
}
    long DigestosGrabados;
    [helpstring("DigestosDisponibles")]
}
    long DigestosDisponibles;
    [helpstring("FechaJornadaActual")]
}
    DATE FechaJornadaActual;
    [helpstring("UmbralMemoriaFiscalLlena")]
}
    long UmbralMemoriaFiscalLlena;
    [helpstring("UmbralMemoriaFiscalCasiLlena")]
}
    long UmbralMemoriaFiscalCasiLlena;
    [helpstring("PrimeraZetaEnMemoriaAuditoria")]
}
    long PrimeraZetaEnMemoriaAuditoria;
    [helpstring("UltimaZetaEnMemoriaAuditoria")]
}
```

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

```

long UltimaZetaEnMemoriaAuditoria;
[helpstring("AlmacenamientoTotalMemoriaAuditoria")
]
double AlmacenamientoTotalMemoriaAuditoria;
[helpstring("AlmacenamientoLibreMemoriaAuditoria")
]
double AlmacenamientoLibreMemoriaAuditoria;
[helpstring("UmbralMemoriaAuditoriaLlena")
]
double UmbralMemoriaAuditoriaLlena;
[helpstring("UmbralMemoriaAuditoriaCasiLlena")
]
double UmbralMemoriaAuditoriaCasiLlena;
[helpstring("AlmacenamientoSecundario")
]
EstadosAlmacenamientoSecundario AlmacenamientoSecundario;
[helpstring("BajasAnteriores")
]
long BajasAnteriores;
[helpstring("BytesMemoriaDinamicaUsada")
]
double BytesMemoriaDinamicaUsada;
[helpstring("BytesMemoriaDinamicaLibre")
]
double BytesMemoriaDinamicaLibre;
[helpstring("BytesAlmacenamientoPrincipal")
]
double BytesAlmacenamientoPrincipal;
[helpstring("BytesAlmacenamientoPrincipalLibre")
]
double BytesAlmacenamientoPrincipalLibre;
[helpstring("BytesAlmacenamientoSecundario")
]
double BytesAlmacenamientoSecundario;
[helpstring("BytesAlmacenamientoSecundarioLibre")
]
double BytesAlmacenamientoSecundarioLibre;

```

CantidadDeZetasRemanentes

Informa la cantidad remanente de registros de *Informes Diarios de Cierre* (cierres 'Z') disponibles para almacenamiento en la memoria fiscal. El controlador fiscal 2G puede almacenar información de hasta 3650 *Informes Diarios de Cierre* (para el titular inicial).

UltimaZeta

Informa la cantidad de registros de *Informes Diarios de Cierre* ya almacenados en la memoria fiscal. Este valor coincide con el número del último *Informe Diario de Cierre* realizado.

UltimaZetaBajada

Informa el número del último *Informe Diario de Cierre* que fue descargado e incluido en un reporte electrónico para su presentación a la AFIP.

Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '1' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00 y SMH/PT-250F v01.00.

UltimaZetaBorrable

Informa el número del último *Informe Diario de Cierre* que será borrado en caso de ser necesario crear espacio de almacenamiento en la CTD (cinta testigo digital, o memoria de auditoría).

Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '1' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00 y SMH/PT-250F v01.00.

FechaUltimaBajada

Informa la fecha en la que se realizó la bajada de AFIP por última vez. Puede venir vacío cuando no se hizo ninguna bajada.

Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.

FechaUltimaZetaBajada

Informa la fecha del último *Informe Diario de Cierre* que fue descargado e incluido en un reporte electrónico para su presentación a la AFIP.

Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.

DigestosGrabados

Informa la cantidad de digestos guardados en la memoria fiscal.

Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.

DigestosDisponibles

Informa la cantidad de digestos que aún pueden almacenarse en la memoria fiscal.

Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.

FechaJornadaActual

Informa la fecha de la jornada fiscal en curso. Es decir, la que se encuentra abierta.

Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.

UmbralMemoriaFiscalLlena

Informa el valor a partir del cual no es posible seguir cerrando jornadas fiscales.

Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.

UmbralMemoriaFiscalCasiLlena

Informa el valor a partir del cual se empieza a advertir que la memoria fiscal está en agotamiento.

Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.

PrimeraZetaEnMemoriaAuditoria

Informa el número del primer *Informe Diario de Cierre* (cierre 'Z') almacenado en la CTD (cinta testigo digital, o memoria de auditoría).

Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.

UltimaZetaEnMemoriaAuditoria

Informa el número del último *Informe Diario de Cierre* (cierre 'Z') almacenado en la CTD (cinta testigo digital, o memoria de auditoría).

Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.

AlmacenamientoTotalMemoriaAuditoria

Informa la cantidad de bytes totales, utilizados, en la CTD (cinta testigo digital, o memoria de auditoría).

Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.

AlmacenamientoLibreMemoriaAuditoria

Informa la cantidad de bytes totales, libres, en la CTD (cinta testigo digital, o memoria de auditoría).

Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.

UmbralMemoriaAuditoriaLlena

Informa el valor a partir del cual no es posible seguir almacenando *Informes Diarios de Cierre* (cierres 'Z') en la CTD (cinta testigo digital, o memoria de auditoría).

Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.

UmbralMemoriaAuditoriaCasiLlena

Informa el valor a partir del cual se empieza a advertir que la CTD (cinta testigo digital, o memoria de auditoría) está en agotamiento.

Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.

AlmacenamientoSecundario

Informa si hay disponible almacenamiento extendido para la CTD (cinta testigo digital, o memoria de auditoría) está en agotamiento.

Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.

Es alguna de las constantes pertenecientes al enum *EstadosAlmacenamientoSecundario*.

<i>AlmacenamientoSecundarioAusente</i>	65
<i>AlmacenamientoSecundarioBloqueado</i>	66
<i>AlmacenamientoSecundarioSoloLectura</i>	82
<i>AlmacenamientoSecundarioOK</i>	79

BajasAnteriores

Informa la cantidad de veces que el controlador fiscal 2G fue dado de baja anteriormente.

Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.

BytesMemoriaDinamicaUsada

Informa la cantidad de bytes utilizados en la memoria RAM del controlador fiscal 2G.

Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los

	<p>modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.</p>
<i>BytesMemoriaDinamicaLibre</i>	<p>Informa la cantidad de bytes disponibles en la memoria RAM del controlador fiscal 2G.</p> <p>Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.</p>
<i>BytesAlmacenamientoPrincipal</i>	<p>Informa la cantidad de bytes totales en el almacenamiento persistente primario.</p> <p>Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.</p>
<i>BytesAlmacenamientoPrincipalLibre</i>	<p>Informa la cantidad de bytes libres en el almacenamiento persistente primario.</p> <p>Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.</p>
<i>BytesAlmacenamientoSecundario</i>	<p>Informa la cantidad de bytes totales en el almacenamiento persistente secundario.</p> <p>Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.</p>
<i>BytesAlmacenamientoSecundarioLibre</i>	<p>Informa la cantidad de bytes disponibles en el almacenamiento persistente secundario.</p> <p>Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.</p>

ConsultarConfiguracionImpresoraFiscal()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta de parámetros de comportamiento del equi-

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

po.

Definido internamente como:

```
ConsultarConfiguracionImpresoraFiscal(
    [in] Configuracion Variable,
    [out, retval] RespuestaConsultarConfiguracionImpresoraFiscal* );
```

Variable Permite especificar el parámetro de comportamiento de la impresora fiscal 2G que se desea consultar. Debe ser alguna de las constantes pertenecientes al enum *Configuracion*.

<i>ImpresionMarco</i>	49
<i>AltoHoja</i>	50
<i>AnchoHoja</i>	51
<i>EstacionReportesXZ</i>	52
<i>ModoImpresion</i>	53
<i>ImpresionUsarColorAlternativo</i>	54
<i>ImpresionColor_Aalternativo</i>	55
<i>TimeoutEnvioRespuestaEnEspera</i>	56
<i>ImpresionCodigo_QR</i>	57
<i>SonidoAviso</i>	65
<i>LimiteBC</i>	66
<i>ImpresionCambio</i>	67
<i>ChequeoDesborde</i>	68
<i>BorradoAutomaticoAuditoria</i>	69
<i>TipoHabilitacion</i>	72
<i>Interlineado</i>	73
<i>ImpresionLeyendas</i>	76
<i>CortePapel</i>	80
<i>ReimpresionCancelados</i>	82
<i>PagoSaldo</i>	83

Ejemplo:

```
Dim respcfg As HasarArgentina.RespuestaConsultarConfiguracionImpresoraFiscal
respcfg = _
    Hasar.ConsultarConfiguracionImpresoraFiscal(HasarArgentina.Configuracion.Interlineado)
Debug.Print("Interlineado = " & respcfg.Valor)
```

La respuesta fiscal al comando fiscal generado por '*ConsultarConfiguracionImpresoraFiscal()*' es la estructura *RespuestaConsultarConfiguracionImpresoraFiscal*, definida internamente como:

```
typedef
struct tagRespuestaConsultarConfiguracionImpresoraFiscal {
    [helpstring("Valor")]
}
    BSTR Valor;
} RespuestaConsultarConfiguracionImpresoraFiscal;
```

Valor | Informa el valor que tiene actualmente el parámetro consultado.

ConsultarConfiguracionRed()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta de datos de su identificación como nodo de una red.

Definido internamente como:

```
ConsultarConfiguracionRed([out, retval] RespuestaConsultarConfiguracionRed* );
```

Ejemplo:

```
Dim resp As HasarArgentina.RespuestaConsultarConfiguracionRed

resp = hasar.ConsultarConfiguracionRed()
Debug.Print("Dirección IP =[" & resp.DireccionIP & "]")
Debug.Print("Máscara      =[" & resp.Mascara & "]")
Debug.Print("Gateway      =[" & resp.Gateway & "]")
```

La respuesta fiscal al comando fiscal generado por '*ConsultarConfiguracionRed()*' es la estructura *RespuestaConsultarConfiguracionRed*, definida internamente como:

```
typedef
struct tagRespuestaConsultarConfiguracionRed {
    [helpstring("DireccionIP")]
}
    BSTR DireccionIP;
    [helpstring("Mascara")]
}
    BSTR Mascara;
    [helpstring("Gateway")]
}
    BSTR Gateway;
    [helpstring("DHCP")]
}
    SiNo DHCP;
} RespuestaConsultarConfiguracionRed;
```

<i>DireccionIP</i>	Informa la dirección IP de la impresora fiscal 2G en la red. Formato: nnn.nnn.nnn.nnn (0 <= nnn <= 255).
<i>Mascara</i>	Informa la máscara de red utilizada. Formato: nnn.nnn.nnn.nnn (0 <= nnn <= 255).
<i>Gateway</i>	Informa la dirección IP de la PC utilizada como puerta de enlace con otras subredes. Formato: nnn.nnn.nnn.nnn (0 <= nnn <= 255).
<i>DHCP</i>	Informa si el controlador fiscal 2G obtuvo la configuración de la red de un servidor de DNS.

ConsultarConfiguracionServidorCorreo()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta de la configuración del servidor de correo electrónico de salida (SMTP) que está utilizando.

Ejemplo:

```
Dim resp As HasarArgentina.RespuestaConsultarConfiguracionServidorCorreo

resp = hasar.ConsultarConfiguracionServidorCorreo()
Debug.Print("Remitente    =[" & resp.DireccionRemitente & "]")
Debug.Print("Dirección IP =[" & resp.DireccionIP & "]")
Debug.Print("Puerto        =[" & resp.Puerto & "]")
```

La respuesta fiscal al comando fiscal generado por 'ConsultarConfiguracionServidorCorreo()' es la estructura *RespuestaConsultarConfiguracionServidorCorreo* , definida internamente como:

```
typedef
struct tagRespuestaConsultarConfiguracionServidorCorreo {
    [helpstring("DireccionIP")]
    BSTR DireccionIP;
    [helpstring("Puerto")]
    long Puerto;
    [helpstring("DireccionRemitente")]
    BSTR DireccionRemitente;
    [helpstring("Usuario")]
    BSTR Usuario;
    [helpstring("Clave")]
    BSTR Clave;
} RespuestaConsultarConfiguracionServidorCorreo;
```

<i>DireccionIP</i>	Informa la dirección IP del servidor SMTP. Formato: nnn.nnn.nnn.nnn (0 <= nnn <= 255). Puede entregar un string vacío (valor por defecto) si nunca se programó una dirección IP del servidor de SMTP.
<i>Puerto</i>	Informa el valor del identificador del puerto utilizado por el servidor SMTP. Por defecto, 25 (si nunca se programó un puerto SMTP).
<i>DireccionRemitente</i>	Informa la dirección de correo electrónico a la que puede responder el receptor de los despachados por la impresora fiscal 2G. Puede entregar un string vacío (valor por defecto) si nunca se programó una dirección de correo electrónico.
<i>Usuario</i>	Informa el usuario para la autenticación del servidor SMTP. Puede entregar un string vacío si no se utiliza autenticación. Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.
<i>Clave</i>	Informa la clave (password) para la autenticación del servidor SMTP. Puede entregar un string vacío si no se utiliza autenticación. Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.

ConsultarDatosInicializacion()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta de datos fiscales del emisor de los comprobantes.

Ejemplo:

```
Dim resp As HasarArgentina.RespuestaConsultarDatosInicializacion

resp = hasar.ConsultarDatosInicializacion
Debug.Print("Razón Social      =[" & resp.RazonSocial & "]")
Debug.Print("Número CUIT       =[" & resp.Cuit & "]")
Debug.Print("Categoría IVA        =[" & resp.ResponsabilidadIVA & "]")
Debug.Print("Ingresos Brutos       =[" & resp.IngBrutos & "]")
Debug.Print("Fecha Inicio Activ. =[" & resp.FechaInicioActividades & "]")
Debug.Print("Número POS           =[" & resp.NumeroPos & "]")
Debug.Print("Código Registro      =[" & resp.Registro & "]")
```

La respuesta fiscal al comando fiscal generado por 'ConsultarDatosInicializacion()' es la estructura *RespuestaConsultarDatosInicializacion*, definida internamente como:

```
typedef
struct tagRespuestaConsultarDatosInicializacion {
    [helpstring("Cuit")]
    BSTR Cuit;
    [helpstring("RazonSocial")]
    BSTR RazonSocial;
    [helpstring("Registro")]
    BSTR Registro;
    [helpstring("NumeroPos")]
    long NumeroPos;
    [helpstring("FechaInicioActividades")]
    DATE FechaInicioActividades;
    [helpstring("IngBrutos")]
    BSTR IngBrutos;
    [helpstring("ResponsabilidadIVA")]
    TiposDeResponsabilidadesImpresor ResponsabilidadIVA;
    [helpstring("FechaInicioActividadesCompleta")]
    DATE FechaInicioActividadesCompleta;
} RespuestaConsultarDatosInicializacion;
```

<i>Cuit</i>	Informa la Clave Unica de Identificación Tributaria (CUIT) del emisor de los comprobantes.						
<i>RazonSocial</i>	Informa el nombre y apellido, o razón social del emisor de los comprobantes.						
<i>Registro</i>	<p>Informa el código de registro que identifica unívocamente a la impresora fiscal 2G. Este código es cargado en fábrica, a la salida de producción.</p> <p>La estructura informativa del código de registro es la siguiente: AABCCnnnnnnnnnn</p> <p>Donde:</p> <table border="1"> <tr> <td><i>AA</i></td> <td>Código de identificación del fabricante de la impresora fiscal 2G (2 caracteres). En el caso de HASAR es siempre HS.</td> </tr> <tr> <td><i>BB</i></td> <td>Código de identificación de la marca (2 caracteres). En el caso de HASAR es siempre HS.</td> </tr> <tr> <td><i>CC</i></td> <td> Código de identificación del modelo (2 caracteres). En el caso de HASAR puede ser: + AA De tratarse el modelo SMH/PT-1000F + AB De tratarse el modelo SMH/PT-250F </td> </tr> </table>	<i>AA</i>	Código de identificación del fabricante de la impresora fiscal 2G (2 caracteres). En el caso de HASAR es siempre HS.	<i>BB</i>	Código de identificación de la marca (2 caracteres). En el caso de HASAR es siempre HS.	<i>CC</i>	Código de identificación del modelo (2 caracteres). En el caso de HASAR puede ser: + AA De tratarse el modelo SMH/PT-1000F + AB De tratarse el modelo SMH/PT-250F
<i>AA</i>	Código de identificación del fabricante de la impresora fiscal 2G (2 caracteres). En el caso de HASAR es siempre HS.						
<i>BB</i>	Código de identificación de la marca (2 caracteres). En el caso de HASAR es siempre HS.						
<i>CC</i>	Código de identificación del modelo (2 caracteres). En el caso de HASAR puede ser: + AA De tratarse el modelo SMH/PT-1000F + AB De tratarse el modelo SMH/PT-250F						

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

	+ AD De tratarse el modelo P-HAS-5100F
	<i>nnnnnnnnnn</i> Número de serie de la impresora fiscal 2G (10 dígitos decimales).
<i>NumeroPOS</i>	Informa el número de punto de venta asignado a la impresora fiscal 2G durante el proceso de inicialización fiscal.
<i>FechaInicioActividades</i>	Informa la fecha de inicio de actividades del emisor de los comprobantes, en el formato: AAMMDD. Donde: AA = Año (00 a 99), MM = Mes (01 a 12), DD = Día (01 a 31) Notar que en el caso de que el año presente ambigüedades (por ejemplo, ¿ El año 20 es 1920 ó 2020 ?) debe utilizarse el campo <i>FechaInicioActividadesCompleta</i> .
<i>IngBrutos</i>	Informa el código de inscripción en Ingresos Brutos del emisor de los comprobantes.
<i>ResponsabilidadIVA</i>	Informa la categoría frente a IVA del emisor de los comprobantes. Los valores posibles son: I = Inscripto , E = Exento , A = No Responsable , M = Montributo , S = Monotributista Social
<i>FechaInicioActividadesCompleta</i>	Informa la fecha de inicio de actividades del emisor de los comprobantes, en el formato: AAAAMMDD. Donde: AAAA = Año (cuatro dígitos), MM = Mes (01 a 12), DD = Día (01 a 31)

ConsultarDocumentoAsociado()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta referido al almacenamiento, en la memoria de trabajo, de datos correspondientes a un comprobante a vincular con el próximo a emitir.

Ejemplo:

```
Dim resp As HasarArgentina.RespuestaConsultarDocumentoAsociado

Debug.Print("Nº POS      =[" & resp.NumeroPos & "]")
Debug.Print("Cód. Comprob. =[" & resp.CodigoComprobante & "]")
Debug.Print("Comprobante Nº =[" & resp.NumeroComprobante & "]")
```

La respuesta fiscal al comando fiscal generado por 'ConsultarDocumentoAsociado()' es la estructura *RespuestaConsultarDocumentoAsociado*, definida internamente como:

```
typedef
struct tagRespuestaConsultarDocumentoAsociado {
    [helpstring("CodigoComprobante")
]
    TiposComprobante CodigoComprobante;
    [helpstring("NumeroPos")
]
    long NumeroPos;
    [helpstring("NumeroComprobante")
]
    long NumeroComprobante;
} RespuestaConsultarDocumentoAsociado;
```

CodigoComprobante

Informa el valor identificatorio asociado a un tipo de comprobante tabulado en la normativa fiscal vigente, y soportado por la impresora fiscal 2G.

El valor '0' (*NoDocumento*) no identifica a ningún tipo de comprobante. La impresora fiscal 2G no emite comprobantes de este tipo. Es un valor interno y propio de la impresora fiscal 2G, que se utiliza en algunos comandos cuando la información del resto de los campos del comando, o de la respuesta, no está asociada a un tipo de comprobante en particular.

Es alguna de las constantes pertenecientes al enum *TiposComprobante*.

***	<i>ComprobanteDonacion</i>	907	***	<i>ReciboM</i>	54
*	<i>FacturaA</i>	1	***	<i>ReciboX</i>	902
*	<i>FacturaB</i>	6	***	<i>RemitoR</i>	91
*	<i>FacturaC</i>	11	***	<i>RemitoX</i>	901
*	<i>FacturaM</i>	51	**	<i>Tique</i>	83
***	<i>Generico</i>	910	**	<i>TiqueFacturaA</i>	81
*	<i>NotaDeCreditoA</i>	3	**	<i>TiqueFacturaB</i>	82
*	<i>NotaDeCreditoB</i>	8	**	<i>TiqueFacturaC</i>	111
*	<i>NotaDeCreditoC</i>	13	**	<i>TiqueFacturaM</i>	118

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

	* <i>NotaDeCreditoM</i>	53	** <i>TiqueNotaCredito</i>	110
	* <i>NotaDeDebitoA</i>	2	** <i>TiqueNotaCreditoA</i>	112
	* <i>NotaDeDebitoB</i>	7	** <i>TiqueNotaCreditoB</i>	113
	* <i>NotaDeDebitoC</i>	12	** <i>TiqueNotaCreditoC</i>	114
	* <i>NotaDeDebitoM</i>	52	** <i>TiqueNotaCreditoM</i>	119
	*** <i>PresupuestoX</i>	903	** <i>TiqueNotaDebitoA</i>	115
	*** <i>ReciboA</i>	4	** <i>TiqueNotaDebitoB</i>	116
	*** <i>ReciboB</i>	9	** <i>TiqueNotaDebitoC</i>	117
	*** <i>ReciboC</i>	15	** <i>TiqueNotaDebitoM</i>	120
	* <i>Disponibles en impresoras fiscal 2G que soporten impresión sobre formularios continuos u hojas sueltas.</i>			
	** <i>Disponibles en impresoras fiscales 2G que soporten impresión sobre rollos de papel continuo.</i>			
	*** <i>Disponibles en todos los modelos de impresoras fiscales 2G.</i>			
<i>NumeroPOS</i>	Informa el número del punto de venta en el cual fue emitido el comprobante asociado.			
<i>NumeroComprobante</i>	Informa el número del comprobante asociado.			

ConsultarEstado()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta de estado general del equipo, e información asociada al último tipo de comprobante emitido/cancelado que se especifique.

Definido internamente como:

```
ConsultarEstado(
    [in, optional, defaultvalue(0)] TiposComprobante CodigoComprobante,
    [out, retval] RespuestaConsultarEstado* );
```

<i>CodigoComprobante</i>	Permite especificar el tipo de comprobante cuya información asociada se desea consultar. Debe ser alguna de las constantes pertenecientes al enum <i>TiposComprobante</i> .			
	*** <i>ComprobanteDonacion</i>	907	*** <i>ReciboM</i>	54
	* <i>FacturaA</i>	1	*** <i>ReciboX</i>	902
	* <i>FacturaB</i>	6	*** <i>RemitoR</i>	91
	* <i>FacturaC</i>	11	*** <i>RemitoX</i>	901
	* <i>FacturaM</i>	51	** <i>Tique</i>	83
	*** <i>Generico</i>	910	** <i>TiqueFacturaA</i>	81

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

* <i>NotaDeCreditoA</i>	3	** <i>TiqueFacturaB</i>	82
* <i>NotaDeCreditoB</i>	8	** <i>TiqueFacturaC</i>	111
* <i>NotaDeCreditoC</i>	13	** <i>TiqueFacturaM</i>	118
* <i>NotaDeCreditoM</i>	53	** <i>TiqueNotaCredito</i>	110
* <i>NotaDeDebitoA</i>	2	** <i>TiqueNotaCreditoA</i>	112
* <i>NotaDeDebitoB</i>	7	** <i>TiqueNotaCreditoB</i>	113
* <i>NotaDeDebitoC</i>	12	** <i>TiqueNotaCreditoC</i>	114
* <i>NotaDeDebitoM</i>	52	** <i>TiqueNotaCreditoM</i>	119
*** <i>PresupuestoX</i>	903	** <i>TiqueNotaDebitoA</i>	115
*** <i>ReciboA</i>	4	** <i>TiqueNotaDebitoB</i>	116
*** <i>ReciboB</i>	9	** <i>TiqueNotaDebitoC</i>	117
*** <i>ReciboC</i>	15	** <i>TiqueNotaDebitoM</i>	120

* Disponibles en impresoras fiscales 2G que soporten impresión sobre formularios continuos u hojas sueltas.

** Disponibles en impresoras fiscales 2G que soporten impresión sobre rollos de papel continuo.

*** Disponibles en todos los modelos de impresoras fiscales 2G.

Si el código de tipo de comprobante especificado en la consulta hace referencia a un tipo de comprobante ya emitido, los campos de la respuesta identificados como número del último comprobante, cantidad comprobantes emitidos y cantidad de comprobantes cancelados reflejarán datos asociados a dicho comprobante. Si en cambio, el código de tipo de comprobante hace referencia a un tipo de comprobante no emitido aún, dichos campos de la respuesta se informarán en cero.

Si se omitiese el tipo de comprobante (o bien, se ingresa como '0' - *No Documento*) y la impresora fiscal 2G no tuviese abierto un comprobante al momento de la consulta, entregará en la respuesta el código del tipo de comprobante correspondiente al último comprobante emitido/cancelado, siendo la numeración y cantidades información asociada al comprobante reportado. De haber un comprobante abierto en la impresora fiscal 2G al momento de la consulta, se entregará en la respuesta el código del tipo de comprobante que se encuentra abierto, siendo la numeración y cantidades información asociada al comprobante reportado.

Ejemplo:

```
Dim resp As HasarArgentina.RespuestaConsultarEstado
```

```
resp = hasar.ConsultarEstado(HasarArgentina.TiposComprobante.TiqueFacturaA)
```

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

```
Debug.Print("Modo Entrenamiento =[" & resp.EstadoAuxiliar.ModoEntrenamiento & "]")
Debug.Print("Estado Interno =[" & resp.EstadoInterno & "]")
Debug.Print("Comprobante en Curso =[" & resp.ComprobanteEnCurso & "]")
Debug.Print("Código últ. Comprob. emitido =[" & resp.CodigoComprobante & "]")
Debug.Print("Número Últ. Comprob. emitido =[" & resp.NumeroUltimoComprobante & "]")
Debug.Print("Último Comprobante Cancelado =[" & _
    resp.EstadoAuxiliar.UltimoComprobanteFueCancelado & "]")
Debug.Print("Comprobantes Emitidos =[" & resp.CantidadEmitidos & "]")
Debug.Print("Comprobantes Cancelados =[" & resp.CantidadCancelados & "]")
Debug.Print("Código Barras Memorizado =[" & resp.EstadoAuxiliar.CodigoBarrasAlmacenado _
    & "]")
Debug.Print("Datos de Cliente Memorizado =[" & resp.EstadoAuxiliar.DatosClienteAlmacenados _
    & "]")
Debug.Print("Memoria Auditoría Casi Llena =[" & _
    resp.EstadoAuxiliar.MemoriaAuditoriaCasiLlena & "]")
Debug.Print("Memoria Auditoría Llena =[" & resp.EstadoAuxiliar.MemoriaAuditoriallena & "]")
Debug.Print("Estado = " & hasar.ObtenerCampoRespuesta("Estado"))
```

La respuesta fiscal al comando fiscal generado por '*ConsultarEstado()*' es la estructura *RespuestaConsultarEstado*, definida internamente como:

```
typedef
struct tagRespuestaConsultarEstado {
    [helpstring("EstadoAuxiliar")
    ]
    EstadoAuxiliar EstadoAuxiliar;
    [helpstring("EstadoInterno")
    ]
    EstadosFiscales EstadoInterno;
    [helpstring("ComprobanteEnCurso")
    ]
    TiposComprobante ComprobanteEnCurso;
    [helpstring("CodigoComprobante")
    ]
    TiposComprobante CodigoComprobante;
    [helpstring("NumeroUltimoComprobante")
    ]
    long NumeroUltimoComprobante;
    [helpstring("CantidadEmitidos")
    ]
    long CantidadEmitidos;
    [helpstring("CantidadCancelados")
    ]
    long CantidadCancelados;
} RespuestaConsultarEstado;
```

EstadoAuxiliar

Informa el estado de la CTD (cinta testigo digital, o memoria de auditoría), que datos asociados a un próximo comprobante a emitir se encuentran almacenados en la memoria de trabajo, y que ocurrió con el último comprobante emitido en forma inmediata anterior.

Se trata de una estructura booleana, definida internamente como:

```
typedef
struct tagEstadoAuxiliar {
    [helpstring("MemoriaAuditoriaLlena")]
}
VARIANT_BOOL MemoriaAuditoriaLlena;
[helpstring("MemoriaAuditoriaCasiLlena")]
}
VARIANT_BOOL MemoriaAuditoriaCasiLlena;
[helpstring("DatosClienteAlmacenados")]
}
VARIANT_BOOL DatosClienteAlmacenados;
[helpstring("CodigoBarrasAlmacenado")]
}
VARIANT_BOOL CodigoBarrasAlmacenado;
[helpstring("ModoEntrenamiento")]
}
VARIANT_BOOL ModoEntrenamiento;
[helpstring("UltimoComprobanteFueCancelado")]
}
VARIANT_BOOL UltimoComprobanteFueCancelado;
[helpstring("CajeroActivo")]
}
VARIANT_BOOL CajeroActivo;
} EstadoAuxiliar;
```

MemoriaAuditoriaLlena

Informa si la CTD (cinta testigo digital, o memoria de auditoría) se ha llenado.

MemoriaAuditoriaCasiLlena

Informa si la CTD (cinta testigo digital, o memoria de auditoría) está a punto de llenarse.

DatosClienteAlmacenados

Informa si hay datos de cliente almacenados en la memoria de trabajo de la impresora fiscal 2G.

CodigoBarrasAlmacenado

Informa si hay un código de barras almacenado en la memoria de trabajo de la impresora fiscal 2G.

ModoEntrenamiento

Informa si la impresora fiscal se encuentra operando en modo entrenamiento, o ya ha sido inicializada.

UltimoComprobanteFueCancelado

Informa si el último comprobante emitido fue cancelado, o fue cerrado exitosamente.

CajeroActivo

Informa, de estar ejecutándose el facturador web integrado, si hay un cajero que haya iniciado su sesión de trabajo.

EstadoInterno

Permite sincronizarse con la impresora fiscal 2G, si dicho sincronismo se ha perdido. Debe entenderse que esta información no es toda la "foto" del estado actual de la impresora fiscal 2G. El estado completo se puede determinar mediante el envío otros comandos de consulta (disponibles en la impresora fiscal 2G).

Informa alguna una las constantes pertenecientes al enum *EstadosFiscales*.

<i>Desconocido</i>	0
<i>NoInicializado</i>	1
<i>InicioJornadaFiscal</i>	2
<i>EnJornadaFiscal</i>	3
<i>Vendiendoltems</i>	4
<i>ImprimiendoTextoFiscal</i>	5
<i>Pagando</i>	6
<i>IngresandoOtrosTributos</i>	7
<i>RealizandoOperacionAjuste</i>	8
<i>RealizandoOperacionGlobalIVA</i>	9
<i>RealizandoOperacionAnticipo</i>	10
<i>ImprimiendoLineasRecibo</i>	11
<i>ImprimiendoTextoNoFiscal</i>	12
<i>CintaAuditoriaCasiLlena</i>	13
<i>CintaAuditoriaLlena</i>	14
<i>ControladorFiscalEsperandoBaja</i>	15
<i>ControladorFiscalDadoDeBaja</i>	16
<i>ControladorFiscalBloqueado</i>	17

ComprobanteEnCurso

Informa el tipo de comprobante que se encuentra abierto al momento de la consulta. De no haber un comprobante abierto se reporta el valor '0' (*NoDocumento*).

El valor '0' (*NoDocumento*) no identifica a ningún tipo de comprobante. La impresora fiscal 2G no emite comprobantes de este tipo. Es un valor interno y propio de la impresora fiscal 2G, que se utiliza en algunos comandos cuando la información del resto de los campos del comando, o de la respuesta, no está asociada a un tipo de comprobante en particular.

Es alguna de las constantes pertenecientes al enum *TiposComprobante*.

***	<i>ComprobanteDonacion</i>	907	***	<i>ReciboM</i>	54
*	<i>FacturaA</i>	1	***	<i>ReciboX</i>	902
*	<i>FacturaB</i>	6	***	<i>RemitoR</i>	91

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

* FacturaC	11	*** RemitoX	901
* FacturaM	51	** Tique	83
*** Generico	910	** TiqueFacturaA	81
* NotaDeCreditoA	3	** TiqueFacturaB	82
* NotaDeCreditoB	8	** TiqueFacturaC	111
* NotaDeCreditoC	13	** TiqueFacturaM	118
* NotaDeCreditoM	53	** TiqueNotaCredito	110
* NotaDeDebitoA	2	** TiqueNotaCreditoA	112
* NotaDeDebitoB	7	** TiqueNotaCreditoB	113
* NotaDeDebitoC	12	** TiqueNotaCreditoC	114
* NotaDeDebitoM	52	** TiqueNotaCreditoM	119
*** PresupuestoX	903	** TiqueNotaDebitoA	115
*** ReciboA	4	** TiqueNotaDebitoB	116
*** ReciboB	9	** TiqueNotaDebitoC	117
*** ReciboC	15	** TiqueNotaDebitoM	120

- * Disponibles en impresoras fiscal 2G que soporten impresión sobre formularios continuos u hojas sueltas.
- ** Disponibles en impresoras fiscales 2G que soporten impresión sobre rollos de papel continuo.
- *** Disponibles en todos los modelos de impresoras fiscales 2G.

CodigoComprobante

Informa el tipo de comprobante consultado puntualmente. Reporta el valor '0' (*NoDocumento*) si no se especificó tipo de comprobante al consultar el estado de la impresora fiscal 2G.

El valor '0' (*NoDocumento*) no identifica a ningún tipo de comprobante. La impresora fiscal 2G no emite comprobantes de este tipo. Es un valor interno y propio de la impresora fiscal 2G, que se utiliza en algunos comandos cuando la información del resto de los campos del comando, o de la respuesta, no está asociada a un tipo de comprobante en particular.

Es alguna de las constantes pertenecientes al enum *TiposComprobante*.

*** ComprobanteDonacion	907	*** ReciboM	54
* FacturaA	1	*** ReciboX	902
* FacturaB	6	*** RemitoR	91
* FacturaC	11	*** RemitoX	901
* FacturaM	51	** Tique	83

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

***	<i>Generico</i>	910	**	<i>TiqueFacturaA</i>	81
*	<i>NotaDeCreditoA</i>	3	**	<i>TiqueFacturaB</i>	82
*	<i>NotaDeCreditoB</i>	8	**	<i>TiqueFacturaC</i>	111
*	<i>NotaDeCreditoC</i>	13	**	<i>TiqueFacturaM</i>	118
*	<i>NotaDeCreditoM</i>	53	**	<i>TiqueNotaCredito</i>	110
*	<i>NotaDeDebitoA</i>	2	**	<i>TiqueNotaCreditoA</i>	112
*	<i>NotaDeDebitoB</i>	7	**	<i>TiqueNotaCreditoB</i>	113
*	<i>NotaDeDebitoC</i>	12	**	<i>TiqueNotaCrediotC</i>	114
*	<i>NotaDeDebitoM</i>	52	**	<i>TiqueNotaCreditoM</i>	119
***	<i>PresupuestoX</i>	903	**	<i>TiqueNotaDebitoA</i>	115
***	<i>ReciboA</i>	4	**	<i>TiqueNotaDebitoB</i>	116
***	<i>ReciboB</i>	9	**	<i>TiqueNotaDebitoC</i>	117
***	<i>ReciboC</i>	15	**	<i>TiqueNotaDebitoM</i>	120

- * Disponibles en impresoras fiscal 2G que soporten impresión sobre formularios continuos u hojas sueltas.
- ** Disponibles en impresoras fiscales 2G que soporten impresión sobre rollos de papel continuo.
- *** Disponibles en todos los modelos de impresoras fiscales 2G.

<i>NumeroUltimoComprobante</i>	Informa el número asociado al tipo de comprobante reportado.
<i>CantidadEmitidos</i>	Informa la cantidad referida únicamente a los comprobantes emitidos durante la jornada fiscal en curso, y del tipo reportado.
<i>CantidadCancelados</i>	Informa la cantidad referida únicamente a los comprobantes cancelados durante la jornada fiscal en curso, y del tipo reportado.

ConsultarEstadoECR()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta de valores asociados a operaciones del facturador web, como el cajero que está registrado. De aplicación sólo en impresoras fiscales 2G con facturador web incorporado, y tiene sentido su uso sólo si el facturador web se está ejecutando.

Definido internamente como:

```
ConsultarEstadoECR([out, retval] RespuestaConsultarEstadoECR* );
```

Ejemplo:

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

```
Dim resp1 As HasarArgentina.RespuestaConsultarEstadoECR
```

```
resp1 = hasar.ConsultarEstadoECR()  
Debug.Print("Nombre Cajero: " & "[" & resp1.NombreCajero & "]")  
Debug.Print("Cajero Nro.: " & "[" & resp1.NumeroCajero & "]")
```

La respuesta fiscal al comando fiscal generado por 'ConsultarEstadoECR()' es la estructura *RespuestaConsultarEstadoECR*, definida internamente como:

```
typedef  
struct tagRespuestaConsultarEstadoECR {  
    [helpstring("NumeroCajero")  
    ]  
    long NumeroCajero;  
    [helpstring("NombreCajero")  
    ]  
    BSTR NombreCajero;  
} RespuestaConsultarEstadoECR;
```

<i>NumeroCajero</i>	Informa el numero del cajero que tiene su sesión de trabajo abierta. Informa cero si ningún cajero inició sesión de trabajo, o el facturador web no se está ejecutando.
<i>NombreCajero</i>	Informa el nombre del cajero que tiene su sesión de trabajo abierta. Informa un string vacío si ningún cajero inició sesión de trabajo, o el facturador web no se está ejecutando.

ConsultarFechaHora()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta de fecha y hora de su reloj de tiempo real.

Definido internamente como:

```
ConsultarFechaHora([out, retval] RespuestaConsultarFechaHora* );
```

Ejemplo:

```
Dim resp As HasarArgentina.RespuestaConsultarFechaHora
```

```
resp = hasar.ConsultarFechaHora()  
Debug.Print("Fecha Actual =[ " & resp.Fecha & "]")  
Debug.Print("Hora Actual =[ " & resp.Hora & "]")
```

La respuesta fiscal al comando fiscal generado por 'ConsultarFechaHora()' es la estructura *RespuestaConsultarFechaHora*, definida internamente como:

```
typedef  
struct tagRespuestaConsultarFechaHora {  
    [helpstring("Fecha")  
    ]  
    DATE Fecha;  
    [helpstring("Hora")  
    ]  
    DATE Hora;  
} RespuestaConsultarFechaHora;
```

Fecha Informa la fecha actual del reloj de tiempo real de la impresora fiscal 2G.

Hora Informa la hora actual del reloj de tiempo real de la impresora fiscal 2G.

ConsultarInformacionAritmetica()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta de información aritmética de una operación en particular, y que ya haya sido registrada en la CTD (cinta testigo digital, o memoria de auditoría). Siempre que se envíe este comando fiscal debe haber un comprobante fiscal abierto.

Definido internamente como:

```
ConsultarInformacionAritmetica(  
    [in] long IndiceAuditoria,  
    [in, optional, defaultvalue(0)] long IndiceDiscriminacionIVA,  
    [out, retval] RespuestaConsultarInformacionAritmetica* );
```

IndiceAuditoria Permite especificar el valor del índice que apunta al registro de nuestro interés en la CTD. Este valor se obtiene de la respuesta a los comandos fiscales que generan una modificación en los totalizadores del documento.

IndiceDiscriminacionIVA Permite especificar el valor del índice que apunta a un registro generado por un desglose impositivo.

Si el monto de la operación registrada corresponde a un prorrateo sobre todas las alícuotas de IVA del comprobante, permite obtener el detalle de cómo fue aplicado en cada alícuota de IVA, utilizando un índice consecutivo en este campo. Así, la primera alícuota de IVA está representada por el índice 16, la segunda por el 17, y así sucesivamente. El índice cero queda reservado para los totales.

Si la operación registrada no generó un prorrateo, se puede especificar cero, o directamente dejar el campo vacío.

Ejemplo:

```
Dim resp2 As HasarArgentina.RespuestaConsultarInformacionAritmetica
```

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

```
resp2 = hasar.ConsultarInformacionAritmetica(14)
Debug.Print("Cantidad = " & resp2.Cantidad)
Debug.Print("P. Unit. Base = " & resp2.PrecioUnitarioBase)
Debug.Print("P. Unit. Total = " & resp2.PrecioUnitarioTotal)
Debug.Print("Cond. IVA = " & resp2.CondicionIVA)
Debug.Print("Alíc. IVA = " & resp2.TasaIVA)
Debug.Print("Monto IVA = " & resp2.MontoIVA)
Debug.Print("Monto IVA Impr. = " & resp2.MontoIVAImpreso)
Debug.Print("Imp. Int. Fijo = " & resp2.ImpuestosInternosFijos)
Debug.Print("Monto II = " & resp2.MontoImpuestosInternos)
Debug.Print("Monto II Impr. = " & resp2.MontoImpuestosInternosImpreso)
Debug.Print("Total = " & resp2.MontoTotal)
Debug.Print("Total Impr.= " & resp2.MontoTotalImpreso)
```

La respuesta fiscal al comando fiscal generado por '*ConsultarInformacionAritmetica()*' es la estructura *RespuestaConsultarInformacionAritmetica* , definida internamente como:

```
typedef
struct tagRespuestaConsultarInformacionAritmetica
[helpstring("Cantidad")
]
double Cantidad;
[helpstring("PrecioUnitarioBase")
]
double PrecioUnitarioBase;
[helpstring("PrecioUnitarioTotal")
]
double PrecioUnitarioTotal;
[helpstring("CondicionIVA")
]
CondicionesIVA CondicionIVA;
[helpstring("TasaIVA")
]
double TasaIVA;
[helpstring("MontoIVA")
]
double MontoIVA;
[helpstring("MontoImpuestosInternos")
]
double MontoImpuestosInternos;
[helpstring("ImpuestosInternosFijos")
]
SiNo ImpuestosInternosFijos;
[helpstring("MontoBase")
]
double MontoBase;
[helpstring("MontoTotal")
]
double MontoTotal;
[helpstring("MontoIVAImpreso")
]
double MontoIVAImpreso;
[helpstring("MontoImpuestosInternosImpreso")
]
double MontoImpuestosInternosImpreso;
[helpstring("MontoBaseImpreso")
]
double MontoBaseImpreso;
[helpstring("MontoTotalImpreso")
]
double MontoTotalImpreso;
} RespuestaConsultarInformacionAritmetica;
```

<i>Cantidad</i>	Informa la cantidad de unidades correspondiente al ítem vendido, o bien el valor -1.000 si se trata de un descuento.
<i>PrecioUnitarioBase</i>	Informa el precio unitario (sin impuestos) correspondiente al ítem vendido, o el monto (sin impuestos) del descuento aplicado. De tratarse de un monto de descuento el valor se reporta en positivo.
<i>PrecioUnitarioTotal</i>	Informa el precio unitario (impuestos incluidos) correspondiente al ítem vendido, o el monto (impuestos incluidos) del descuento aplicado. De tratarse de un monto de descuento el valor se reporta en positivo.
<i>CondicionIVA</i>	Informa la condición de IVA que afecta al ítem vendido, o al monto de descuento aplicado.
<i>TasaIVA</i>	Informa la alícuota de IVA que afecta al ítem vendido, o al monto de descuento aplicado.
<i>MontoIVA</i>	Informa el monto de IVA resultante de aplicar la alícuota de IVA indicada en el campo ' <i>TasaIVA</i> ' sobre el campo ' <i>PrecioUnitarioBase</i> '. De tratarse de un descuento el monto de IVA se mostrará como un valor negativo.
<i>MontoImpuestosInternos</i>	Informa el monto (fijo o variable) de impuestos internos que afecta al ítem vendido, o al descuento aplicado. De tratarse de un descuento el monto de impuestos internos se mostrará como un valor negativo.
<i>ImpuestosInternosFijos</i>	Informa un valor de afirmación o negación, respecto de considerar al monto de impuestos internos como fijo (no afectado por descuentos). Valores posibles: Si/No.
<i>MontoBase</i>	Informa el precio unitario (sin impuestos) correspondiente al ítem vendido, o el monto (sin impuestos) del descuento aplicado, truncado y redondeado a dos decimales. De tratarse de un monto de descuento el valor se reporta en negativo.
<i>MontoTotal</i>	Informa el monto total (impuestos incluidos) correspondiente al ítem vendido, o el monto total (impuestos incluidos) del descuento aplicado, truncado y redondeado a dos decimales. De tratarse de un monto de descuento el valor se reporta en negativo.
<i>MontoIVAImpreso</i>	Informa el monto de IVA impreso, correspondiente al ítem vendido o al descuento aplicado. De tratarse de un descuento el valor de IVA se reporta en negativo.
<i>MontoImpuestosInternosImpreso</i>	Informa el monto de impuestos internos impreso, correspondiente al ítem

<i>MontoBaseImpreso</i>	vendido o al descuento aplicado. De tratarse de un descuento el valor de impuesto interno se reporta en negativo.
<i>MontoTotalImpreso</i>	Informa el precio unitario impreso (sin impuestos) correspondiente al ítem vendido, o el monto impreso (sin impuestos) del descuento aplicado, truncado y redondeado a dos decimales. De tratarse de un monto de descuento el valor se reporta en negativo.
<i>MontoTotalImpreso</i>	Informa el monto total impreso (impuestos incluidos) correspondiente al ítem vendido, o el monto total impreso (impuestos incluidos) del descuento aplicado, truncado y redondeado a dos decimales. De tratarse de un monto de descuento el valor se reporta en negativo.

ConsultarInformacionBloqueo()

No disponible en los modelos de impresoras fiscales 2G: SMH/PT-1000F v01.00 y SMH/PT-250F v01.00

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta de información asociada al bloqueo actual del equipo, si se estuviese en presencia de uno.

Definido internamente como:

```
ConsultarInformacionBloqueo([out, retval] RespuestaConsultarInformacionBloqueo* );
```

Ejemplo:

```
Dim resp2 As HasarArgentina.RespuestaConsultarInformacionBloqueo  
  
resp2 = hasar.ConsultarInformacionBloqueo  
Debug.Print("Cant. Bloqueos = " & resp2.BloqueosAnteriores)  
Debug.Print("Cód. Bloqueo = " & resp2.CodigoBloqueo)  
Debug.Print("Contexto = " & resp2.Contexto)  
Debug.Print("Descripción = " & resp2.Descripcion)  
Debug.Print("Máx. Cant. Bloqueos = " & resp2.MaximaCantidadBloqueos)
```

La respuesta fiscal al comando fiscal generado por 'ConsultarInformacionBloqueo()' es la estructura *RespuestaConsultarInformacionBloqueo*, definida internamente como:

```
typedef  
struct tagRespuestaConsultarInformacionBloqueo {  
    [helpstring("CodigoBloqueo")  
    ]  
    BSTR CodigoBloqueo;  
    [helpstring("Descripcion")  
    ]  
    BSTR Descripcion;  
    [helpstring("Contexto")  
    ]  
    BSTR Contexto;  
    [helpstring("BloqueosAnteriores")  
    ]  
    long BloqueosAnteriores;  
    [helpstring("MaximaCantidadBloqueos")  
    ]  
    long MaximaCantidadBloqueos;  
} RespuestaConsultarInformacionBloqueo;
```

<i>CodigoBloqueo</i>	Informa el identificador de error asociado al incidente que se ha producido.
<i>Descripcion</i>	Informa una breve descripción del incidente que se producido a raíz del último comando enviado a la impresora fiscal 2G.
<i>Contexto</i>	Informa una breve descripción del contexto en el cual el incidente se ha producido.
<i>BloqueosAnteriores</i>	Informa la cantidad de veces que el controlador fiscal 2G se ha bloqueado con anterioridad. Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.
<i>MaximaCantidadBloqueos</i>	Informa la máxima cantidad de bloqueos admitida. Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.

ConsultarParametroHardware()

No disponible en los modelos de impresoras fiscales 2G: SMH/PT-1000F v01.00 y SMH/PT-250F v01.00

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta de parámetros asociados al hardware del equipo, los cuales son mostrados como cifras de utilización.

Definidos internamente como:

```
ConsultarParametroHardware (
    [in] ParametrosHardware Parametro,
    [out, retval] RespuestaConsultarParametroHardware* );
```

<i>Parametro</i>	Permite especificar el parámetro de hardware cuyo valor se desea consultar. Debe ser alguna de las constantes pertenecientes al enum <i>ParametrosHardware</i> .												
	<table border="1"> <tr> <td><i>CargandoBateria</i></td> <td>67</td> </tr> <tr> <td><i>NivelBateria</i></td> <td>78</td> </tr> <tr> <td><i>UsoCuchilla</i></td> <td>99</td> </tr> <tr> <td><i>UsoCabezal</i></td> <td>104</td> </tr> <tr> <td><i>UsoPapel</i></td> <td>112</td> </tr> <tr> <td><i>TiempoOperacion</i></td> <td>116</td> </tr> </table>	<i>CargandoBateria</i>	67	<i>NivelBateria</i>	78	<i>UsoCuchilla</i>	99	<i>UsoCabezal</i>	104	<i>UsoPapel</i>	112	<i>TiempoOperacion</i>	116
<i>CargandoBateria</i>	67												
<i>NivelBateria</i>	78												
<i>UsoCuchilla</i>	99												
<i>UsoCabezal</i>	104												
<i>UsoPapel</i>	112												
<i>TiempoOperacion</i>	116												

Ejemplo:

```
Dim resp2 As HasarArgentina.RespuestaConsultarParametroHardware

resp2 = hasar.ConsultarParametroHardware(HasarArgentina.ParametrosHardware.NivelBateria)
Debug.Print("Nivel Bateria Actual = " & resp2.Actual)
Debug.Print("Nivel Bateria Acumulado = " & resp2.Acumulado)
Debug.Print("Referencia = " & resp2.Referencia)
Debug.Print("Unidades = " & resp2.Unidades)
```

La respuesta fiscal al comando fiscal generado por 'ConsultarParametroHardware()' es la estructura *Respuesta-ConsultarParametroHardware* , definida internamente como:

```
typedef
struct tagRespuestaConsultarParametroHardware {
    [helpstring("Unidades")]
    ]
    Unidades Unidades;
    [helpstring("Actual")]
    ]
    long Actual;
    [helpstring("Acumulado")]
    ]
    long Acumulado;
    [helpstring("Referencia")]
    ]
    long Referencia;
} RespuestaConsultarParametroHardware;
```

<i>Unidades</i>	<p>Informa la unidad de medida en que se reporta la información.</p> <p>Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.</p> <p>Informa una de las constantes pertenecientes al enum <i>Unidades</i>.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td><i>EstadoBinario</i></td><td>66</td></tr> <tr><td><i>Cortes</i></td><td>67</td></tr> <tr><td><i>Horas</i></td><td>72</td></tr> <tr><td><i>Milímetros</i></td><td>77</td></tr> <tr><td><i>NivelCarga</i></td><td>78</td></tr> <tr><td><i>Paginas</i></td><td>80</td></tr> <tr><td><i>Caracteres</i></td><td>99</td></tr> </table>	<i>EstadoBinario</i>	66	<i>Cortes</i>	67	<i>Horas</i>	72	<i>Milímetros</i>	77	<i>NivelCarga</i>	78	<i>Paginas</i>	80	<i>Caracteres</i>	99
<i>EstadoBinario</i>	66														
<i>Cortes</i>	67														
<i>Horas</i>	72														
<i>Milímetros</i>	77														
<i>NivelCarga</i>	78														
<i>Paginas</i>	80														
<i>Caracteres</i>	99														
<i>Actual</i>	Informa el valor actual de la medición.														

	Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.
<i>Acumulado</i>	Informa el valor acumulado total, valor que puede estar vacío.
	Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.
<i>Referencia</i>	Informa el valor de referencia del fabricante, valor que puede estar vacío.
	Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.

ConsultarSubtotal()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta de acumulados globales de comprobante en curso de emisión.

Definido internamente como:

```
ConsultarSubtotal(
    [in, optional, defaultvalue(78)] ImpresionSubtotal Impresion,
    [in, optional, defaultvalue(48)] ModosDeDisplay ModoDisplay,
    [out, retval] RespuestaConsultarSubtotal* );
```

<i>Impresion</i>	Permite especificar si se imprime la leyenda “SUBTOT. nnnnnn.nn”, que es total acumulado hasta el momento en el comprobante. Debe ser alguna de las constantes pertenecientes al enum <i>ImpresionSubtotal</i> .						
	<table border="1"> <tr> <td><i>NoImprimeSubtotal</i></td> <td>78</td> </tr> <tr> <td><i>ImprimeSubtotal</i></td> <td>80</td> </tr> </table>	<i>NoImprimeSubtotal</i>	78	<i>ImprimeSubtotal</i>	80		
<i>NoImprimeSubtotal</i>	78						
<i>ImprimeSubtotal</i>	80						
	De indicarse impresión del subtotal, ésta sólo se hace efectiva en Tiques.						
<i>ModoDisplay</i>	Permite especificar el uso de un visor de cliente conectado a la impresora fiscal 2G. Debe ser alguna de las constantes pertenecientes al enum <i>ModosDeDisplay</i> .						
	<table border="1"> <tr> <td><i>DisplayNo</i></td> <td>48</td> </tr> <tr> <td><i>DisplaySi</i></td> <td>49</td> </tr> <tr> <td><i>DisplayRep</i></td> <td>50</td> </tr> </table>	<i>DisplayNo</i>	48	<i>DisplaySi</i>	49	<i>DisplayRep</i>	50
<i>DisplayNo</i>	48						
<i>DisplaySi</i>	49						
<i>DisplayRep</i>	50						

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

No hay, hasta el momento, una impresora fiscal 2G homologada que soporte la conexión de un dispositivo como el mencionado.

Ejemplo:

```
Dim resp As HasarArgentina.RespuestaConsultarSubtotal

resp = hasar.ConsultarSubtotal(HasarArgentina.ImpresionSubtotal.NoImprimeSubtotal, _
    HasarArgentina.ModosDeDisplay.DisplayNo)
Debug.Print("Monto Total      =[" & resp.Subtotal & "]"")
Debug.Print("Monto Pagado     =[" & resp.MontoPagado & "]"")
Debug.Print("Monto Base       =[" & resp.MontoBase & "]"")
Debug.Print("Monto IVA        =[" & resp.MontoIVA & "]"")
Debug.Print("Monto Imp. Int.    =[" & resp.MontoImpInternos & "]"")
Debug.Print("Monto Otros Trib. =[" & resp.MontoOtrosTributos & "]"")
```

La respuesta fiscal al comando fiscal generado por '*ConsultarSubtotal()*' es la estructura *RespuestaConsultarSubtotal*, definida internamente como:

```
typedef
struct tagRespuestaConsultarSubtotal {
    [helpstring("CantidadItems")]
    double CantidadItems;
    [helpstring("Subtotal")]
    double Subtotal;
    [helpstring("MontoIVA")]
    double MontoIVA;
    [helpstring("MontoPagado")]
    double MontoPagado;
    [helpstring("MontoOtrosTributos")]
    double MontoOtrosTributos;
    [helpstring("MontoImpInternos")]
    double MontoImpInternos;
    [helpstring("MontoBase")]
    double MontoBase;
    [helpstring("AjusteRedondeo")]
    double AjusteRedondeo;
} RespuestaConsultarSubtotal;
```

CantidadItems

Informa la suma de la cantidad de unidades indicada en cada ítem registrado hasta el momento de la consulta.

<i>Subtotal</i>	Informa la suma del resto de los montos parciales reportados en la respuesta del comando fiscal generado por ' <i>ConsultarSubtotal()</i> '; excepto el subtotal pagado (no incluido en esta suma). Este monto coincide con el importe final del comprobante si sucediese que a continuación de esta consulta se enviara a la impresora fiscal 2G el comando fiscal de cierre del comprobante.
<i>MontoIVA</i>	Informa el monto de IVA acumulado hasta el momento de la consulta, y considerando a todas las alícuotas intervinientes en la conformación del comprobante.
<i>MontoPagado</i>	Informa el monto acumulado de pagos realizados hasta el momento de la consulta.
<i>MontoOtrosTributos</i>	Informa el monto acumulado de tributos registrados hasta el momento de la consulta. No incluye la suma de impuestos internos por ítem.
<i>MontoImpInternos</i>	Informa el monto acumulado de impuestos internos por ítem, hasta el momento de la consulta.
<i>MontoBase</i>	Informa el monto de bases imponibles acumulado hasta el momento de la consulta, considerando e incluyendo las correspondientes a todas las alícuotas de IVA intervinientes en la conformación del comprobante.
<i>AjusteRedondeo</i>	Informa las potenciales diferencias entre los montos impresos y los montos internos calculados con mayor precisión, que son reportados por este comando fiscal. Informa las diferencias entre los montos reales y los montos impresos, limitados y redondeados a dos decimales a fines de la impresión. En el caso de los comprobantes "A", esta diferencia surge entre la suma real de las bases imponibles y la suma de los montos impresos. En el caso de comprobantes "B/C", esta diferencia se da en el monto final. En ambos casos, lo impreso puede diferir de lo reportado por este comando fiscal, por lo cual se informa el ajuste que será eventualmente impreso al finalizar el comprobante; independientemente de lo que se vea impreso en un momento dado en el comprobante, los montos reportados son los que se utilizarán para cerrar el comprobante y acumular.

ConsultarUltimaConexion()

No disponible en los modelos de impresoras fiscales 2G: SMH/PT-1000F v01.00 y SMH/PT-250F v01.00.

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta de información sobre la última conexión que se realizó con la impresora fiscal 2G.

Definido internamente como:

```
ConsultarUltimaConexion([out, retval] RespuestaConsultarUltimaConexion* );
```

Ejemplo:

```
Dim ultconex As HasarArgentina.RespuestaConsultarUltimaConexion
```

```
ultconex = hasar.ConsultarUltimaConexion()  
Debug.Print("Interfaz = " & ultconex.IdentificadorConexion)  
Debug.Print("Fecha = " & ultconex.Fecha)  
Debug.Print("Hora = " & ultconex.Hora)
```

La respuesta fiscal al comando fiscal generado por 'ConsultarUltimaConexion()' es la estructura *RespuestaConsultarUltimaConexion*, definida internamente como:

```
typedef  
struct tagRespuestaConsultarUltimaConexion {  
    [helpstring("IdentificadorConexion")  
]  
    BSTR IdentificadorConexion;  
    [helpstring("Fecha")  
]  
    DATE Fecha;  
    [helpstring("Hora")  
]  
    DATE Hora;  
} RespuestaConsultarUltimaConexion;
```

<i>IdentificadorConexion</i>	Informa si la última conexión fue realizada vía puerto serie (RS-232 o USB), o ethernet. Por ejemplo, 'HTTP (10.0.7.238)'.
<i>Fecha</i>	Informa la fecha en que fue realizada la última conexión a la impresora fiscal 2G, en el formato: DD/MM/AAAA.
<i>Hora</i>	Informa la fecha en que fue realizada la última conexión a la impresora fiscal 2G, en el formato: HH:MM:SS.

ConsultarUltimoError()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta de información sobre el error producido, y referido al comando fiscal enviado inmediatamente antes. El comando fiscal generado por este método debería enviarse a la impresora fiscal 2G cada vez que, por ejemplo, un comando fiscal es rechazado.

Definido internamente como:

```
ConsultarUltimoError([out, retval] RespuestaConsultarUltimoError* );
```

Ejemplo:

```
Dim resp As HasarArgentina.RespuestaConsultarUltimoError
```

```
resp = hasar.ConsultarUltimoError()  
Debug.Print("Parámetro N°      =[" & resp.NumeroParametro & "]")  
Debug.Print("Nombre Parámetro =[" & resp.NombreParametro & "]")  
Debug.Print("Último Error      =[" & resp.UltimoError & "]")  
Debug.Print("Descripción       =[" & resp.Descripcion & "]")  
Debug.Print("Contexto         =[" & resp.Contexto & "]")
```

La respuesta fiscal al comando fiscal generado por 'ConsultarUltimarError()' es la estructura *RespuestaConsultarUltimarError* , definida internamente como:

```
typedef  
struct tagRespuestaConsultarUltimoError {  
    [helpstring("UltimoError")  
]  
    BSTR UltimoError;  
    [helpstring("NumeroParametro")  
]  
    long NumeroParametro;  
    [helpstring("Descripcion")  
]  
    BSTR Descripcion;  
    [helpstring("Contexto")  
]  
    BSTR Contexto;  
    [helpstring("NombreParametro")  
]  
    BSTR NombreParametro;  
} RespuestaConsultarUltimoError;
```

<i>UltimoError</i>	Informa el identificador de error asociado al incidente que se ha producido.
<i>NumeroParametro</i>	Informa cual de los campos de información del comando fiscal enviado se asocia a la generación del incidente. El valor cero debe interpretarse como un error asociado a la ejecución del comando fiscal en sí, y no a alguno de sus campos de información.
<i>Descripcion</i>	Informa una breve descripción del incidente que se producido a raíz del último comando fiscal enviado a la impresora fiscal 2G.
<i>Contexto</i>	Informa una breve descripción del contexto en el cual el incidente se ha producido. El mensaje de contexto puede incluir un identificador de estado del parser interno de la impresora fiscal 2G.
<i>NombreParametro</i>	Informa el nombre que identifica al campo asociado al incidente que se reporta.

ConsultarVersion()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta sobre información de identificación como equipo fiscal. Esta información es importante para el sistema de facturación ya que su comportamiento debe

ajustarse a:

- El conjunto de comandos fiscales soportados por la impresora fiscal 2G identificada.
- La sintaxis de los comandos/respuestas fiscales que forman parte del conjunto soportado.
- Como reacciona la impresora fiscal 2G ante el comando fiscal recibido.
- Las prestaciones disponibles en la impresora fiscal 2G.

Definido internamente como:

```
ConsultarVersion([out, retval] RespuestaConsultarVersion* );
```

Ejemplo:

```
Dim resp As HasarArgentina.RespuestaConsultarVersion

resp = hasar.ConsultarVersion()
Debug.Print("Producto      =[" & resp.NombreProducto & "]")
Debug.Print("Marca         =[" & resp.Marca & "]")
Debug.Print("Modelo y Versión  =[" & resp.Version & "]")
Debug.Print("Fecha de Firmware =[" & resp.FechaFirmware & "]")
Debug.Print("Motor Versión     =[" & resp.VersionMotor & "]")
Debug.Print("Protocolo Versión =[" & resp.VersionProtocolo & "]")
```

La respuesta fiscal al comando fiscal generado por 'ConsultarVersion()' es la estructura *RespuestaConsultarVersion*, definida internamente como:

```
typedef
struct tagRespuestaConsultarVersion {
    [helpstring("Version")]
}
    BSTR Version;
    [helpstring("Marca")]
}
    BSTR Marca;
    [helpstring("NombreProducto")]
}
    BSTR NombreProducto;
    [helpstring("VersionMotor")]
}
    BSTR VersionMotor;
    [helpstring("FechaFirmware")]
}
    BSTR FechaFirmware;
    [helpstring("VersionProtocolo")]
}
    long VersionProtocolo;
} RespuestaConsultarVersion;
```

Version

Informa la identidad de la impresora fiscal 2G. Esta identidad consiste en el modelo de impresora fiscal 2G y en la versión de firmware fiscal que posee.

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

<i>Marca</i>	Informa la marca con que fue homologada la impresora fiscal 2G.
<i>NombreProducto</i>	Informa la identificación interna del conjunto de elementos intervinientes, a nivel software, como pilares del firmware fiscal.
<i>VersionMotor</i>	Informa la versión interna del motor de desarrollo del firmware fiscal.
<i>Fecha Firmware</i>	Informa la fecha y hora de creación del firmware fiscal homologado.
<i>VersionProtocolo</i>	Informa el número de versión del protocolo de comunicaciones, para así poder identificar si ciertos campos o comandos fiscales están disponibles. Las herramientas de diálogo que utilice el sistema de facturación deberán soportar un protocolo mayor o igual al que reporta la impresora fiscal 2G, para resultar compatibles.

ConsultarVigenciaCertificados()

No disponible en los modelos de impresoras fiscales 2G: SMH/PT-1000F v01.00, SMH/PT-250F v01.00, y P-HAS-5100-FAR v01.00.

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta sobre la vigencia actual del certificado digital más próximo a vencer, o el que ya esté vencido.

Definido internamente como:

```
ConsultarVigenciaCertificados([out, retval] RespuestaConsultarVigenciaCertificados* );
```

Ejemplo:

```
Dim resp2 As HasarArgentina.RespuestaConsultarVigenciaCertificados  
  
resp2 = hasar.ConsultarVigenciaCertificados  
Debug.Print("Fecha desde = " & resp2.FechaDesde)  
Debug.Print("Fecha hasta = " & resp2.FechaHasta)  
Debug.Print("Hora desde = " & resp2.HoraDesde)  
Debug.Print("Hora hasta = " & resp2.HoraHasta)  
Debug.Print("Nombre = " & resp2.Nombre)
```

La respuesta fiscal al comando fiscal generado por 'ConsultarVigenciaCertificados()' es la estructura *RespuestaConsultarVigenciaCertificados*, definida internamente como:

```
typedef
struct tagRespuestaConsultarVigenciaCertificados {
    [helpstring("Nombre")]
    ]
    BSTR Nombre;
    [helpstring("FechaDesde")]
    ]
    DATE FechaDesde;
    [helpstring("HoraDesde")]
    ]
    DATE HoraDesde;
    [helpstring("FechaHasta")]
    ]
    DATE FechaHasta;
    [helpstring("HoraHasta")]
    ]
    DATE HoraHasta;
    [helpstring("CRTValido")]
    ]
    SiNo CRTValido;
    [helpstring("CadenaAFIPValida")]
    ]
    SiNo CadenaAFIPValida;
} RespuestaConsultarVigenciaCertificados;
```

<i>Nombre</i>	<p>Informa la identificación (nombre) del certificado digital a vencer, o vencido.</p> <p>Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.</p>
<i>FechaDesde</i>	<p>Informa la fecha inicial de validez del certificado digital.</p> <p>Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.</p>
<i>HoraDesde</i>	<p>Informa la hora inicial de vigencia del certificado digital.</p> <p>Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.</p>
<i>FechaHasta</i>	<p>Informa la fecha final de vigencia del certificado digital.</p> <p>Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.</p>
<i>HoraHasta</i>	<p>Informa la hora de finalización de vigencia del certificado digital..</p> <p>Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo</p>

<i>CRTValido</i>	<p>'2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.</p> <p>Informa la validez del CRT.</p> <p>Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.</p>				
<i>CadenaAFIPValida</i>	<p>Informa la validez de la cadena de certificación.</p> <p>Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. No se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.</p> <p>Es alguna de las constantes pertenecientes al enum <i>SiNo</i>.</p> <table><tr><td><i>No</i></td><td>78</td></tr><tr><td><i>Si</i></td><td>83</td></tr></table>	<i>No</i>	78	<i>Si</i>	83
<i>No</i>	78				
<i>Si</i>	83				

ConsultarZetaBorrable()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta que retorna hasta que número de *Informe Diario de Cierre* (cierre 'Z') podrá borrarse información de la CTD (cinta testigo digital, o memoria de auditoría), en el supuesto que fuese requerido espacio para almacenar nueva información.

Definido internamente como:

```
ConsultarZetaBorrable([out, retval] RespuestaConsultarZetaBorrable* );
```

Ejemplo:

```
Dim resp As HasarArgentina.RespuestaConsultarZetaBorrable

resp = hasar.ConsultarZetaBorrable()
Debug.Print("Borrar hasta 'Z' Nº =[" & resp.NumeroZeta & "]")
```

La respuesta fiscal al comando fiscal generado por '*ConsultarZetaBorrable()*' es la estructura *RespuestaConsultarZetaBorrable*, definida internamente como:

```
typedef
struct tagRespuestaConsultarZetaBorrable {
    [helpstring("NumeroZeta")]
}
    long NumeroZeta;
} RespuestaConsultarZetaBorrable;
```

<i>NumeroZeta</i>	Informa el número del <i>Informe Diario de Cierre</i> (cierre 'Z') marcado como límite para el borrado de jornadas fiscales.
-------------------	--

ConsultarZona()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta de textos almacenados en la memoria de trabajo del equipo. Se pueden consultar líneas de fantasía, encabezado, domicilio del emisor, y/o pie del comprobante.

Definido internamente como:

```
ConsultarZona(
    [in] long NumeroLinea,
    [in] TiposDeEstacion Estacion,
    [in] ZonasDeLineasDeUsuario IdentificadorZona,
    [out, retval] RespuestaConsultarZona* );
```

<i>NumeroLinea</i>	<p>Permite especificar el número de línea, dentro de la zona indicada en el argumento <i>IdentificadorZona</i>. Los valores posibles son:</p> <table border="1"> <tr><td><i>Zona1Cola</i></td><td>Líneas 1 a 4.</td></tr> <tr><td><i>Zona1Encabezado</i></td><td>Líneas 1 a 3.</td></tr> <tr><td><i>Zona2Cola</i></td><td>Líneas 1 a 6.</td></tr> <tr><td><i>Zona2Encabezado</i></td><td>Líneas 1 a 3.</td></tr> <tr><td><i>ZonaDomicilioEmisor</i></td><td>Líneas 1 a 4.</td></tr> <tr><td><i>ZonaFantasia</i></td><td>Líneas 1 y 2.</td></tr> </table>	<i>Zona1Cola</i>	Líneas 1 a 4.	<i>Zona1Encabezado</i>	Líneas 1 a 3.	<i>Zona2Cola</i>	Líneas 1 a 6.	<i>Zona2Encabezado</i>	Líneas 1 a 3.	<i>ZonaDomicilioEmisor</i>	Líneas 1 a 4.	<i>ZonaFantasia</i>	Líneas 1 y 2.
<i>Zona1Cola</i>	Líneas 1 a 4.												
<i>Zona1Encabezado</i>	Líneas 1 a 3.												
<i>Zona2Cola</i>	Líneas 1 a 6.												
<i>Zona2Encabezado</i>	Líneas 1 a 3.												
<i>ZonaDomicilioEmisor</i>	Líneas 1 a 4.												
<i>ZonaFantasia</i>	Líneas 1 y 2.												
<i>Estacion</i>	<p>Permite especificar la estación de impresión asociada a la línea y zona consultada. Debe ser alguna de las constantes pertenecientes al enum <i>TiposDeEstacion</i>.</p> <table border="1"> <tr><td><i>Estacion_Por_Defecto</i></td><td>68</td></tr> <tr><td><i>Estacion_Slip</i></td><td>83</td></tr> <tr><td><i>Estacion_Ticket</i></td><td>84</td></tr> </table>	<i>Estacion_Por_Defecto</i>	68	<i>Estacion_Slip</i>	83	<i>Estacion_Ticket</i>	84						
<i>Estacion_Por_Defecto</i>	68												
<i>Estacion_Slip</i>	83												
<i>Estacion_Ticket</i>	84												
<i>IdentificadorZona</i>	<p>Permite especificar la zona del comprobante a la que pertenece la línea que se consulta. Debe ser alguna de las constantes que pertenecen al enum <i>ZonasDeLineasDeUsuario</i>.</p> <table border="1"> <tr><td><i>Zona1Cola</i></td><td>84</td></tr> <tr><td><i>Zona1Encabezado</i></td><td>72</td></tr> <tr><td><i>Zona2Cola</i></td><td>116</td></tr> <tr><td><i>Zona2Encabezado</i></td><td>104</td></tr> <tr><td><i>ZonaDomicilioEmisor</i></td><td>79</td></tr> <tr><td><i>ZonaFantasia</i></td><td>70</td></tr> </table>	<i>Zona1Cola</i>	84	<i>Zona1Encabezado</i>	72	<i>Zona2Cola</i>	116	<i>Zona2Encabezado</i>	104	<i>ZonaDomicilioEmisor</i>	79	<i>ZonaFantasia</i>	70
<i>Zona1Cola</i>	84												
<i>Zona1Encabezado</i>	72												
<i>Zona2Cola</i>	116												
<i>Zona2Encabezado</i>	104												
<i>ZonaDomicilioEmisor</i>	79												
<i>ZonaFantasia</i>	70												

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

Ejemplo:

```
Dim resp As HasarArgentina.RespuestaConsultarZona

resp = hasar.ConsultarZona(3, HasarArgentina.TiposDeEstacion.EstacionPorDefecto, _
    HasarArgentina.ZonasDeLineasDeUsuario.Zona1Encabezado)
Debug.Print("Texto      =[" & resp.Descripcion & "]")
Debug.Print("Centrado   =[" & resp.Atributos.Centrado & "]")
Debug.Print("Doble Ancho =[" & resp.Atributos.DobleAncho & "]")
Debug.Print("Negrita    =[" & resp.Atributos.Negrita & "]")
Debug.Print("Borrado    ? =[" & resp.Atributos.BorradoTexto & "]")
```

La respuesta fiscal al comando fiscal generado por 'ConsultarZona()' es la estructura *RespuestaConsultarZona*, definida internamente como:

```
typedef
struct tagRespuestaConsultarZona {
    [helpstring("Atributos")]
}
    AtributosDeTexto Atributos;
    [helpstring("Descripcion")]
}
    BSTR Descripcion;
    [helpstring("MaximaCantidadLineas")]
}
    long MaximaCantidadLineas;
} RespuestaConsultarZona;
```

Atributos

Informa el estilo de impresión del texto.

Es una estructura booleana definida internamente como *AtributosDeTexto*.

```
typedef
struct tagAtributosDeTexto {
    [helpstring("BorradoTexto")]
}
    VARIANT_BOOL BorradoTexto;
    [helpstring("DobleAncho")]
}
    VARIANT_BOOL DobleAncho;
    [helpstring("Centrado")]
}
    VARIANT_BOOL Centrado;
    [helpstring("Negrita")]
}
    VARIANT_BOOL Negrita;
} AtributosDeTexto;
```

BorradoTexto

Informa si la línea de usuario consultada tiene texto almacenado en la memoria de trabajo de la impresora fiscal 2G.

<i>DobleAncho</i>	Informa si el texto almacenado se imprimiría en doble ancho.
<i>Centrado</i>	Informa si el texto almacenado se imprimiría centrado.
<i>Negrita</i>	Informa si el texto almacenado se imprimiría en negrita.
<i>Descripcion</i>	Informa el texto almacenado en la memoria de trabajo de la impresora fiscal 2G, asociado a la línea, zona y estación de impresión que se ha consultado.
<i>MaximaCantidaLineas</i>	Informa la cantidad máxima de líneas que se pueden imprimir en la zona solicitada. Esta información se encuentra disponible sólo en controladores fiscales 2G con protocolo '2' o superior. Este campo no se encuentra disponible en los modelos SMH/PT-1000F v01.00, SMH/PT-250F v01.00 y SMH/PT-5100F v01.00.

ContinuarConsultaAcumulados()

Permite enviar a la impresora fiscal 2G el comando fiscal de pedido del siguiente registro de información, cuando el primer registro fue obtenido con alguno de los comandos fiscales generados por: 'ConsultarAcumuladosCierreZ()', 'ConsultarAcumuladosComprobante()', y/o 'ConsultarAcumuladosMemoriaDeTrabajo()'.

El comando fiscal generado por 'ContinuarConsultaAcumulados()' se debe enviar a la impresora fiscal 2G tantas veces como sea necesario, hasta obtener la totalidad de registros disponibles.

Definido internamente como:

```
ContinuarConsultaAcumulados([out, retval] RespuestaContinuarConsultaAcumulados* );
```

Ejemplo:

```
Dim comprob2 As HasarArgentina.RespuestaContinuarConsultaAcumulados
comprob2 = hasar.ContinuarConsultaAcumulados

If (comprob2.Registro = HasarArgentina.TiposDeRegistroInforme.RegistroDetalladoDNFH) Then
    Debug.Print("Cód. Comprob. =[" & comprob2.RegDNFH.CodigoComprobante & "]")
    Debug.Print("Desde Nº     =[" & comprob2.RegDNFH.NumeroInicial & "]")
    Debug.Print("Hasta Nº       =[" & comprob2.RegDNFH.NumeroFinal & "]")
    Debug.Print("TOTAL          =[" & comprob2.RegDNFH.Total & "]")
End If
```

La respuesta fiscal al comando fiscal generado por 'ContinuarConsultaAcumulados()' es la estructura *RespuestaContinuarConsultaAcumulados*, definida internamente como:

```
typedef
struct tagRespuestaContinuarConsultaAcumulados {
    [helpstring("Registro")]
}
    TiposDeRegistroInforme Registro;
    [helpstring("RegDF")]
}
    ContinuarConsultaAcumuladosRegDF RegDF;
    [helpstring("RegDNFH")]
}
    ContinuarConsultaAcumuladosRegDNFH RegDNFH;
    [helpstring("RegDNFH_NoAcum")]
}
    ContinuarConsultaAcumuladosRegDNFH_NoAcum RegDNFH_NoAcum;
    [helpstring("RegGlobal")]
}
    ContinuarConsultaAcumuladosRegGlobal RegGlobal;
} RespuestaContinuarConsultaAcumulados;
```

Registro

Informa el tipo de registro que se está entregando como respuesta fiscal a la consulta de acumulados que se ha realizado.

<i>RegistroFinal</i>	48
<i>RegistroDetalladoDF</i>	49
<i>RegistroDetalladoDNFH</i>	50
<i>RegistroDetalladoDNFHNoAcum</i>	51
<i>RegistroGlobal</i>	52

Si *Registro = RegistroFinal*, entonces no hay más información de acumulados.

RegDF

Informa datos asociados a los documentos fiscales emitidos.

Cuando *Registro = RegistroDetalladoDF* la respuesta fiscal continúa con una estructura definida internamente como:

```
typedef
struct tagContinuarConsultaAcumuladosRegDF {
    [helpstring("CodigoComprobante")]
}
    TiposComprobante CodigoComprobante;
    [helpstring("NumeroInicial")]
}
    long NumeroInicial;
    [helpstring("NumeroFinal")]
}
    long NumeroFinal;
    [helpstring("CantidadCancelados")]
}
    long CantidadCancelados;
    [helpstring("Total")]
}
```

```
double Total;  
[helpstring("TotalExento")  
]  
double TotalExento;  
[helpstring("TotalNoGravado")  
]  
double TotalNoGravado;  
[helpstring("TotalGravado")  
]  
double TotalGravado;  
[helpstring("TotalIVA")  
]  
double TotalIVA;  
[helpstring("TotalTributos")  
]  
double TotalTributos;  
[helpstring("AlicuotaIVA_1")  
]  
double AlicuotaIVA_1;  
[helpstring("MontoIVA_1")  
]  
double MontoIVA_1;  
[helpstring("AlicuotaIVA_2")  
]  
double AlicuotaIVA_2;  
[helpstring("MontoIVA_2")  
]  
double MontoIVA_2;  
[helpstring("AlicuotaIVA_3")  
]  
double AlicuotaIVA_3;  
[helpstring("MontoIVA_3")  
]  
double MontoIVA_3;  
[helpstring("AlicuotaIVA_4")  
]  
double AlicuotaIVA_4;  
[helpstring("MontoIVA_4")  
]  
double MontoIVA_4;  
[helpstring("AlicuotaIVA_5")  
]  
double AlicuotaIVA_5;  
[helpstring("MontoIVA_5")  
]  
double MontoIVA_5;  
[helpstring("AlicuotaIVA_6")  
]  
double AlicuotaIVA_6;  
[helpstring("MontoIVA_6")  
]  
double MontoIVA_6;  
[helpstring("AlicuotaIVA_7")  
]  
]
```

```
double AlicuotaIVA_7;  
[helpstring("MontoIVA_7")  
]  
double MontoIVA_7;  
[helpstring("AlicuotaIVA_8")  
]  
double AlicuotaIVA_8;  
[helpstring("MontoIVA_8")  
]  
double MontoIVA_8;  
[helpstring("AlicuotaIVA_9")  
]  
double AlicuotaIVA_9;  
[helpstring("MontoIVA_9")  
]  
double MontoIVA_9;  
[helpstring("AlicuotaIVA_10")  
]  
double AlicuotaIVA_10;  
[helpstring("MontoIVA_10")  
]  
double MontoIVA_10;  
[helpstring("CodigoTributo1")  
]  
TiposTributos CodigoTributo1;  
[helpstring("ImporteTributo1")  
]  
double ImporteTributo1;  
[helpstring("CodigoTributo2")  
]  
TiposTributos CodigoTributo2;  
[helpstring("ImporteTributo2")  
]  
double ImporteTributo2;  
[helpstring("CodigoTributo3")  
]  
TiposTributos CodigoTributo3;  
[helpstring("ImporteTributo3")  
]  
double ImporteTributo3;  
[helpstring("CodigoTributo4")  
]  
TiposTributos CodigoTributo4;  
[helpstring("ImporteTributo4")  
]  
double ImporteTributo4;  
[helpstring("CodigoTributo5")  
]  
TiposTributos CodigoTributo5;  
[helpstring("ImporteTributo5")  
]  
double ImporteTributo5;  
[helpstring("CodigoTributo6")  
]  
TiposTributos CodigoTributo6;  
[helpstring("ImporteTributo6")  
]
```

```

double ImporteTributo6;
[helpstring("CodigoTributo7")
]
TiposTributos CodigoTributo7;
[helpstring("ImporteTributo7")
]
double ImporteTributo7;
[helpstring("CodigoTributo8")
]
TiposTributos CodigoTributo8;
[helpstring("ImporteTributo8")
]
double ImporteTributo8;
[helpstring("CodigoTributo9")
]
TiposTributos CodigoTributo9;
[helpstring("ImporteTributo9")
]
double ImporteTributo9;
[helpstring("CodigoTributo10")
]
TiposTributos CodigoTributo10;
[helpstring("ImporteTributo10")
]
double ImporteTributo10;
[helpstring("CodigoTributo11")
]
TiposTributos CodigoTributo11;
[helpstring("ImporteTributo11")
]
double ImporteTributo11;
} ContinuarConsultaAcumuladosRegDF;
    
```

CodigoComprobante

Informa un código de tipo de comprobante fiscal.

El valor '0' (*NoDocumento*) no identifica a ningún tipo de comprobante. La impresora fiscal 2G no emite comprobantes de este tipo. Es un valor interno y propio de la impresora fiscal 2G, que se utiliza en algunos comandos cuando la información del resto de los campos del comando, o de la respuesta, no está asociada a un tipo de comprobante en particular.

El valor informado es alguna de las constantes pertenecientes al enum *TiposComprobante*.

***	<i>ComprobanteDonacion</i>	907	***	<i>ReciboM</i>	54
*	<i>FacturaA</i>	1	***	<i>ReciboX</i>	902
*	<i>FacturaB</i>	6	***	<i>RemitoR</i>	91
*	<i>FacturaC</i>	11	***	<i>RemitoX</i>	901
*	<i>FacturaM</i>	51	**	<i>Tique</i>	83
***	<i>Generico</i>	910	**	<i>TiqueFacturaA</i>	81

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

	* NotaDeCreditoA	3	** TiqueFacturaB	82
	* NotaDeCreditoB	8	** TiqueFacturaC	111
	* NotaDeCreditoC	13	** TiqueFacturaM	118
	* NotaDeCreditoM	53	** TiqueNotaCredito	110
	* NotaDeDebitoA	2	** TiqueNotaCreditoA	112
	* NotaDeDebitoB	7	** TiqueNotaCreditoB	113
	* NotaDeDebitoC	12	** TiqueNotaCreditoC	114
	* NotaDeDebitoM	52	** TiqueNotaCreditoM	119
	*** PresupuestoX	903	** TiqueNotaDebitoA	115
	*** ReciboA	4	** TiqueNotaDebitoB	116
	*** ReciboB	9	** TiqueNotaDebitoC	117
	*** ReciboC	15	** TiqueNotaDebitoM	120
	* Disponibles en impresoras fiscal 2G que soporten impresión sobre formularios continuos u hojas sueltas.			
	** Disponibles en impresoras fiscales 2G que soporten impresión sobre rollos de papel continuo.			
	*** Disponibles en todos los modelos de impresoras fiscales 2G.			
<i>NumeroInicial</i>	Informa el número inicial del rango de comprobantes del tipo informado en el segundo campo del registro, que abarca la consulta realizada.			
<i>NumeroFinal</i>	Informa el número final del rango de comprobantes del tipo informado en el segundo campo del registro, que abarca la consulta realizada.			
<i>CantidadCancelados</i>	Informa la cantidad de comprobantes del tipo reportado en el segundo campo del registro, que fueron cancelados.			
<i>Total</i>	Informa el importe total; acumulado correspondiente a los documentos del tipo informado en el segundo campo del registro, y para el rango numérico detallado. El importe total reportado incluye IVA y otros tributos, de corresponder.			
<i>TotalExento</i>	Informa el importe total exento; acumulado correspondiente a los documentos de tipo informado en el segundo campo del registro, y para el rango numérico informado.			
<i>TotalNoGravado</i>	Informa el importe total no gravado; acumulado correspondiente a los documentos de tipo informado en el segundo campo del re-			

<i>TotalGravado</i>	gistro, y para el rango numérico informado. Informa el importe total gravado; acumulado correspondiente a los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado.
<i>TotalIVA</i>	Informa el importe total de IVA; acumulado correspondiente a los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado.
<i>TotalTributos</i>	Informa el importe total de otros tributos; acumulado correspondiente a los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado.
<i>AlicuotaIVA_1</i>	Informa una de las alícuotas de IVA empleada en alguno de los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado.
<i>MontoIVA_1</i>	Informa el importe total de IVA para la alícuota reportada en el campo anterior del registro; acumulado correspondiente a los documentos de tipo informado en el segundo campo del registro, y para el rango numérico informado.
<i>AlicuotaIVA_2</i>	Informa una de las alícuotas de IVA empleada en alguno de los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado.
<i>MontoIVA_2</i>	Informa el importe total de IVA para la alícuota reportada en el campo anterior del registro; acumulado correspondiente a los documentos de tipo informado en el segundo campo del registro, y para el rango numérico informado.
<i>AlicuotaIVA_3</i>	Informa una de las alícuotas de IVA empleada en alguno de los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado.
<i>MontoIVA_3</i>	Informa el importe total de IVA para la alícuota reportada en el campo anterior del registro; acumulado correspondiente a los documentos de tipo informado en el segundo campo del registro, y para el rango numérico informado.
<i>AlicuotaIVA_4</i>	Informa una de las alícuotas de IVA empleada en alguno de los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado.
<i>MontoIVA_4</i>	Informa el importe total de IVA para la alícuota reportada en el

	<p>campo anterior del registro; acumulado correspondiente a los documentos de tipo informado en el segundo campo del registro, y para el rango numérico informado.</p>
<i>AlicuotaIVA_5</i>	<p>Informa una de las alícuotas de IVA empleada en alguno de los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado.</p>
<i>MontaIVA_5</i>	<p>Informa el importe total de IVA para la alícuota reportada en el campo anterior del registro; acumulado correspondiente a los documentos de tipo informado en el segundo campo del registro, y para el rango numérico informado.</p>
<i>AlicuotaIVA_6</i>	<p>Informa una de las alícuotas de IVA empleada en alguno de los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado.</p>
<i>MontaIVA_6</i>	<p>Informa el importe total de IVA para la alícuota reportada en el campo anterior del registro; acumulado correspondiente a los documentos de tipo informado en el segundo campo del registro, y para el rango numérico informado.</p>
<i>AlicuotaIVA_7</i>	<p>Informa una de las alícuotas de IVA empleada en alguno de los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado.</p>
<i>MontaIVA_7</i>	<p>Informa el importe total de IVA para la alícuota reportada en el campo anterior del registro; acumulado correspondiente a los documentos de tipo informado en el segundo campo del registro, y para el rango numérico informado.</p>
<i>AlicuotaIVA_8</i>	<p>Informa una de las alícuotas de IVA empleada en alguno de los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado.</p>
<i>MontaIVA_8</i>	<p>Informa el importe total de IVA para la alícuota reportada en el campo anterior del registro; acumulado correspondiente a los documentos de tipo informado en el segundo campo del registro, y para el rango numérico informado.</p>
<i>AlicuotaIVA_9</i>	<p>Informa una de las alícuotas de IVA empleada en alguno de los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado.</p>

<i>MontoIVA_9</i>	Informa el importe total de IVA para la alícuota reportada en el campo anterior del registro; acumulado correspondiente a los documentos de tipo informado en el segundo campo del registro, y para el rango numérico informado.																								
<i>AlicuotaIVA_10</i>	Informa una de las alícuotas de IVA empleada en alguno de los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado.																								
<i>MontoIVA_10</i>	Informa el importe total de IVA para la alícuota reportada en el campo anterior del registro; acumulado correspondiente a los documentos de tipo informado en el segundo campo del registro, y para el rango numérico informado.																								
<i>CodigoTributo1</i>	<p>Informa un código de tributo empleado en alguno de los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado. Se reporta siempre el valor (no el identificador). El valor '0' (<i>SinImpuestos</i>) indica la ausencia de otros tributos.</p> <p>Es una de las constantes pertenecientes al enum <i>TiposTributos</i>.</p> <table border="1"> <tr><td><i>SinImpuestos</i></td><td>0</td></tr> <tr><td><i>ImpuestosNacionales</i></td><td>1</td></tr> <tr><td><i>ImpuestosProvinciales</i></td><td>2</td></tr> <tr><td><i>ImpuestosMunicipales</i></td><td>3</td></tr> <tr><td><i>ImpuestosInternos</i></td><td>4</td></tr> <tr><td><i>IIBB</i></td><td>5</td></tr> <tr><td><i>PercepcionIVA</i></td><td>6</td></tr> <tr><td><i>PercepcionIIBB</i></td><td>7</td></tr> <tr><td><i>PercepcionImpuestosMunicipales</i></td><td>8</td></tr> <tr><td><i>OtrasPercepciones</i></td><td>9</td></tr> <tr><td><i>ImpuestoInternaltem</i></td><td>10</td></tr> <tr><td><i>OtrosTributos</i></td><td>99</td></tr> </table>	<i>SinImpuestos</i>	0	<i>ImpuestosNacionales</i>	1	<i>ImpuestosProvinciales</i>	2	<i>ImpuestosMunicipales</i>	3	<i>ImpuestosInternos</i>	4	<i>IIBB</i>	5	<i>PercepcionIVA</i>	6	<i>PercepcionIIBB</i>	7	<i>PercepcionImpuestosMunicipales</i>	8	<i>OtrasPercepciones</i>	9	<i>ImpuestoInternaltem</i>	10	<i>OtrosTributos</i>	99
<i>SinImpuestos</i>	0																								
<i>ImpuestosNacionales</i>	1																								
<i>ImpuestosProvinciales</i>	2																								
<i>ImpuestosMunicipales</i>	3																								
<i>ImpuestosInternos</i>	4																								
<i>IIBB</i>	5																								
<i>PercepcionIVA</i>	6																								
<i>PercepcionIIBB</i>	7																								
<i>PercepcionImpuestosMunicipales</i>	8																								
<i>OtrasPercepciones</i>	9																								
<i>ImpuestoInternaltem</i>	10																								
<i>OtrosTributos</i>	99																								
<i>ImporteTributo1</i>	Informa el importe total del tributo reportado en el campo anterior; acumulado correspondiente a los documentos del tipo informado en el segundo campo del registro, y para el rango numérico informado.																								
<i>CodigoTributo2</i>	Informa un código de tributo empleado en alguno de los documentos de tipo reportado en el segundo campo del registro, y																								

para el rango numérico informado. Se reporta siempre el valor (no el identificador). El valor '0' (*SinImpuestos*) indica la ausencia de otros tributos.

Es una de las constantes pertenecientes al enum *TiposTributos*.

<i>SinImpuestos</i>	0
<i>ImpuestosNacionales</i>	1
<i>ImpuestosProvinciales</i>	2
<i>ImpuestosMunicipales</i>	3
<i>ImpuestosInternos</i>	4
<i>IIBB</i>	5
<i>PercepcionIVA</i>	6
<i>PercepcionIIBB</i>	7
<i>PercepcionImpuestosMunicipales</i>	8
<i>OtrasPercepciones</i>	9
<i>ImpuestoInternoltem</i>	10
<i>OtrosTributos</i>	99

ImporteTributo2

Informa el importe total del tributo reportado en el campo anterior; acumulado correspondiente a los documentos del tipo informado en el segundo campo del registro, y para el rango numérico informado.

CodigoTributo3

Informa un código de tributo empleado en alguno de los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado. Se reporta siempre el valor (no el identificador). El valor '0' (*SinImpuestos*) indica la ausencia de otros tributos.

Es una de las constantes pertenecientes al enum *TiposTributos*.

<i>SinImpuestos</i>	0
<i>ImpuestosNacionales</i>	1
<i>ImpuestosProvinciales</i>	2
<i>ImpuestosMunicipales</i>	3
<i>ImpuestosInternos</i>	4
<i>IIBB</i>	5
<i>PercepcionIVA</i>	6
<i>PercepcionIIBB</i>	7

	<table border="1"> <tr> <td><i>PercepcionImpuestosMunicipales</i></td> <td align="right">8</td> </tr> <tr> <td><i>OtrasPercepciones</i></td> <td align="right">9</td> </tr> <tr> <td><i>ImpuestoInternoltem</i></td> <td align="right">10</td> </tr> <tr> <td><i>OtrosTributos</i></td> <td align="right">99</td> </tr> </table>	<i>PercepcionImpuestosMunicipales</i>	8	<i>OtrasPercepciones</i>	9	<i>ImpuestoInternoltem</i>	10	<i>OtrosTributos</i>	99																
<i>PercepcionImpuestosMunicipales</i>	8																								
<i>OtrasPercepciones</i>	9																								
<i>ImpuestoInternoltem</i>	10																								
<i>OtrosTributos</i>	99																								
<i>ImporteTributo3</i>	<p>Informa el importe total del tributo reportado en el campo anterior; acumulado correspondiente a los documentos del tipo informado en el segundo campo del registro, y para el rango numérico informado.</p>																								
<i>CodigoTributo4</i>	<p>Informa un código de tributo empleado en alguno de los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado. Se reporta siempre el valor (no el identificador). El valor '0' (<i>SinImpuestos</i>) indica la ausencia de otros tributos.</p> <p>Es una de las constantes pertenecientes al enum <i>TiposTributos</i>.</p> <table border="1"> <tr> <td><i>SinImpuestos</i></td> <td align="right">0</td> </tr> <tr> <td><i>ImpuestosNacionales</i></td> <td align="right">1</td> </tr> <tr> <td><i>ImpuestosProvinciales</i></td> <td align="right">2</td> </tr> <tr> <td><i>ImpuestosMunicipales</i></td> <td align="right">3</td> </tr> <tr> <td><i>ImpuestosInternos</i></td> <td align="right">4</td> </tr> <tr> <td><i>IIBB</i></td> <td align="right">5</td> </tr> <tr> <td><i>PercepcionIVA</i></td> <td align="right">6</td> </tr> <tr> <td><i>PercepcionIIBB</i></td> <td align="right">7</td> </tr> <tr> <td><i>PercepcionImpuestosMunicipales</i></td> <td align="right">8</td> </tr> <tr> <td><i>OtrasPercepciones</i></td> <td align="right">9</td> </tr> <tr> <td><i>ImpuestoInternoltem</i></td> <td align="right">10</td> </tr> <tr> <td><i>OtrosTributos</i></td> <td align="right">99</td> </tr> </table>	<i>SinImpuestos</i>	0	<i>ImpuestosNacionales</i>	1	<i>ImpuestosProvinciales</i>	2	<i>ImpuestosMunicipales</i>	3	<i>ImpuestosInternos</i>	4	<i>IIBB</i>	5	<i>PercepcionIVA</i>	6	<i>PercepcionIIBB</i>	7	<i>PercepcionImpuestosMunicipales</i>	8	<i>OtrasPercepciones</i>	9	<i>ImpuestoInternoltem</i>	10	<i>OtrosTributos</i>	99
<i>SinImpuestos</i>	0																								
<i>ImpuestosNacionales</i>	1																								
<i>ImpuestosProvinciales</i>	2																								
<i>ImpuestosMunicipales</i>	3																								
<i>ImpuestosInternos</i>	4																								
<i>IIBB</i>	5																								
<i>PercepcionIVA</i>	6																								
<i>PercepcionIIBB</i>	7																								
<i>PercepcionImpuestosMunicipales</i>	8																								
<i>OtrasPercepciones</i>	9																								
<i>ImpuestoInternoltem</i>	10																								
<i>OtrosTributos</i>	99																								
<i>ImporteTributo4</i>	<p>Informa el importe total del tributo reportado en el campo anterior; acumulado correspondiente a los documentos del tipo informado en el segundo campo del registro, y para el rango numérico informado.</p>																								
<i>CodigoTributo5</i>	<p>Informa un código de tributo empleado en alguno de los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado. Se reporta siempre el valor (no el identificador). El valor '0' (<i>SinImpuestos</i>) indica la ausencia de otros tributos.</p>																								

	Es una de las constantes pertenecientes al enum <i>TiposTributos</i> .																								
	<table border="1"> <tr><td><i>SinImpuestos</i></td><td>0</td></tr> <tr><td><i>ImpuestosNacionales</i></td><td>1</td></tr> <tr><td><i>ImpuestosProvinciales</i></td><td>2</td></tr> <tr><td><i>ImpuestosMunicipales</i></td><td>3</td></tr> <tr><td><i>ImpuestosInternos</i></td><td>4</td></tr> <tr><td><i>IIBB</i></td><td>5</td></tr> <tr><td><i>PercepcionIVA</i></td><td>6</td></tr> <tr><td><i>PercepcionIIBB</i></td><td>7</td></tr> <tr><td><i>PercepcionImpuestosMunicipales</i></td><td>8</td></tr> <tr><td><i>OtrasPercepciones</i></td><td>9</td></tr> <tr><td><i>ImpuestoInternoltem</i></td><td>10</td></tr> <tr><td><i>OtrosTributos</i></td><td>99</td></tr> </table>	<i>SinImpuestos</i>	0	<i>ImpuestosNacionales</i>	1	<i>ImpuestosProvinciales</i>	2	<i>ImpuestosMunicipales</i>	3	<i>ImpuestosInternos</i>	4	<i>IIBB</i>	5	<i>PercepcionIVA</i>	6	<i>PercepcionIIBB</i>	7	<i>PercepcionImpuestosMunicipales</i>	8	<i>OtrasPercepciones</i>	9	<i>ImpuestoInternoltem</i>	10	<i>OtrosTributos</i>	99
<i>SinImpuestos</i>	0																								
<i>ImpuestosNacionales</i>	1																								
<i>ImpuestosProvinciales</i>	2																								
<i>ImpuestosMunicipales</i>	3																								
<i>ImpuestosInternos</i>	4																								
<i>IIBB</i>	5																								
<i>PercepcionIVA</i>	6																								
<i>PercepcionIIBB</i>	7																								
<i>PercepcionImpuestosMunicipales</i>	8																								
<i>OtrasPercepciones</i>	9																								
<i>ImpuestoInternoltem</i>	10																								
<i>OtrosTributos</i>	99																								
<i>ImporteTributo5</i>	Informa el importe total del tributo reportado en el campo anterior; acumulado correspondiente a los documentos del tipo informado en el segundo campo del registro, y para el rango numérico informado.																								
<i>CodigoTributo6</i>	Informa un código de tributo empleado en alguno de los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado. Se reporta siempre el valor (no el identificador). El valor '0' (<i>SinImpuestos</i>) indica la ausencia de otros tributos.																								
	Es una de las constantes pertenecientes al enum <i>TiposTributos</i> .																								
	<table border="1"> <tr><td><i>SinImpuestos</i></td><td>0</td></tr> <tr><td><i>ImpuestosNacionales</i></td><td>1</td></tr> <tr><td><i>ImpuestosProvinciales</i></td><td>2</td></tr> <tr><td><i>ImpuestosMunicipales</i></td><td>3</td></tr> <tr><td><i>ImpuestosInternos</i></td><td>4</td></tr> <tr><td><i>IIBB</i></td><td>5</td></tr> <tr><td><i>PercepcionIVA</i></td><td>6</td></tr> <tr><td><i>PercepcionIIBB</i></td><td>7</td></tr> <tr><td><i>PercepcionImpuestosMunicipales</i></td><td>8</td></tr> <tr><td><i>OtrasPercepciones</i></td><td>9</td></tr> <tr><td><i>ImpuestoInternoltem</i></td><td>10</td></tr> </table>	<i>SinImpuestos</i>	0	<i>ImpuestosNacionales</i>	1	<i>ImpuestosProvinciales</i>	2	<i>ImpuestosMunicipales</i>	3	<i>ImpuestosInternos</i>	4	<i>IIBB</i>	5	<i>PercepcionIVA</i>	6	<i>PercepcionIIBB</i>	7	<i>PercepcionImpuestosMunicipales</i>	8	<i>OtrasPercepciones</i>	9	<i>ImpuestoInternoltem</i>	10		
<i>SinImpuestos</i>	0																								
<i>ImpuestosNacionales</i>	1																								
<i>ImpuestosProvinciales</i>	2																								
<i>ImpuestosMunicipales</i>	3																								
<i>ImpuestosInternos</i>	4																								
<i>IIBB</i>	5																								
<i>PercepcionIVA</i>	6																								
<i>PercepcionIIBB</i>	7																								
<i>PercepcionImpuestosMunicipales</i>	8																								
<i>OtrasPercepciones</i>	9																								
<i>ImpuestoInternoltem</i>	10																								

	<i>OtrosTributos</i>	99																								
<i>ImporteTributo6</i>	Informa el importe total del tributo reportado en el campo anterior; acumulado correspondiente a los documentos del tipo informado en el segundo campo del registro, y para el rango numérico informado.																									
<i>CodigoTributo7</i>	<p>Informa un código de tributo empleado en alguno de los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado. Se reporta siempre el valor (no el identificador). El valor '0' (<i>SinImpuestos</i>) indica la ausencia de otros tributos.</p> <p>Es una de las constantes pertenecientes al enum <i>TiposTributos</i>.</p> <table border="1"> <tr><td><i>SinImpuestos</i></td><td align="right">0</td></tr> <tr><td><i>ImpuestosNacionales</i></td><td align="right">1</td></tr> <tr><td><i>ImpuestosProvinciales</i></td><td align="right">2</td></tr> <tr><td><i>ImpuestosMunicipales</i></td><td align="right">3</td></tr> <tr><td><i>ImpuestosInternos</i></td><td align="right">4</td></tr> <tr><td><i>IIBB</i></td><td align="right">5</td></tr> <tr><td><i>PercepcionIVA</i></td><td align="right">6</td></tr> <tr><td><i>PercepcionIIBB</i></td><td align="right">7</td></tr> <tr><td><i>PercepcionImpuestosMunicipales</i></td><td align="right">8</td></tr> <tr><td><i>OtrasPercepciones</i></td><td align="right">9</td></tr> <tr><td><i>ImpuestoInternaltem</i></td><td align="right">10</td></tr> <tr><td><i>OtrosTributos</i></td><td align="right">99</td></tr> </table>		<i>SinImpuestos</i>	0	<i>ImpuestosNacionales</i>	1	<i>ImpuestosProvinciales</i>	2	<i>ImpuestosMunicipales</i>	3	<i>ImpuestosInternos</i>	4	<i>IIBB</i>	5	<i>PercepcionIVA</i>	6	<i>PercepcionIIBB</i>	7	<i>PercepcionImpuestosMunicipales</i>	8	<i>OtrasPercepciones</i>	9	<i>ImpuestoInternaltem</i>	10	<i>OtrosTributos</i>	99
<i>SinImpuestos</i>	0																									
<i>ImpuestosNacionales</i>	1																									
<i>ImpuestosProvinciales</i>	2																									
<i>ImpuestosMunicipales</i>	3																									
<i>ImpuestosInternos</i>	4																									
<i>IIBB</i>	5																									
<i>PercepcionIVA</i>	6																									
<i>PercepcionIIBB</i>	7																									
<i>PercepcionImpuestosMunicipales</i>	8																									
<i>OtrasPercepciones</i>	9																									
<i>ImpuestoInternaltem</i>	10																									
<i>OtrosTributos</i>	99																									
<i>ImporteTributo7</i>	Informa el importe total del tributo reportado en el campo anterior; acumulado correspondiente a los documentos del tipo informado en el segundo campo del registro, y para el rango numérico informado.																									
<i>CodigoTributo8</i>	<p>Informa un código de tributo empleado en alguno de los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado. Se reporta siempre el valor (no el identificador). El valor '0' (<i>SinImpuestos</i>) indica la ausencia de otros tributos.</p> <p>Es una de las constantes pertenecientes al enum <i>TiposTributos</i>.</p> <table border="1"> <tr><td><i>SinImpuestos</i></td><td align="right">0</td></tr> </table>		<i>SinImpuestos</i>	0																						
<i>SinImpuestos</i>	0																									

	<table border="1"> <tr><td><i>ImpuestosNacionales</i></td><td>1</td></tr> <tr><td><i>ImpuestosProvinciales</i></td><td>2</td></tr> <tr><td><i>ImpuestosMunicipales</i></td><td>3</td></tr> <tr><td><i>ImpuestosInternos</i></td><td>4</td></tr> <tr><td><i>IIBB</i></td><td>5</td></tr> <tr><td><i>PercepcionIVA</i></td><td>6</td></tr> <tr><td><i>PercepcionIIBB</i></td><td>7</td></tr> <tr><td><i>PercepcionImpuestosMunicipales</i></td><td>8</td></tr> <tr><td><i>OtrasPercepciones</i></td><td>9</td></tr> <tr><td><i>ImpuestoInternoltem</i></td><td>10</td></tr> <tr><td><i>OtrosTributos</i></td><td>99</td></tr> </table>	<i>ImpuestosNacionales</i>	1	<i>ImpuestosProvinciales</i>	2	<i>ImpuestosMunicipales</i>	3	<i>ImpuestosInternos</i>	4	<i>IIBB</i>	5	<i>PercepcionIVA</i>	6	<i>PercepcionIIBB</i>	7	<i>PercepcionImpuestosMunicipales</i>	8	<i>OtrasPercepciones</i>	9	<i>ImpuestoInternoltem</i>	10	<i>OtrosTributos</i>	99		
<i>ImpuestosNacionales</i>	1																								
<i>ImpuestosProvinciales</i>	2																								
<i>ImpuestosMunicipales</i>	3																								
<i>ImpuestosInternos</i>	4																								
<i>IIBB</i>	5																								
<i>PercepcionIVA</i>	6																								
<i>PercepcionIIBB</i>	7																								
<i>PercepcionImpuestosMunicipales</i>	8																								
<i>OtrasPercepciones</i>	9																								
<i>ImpuestoInternoltem</i>	10																								
<i>OtrosTributos</i>	99																								
<i>ImporteTributo8</i>	Informa el importe total del tributo reportado en el campo anterior; acumulado correspondiente a los documentos del tipo informado en el segundo campo del registro, y para el rango numérico informado.																								
<i>CodigoTributo9</i>	<p>Informa un código de tributo empleado en alguno de los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado. Se reporta siempre el valor (no el identificador). El valor '0' (<i>SinImpuestos</i>) indica la ausencia de otros tributos.</p> <p>Es una de las constantes pertenecientes al enum <i>TiposTributos</i>.</p> <table border="1"> <tr><td><i>SinImpuestos</i></td><td>0</td></tr> <tr><td><i>ImpuestosNacionales</i></td><td>1</td></tr> <tr><td><i>ImpuestosProvinciales</i></td><td>2</td></tr> <tr><td><i>ImpuestosMunicipales</i></td><td>3</td></tr> <tr><td><i>ImpuestosInternos</i></td><td>4</td></tr> <tr><td><i>IIBB</i></td><td>5</td></tr> <tr><td><i>PercepcionIVA</i></td><td>6</td></tr> <tr><td><i>PercepcionIIBB</i></td><td>7</td></tr> <tr><td><i>PercepcionImpuestosMunicipales</i></td><td>8</td></tr> <tr><td><i>OtrasPercepciones</i></td><td>9</td></tr> <tr><td><i>ImpuestoInternoltem</i></td><td>10</td></tr> <tr><td><i>OtrosTributos</i></td><td>99</td></tr> </table>	<i>SinImpuestos</i>	0	<i>ImpuestosNacionales</i>	1	<i>ImpuestosProvinciales</i>	2	<i>ImpuestosMunicipales</i>	3	<i>ImpuestosInternos</i>	4	<i>IIBB</i>	5	<i>PercepcionIVA</i>	6	<i>PercepcionIIBB</i>	7	<i>PercepcionImpuestosMunicipales</i>	8	<i>OtrasPercepciones</i>	9	<i>ImpuestoInternoltem</i>	10	<i>OtrosTributos</i>	99
<i>SinImpuestos</i>	0																								
<i>ImpuestosNacionales</i>	1																								
<i>ImpuestosProvinciales</i>	2																								
<i>ImpuestosMunicipales</i>	3																								
<i>ImpuestosInternos</i>	4																								
<i>IIBB</i>	5																								
<i>PercepcionIVA</i>	6																								
<i>PercepcionIIBB</i>	7																								
<i>PercepcionImpuestosMunicipales</i>	8																								
<i>OtrasPercepciones</i>	9																								
<i>ImpuestoInternoltem</i>	10																								
<i>OtrosTributos</i>	99																								
<i>ImporteTributo9</i>	Informa el importe total del tributo reportado en el campo anterior; acumulado correspondiente a los documentos del tipo																								

CodigoTributo10

informado en el segundo campo del registro, y para el rango numérico informado.

Informa un código de tributo empleado en alguno de los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado. Se reporta siempre el valor (no el identificador). El valor '0' (*SinImpuestos*) indica la ausencia de otros tributos.

Es una de las constantes pertenecientes al enum *TiposTributos*.

<i>SinImpuestos</i>	0
<i>ImpuestosNacionales</i>	1
<i>ImpuestosProvinciales</i>	2
<i>ImpuestosMunicipales</i>	3
<i>ImpuestosInternos</i>	4
<i>IIBB</i>	5
<i>PercepcionIVA</i>	6
<i>PercepcionIIBB</i>	7
<i>PercepcionImpuestosMunicipales</i>	8
<i>OtrasPercepciones</i>	9
<i>ImpuestoInternoltem</i>	10
<i>OtrosTributos</i>	99

ImporteTributo10

Informa el importe total del tributo reportado en el campo anterior; acumulado correspondiente a los documentos del tipo informado en el segundo campo del registro, y para el rango numérico informado.

CodigoTributo11

Informa un código de tributo empleado en alguno de los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado. Se reporta siempre el valor (no el identificador). El valor '0' (*SinImpuestos*) indica la ausencia de otros tributos.

Es una de las constantes pertenecientes al enum *TiposTributos*.

<i>SinImpuestos</i>	0
<i>ImpuestosNacionales</i>	1
<i>ImpuestosProvinciales</i>	2
<i>ImpuestosMunicipales</i>	3

<i>ImpuestosInternos</i>	4
<i>IIBB</i>	5
<i>PercepcionIVA</i>	6
<i>PercepcionIIBB</i>	7
<i>PercepcionImpuestosMunicipales</i>	8
<i>OtrasPercepciones</i>	9
<i>ImpuestoInternoltem</i>	10
<i>OtrosTributos</i>	99

ImporteTributo11 Informa el importe total del tributo reportado en el campo anterior; acumulado correspondiente a los documentos del tipo informado en el segundo campo del registro, y para el rango numérico informado.

RegDNFH

Informa datos asociados a los documentos no fiscales homologados emitidos que acumulan totales.

Cuando *Registro = RegistroDetalladoDNFH* la respuesta fiscal continúa con una estructura definida internamente como:

```
typedef
struct tagContinuarConsultaAcumuladosRegDNFH {
    [helpstring("CodigoComprobante")]
}
    TiposComprobante CodigoComprobante;
    [helpstring("NumeroInicial")]
}
    long NumeroInicial;
    [helpstring("NumeroFinal")]
}
    long NumeroFinal;
    [helpstring("Total")]
}
    double Total;
} ContinuarConsultaAcumuladosRegDNFH;
```

CodigoComprobante Informa un código de tipo de comprobante correspondiente a un DNFH (Documento No Fiscal Homologado) que genera acumulados.

El valor '0' (*NoDocumento*) no identifica a ningún tipo de comprobante. La impresora fiscal 2G no emite comprobantes de este tipo. Es un valor interno y propio de la impresora fiscal 2G, que se utiliza en algunos comandos fiscales cuando la información del resto de los campos del comando fiscal, o de la respuesta fiscal, no está asociada a un tipo de comprobante en particular.

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

Es una de las constantes pertenecientes al enum *TiposComprobante*.

***	<i>ComprobanteDonacion</i>	907	***	<i>ReciboM</i>	54
*	<i>FacturaA</i>	1	***	<i>ReciboX</i>	902
*	<i>FacturaB</i>	6	***	<i>RemitoR</i>	91
*	<i>FacturaC</i>	11	***	<i>RemitoX</i>	901
*	<i>FacturaM</i>	51	**	<i>Tique</i>	83
***	<i>Generico</i>	910	**	<i>TiqueFacturaA</i>	81
*	<i>NotaDeCreditoA</i>	3	**	<i>TiqueFacturaB</i>	82
*	<i>NotaDeCreditoB</i>	8	**	<i>TiqueFacturaC</i>	111
*	<i>NotaDeCreditoC</i>	13	**	<i>TiqueFacturaM</i>	118
*	<i>NotaDeCreditoM</i>	53	**	<i>TiqueNotaCredito</i>	110
*	<i>NotaDeDebitoA</i>	2	**	<i>TiqueNotaCreditoA</i>	112
*	<i>NotaDeDebitoB</i>	7	**	<i>TiqueNotaCreditoB</i>	113
*	<i>NotaDeDebitoC</i>	12	**	<i>TiqueNotaCreditoC</i>	114
*	<i>NotaDeDebitoM</i>	52	**	<i>TiqueNotaCreditoM</i>	119
***	<i>PresupuestoX</i>	903	**	<i>TiqueNotaDebitoA</i>	115
***	<i>ReciboA</i>	4	**	<i>TiqueNotaDebitoB</i>	116
***	<i>ReciboB</i>	9	**	<i>TiqueNotaDebitoC</i>	117
***	<i>ReciboC</i>	15	**	<i>TiqueNotaDebitoM</i>	120

- * Disponibles en impresoras fiscal 2G que soporten impresión sobre formularios continuos u hojas sueltas.
- ** Disponibles en impresoras fiscales 2G que soporten impresión sobre rollos de papel continuo.
- *** Disponibles en todos los modelos de impresoras fiscales 2G.

NumeroInicial

Informa el número inicial del rango de comprobantes del tipo reportado en el segundo campo del registro.

NumeroFinal

Informa el número final del rango de comprobantes del tipo reportado en el segundo campo del registro.

Total

Informa el importe total; acumulado correspondiente a los documentos de tipo reportado en el segundo campo del registro, y para el rango numérico informado.

RegDNFH_NoAcum

Informa datos asociados a los documentos no fiscales homologados emitidos que no acumulan totales.

Cuando *Registro = RegistroDetalladoDNFHNoAcum* la respuesta fiscal continúa con una estructura definida internamente como:

```
typedef
struct tagContinuarConsultaAcumuladosRegDNFH_NoAcum {
    [helpstring("CodigoComprobante")]
}
    TiposComprobante CodigoComprobante;
    [helpstring("NumeroInicial")]
}
    long NumeroInicial;
    [helpstring("NumeroFinal")]
}
    long NumeroFinal;
} ContinuarConsultaAcumuladosRegDNFH_NoAcum;
```

CodigoComprobante Informa un código de tipo de comprobante correspondiente a un DNFH (Documento No Fiscal Homologado) que no genera acumulados.

El valor '0' (*NoDocumento*) no identifica a ningún tipo de comprobante. La impresora fiscal 2G no emite comprobantes de este tipo. Es un valor interno y propio de la impresora fiscal 2G, que se utiliza en algunos comandos fiscales cuando la información del resto de los campos del comando fiscal, o de la respuesta fiscal, no está asociada a un tipo de comprobante en particular.

Es una de las constantes pertenecientes al enum *TiposComprobante*.

***	<i>ComprobanteDonacion</i>	907	***	<i>ReciboM</i>	54
*	<i>FacturaA</i>	1	***	<i>ReciboX</i>	902
*	<i>FacturaB</i>	6	***	<i>RemitoR</i>	91
*	<i>FacturaC</i>	11	***	<i>RemitoX</i>	901
*	<i>FacturaM</i>	51	**	<i>Tique</i>	83
***	<i>Generico</i>	910	**	<i>TiqueFacturaA</i>	81
*	<i>NotaDeCreditoA</i>	3	**	<i>TiqueFacturaB</i>	82
*	<i>NotaDeCreditoB</i>	8	**	<i>TiqueFacturaC</i>	111
*	<i>NotaDeCreditoC</i>	13	**	<i>TiqueFacturaM</i>	118
*	<i>NotaDeCreditoM</i>	53	**	<i>TiqueNotaCredito</i>	110
*	<i>NotaDeDebitoA</i>	2	**	<i>TiqueNotaCreditoA</i>	112
*	<i>NotaDeDebitoB</i>	7	**	<i>TiqueNotaCreditoB</i>	113
*	<i>NotaDeDebitoC</i>	12	**	<i>TiqueNotaCreditoC</i>	114
*	<i>NotaDeDebitoM</i>	52	**	<i>TiqueNotaCreditoM</i>	119
***	<i>PresupuestoX</i>	903	**	<i>TiqueNotaDebitoA</i>	115

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

	<table border="1"> <tr> <td>*** ReciboA</td> <td>4</td> <td>** TiqueNotaDebitoB</td> <td>116</td> </tr> <tr> <td>*** ReciboB</td> <td>9</td> <td>** TiqueNotaDebitoC</td> <td>117</td> </tr> <tr> <td>*** ReciboC</td> <td>15</td> <td>** TiqueNotaDebitoM</td> <td>120</td> </tr> </table>	*** ReciboA	4	** TiqueNotaDebitoB	116	*** ReciboB	9	** TiqueNotaDebitoC	117	*** ReciboC	15	** TiqueNotaDebitoM	120
*** ReciboA	4	** TiqueNotaDebitoB	116										
*** ReciboB	9	** TiqueNotaDebitoC	117										
*** ReciboC	15	** TiqueNotaDebitoM	120										
	<table border="1"> <tr> <td>*</td> <td>Disponibles en impresoras fiscal 2G que soporten impresión sobre formularios continuos u hojas sueltas.</td> </tr> <tr> <td>**</td> <td>Disponibles en impresoras fiscales 2G que soporten impresión sobre rollos de papel continuo.</td> </tr> <tr> <td>***</td> <td>Disponibles en todos los modelos de impresoras fiscales 2G.</td> </tr> </table>	*	Disponibles en impresoras fiscal 2G que soporten impresión sobre formularios continuos u hojas sueltas.	**	Disponibles en impresoras fiscales 2G que soporten impresión sobre rollos de papel continuo.	***	Disponibles en todos los modelos de impresoras fiscales 2G.						
*	Disponibles en impresoras fiscal 2G que soporten impresión sobre formularios continuos u hojas sueltas.												
**	Disponibles en impresoras fiscales 2G que soporten impresión sobre rollos de papel continuo.												
***	Disponibles en todos los modelos de impresoras fiscales 2G.												
<i>NumeroInicial</i>	Informa el número inicial del rango de comprobantes del tipo reportado en el segundo campo del registro.												
<i>NumeroFinal</i>	Informa el número final del rango de comprobantes del tipo reportado en el segundo campo del registro.												
<i>RegGlobal</i>	<p>Informa datos globales asociados a todos los tipos de comprobantes emitidos.</p> <p>Cuando <i>Registro = RegistroGlobal</i> la respuesta fiscal continúa con una estructura definida internamente como:</p> <pre>typedef struct tagContinuarConsultaAcumuladosRegGlobal { [helpstring("DF_CantidadCancelados")] } long DF_CantidadCancelados; [helpstring("DF_CantidadEmitidos")] } long DF_CantidadEmitidos; [helpstring("DF_Total")] } double DF_Total; [helpstring("DF_TotalExento")] } double DF_TotalExento; [helpstring("DF_TotalNoGravado")] } double DF_TotalNoGravado; [helpstring("DF_TotalGravado")] } double DF_TotalGravado; [helpstring("DF_TotalIVA")] } double DF_TotalIVA; [helpstring("DF_TotalTributos")] } double DF_TotalTributos; [helpstring("DNFH_Total")] } double DNFH_Total; } ContinuarConsultaAcumuladosRegGlobal;</pre>												
<i>DF_CantidadCancelados</i>	Informa la cantidad total de comprobantes fiscales que fueron cancelados hasta el momento de la consulta.												

<i>DF_CantidadEmitidos</i>	<p>Informa la cantidad total de comprobantes fiscales que fueron emitidos hasta el momento de la consulta.</p>
<i>DF_Total</i>	<p>Informa el importe total; acumulado correspondiente a DF (Documentos Fiscales). El importe total reportado incluye IVA y otros tributos, de corresponder.</p> <p>Tener en cuenta que los tiques notas de crédito son Documentos Fiscales, por lo tanto en este campo se reporta la suma de todos los DF, menos la suma de los tiques nota de crédito emitidos.</p>
<i>DF_TotalExento</i>	<p>Informa el importe total exento; acumulado correspondiente a DF (Documentos Fiscales). El importe total reportado incluye IVA y otros tributos, de corresponder.</p> <p>Tener en cuenta que los tiques notas de crédito son Documentos Fiscales, por lo tanto en este campo se reporta la suma de todos los DF, menos la suma de los tiques nota de crédito emitidos.</p>
<i>DF_TotalNoGravado</i>	<p>Informa el importe total no gravado; acumulado correspondiente a DF (Documentos Fiscales). El importe total reportado incluye IVA y otros tributos, de corresponder.</p> <p>Tener en cuenta que los tiques notas de crédito son Documentos Fiscales, por lo tanto en este campo se reporta la suma de todos los DF, menos la suma de los tiques nota de crédito emitidos.</p>
<i>DF_TotalGravado</i>	<p>Informa el importe total gravado; acumulado correspondiente a DF (Documentos Fiscales). El importe total reportado incluye IVA y otros tributos, de corresponder.</p> <p>Tener en cuenta que los tiques notas de crédito son Documentos Fiscales, por lo tanto en este campo se reporta la suma de todos los DF, menos la suma de los tiques nota de crédito emitidos.</p>
<i>DF_TotalIVA</i>	<p>Informa el importe total de IVA; acumulado correspondiente a DF (Documentos Fiscales).</p> <p>Tener en cuenta que los tiques notas de crédito son Documentos Fiscales, por lo tanto en este campo se reporta la suma de todos los DF, menos la suma de los tiques nota de crédito emitidos.</p>

<i>DF_TotalTributos</i>	<p>Informa el importe total de otros tributos; acumulado correspondiente a DF (Documentos Fiscales).</p> <p>Tener en cuenta que los tiques notas de crédito son Documentos Fiscales, por lo tanto en este campo se reporta la suma de todos los DF, menos la suma de los tiques nota de crédito emitidos.</p>
<i>DNFH_Total</i>	<p>Informa el importe total acumulado correspondiente a DNFH (Documentos No Fiscales Homologados) que generan acumulados.</p>

CopiarComprobante()

Permite enviar a la impresora fiscal 2G el comando fiscal de pedido de reimpresión de cualquier comprobante, emitido previamente, y sin necesidad de que el mismo pertenezca a la jornada fiscal/comercial en curso.

Definido internamente como:

```
CopiarComprobante(
    [in] TiposComprobante CodigoComprobante,
    [in] long NumeroComprobante,
    [in, optional, defaultvalue(83)] SiNo Imprimir,
    [out, retval] RespuestaCopiarComprobante* );
```

CodigoComprobante

Permite especificar el tipo de comprobante que se desea reimprimir. Debe ser alguna de las constantes pertenecientes al enum *TiposComprobante*.

*** <i>ComprobanteDonacion</i>	907	*** <i>ReciboM</i>	54
* <i>FacturaA</i>	1	*** <i>ReciboX</i>	902
* <i>FacturaB</i>	6	*** <i>RemitoR</i>	91
* <i>FacturaC</i>	11	*** <i>RemitoX</i>	901
* <i>FacturaM</i>	51	** <i>Tique</i>	83
*** <i>Generico</i>	910	** <i>TiqueFacturaA</i>	81
* <i>NotaDeCreditoA</i>	3	** <i>TiqueFacturaB</i>	82
* <i>NotaDeCreditoB</i>	8	** <i>TiqueFacturaC</i>	111
* <i>NotaDeCreditoC</i>	13	** <i>TiqueFacturaM</i>	118
* <i>NotaDeCreditoM</i>	53	** <i>TiqueNotaCredito</i>	110
* <i>NotaDeDebitoA</i>	2	** <i>TiqueNotaCreditoA</i>	112
* <i>NotaDeDebitoB</i>	7	** <i>TiqueNotaCreditoB</i>	113
* <i>NotaDeDebitoC</i>	12	** <i>TiqueNotaCrediotC</i>	114

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

	* <i>NotaDeDebitoM</i>	52	** <i>TiqueNotaCreditoM</i>	119				
	*** <i>PresupuestoX</i>	903	** <i>TiqueNotaDebitoA</i>	115				
	*** <i>ReciboA</i>	4	** <i>TiqueNotaDebitoB</i>	116				
	*** <i>ReciboB</i>	9	** <i>TiqueNotaDebitoC</i>	117				
	*** <i>ReciboC</i>	15	** <i>TiqueNotaDebitoM</i>	120				
	<hr/> <p>* Disponibles en impresoras fiscales 2G que soporten impresión sobre formularios continuos u hojas sueltas.</p> <p>** Disponibles en impresoras fiscales 2G que soporten impresión sobre rollos de papel continuo.</p> <p>*** Disponibles en todos los modelos de impresoras fiscales 2G.</p>							
<i>NumeroComprobante</i>	Permite especificar el número del comprobante que se desea reimprimir.							
<i>Imprimir</i>	<p>No disponible en los modelos de impresoras fiscales 2G: SMH/PT-1000F v01.00 y SMH/PT-250F v01.00.</p> <p>Disponible en impresoras fiscales 2G con protocolo '1' o superior.</p> <p>Permite especificar si efectivamente se reimprime el comprobante, o si solamente se retorna información de acceso a un recurso de la impresora fiscal 2G, para la visualización del comprobante (por ejemplo, en un navegador de internet). Debe ser alguna de las constantes pertenecientes al enum <i>SiNo</i>.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tbody> <tr> <td><i>No</i></td> <td>78</td> </tr> <tr> <td><i>Si</i></td> <td>83</td> </tr> </tbody> </table>				<i>No</i>	78	<i>Si</i>	83
<i>No</i>	78							
<i>Si</i>	83							

Ejemplo:

```
Dim resp As HasarArgentina.RespuestaCopiarComprobante

resp = hasar.CopiarComprobante(HasarArgentina.TiposComprobante.TiqueFacturaC, 101, _
    HasarArgentina.SiNo.No)
Debug.Print("Indice = " & "[" & resp.Indice & "]")
Debug.Print("Indice = " & "[" & resp.Zeta & "]")
```

La respuesta fiscal al comando fiscal generado por '*CopiarComprobante()*' es la estructura *RespuestaCopiarComprobante*, definida internamente como:

```
typedef  
struct tagRespuestaCopiarComprobante {  
    [helpstring("Zeta")  
]  
    long Zeta;  
    [helpstring("Indice")  
]  
    long Indice;  
} RespuestaCopiarComprobante;
```

<i>Zeta</i>	Informa el número de <i>Informe Diario de Cierre</i> (cierre 'Z') en el que se encuentra el comprobante solicitado. Puede utilizarse junto con el campo <i>Indice</i> para hacer un llamado por HTTP a la impresora fiscal 2G para ver el comprobante (ver descripción del próximo campo de la estructura).
<i>Indice</i>	<p>Este índice puede utilizarse junto al anterior (número de <i>Informe Diario de Cierre</i>) para hacer un llamado por HTTP a la impresora fiscal 2G y ver el comprobante en cuestión.</p> <p>Por ejemplo, si la impresora fiscal 2G cuya dirección IP es '192.168.1.5' retornase, para un determinado comprobante, <i>Zeta</i> = 1 e <i>Indice</i> = 1, podría verse el mismo mediante un navegador de internet, accediendo a:</p> <p style="text-align: center;">http://192.168.1.5/verdoc.html?ZETA=1&DOCID=1</p> <p>La impresora fiscal 2G solicitará una clave de acceso; por defecto '9999' (y no especificar usuario alguno).</p>

CrearCodigoQR()

Disponible solamente en impresoras desarrolladas para el GCABA (Gobierno de la Ciudad Autónoma de Buenos Aires).

Permite enviar a la impresora fiscal 2G el comando fiscal de creación de un código QR, el cual será impreso por el comando fiscal generado por '*CerrarDocumento()*'. El texto que conformará el código QR es almacenado en la memoria de trabajo de la impresora fiscal 2G.

Definido internamente como:

```
CrearCodigoQR([in] BSTR Informacion);
```

<i>Informacion</i>	String de hasta 255 caracteres, que contiene la información que desea mostrarse cuando se escanee el código QR.
--------------------	---

Ejemplo:

```
hasar.CrearCodigoQR("www.grupohasar.com/GCABA - Tecnología al servicio del ciudadano")
```

El ejemplo anterior genera un código QR como el siguiente:



EliminarCodigoQR()

Disponible solamente en impresoras desarrolladas para el GCABA (Gobierno de la Ciudad Autónoma de Buenos Aires).

Permite enviar a la impresora fiscal 2G el comando fiscal de eliminación de la memoria de trabajo, del texto con el cual la impresora fiscal 2G construye el código QR a imprimir al cerrar un comprobante.

Definido internamente como:

```
EliminarCodigoQR();
```

Ejemplo:

```
hasar.EliminarCodigoQR()
```

EliminarLogoAdicional()

Disponible solamente en impresoras desarrolladas para el GCABA (Gobierno de la Ciudad Autónoma de Buenos Aires).

Permite enviar a la impresora fiscal 2G el comando fiscal de eliminación de la memoria de trabajo, de uno de los logotipos adicionales que en ella se han almacenado.

Definido internamente como:

```
EliminarLogoAdicional([in] long NumeroLogo);
```

<i>NumeroLogo</i>	Permite especificar el número de logotipo adicional que se desea eliminar de la memoria de trabajo de la impresora fiscal 2G. Debe ser un valor comprendido entre '0' y '10', reservándose el valor '0' para la eliminación del logotipo comercial del emisor.
-------------------	--

Ejemplo:

```
hasar.EliminarLogoAdicional(3)
```

EliminarLogoEmisor()

Permite enviar a la impresora fiscal 2G el comando fiscal que elimina de la memoria de trabajo, el logotipo comercial del emisor (el cual se imprime al comienzo del encabezado de los comprobantes).

El comando fiscal generado por *'EliminarLogoEmisor()'* requiere que, previamente, se envíe el comando fiscal generado por *'CerrarJornadaFiscal() -Informe Diario de Cierre (cierre 'Z')'*.

Definido internamente como:

```
EliminarLogoEmisor();
```

Ejemplo:

```
hasar.EliminarLogoEmisor()
```

EnviarDocumentoCorreo()

Permite enviar a la impresora fiscal 2G el comando fiscal de envío por correo electrónico, al destinatario que se especifique, de un comprobante almacenado en la CTD (cinta testigo digital, o memoria de auditoría) del equipo fiscal.

Esta alternativa de envío es válida cuando el comprobante en cuestión no quiere ser enviado en el momento en que se ejecuta el comando de cierre del comprobante durante su emisión, o quiere enviarse una nueva copia al mismo u otro destinatario.

El comprobante seleccionado para enviar por correo electrónico es extraído de la memoria de auditoría (CTD, o cinta testigo digital) y convertido al formato HTML. Es este último archivo el que se adjunta al correo electrónico a enviar.

Las solicitudes de envío de comprobantes por correo electrónico se encolan internamente y se envían en función de la actividad del servidor de correo electrónico (activo/inactivo). Este tratamiento diferido se hace para que el sistema de facturación no tenga que esperar el éxito o fracaso del envío del correo electrónico solicitado. Ante cualquier error que se produzca con el envío del correo electrónico solicitado (servidor caído, correo electrónico rechazado, etc.), el sistema de facturación no tiene ofrecido un mecanismo (por parte de la impresora fiscal 2G) para notificarse del incidente.

Definido internamente como:

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

```

EnviarDocumentoCorreo(
    [in] TiposComprobante CodigoComprobante,
    [in] long NumeroComprobante,
    [in] BSTR DireccionEMail);
    
```

CodigoComprobante

Permite especificar el tipo del comprobante que se desea enviar por correo electrónico. Debe ser alguna de las constantes pertenecientes al enum *TiposComprobante*.

***	<i>ComprobanteDonacion</i>	907	***	<i>ReciboM</i>	54
*	<i>FacturaA</i>	1	***	<i>ReciboX</i>	902
*	<i>FacturaB</i>	6	***	<i>RemitoR</i>	91
*	<i>FacturaC</i>	11	***	<i>RemitoX</i>	901
*	<i>FacturaM</i>	51	**	<i>Tique</i>	83
***	<i>Generico</i>	910	**	<i>TiqueFacturaA</i>	81
*	<i>NotaDeCreditoA</i>	3	**	<i>TiqueFacturaB</i>	82
*	<i>NotaDeCreditoB</i>	8	**	<i>TiqueFacturaC</i>	111
*	<i>NotaDeCreditoC</i>	13	**	<i>TiqueFacturaM</i>	118
*	<i>NotaDeCreditoM</i>	53	**	<i>TiqueNotaCredito</i>	110
*	<i>NotaDeDebitoA</i>	2	**	<i>TiqueNotaCreditoA</i>	112
*	<i>NotaDeDebitoB</i>	7	**	<i>TiqueNotaCreditoB</i>	113
*	<i>NotaDeDebitoC</i>	12	**	<i>TiqueNotaCreditoC</i>	114
*	<i>NotaDeDebitoM</i>	52	**	<i>TiqueNotaCreditoM</i>	119
***	<i>PresupuestoX</i>	903	**	<i>TiqueNotaDebitoA</i>	115
***	<i>ReciboA</i>	4	**	<i>TiqueNotaDebitoB</i>	116
***	<i>ReciboB</i>	9	**	<i>TiqueNotaDebitoC</i>	117
***	<i>ReciboC</i>	15	**	<i>TiqueNotaDebitoM</i>	120

- * Disponibles en impresoras fiscales 2G que soporten impresión sobre formularios continuos u hojas sueltas.
- ** Disponibles en impresoras fiscales 2G que soporten impresión sobre rollos de papel continuo.
- *** Disponibles en todos los modelos de impresoras fiscales 2G.

NumeroComprobante

Permite especificar el número del comprobante que se desea enviar por correo electrónico.

DireccionEMail

Permite especificar la dirección de correo electrónico del destinatario del documento que se desea enviar.

El comando fiscal generado por '*EnviarDocumentoCorreo()*' requiere que la impresora fiscal 2G cuente con una

correcta configuración del servidor de correo electrónico saliente (SMTP) que se va a utilizar (ver comando fiscal generado por '*ConfigurarServidorCorreo()*'). El servicio de SMTP a emplear puede requerir, o no, autenticación.

Ejemplo:

```
hasar.EnviarDocumentoCorreo(HasarArgentina.TiposComprobante.TiqueFacturaC, 75, _
    "<ricardo.cardenes@hasar.com>")
```

Notar que el destinatario del correo electrónico debe especificarse encerrado entre los caracteres '<' y '>'.

ImprimirAjuste()

Permite enviar a la impresora fiscal 2G el comando fiscal de impresión de ajustes impositivos (positivos/negativos), o de impresión de bonificaciones generales. La diferencia entre un ajuste negativo y una bonificación general reside en el código matrix que se debe informar al utilizar '*ImprimirAjuste()*'.

Definido internamente como:

```
ImprimirAjuste(
    [in] BSTR Descripcion,
    [in] double Monto,
    [in] ModosDeDisplay ModoDisplay,
    [in] ModosDePrecio ModoBaseTotal,
    [in] BSTR CodigoProducto,
    [in] TiposDeAjustes Operacion,
    [out, retval] RespuestaImprimirAjuste* );
```

<i>Descripción</i>	Permite especificar el texto a imprimir, asociado a la operación que se realiza.						
<i>Monto</i>	Permite especificar el monto asociado a la operación que se realiza.						
<i>ModoDisplay</i>	Permite especificar el modo de uso del visor de cliente conectado a la impresora fiscal 2G. Debe ser alguna de las constantes pertenecientes al enum <i>ModosDeDisplay</i> . <table border="1" data-bbox="821 1585 1054 1738"> <tr> <td><i>DisplayNo</i></td> <td>48</td> </tr> <tr> <td><i>DisplaySi</i></td> <td>49</td> </tr> <tr> <td><i>DisplayRep</i></td> <td>50</td> </tr> </table>	<i>DisplayNo</i>	48	<i>DisplaySi</i>	49	<i>DisplayRep</i>	50
<i>DisplayNo</i>	48						
<i>DisplaySi</i>	49						
<i>DisplayRep</i>	50						
<i>ModoBaseTotal</i>	En impresoras fiscales 2G que no soportan conexión de visor de cliente, especificar <i>DisplayNo</i> . Permite especificar si el monto asociado a la operación que se realiza es una base imponible, o incluye los impuestos que le afecten (monto total). Debe ser alguna de las constantes pertenecientes al enum <i>ModosDePrecio</i> .						

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

	<table border="1"> <tr> <td><i>ModoPrecioBase</i></td> <td style="text-align: center;">66</td> </tr> <tr> <td><i>ModoPrecioTotal</i></td> <td style="text-align: center;">84</td> </tr> </table>	<i>ModoPrecioBase</i>	66	<i>ModoPrecioTotal</i>	84		
<i>ModoPrecioBase</i>	66						
<i>ModoPrecioTotal</i>	84						
<i>CodigoProducto</i>	<p>Permite especificar el código matrix asociado a la operación que se realiza. Debe ser alguno de los especificados en la tabla siguiente:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">7790001001030</td> <td style="text-align: center;">Bonificación general.</td> </tr> <tr> <td style="text-align: center;">7790001001139</td> <td style="text-align: center;">Ajuste impositivo.</td> </tr> </table>	7790001001030	Bonificación general.	7790001001139	Ajuste impositivo.		
7790001001030	Bonificación general.						
7790001001139	Ajuste impositivo.						
<i>Operacion</i>	<p>Permite especificar la operación que se realiza. Debe ser alguna de las constantes pertenecientes al enum <i>TiposDeAjustes</i>.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;"><i>Bonificacion_General</i></td> <td style="text-align: center;">66</td> </tr> <tr> <td style="text-align: center;"><i>Ajuste_Neg</i></td> <td style="text-align: center;">68</td> </tr> <tr> <td style="text-align: center;"><i>Ajuste_Pos</i></td> <td style="text-align: center;">82</td> </tr> </table> <p>El comando fiscal generado por '<i>ImprimirAjuste()</i>' puede enviarse a la impresora fiscal 2G cuantas veces resulte necesario. Una vez enviado el primer comando fiscal generado por '<i>ImprimirAjuste()</i>' queda inhabilitada la impresión de nuevos ítems.</p> <p>Las leyendas 'AJUSTE', y 'BONIFICACION', no son modificables, ni se puede evitar su impresión. El valor entre paréntesis, en cada línea que el monto de la operación es prorrateada, indica la alícuota de IVA por la cual está afectado. Si este valor es cero, el monto puede estar afectado por alícuota 0 %, ser un monto no gravado, o estar exento de IVA. Si en la misma línea que se imprime la alícuota de IVA se imprime un valor entre corchetes, por ejemplo [90.91], debe interpretarse que el monto está también afectado por impuestos internos. Este valor no indica porcentaje alguno de impuesto interno en forma directa. Los impuestos internos sólo se mostrarán en el caso de ser variables (los impuestos internos fijos no pueden estar afectados por ajustes o bonificaciones). Notar que salvo el caso del recargo (es afectado por una única alícuota de IVA) o del anticipo/seña (ocurre lo mismo), los ajustes y bonificaciones se imprimen detallando el prorrateo sobre cada alícuota de IVA del comprobante.</p>	<i>Bonificacion_General</i>	66	<i>Ajuste_Neg</i>	68	<i>Ajuste_Pos</i>	82
<i>Bonificacion_General</i>	66						
<i>Ajuste_Neg</i>	68						
<i>Ajuste_Pos</i>	82						

Ejemplo:

```
hasar.ImprimirAjuste("Ajuste Pos. ...", 0.01, HasarArgentina.ModosDeDisplay.DisplayNo, _
    HasarArgentina.ModosDePrecio.ModoPrecioBase, "7790001001030", _
    HasarArgentina.TiposDeAjustes.AjustePos)
```

La respuesta fiscal al comando fiscal generado por '*ImprimirAjuste()*' es la estructura *RespuestaImprimirAjuste*, definida internamente como:

```
typedef
struct tagRespuestaImprimirAjuste {
    [helpstring("IndiceAuditoria")]
}
    long IndiceAuditoria;
    [helpstring("CantidadIVAs")]
}
    long CantidadIVAs;
} RespuestaImprimirAjuste;
```

<i>IndiceAuditoria</i>	Informa el orden dentro del documento en emisión, de acuerdo a como está registrada la operación en la CTD (cinta de auditoría electrónica, o memoria de auditoría). Puede ser utilizado para consultar mayor información sobre la operación, por ejemplo mediante el comando fiscal ' <i>ConsultarInformacionAritmetica</i> '.
<i>CantidadIVAs</i>	Informa la cantidad de IVAs de la operación a ser consultada. Puede ser utilizado para consultar mayor información sobre la operación, por ejemplo mediante el comando fiscal ' <i>ConsultarInformacionAritmetica</i> '.

ImprimirAnticipoBonificacionEnvases()

Permite enviar a la impresora fiscal 2G el comando fiscal de impresión de bonificaciones sobre impuestos, recargos financieros, cobro de anticipos/señas, descuento de anticipos/señas, y/o devoluciones de envases.

Definido internamente como:

```
ImprimirAnticipoBonificacionEnvases (
    [in] BSTR Descripcion,
    [in] double Monto,
    [in] CondicionesIVA CondicionIVA,
    [in] double AlicuotaIVA,
    [in] ModosDeImpuestosInternos TipoImpuestoInterno,
    [in] double MagnitudImpuestoInterno,
    [in] ModosDeDisplay ModoDisplay,
    [in] ModosDePrecio ModoBaseTotal,
    [in] BSTR Codigoproducto,
    [in] TiposDeOperacionesGlobalesIVA Operacion,
    [out, retval] RespuestaImprimirAnticipoBonificacionEnvases* );
```

<i>Descripción</i>	Permite especificar el texto a imprimir, asociado a la operación que se realiza.				
<i>Monto</i>	Permite especificar el monto asociado a la operación que se realiza.				
<i>CondicionIVA</i>	Permite especificar la condición de IVA que caracteriza a la operación que se realiza. Debe ser alguna de las constantes pertenecientes al enum <i>CondicionesIVA</i> . <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>No_Gravado</td> <td>1</td> </tr> <tr> <td>Exento</td> <td>2</td> </tr> </table>	No_Gravado	1	Exento	2
No_Gravado	1				
Exento	2				

Gravado | 7

AlicuotaIVA

Permite especificar el valor porcentual de alícuota de IVA que afecta a la operación que se realiza.

TipImpuestoInterno

Permite especificar el tipo de impuesto interno que afecta a la operación que se realiza. Debe ser alguna de las constantes pertenecientes al enum *ModosDeImpuestosInternos*.

IIFijoMonto	36
IIVariablePorcentual	37
IIFijoKIVA	43
IIVariableKIVA	48

MagnitudImpuestoInterno

Permite especificar la magnitud del impuesto interno (porcentaje, monto, o coeficiente) en relación coherente al tipo especificado.

ModoDisplay

Permite especificar el modo de uso del visor de cliente conectado a la impresora fiscal 2G. Debe ser alguna de las constantes pertenecientes al enum *ModosDeDisplay*.

<i>DisplayNo</i>	48
<i>DisplaySi</i>	49
<i>DisplayRep</i>	50

En impresoras fiscales 2G que no soportan conexión de visor de cliente, especificar *DisplayNo*.

ModoBaseTotal

Permite especificar si el monto asociado a la operación que se realiza es una base imponible, o incluye los impuestos que le afecten (monto total). Debe ser alguna de las constantes pertenecientes al enum *ModosDePrecio*.

<i>ModoPrecioBase</i>	66
<i>ModoPrecioTotal</i>	84

CodigoProducto

Permite especificar el código matrix asociado a la operación que se realiza. Debe ser alguno de los especificados en la tabla siguiente:

7790001001030	Bonificación sobre alícuota IVA.
7790001001047	Recargo financiero.
7790001001856	Cobro/descuento de anticipo/seña.

Operacion

Permite especificar la operación que se realiza. Debe ser alguna de las constantes pertenecientes al enum *TiposDeOperacionesGlobalesIVA*.

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

<i>CobroAnticipo</i>	65
<i>BonificacionIVA</i>	66
<i>DescuentoAnticipo</i>	68
<i>DevolucionEnvases</i>	69
<i>RecargoIVA</i>	82

Una vez enviado el primer comando fiscal generado por '*ImprimirAnticipoBonificacionEnvases()*' queda inhabilitada la impresión de nuevos ítems (no se puede continuar con más ventas en el comprobante actual).

En el caso de emisión de *Presupuestos* 'X' no se acepta el comando fiscal generado por '*ImprimirAnticipoBonificacionEnvases()*' si la operación declarada es un cobro de anticipo/seña (operación positiva).

Si el comando '*ImprimirAnticipoBonificacionEnvases()*' se emplea para realizar una operación de cobro/descuento de anticipo/seña, a continuación sólo podrán realizarse más operaciones del mismo tipo. Quedan inhabilitadas las operaciones de descuento o ajustes, en forma posterior. La normativa fiscal vigente no permite la aplicación de operaciones de descuento/ajuste sobre montos correspondientes a anticipos/señas.

Las leyendas: 'RECARGO', 'BONIFICACION', y 'DEVOLUCION DE ENVASES' no son modificables, ni se puede evitar su impresión.

El valor entre paréntesis, en la misma línea que el monto de la operación, indica la alícuota de IVA por la cual está afectado. Si este valor es cero, el monto puede estar afectado por alícuota 0%, ser un monto no gravado, o estar exento de IVA.

Si en la misma línea que se imprime la alícuota de IVA se imprime un valor entre corchetes, por ejemplo [90.91], debe interpretarse que el monto está también afectado por impuestos internos. Este valor no indica porcentaje alguno de impuesto interno en forma directa (es un coeficiente, sólo de interés para la AFIP).

Ejemplo:

```
hasar.ImprimirAnticipoBonificacionEnvases("Devolución Envases ...", 12.4, _  
    HasarArgentina.CondicionesIVA.Gravado, 21.0, _  
    HasarArgentina.ModosDeImpuestosInternos.IIVariablePorcentual, 0.0, _  
    HasarArgentina.ModosDeDisplay.DisplayNo, HasarArgentina.ModosDePrecio.ModoprecioBase, _  
    "7790001001030", HasarArgentina.TiposDeOperacionesGlobalesIVA.DevolucionEnvases)
```

La respuesta fiscal al comando fiscal generado por '*ImprimirAnticipoBonificacionEnvases()*' es la estructura *RespuestaImprimirAnticipoBonificacionEnvases*, definida internamente como:

```
typedef  
struct tagRespuestaImprimirAnticipoBonificacionEnvases {  
    [helpstring("IndiceAuditoria")  
]  
    long IndiceAuditoria;  
} RespuestaImprimirAnticipoBonificacionEnvases;
```

<i>IndiceAuditoria</i>	Informa el orden dentro del documento actual, de acuerdo a como está registrada la operación en la CTD (cinta de auditoría electrónica, o memoria de auditoría). Puede ser utilizado para consultar mayor información sobre la operación, por ejemplo mediante el comando fiscal generado por ' <i>ConsultarInformacionAritmetica()</i> '.
------------------------	--

ImprimirConceptoRecibo()

Permite enviar a la impresora fiscal 2G el comando fiscal de impresión de textos (conceptos) aclaratorios de los motivos por los cuales se emite el *Recibo 'X'*.

Definido internamente como:

```
ImprimirConceptoRecibo([in] BSTR Descripcion);
```

<i>Descripcion</i>	Permite especificar una línea de concepto por el cual se emite el <i>Recibo 'X'</i> . El texto será impreso cuando la impresora fiscal 2G reciba el comando fiscal generado por ' <i>CerrarDocumento()</i> '.
--------------------	---

El comando fiscal generado por '*ImprimirConceptoRecibo()*' sólo es aceptado cuando se encuentra abierto un *Recibo 'X'*, y ya se ha generado el monto del mismo. El envío de este comando fiscal debe ser:

- Posterior al comando fiscal generado por '*ImprimirItem()*'.
- Previo a los comandos fiscales generados por '*ImprimirPago()*' y/o '*CerrarDocumento()*'.

La cantidad de veces consecutivas que puede enviarse el comando fiscal generado por '*ImprimirConceptoRecibo()*' no se encuentra limitada. Sin embargo, es obligatorio que, al menos en una oportunidad, este comando sea enviado a la impresora fiscal 2G, para que el *Recibo 'X'* pueda cerrarse.

Ejemplo:

```
hasar.ImprimirConceptoRecibo("Alquiler del mes de Julio de 2019.")
```

ImprimirDescuentoItem()

Permite enviar a la impresora fiscal 2G el comando fiscal de impresión de descuento, aplicado sobre el último ítem impreso (comando fiscal inmediato anterior). Se puede enviar una sola vez por cada ítem.

Definido internamente como:

```
ImprimirDescuentoItem(  
    [in] BSTR Descripcion,  
    [in] double Monto,  
    [in] ModosDeDisplay ModoDisplay,  
    [in] ModosDePrecio ModoBaseTotal,  
    [out, retval] RespuestaImprimirDescuentoItem* );
```

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

<i>Descripcion</i>	Permite especificar el texto a imprimir, asociado a la operación que se realiza.						
<i>Monto</i>	Permite especificar el monto asociado a la operación que se realiza.						
<i>ModoDisplay</i>	<p>Permite especificar el modo de uso del visor de cliente conectado a la impresora fiscal 2G. Debe ser alguna de las constantes pertenecientes al enum <i>ModosDeDisplay</i>.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td><i>DisplayNo</i></td> <td>48</td> </tr> <tr> <td><i>DisplaySi</i></td> <td>49</td> </tr> <tr> <td><i>DisplayRep</i></td> <td>50</td> </tr> </table> <p>En impresoras fiscales 2G que no soportan conexión de visor de cliente, especificar <i>DisplayNo</i>.</p>	<i>DisplayNo</i>	48	<i>DisplaySi</i>	49	<i>DisplayRep</i>	50
<i>DisplayNo</i>	48						
<i>DisplaySi</i>	49						
<i>DisplayRep</i>	50						
<i>ModoBaseTotal</i>	<p>Permite especificar si el monto asociado a la operación que se realiza es una base imponible, o incluye los impuestos que le afecten (monto total). Debe ser alguna de las constantes pertenecientes al enum <i>ModosDePrecio</i>.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td><i>ModoPrecioBase</i></td> <td>66</td> </tr> <tr> <td><i>ModoPrecioTotal</i></td> <td>84</td> </tr> </table>	<i>ModoPrecioBase</i>	66	<i>ModoPrecioTotal</i>	84		
<i>ModoPrecioBase</i>	66						
<i>ModoPrecioTotal</i>	84						

La leyenda 'DESCUENTO SOBRE ULTIMA VENTA' no es modificable, ni es posible evitar su impresión. El valor entre paréntesis, en la misma línea que la descripción del descuento, indica la alícuota de IVA por la cual está afectado. Si este valor es cero, el descuento puede estar afectado por alícuota 0%, ser un descuento no gravado, o estar exento de IVA.

Si en la misma línea que se imprime la alícuota de IVA se imprime un valor entre corchetes, por ejemplo [90.91], debe interpretarse que el descuento está también afectado por impuestos internos. Este valor no indica porcentaje alguno de impuesto interno en forma directa (es un coeficiente, sólo de interés para la AFIP).

Ya sea que el monto del descuento incluya (monto final) o no (monto base) los impuestos que lo afectan, la impresora fiscal 2G considerará a los efectos del cálculo, las tasas de los impuestos que afectan al ítem sobre el cual se aplica el descuento.

Ejemplo:

```
Dim respdescitem As HasarArgentina.RespuestaImprimirDescuentoItem
respdescitem = hasar.ImprimirDescuentoItem("Oferta Uno ...", 42.4, _
    HasarArgentina.ModosDeDisplay.DisplayNo, HasarArgentina.ModosDePrecio.ModoPrecioBase)
Debug.Print("Indice =[" & respdescitem.IndiceAuditoria & "]")
```

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

La respuesta fiscal al comando fiscal generado por '*ImprimirDescuentoItem()*' es la estructura *RespuestaImprimirDescuentoItem*, definida internamente como:

```
typedef
struct tagRespuestaImprimirDescuentoItem {
    [helpstring("IndiceAuditoria")]
}
    long IndiceAuditoria;
} RespuestaImprimirDescuentoItem;
```

<i>IndiceAuditoria</i>	<p>Informa el orden dentro del documento actual, de acuerdo a como está registrada la operación en la CTD (cinta de auditoría electrónica, o memoria de auditoría). Este valor puede ser utilizado para consultar mayor información sobre la operación, por ejemplo mediante el comando fiscal generado por '<i>ConsultarInformacionAritmetica()</i>'.</p> <p>No disponible en las impresoras fiscales 2G con protocolo '0' (SMH/PT-1000F v01.00 y SMH/PT-250F v01.00).</p>
------------------------	---

ImprimirItem()

Permite enviar a la impresora fiscal 2G el comando fiscal de impresión de un ítem.

Definido internamente como:

```
ImprimirItem(
    [in] BSTR Descripcion,
    [in] double Cantidad,
    [in] double PrecioUnitario,
    [in] CondicionesIVA CondicionIVA,
    [in] double AlicuotaIVA,
    [in] ModosDeMonto OperacionMonto,
    [in] ModosDeImpuestosInternos TipoImpuestoInterno,
    [in] double MagnitudImpuestoInterno,
    [in] ModosDeDisplay ModoDisplay,
    [in] ModosDePrecio ModoBaseTotal,
    [in, optional, defaultvalue(1)] long UnidadReferencia,
    [in, optional, defaultvalue("")] BSTRCodigoProducto,
    [in, optional, defaultvalue("")] BSTRCodigoInterno,
    [in, optional, defaultvalue(7)] UnidadesMedida UnidadMedida,
    [out, retval] RespuestaImprimirItem* );
```

<i>Descripcion</i>	Permite especificar el texto que describe al ítem que se va a imprimir. En caso de ser necesario extender la descripción del ítem, en forma previa a este comando fiscal se puede enviar (hasta un máximo de 4 veces consecutivas) el comando fiscal generado por ' <i>ImprimirTextoFiscal()</i> '.
<i>Cantidad</i>	Permite especificar la cantidad de unidades que se venden del ítem a imprimir.
<i>PrecioUnitario</i>	Permite especificar el precio por unidad de medida del ítem a imprimir.

CondicionIVA

Permite especificar la condición de IVA que caracteriza a la operación que se realiza. Debe ser alguna de las constantes pertenecientes al enum *CondicionesIVA*.

<i>No_Gravado</i>	1
<i>Exento</i>	2
<i>Gravado</i>	7

AlicuotaIVA

Permite especificar el valor porcentual de alícuota de IVA que afecta a la operación que se realiza.

OperacionMonto

Permite especificar si la venta del ítem es positiva/negativa. Debe ser alguna de las constantes pertenecientes al enum *ModosDeMonto*.

<i>ModoSumaMonto</i>	77
<i>ModoRestaMonto</i>	109

No se permite la inclusión de ítems negativos en el comprobante, salvo que se los haya incluido en positivo previamente, y en cantidad suficiente como para poder efectuar la corrección mediante la operación de sustracción. No se permite el canje o devolución de ítems no preexistentes en el comprobante en curso de emisión.

TipolImpuestoInterno

Permite especificar el tipo de impuesto interno que afecta a la operación que se realiza. Debe ser alguna de las constantes pertenecientes al enum *ModosDeImpuestosInternos*.

<i>IFijoMonto</i>	36
<i>IVariablePorcentual</i>	37
<i>IFijoKIVA</i>	43
<i>IVariableKIVA</i>	48

MagnitudImpuestoInterno

Permite especificar la magnitud del impuesto interno (porcentaje, monto, o coeficiente) en relación coherente al tipo especificado.

ModoDisplay

Permite especificar el modo de uso del visor de cliente conectado a la impresora fiscal 2G. Debe ser alguna de las constantes pertenecientes al enum *ModosDeDisplay*.

<i>DisplayNo</i>	48
<i>DisplaySi</i>	49
<i>DisplayRep</i>	50

En impresoras fiscales 2G que no soportan conexión de visor de cliente, especificar *DisplayNo*.

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

ModoBaseTotal Permite especificar si el monto asociado a la operación que se realiza es una base imponible, o incluye los impuestos que le afecten (monto total). Debe ser alguna de las constantes pertenecientes al enum *ModosDePrecio*.

<i>ModoPrecioBase</i>	66
<i>ModoPrecioTotal</i>	84

UnidadReferencia Permite especificar la cantidad de unidades que componen la unidad de medida. Por ejemplo, 8 en el caso de un pack de gaseosas.

CodigoProducto Permite especificar el código mátrix (de barras) del ítem a imprimir. De no poseer uno, obligatoriamente deberá especificarse un código interno (sólo entonces este argumento podrá omitirse).

CodigoInterno Permite especificar el código interno (definido por el usuario) del ítem a imprimir. Se puede omitir, únicamente, si se especifica el código mátrix del ítem.

UnidadMedida Permite especificar la unidad de medida, asociada a la cantidad de unidades vendidas del ítem. Debe ser alguna de las constantes pertenecientes al enum *Unidades-Medida*.

<i>Ajustes</i>	91	<i>MetroCubico</i>	4
<i>Anulacion</i>	96	<i>Microcurie</i>	50
<i>Bonificacion</i>	99	<i>Microgramo</i>	33
<i>Centimetro</i>	20	<i>Milicurie</i>	49
<i>CentimetroCubico</i>	27	<i>Miligramo</i>	41
<i>Curie</i>	48	<i>Mililitro</i>	47
<i>DecametroCubico</i>	30	<i>Milimetro</i>	15
<i>Docena</i>	9	<i>MilimetroCubico</i>	16
<i>Donacion</i>	90	<i>Millar</i>	11
<i>Gramo</i>	14	<i>MUIACTANT</i>	36
<i>GramoActivo</i>	22	<i>MUIACTHOR</i>	26
<i>GramoBase</i>	23	<i>MUIACTIG</i>	55
<i>Gruesa</i>	54	<i>Nanogramo</i>	34
<i>Hectolitro</i>	18	<i>OtrasUnidades</i>	98
<i>HectometroCubico</i>	31	<i>Pack</i>	62
<i>Horma</i>	63	<i>Par</i>	8
<i>JuegoPaqueteMazoNaipes</i>	25	<i>Picogramo</i>	35
<i>Kilo</i>	1	<i>Quilate</i>	10

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

<i>KilogramoActivo</i>	21	<i>SenasAnticipos</i>	97
<i>KilogramoBase</i>	53	<i>SinDescripcion</i>	0
<i>KilogramoBruto</i>	61	<i>Tonelada</i>	29
<i>Kilometro</i>	17	<i>UInterActHormonal</i>	51
<i>KilometroCubico</i>	32	<i>UIACTANT</i>	28
<i>KWH</i>	6	<i>UIACTHOR</i>	24
<i>Litro</i>	5	<i>UIACTIG</i>	37
<i>MegaUInterActAantib</i>	12	<i>Unidad</i>	7
<i>MegaUInterActHor</i>	52	<i>UnidadInternaActInmung</i>	13
<i>MegaUnidadIntActInmun g</i>	19		
<i>Metro</i>	2		
<i>MetroCuadrado</i>	3		

En el *Documento Donación*, si bien el comando fiscal generado por '*ImprimirItem()*' es aceptado, sólo se imprime la descripción del mismo, y se emplea la cantidad multiplicada por el precio unitario (monto final) para generar el importe total del comprobante.

En el *Recibo 'X'*, si bien el comando fiscal generado por '*ImprimirItem()*' es aceptado, sólo se emplea la cantidad multiplicada por el precio unitario (monto final) para generar el total del comprobante. Se puede enviar un solo comando fiscal para generar el importe total del comprobante. Especificar: cantidad '1', precio unitario igual al monto total deseado en el recibo, e impuestos que lo afectan en cero.

En el *Remito 'R' ó 'X'*, si bien el comando fiscal generado por '*ImprimirItem()*' es aceptado, no se imprime el monto final de cada ítem. Sí, en cambio, se imprime el resto de información.

El valor entre paréntesis, en la misma línea que se imprime la descripción del ítem, indica la alícuota de IVA por la cual está afectado. Si este valor es cero, el producto puede estar afectado por alícuota 0%, ser un ítem no gravado, o estar exento de IVA.

Si en la misma línea que se imprime la alícuota de IVA se imprime un valor entre corchetes, por ejemplo [90.91], debe interpretarse que el producto está también afectado por impuestos internos. Este valor entre corchetes no indica porcentaje alguno de impuesto interno en forma directa (sólo se trata de un coeficiente de interés para la AFIP).

Ejemplo:

```
Dim respitem As HasarArgentina.RespuestaImprimirItem
respitem = hasar.ImprimirItem("Producto Uno ...", 1, 100.0, -
    HasarArgentina.CondicionesIVA.Gravado, 21.0, _
    HasarArgentina.ModosDeMonto.ModoSumaMonto, _
```

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

```
HasarArgentina.ModosDeImpuestosInternos.IIVariablePorcentual, 10.0, _
HasarArgentina.ModosDeDisplay.DisplayNo, HasarArgentina.ModosDePrecio.ModoPrecioBase, _
1, "779123450001", "00001", HasarArgentina.UnidadesMedida.Unidad)
Debug.Print("Indice =[" & respitem.IndiceAuditoria & "]")
```

La respuesta fiscal al comando fiscal generado por '*ImprimirItem()*' es la estructura *RespuestaImprimirItem*, definida internamente como:

```
typedef
struct tagRespuestaImprimirItem {
    [helpstring("IndiceAuditoria")]
}
    long IndiceAuditoria;
} RespuestaImprimirItem;
```

<i>IndiceAuditoria</i>	<p>Informa el orden dentro del documento actual, como registro en la CTD (cinta de auditoría electrónica, o memoria de auditoría). Puede ser utilizado para consultar mayor información sobre la operación, por ejemplo mediante el comando fiscal generado por '<i>ConsultarInformacionAritmetica()</i>', o para anular la venta con el comando fiscal generado por '<i>AnularItem()</i>'.</p> <p>No disponible en impresoras fiscales 2G con protocolo '0' (SMH/PT-1000F v01.00 y SMH/PT-250F v01.00).</p>
------------------------	--

ImprimirOtrosTributos()

Permite enviar a la impresora fiscal 2G el comando fiscal de impresión de tributos diferentes al IVA, o al impuesto interno asociado al ítem.

Definido internamente como:

```
ImprimirOtrosTributos(
    [in] TiposTributos Codigo,
    [in] BSTR Descripcion,
    [in] double BaseImponible,
    [in] double Importe);
```

<i>Codigo</i>	Permite especificar el tipo de tributo que se desea imprimir en el comprobante. Debe ser alguna de las constantes pertenecientes al enum <i>TiposTributos</i> .			
	<i>IIBB</i>	5	<i>OtrasPrcepciones</i>	9
	<i>ImpuestoInternoltem</i>	10	<i>OtrosTributos</i>	99
	<i>ImpuestosInternos</i>	4	<i>PercepcionIIBB</i>	7
	<i>ImpuestosMunicipales</i>	3	<i>PercepcionImpuestosMunicipales</i>	8
	<i>ImpuestosNacionales</i>	1	<i>PercepcionIVA</i>	6

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

	<i>ImpuestosProvinciales</i>	2	<i>SinImpuestos</i>	0
<i>Descripcion</i>	Permite especificar el texto a imprimir como identificación del tributo.			
<i>BaseImponible</i>	Permite especificar el monto utilizado como base para la determinación del importe del tributo.			
<i>Importe</i>	Permite especificar el importe del tributo, calculado por el sistema de facturación aplicando las reglas que la normativa correspondiente indique.			

El comando fiscal generado por '*ImprimirOtrosTributos()*' puede enviarse a la impresora fiscal 2G cuantas veces resulte necesario. Una vez enviado el primer comando fiscal generado por '*ImprimirOtrosTributos()*' queda inhabilitada la impresión de nuevos ítems, u operaciones de descuentos, ajustes, bonificaciones, etc.

El comando fiscal generado por '*ImprimirOtrosTributos()*' no es aceptado en los Documentos No Fiscales Homologados (DNFH) Donación y Remitos 'R' o 'X'.

Ejemplo:

```
hasar.ImprimirOtrosTributos(HasarArgentina.TiposTributos.IIBB, "Otro Tributo Uno ...", 81.21, _
4.06)
```

ImprimirPago()

Permite enviar a la impresora fiscal 2G el comando de impresión de un medio de pago, utilizado para saldar el monto total del comprobante, o especificar un monto parcial de dicho total.

Definido Internamente como:

```
ImprimirPago(
    [in] BSTR Descripcion,
    [in] double Monto,
    [in] ModosDePago Operacion,
    [in, optional, defaultvalue(48)] ModosDeDisplay ModoDisplay,
    [in, optional, defaultvalue("")] BSTR DescripcionAdicional,
    [in, optional, defaultvalue(8)] TiposPagoCodigoFormaPago,
    [in, optional, defaultvalue(0)] long Cuotas,
    [in, optional, defaultvalue("")] BSTR Cupones,
    [in, optional, defaultvalue("")] BSTR Referencia,
    [out, retval] RespuestaImprimirPago* );
```

<i>Descripcion</i>	Permite especificar el texto a imprimir como identificación del medio de pago, o anulación del mismo.
--------------------	---

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

<i>Monto</i>	Permite especificar el monto que se está pagando/anulando.																																																				
<i>Operacion</i>	Permite especificar si la operación es un pago, o la anulación de un pago previo y realizado en el comprobante en curso. Debe ser alguna de las constantes pertenecientes al enum <i>ModosDePago</i> . <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td><i>Anular</i></td> <td>82</td> </tr> <tr> <td><i>Pagar</i></td> <td>84</td> </tr> </table>	<i>Anular</i>	82	<i>Pagar</i>	84																																																
<i>Anular</i>	82																																																				
<i>Pagar</i>	84																																																				
<i>ModoDisplay</i>	Permite especificar el modo de uso del visor de cliente conectado a la impresora fiscal 2G. Debe ser alguna de las constantes pertenecientes al enum <i>ModosDeDisplay</i> . <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td><i>DisplayNo</i></td> <td>48</td> </tr> <tr> <td><i>DisplaySi</i></td> <td>49</td> </tr> <tr> <td><i>DisplayRep</i></td> <td>50</td> </tr> </table> <p>En impresoras fiscales 2G que no soportan conexión de visor de cliente, especificar <i>DisplayNo</i>.</p>	<i>DisplayNo</i>	48	<i>DisplaySi</i>	49	<i>DisplayRep</i>	50																																														
<i>DisplayNo</i>	48																																																				
<i>DisplaySi</i>	49																																																				
<i>DisplayRep</i>	50																																																				
<i>DescripcionAdicional</i>	Permite especificar un texto que se imprimirá como extensión de la descripción que identifica al medio de pago.																																																				
<i>CodigoFormaPago</i>	Permite especificar el tipo de pago. Debe ser alguna de las constantes pertenecientes al enum <i>TiposPago</i> . <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td><i>Cambio</i></td> <td>0</td> <td><i>LetraDeCambio</i></td> <td>13</td> </tr> <tr> <td><i>CartaDeCreditoDocumentario</i></td> <td>1</td> <td><i>MedioDePagoDeComercioExterior</i></td> <td>14</td> </tr> <tr> <td><i>CartaDeCreditoSimple</i></td> <td>2</td> <td><i>OrdenDePagoDocumentaria</i></td> <td>15</td> </tr> <tr> <td><i>Cheque</i></td> <td>3</td> <td><i>OrdenDePagoSimple</i></td> <td>16</td> </tr> <tr> <td><i>ChequeCancelatorios</i></td> <td>4</td> <td><i>OtrosMediosPago</i></td> <td>99</td> </tr> <tr> <td><i>CreditoDocumentario</i></td> <td>5</td> <td><i>PagoContraReembolso</i></td> <td>17</td> </tr> <tr> <td><i>CuentaCorriente</i></td> <td>6</td> <td><i>RemesaDocumentaria</i></td> <td>18</td> </tr> <tr> <td><i>Deposito</i></td> <td>7</td> <td><i>RemesaSimple</i></td> <td>19</td> </tr> <tr> <td><i>Efectivo</i></td> <td>8</td> <td><i>TarjetaDeCredito</i></td> <td>20</td> </tr> <tr> <td><i>EndosoDeCheque</i></td> <td>9</td> <td><i>TarjetaDeDebito</i></td> <td>21</td> </tr> <tr> <td><i>FacturaDeCredito</i></td> <td>10</td> <td><i>Ticket</i></td> <td>22</td> </tr> <tr> <td><i>GarantiaBancaria</i></td> <td>11</td> <td><i>TransferenciaBancaria</i></td> <td>23</td> </tr> <tr> <td><i>Giro</i></td> <td>12</td> <td><i>TransferenciaNoBancaria</i></td> <td>24</td> </tr> </table>	<i>Cambio</i>	0	<i>LetraDeCambio</i>	13	<i>CartaDeCreditoDocumentario</i>	1	<i>MedioDePagoDeComercioExterior</i>	14	<i>CartaDeCreditoSimple</i>	2	<i>OrdenDePagoDocumentaria</i>	15	<i>Cheque</i>	3	<i>OrdenDePagoSimple</i>	16	<i>ChequeCancelatorios</i>	4	<i>OtrosMediosPago</i>	99	<i>CreditoDocumentario</i>	5	<i>PagoContraReembolso</i>	17	<i>CuentaCorriente</i>	6	<i>RemesaDocumentaria</i>	18	<i>Deposito</i>	7	<i>RemesaSimple</i>	19	<i>Efectivo</i>	8	<i>TarjetaDeCredito</i>	20	<i>EndosoDeCheque</i>	9	<i>TarjetaDeDebito</i>	21	<i>FacturaDeCredito</i>	10	<i>Ticket</i>	22	<i>GarantiaBancaria</i>	11	<i>TransferenciaBancaria</i>	23	<i>Giro</i>	12	<i>TransferenciaNoBancaria</i>	24
<i>Cambio</i>	0	<i>LetraDeCambio</i>	13																																																		
<i>CartaDeCreditoDocumentario</i>	1	<i>MedioDePagoDeComercioExterior</i>	14																																																		
<i>CartaDeCreditoSimple</i>	2	<i>OrdenDePagoDocumentaria</i>	15																																																		
<i>Cheque</i>	3	<i>OrdenDePagoSimple</i>	16																																																		
<i>ChequeCancelatorios</i>	4	<i>OtrosMediosPago</i>	99																																																		
<i>CreditoDocumentario</i>	5	<i>PagoContraReembolso</i>	17																																																		
<i>CuentaCorriente</i>	6	<i>RemesaDocumentaria</i>	18																																																		
<i>Deposito</i>	7	<i>RemesaSimple</i>	19																																																		
<i>Efectivo</i>	8	<i>TarjetaDeCredito</i>	20																																																		
<i>EndosoDeCheque</i>	9	<i>TarjetaDeDebito</i>	21																																																		
<i>FacturaDeCredito</i>	10	<i>Ticket</i>	22																																																		
<i>GarantiaBancaria</i>	11	<i>TransferenciaBancaria</i>	23																																																		
<i>Giro</i>	12	<i>TransferenciaNoBancaria</i>	24																																																		
<i>Cuotas</i>	Permite especificar la cantidad de cuotas aplicadas al medio de pago/anulación que se indica en este comando fiscal.																																																				

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

<i>Cupones</i>	Permite especificar la numeración de los cupones afectados al pago/anulación que se indica en este comando fiscal.
<i>Referencia</i>	Permite especificar un número que será almacenado en la CTD (cinta testigo digital, o memoria de auditoría). Puede usarse para guardar un número de autorización de tarjeta, o una referencia interna que puede recuperarse en otro momento. Este dato no se imprime en el comprobante.

El comando fiscal generado por *'ImprimirPago()'* se puede enviar hasta un máximo de cinco veces por comprobante, independientemente de la modalidad de la operación (pago o anulación de pago) especificada en el comando fiscal.

Si el quinto comando fiscal generado por *'ImprimirPago()'* no salda el importe total pendiente a saldar, igualmente se aceptará, pero no será admitido un sexto medio de pago. Al cerrar el comprobante se imprimirá automáticamente la leyenda 'Saldo' asociada al monto faltante para completar el pago exacto.

Si el quinto comando fiscal generado por *'ImprimirPago()'* excede el importe total pendiente a saldar, igualmente se aceptará, y no se admitirán más medios de pago. Al cerrar el comprobante se imprimirá automáticamente la leyenda 'CAMBIO \$ nnnnnn.nn' para mostrar que debe entregarse un vuelto.

La impresión que provoca el comando fiscal generado por *'ImprimirPago()'* es diferida por la impresora fiscal 2G hasta en que recibe el comando fiscal generado por *'CerrarDocumento()'*.

La impresión de la leyenda 'RECIBI(MOS)' (o 'ENTREGAMOS:', en el caso de los *Tiques Nota de Crédito*) no es modificable, ni puede evitarse su impresión.

Ejemplo:

```
Dim resppago As HasarArgentina.RespuestaImprimirPago

resppago = hasar.ImprimirPago("Medio de Pago Uno ...", 124.07, _
    HasarArgentina.ModosDePago.Pagar, _
    HasarArgentina.ModosDeDisplay.DisplayNo, "", HasarArgentina.TiposPago.Cheque, 0, "")
Debug.Print("Saldo =[ " & resppago.Saldo & "]" )
```

La respuesta fiscal al comando fiscal generado por *'ImprimirPago()'* es la estructura *RespuestaImprimirPago*, definida internamente como:

```
typedef
struct tagRespuestaImprimirPago {
    [helpstring("Saldo")]
}
    double Saldo;
} RespuestaImprimirPago;
```

<i>Saldo</i>	Informa un importe que se interpreta de la siguiente manera:	
<i>Positivo</i>	Prefijo '+', o ausencia de signo. Importe que debe ser saldado antes de cerrar el comprobante. De no hacerlo, la impresora fiscal 2G asignará el importe a un pago por defecto en el momento del cierre.	
<i>0.00</i>	Se ha saldado el importe total del comprobante. No se requieren más medios de pago (aunque éstos resulten aceptados por la impresora fiscal 2G).	
<i>Negativo</i>	Prefijo '-'. El pago realizado excede el monto total del comprobante, o excede el importe pendiente de saldo. El valor reportado debe interpretarse como vuelto, o cambio.	

ImprimirTextoFiscal()

Permite enviar a la impresora fiscal 2G el comando fiscal de impresión de texto fiscal (extensión de la descripción del ítem). Este comando se debe enviar en forma previa al comando fiscal generado por '*ImprimirItem()*', y hasta un máximo 4 veces.

Definido internamente como:

```
ImprimirTextoFiscal(
    [in, out] AtributosDeTexto* Atributos,
    [in] BSTR Texto,
    [in, optional, defaultvalue(48)] ModosDeDisplay ModoDisplay);
```

Atributos Permite especificar el estilo de impresión del texto (centrado, negrita, doble ancho, borrado) mediante una estructura booleana, que internamente se encuentra definida como tipo *AtributosDeTexto*.

```
typedef
struct tagAtributosDeTexto {
    [helpstring("BorradoTexto")]
}
    VARIANT_BOOL BorradoTexto;
    [helpstring("DobleAncho")]
}
    VARIANT_BOOL DobleAncho;
    [helpstring("Centrado")]
}
    VARIANT_BOOL Centrado;
    [helpstring("Negrita")]
}
    VARIANT_BOOL Negrita;
} AtributosDeTexto;
```

Texto Permite especificar el texto que se imprimirá como una extensión de la descripción del próximo ítem a imprimir.

ModoDisplay Permite especificar el modo de uso del visor de cliente conectado a la impresora fiscal 2G. Debe ser alguna de las constantes pertenecientes al enum *ModosDeDisplay*.

<i>DisplayNo</i>	48
<i>DisplaySi</i>	49
<i>DisplayRep</i>	50

En impresoras fiscales 2G que no soportan conexión de visor de cliente, especificar *DisplayNo*.

Las líneas de texto fiscal no se imprimen hasta que la impresora fiscal 2G recibe, y acepta, el comando fiscal generado por '*ImprimirItem()*' (impresión de ítem).

Las líneas de texto fiscal son almacenadas temporalmente en la memoria de trabajo de la impresora fiscal 2G. Si el comando de impresión de ítem fuese rechazado, las líneas de texto fiscal son eliminadas de la memoria de trabajo de la impresora fiscal 2G.

Las líneas de texto fiscal no se imprimen en los siguientes comprobantes: *Recibo 'X'*, y *Documento Donación*.

Ejemplo:

```
Dim estilo As HasarArgentina.AtributosDeTexto

estilo.Centrado = False
estilo.DobleAncho = False
estilo.BorradoTexto = False
estilo.Negrita = False

hasar.ImprimirTextoFiscal(estilo, "Hasta agotar stock ...", _
    HasarArgentina.ModosDeDisplay.DisplayNo)
```

ImprimirTextoGenerico()

Permite enviar a la impresora fiscal 2G el comando fiscal de impresión de texto, aceptado únicamente en el cuerpo un *Documento Genérico*, abierto.

Definido internamente como:

```
ImprimirTextoGenerico(
    [in, out] AtributosDeTexto* Atributos,
    [in] BSTR Texto,
    [in, optional, defaultvalue(48)] ModosDeDisplay ModoDisplay);
```

Atributos Permite especificar el estilo de impresión del texto (centrado, negrita, doble ancho, borrado) mediante una estructura booleana, que internamente se encuentra definida como tipo *Atribu-*

tosDeTexto.

```
typedef
struct tagAtributosDeTexto {
    [helpstring("BorradoTexto")]
}
    VARIANT_BOOL BorradoTexto;
    [helpstring("DobleAncho")]
}
    VARIANT_BOOL DobleAncho;
    [helpstring("Centrado")]
}
    VARIANT_BOOL Centrado;
    [helpstring("Negrita")]
}
    VARIANT_BOOL Negrita;
} AtributosDeTexto;
```

Texto Permite especificar el texto que se imprimirá en el cuerpo del *Documento Genérico*.

ModoDisplay Permite especificar el modo de uso del visor de cliente conectado a la impresora fiscal 2G. Debe ser alguna de las constantes pertenecientes al enum *ModosDeDisplay*.

<i>DisplayNo</i>	48
<i>DisplaySi</i>	49
<i>DisplayRep</i>	50

En impresoras fiscales 2G que no soportan conexión de visor de cliente, especificar *DisplayNo*.

Ejemplo:

```
Dim estilo As HasarArgentina.AtributosDeTexto

estilo.Centrado = False
estilo.DobleAncho = False
estilo.BorradoTexto = False
estilo.Negrita = False

hasar.ImprimirTextoGenerico(estilo, "Hasta agotar stock ...", _
    HasarArgentina.ModosDeDisplay.DisplayNo)
```

ObtenerCampoRespuesta()

Habiendo enviado cualquier comando fiscal, a la impresora fiscal 2G, y cuando a respuesta fiscal al mismo contenga campos de información adicionales al de estado de impresora y estado fiscal, el método '*ObtenerCampoRespuesta()*' permite extraer la información de cada uno de esos campos adicionales.

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

Definido internamente como:

```
ObtenerCampoRespuesta(  
    [in] BSTR NombreCampo,  
    [out, retval] BSTR* );
```

<i>NombreCampo</i>	<p>Permite especificar el nombre del campo, cuya información se desea extraer, y perteneciente a la respuesta fiscal del comando fiscal enviado inmediatamente antes de la primera obtención de información, sobre dicha respuesta fiscal.</p> <p>El string que se indique debe coincidir exactamente con el nombre indicado en los campos de salida, informados en el manual de comandos de la impresora fiscal 2G, y que forman parte de la respuesta fiscal al comando fiscal enviado.</p>
--------------------	---

Si bien *'ObtenerCampoRespuesta()'* se encuentra disponible para aquellos lenguajes que no tengan compatibilidad con los tipos de datos ofrecidos por el OCX 2G (como VFP, que no soporta estructuras), también puede ser utilizado desde cualquier otros lenguaje de desarrollo de software de gestión para puntos de venta.

'ObtenerCampoRespuesta()' debe invocarse tantas veces como campos contenga la respuesta fiscal, o resulten de interés para el sistema de facturación.

Si el campo consultado no existiese el método *'ObtenerCampoRespuesta()'* retorna un string vacío.

Ejemplo:

```
Dim subt As HasarArgentina.RespuestaConsultarSubtotal  
  
subt = hasar.ConsultarSubtotal(HasarArgentina.ImpresionSubtotal.NoImprimeSubtotal, _  
    HasarArgentina.ModosDeDisplay.DisplayNo)  
Debug.Print("Total Comprob. =[" & hasar.ObtenerCampoRespuesta("Subtotal") & "]")
```

La respuesta fiscal al comando fiscal generado por *'ConsultarSubtotal()'* es la estructura *RespuestaConsultarSubtotal*, definida internamente como:

```
typedef  
struct tagRespuestaConsultarSubtotal {  
    [helpstring("CantidadItems")  
]  
    double CantidadItems;  
    [helpstring("Subtotal")  
]  
    double Subtotal;  
    [helpstring("MontoIVA")  
]  
    double MontoIVA;  
    [helpstring("MontoPagado")  
]  
    double MontoPagado;  
    [helpstring("MontoOtrosTributos")  
]
```

Grupo HASAR – Compañía HASAR SAIC – OCX 2G
Manual del usuario - Rev 004

```

double MontoOtrosTributos;
[helpstring("MontoImpInternos")
]
double MontoImpInternos;
[helpstring("MontoBase")
]
double MontoBase;
[helpstring("AjusteRedondeo")
]
double AjusteRedondeo;
} RespuestaConsultarSubtotal;

```

<i>CantidadItems</i>	Informa la suma de la cantidad de unidades indicada en cada ítem registrado hasta el momento de la consulta.
<i>Subtotal</i>	Informa la suma del resto de los montos parciales reportados en la respuesta fiscal del comando fiscal generado por ' <i>ConsultarSubtotal()</i> '; excepto el subtotal pagado (no incluido en esta suma). Este monto coincide con el importe final del comprobante si sucediese que a continuación de esta consulta se enviara a la impresora fiscal 2G el comando fiscal de cierre del comprobante.
<i>MontoIVA</i>	Informa el monto de IVA acumulado hasta el momento de la consulta, y considerando a todas las alícuotas intervinientes en la conformación del comprobante.
<i>MontoPagado</i>	Informa el monto acumulado de pagos realizados hasta el momento de la consulta.
<i>MontoOtrosTributos</i>	Informa el monto acumulado de tributos registrados hasta el momento de la consulta. No incluye la suma de impuestos internos por ítem.
<i>MontoImpInternos</i>	Informa el monto acumulado de impuestos internos por ítem, hasta el momento de la consulta.
<i>MontoBase</i>	Informa el monto de bases imponibles acumulado hasta el momento de la consulta, considerando e incluyendo las correspondientes a todas las alícuotas de IVA intervinientes en la conformación del comprobante.
<i>AjusteRedondeo</i>	Informa las potenciales diferencias entre los montos impresos y los montos internos calculados con mayor precisión, que son reportados por este comando. Este campo reporta las diferencias entre los montos reales y los montos impresos, limitados y redondeados a dos decimales a fines de la impresión. En el caso de los comprobantes "A", esta diferencia surge entre la suma real de las bases imponibles y la suma de los montos impresos. En el caso de comprobantes "B/C", esta diferencia se da en el monto final. En ambos casos, lo impreso puede diferir de lo reportado por este comando, por lo cual se informa el ajuste que será eventualmente impreso al finalizar el comprobante; independientemente de lo que se vea impreso en un momento dado en el comprobante, los montos reportados son los que se utilizarán para cerrar el comprobante y acumular. No disponible en impresoras fiscales 2G con protocolo '0' (SMH/PT-1000F v01.00 y

|SMH/PT-250F v01.00).

ObtenerPrimerBloqueAuditoria()

Permite enviar a la impresora fiscal 2G el comando fiscal de solicitud de primer registro de información almacenada en la CTD (cinta testigo digital, o memoria de auditoría). La consulta puede ser por rango de fechas de *Informes Diarios de Cierre* (cierre 'Z'), o por rango de numeración de los mismos.

Definido internamente como:

```
ObtenerPrimerBloqueAuditoria (
    [in] BSTR ZetaFechaInicial,
    [in] BSTR ZetaFechaFinal,
    [in] TipoReporteZ TipoReporte,
    [in] CompresionDeInformacion Zipea,
    [in] XMLsPorBajada XMLUnico,
    [out, retval] RespuestaObtenerPrimerBloqueAuditoria* );
```

<i>ZetaFechaInicial</i>	Permite especificar el valor inicial del rango de descarga de información. El valor a especificar puede ser una fecha o un número de <i>Informe Diario de Cierre</i> (cierre 'Z').				
<i>ZetaFechaFinal</i>	Permite especificar el valor final del rango de descarga de información. El valor a especificar puede ser una fecha o un número de <i>Informe Diario de Cierre</i> (cierre 'Z').				
<i>TipoReporte</i>	Permite especificar el tipo de información ingresada en los campos correspondientes al inicio y final del rango de búsqueda. Debe ser alguna de las constantes pertenecientes al enum <i>TipoReporteZ</i> . <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td><i>ReporteZFecha</i></td> <td>70</td> </tr> <tr> <td><i>ReporteZNumero</i></td> <td>90</td> </tr> </table>	<i>ReporteZFecha</i>	70	<i>ReporteZNumero</i>	90
<i>ReporteZFecha</i>	70				
<i>ReporteZNumero</i>	90				
<i>Zipea</i>	Permite especificar si la información a descargar se comprimirá sobre un archivo en formato ZIP. De no comprimirse, la descarga consistirá en un archivo en formato XML nativo. Debe ser alguna de las constantes pertenecientes al enum <i>CompresionDeInformacion</i> . <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td><i>NoComprime</i></td> <td>78</td> </tr> <tr> <td><i>Comprime</i></td> <td>80</td> </tr> </table>	<i>NoComprime</i>	78	<i>Comprime</i>	80
<i>NoComprime</i>	78				
<i>Comprime</i>	80				
<i>XMLUnico</i>	Permite especificar si se generará un único archivo en formato XML nativo para todas las jornadas fiscales/comerciales descargadas, o si deberá generarse un archivo en formato XML nativo por cada jornada fiscal. Debe ser alguna de las constantes pertenecientes al enum <i>XMLsPorBajada</i> . <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td><i>VariosXLMs</i></td> <td>78</td> </tr> <tr> <td><i>XMLUnico</i></td> <td>80</td> </tr> </table>	<i>VariosXLMs</i>	78	<i>XMLUnico</i>	80
<i>VariosXLMs</i>	78				
<i>XMLUnico</i>	80				

Ejemplo:

```
Dim bloque1 As HasarArgentina.RespuestaObtenerPrimerBloqueAuditoria

bloque1 = hasar.ObtenerPrimerBloqueAuditoria("1", "10", _
                                             HasarArgentina.TipoReporteZ.ReporteZNumero, _
                                             HasarArgentina.CompresionDeInformacion.NoComprime, _
                                             HasarArgentina.XMLsPorBajada.XMLUnico)
Debug.Print("Primer bloque =[ " & bloque1.Informacion & " ]")
```

La respuesta fiscal al comando fiscal generado por 'ObtenerPrimerBloqueAuditoria()' es la estructura *RespuestaObtenerPrimerBloqueAuditoria*, definida internamente como:

```
typedef
struct tagRespuestaObtenerPrimerBloqueAuditoria {
    [helpstring("Registro")]
}
    IdentificadorBloque Registro;
    [helpstring("Informacion")]
}
    BSTR Informacion;
} RespuestaObtenerPrimerBloqueAuditoria;
```

<i>Registro</i>	<p>Informa si se trata de la descarga del último bloque de información ('<i>BloqueFinal</i>'), o aún hay más ('<i>BloqueInformacion</i>').</p> <p>Es una de las constantes pertenecientes al enum <i>IdentificadorBloque</i>.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td><i>BloqueFinal</i></td> <td align="center">0</td> </tr> <tr> <td><i>BloqueInformacion</i></td> <td align="center">1</td> </tr> </table>	<i>BloqueFinal</i>	0	<i>BloqueInformacion</i>	1
<i>BloqueFinal</i>	0				
<i>BloqueInformacion</i>	1				
<i>Informacion</i>	<p>Informa el primer bloque de información de la CTD (cinta testigo digital, o memoria de auditoría) en el formato indicado por el campo correspondiente a compresión (o no) de información, y dependiendo del tamaño del mismo. De corresponder, debe enviarse posteriormente el comando fiscal generado por '<i>ObtenerSiguieteBloqueAuditoría()</i>', tantas veces como sea necesario hasta descargar la información solicitada en forma completa.</p> <p>Cada bloque que se descarga debe grabarse en un archivo de texto cuyo contenido es ASCII85 (se indicó comprimir) o XML (se indicó no comprimir).</p>				

ObtenerPrimerBloqueDocumento()

Permite enviar a la impresora fiscal 2G el comando fiscal de solicitud de primer registro de información almacenada en la CTD (cinta testigo digital, o memoria de auditoría), correspondiente a determinado tipo de comprobante. La consulta sólo puede ser por rango de números de comprobantes.

Definido internamente como:

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

```

ObtenerPrimerBloqueDocumento(
    [in] long NumeroInicial,
    [in] long NumeroFinal,
    [in] TiposComprobante CodigoComprobante,
    [in] CompresionDeInformacion Zipea,
    [in] XMLsPorBajada XMLUnico,
    [out, retval] RespuestaObtenerPrimerBloqueDocumento* );
    
```

NumeroInicial Permite especificar el número inicial de comprobante para el rango de descarga de información.

NumeroFinal Permite especificar el número final de comprobante para el rango de descarga de información.

CodigoComprobante Permite especificar el tipo de comprobante al que corresponde el rango de descarga de información. Debe ser alguna de las constantes pertenecientes al enum *TiposComprobante*.

*** ComprobanteDonacion	907	*** ReciboM	54
* FacturaA	1	*** ReciboX	902
* FacturaB	6	*** RemitoR	91
* FacturaC	11	*** RemitoX	901
* FacturaM	51	** Tique	83
*** Generico	910	** TiqueFacturaA	81
* NotaDeCreditoA	3	** TiqueFacturaB	82
* NotaDeCreditoB	8	** TiqueFacturaC	111
* NotaDeCreditoC	13	** TiqueFacturaM	118
* NotaDeCreditoM	53	** TiqueNotaCredito	110
* NotaDeDebitoA	2	** TiqueNotaCreditoA	112
* NotaDeDebitoB	7	** TiqueNotaCreditoB	113
* NotaDeDebitoC	12	** TiqueNotaCrediotC	114
* NotaDeDebitoM	52	** TiqueNotaCreditoM	119
*** PresupuestoX	903	** TiqueNotaDebitoA	115
*** ReciboA	4	** TiqueNotaDebitoB	116
*** ReciboB	9	** TiqueNotaDebitoC	117
*** ReciboC	15	** TiqueNotaDebitoM	120

* Disponibles en impresoras fiscales 2G que soporten impresión sobre formularios continuos u hojas sueltas.

** Disponibles en impresoras fiscales 2G que soporten impresión sobre rollos de papel continuo.

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

*** Disponibles en todos los modelos de impresoras fiscales 2G.

Zipea

Permite especificar si la información a descargar se comprimirá sobre un archivo en formato ZIP. De no comprimirse, la descarga consistirá en un archivo en formato XML nativo. Debe ser alguna de las constantes pertenecientes al enum *CompresionDeInformacion*.

<i>NoComprime</i>	78
<i>Comprime</i>	80

XMLUnico

Permite especificar si se generará un único archivo en formato XML nativo para todos los comprobantes descargados, o si deberá generarse un archivo en formato XML nativo por cada comprobante. Debe ser alguna de las constantes pertenecientes al enum *XMLsPorBajada*.

<i>VariosXMLs</i>	78
<i>XMLUnico</i>	80

Ejemplo:

```
Dim bloque1 As HasarArgentina.RespuestaObtenerPrimerBloqueDocumento
bloque1 = hasar.ObtenerPrimerBloqueDocumento(1, 5, _
    HasarArgentina.TiposComprobante.TiqueFacturaA, _
    HasarArgentina.CompresionDeInformacion.NoComprime, _
    HasarArgentina.XMLsPorBajada.XMLUnico)
Debug.Print("Primer bloque =[ " & bloque1.Informacion & "]" )
```

La respuesta fiscal al comando fiscal generado por '*ObtenerPrimerBloqueDocumento()*' es la estructura *RespuestaObtenerPrimerBloqueDocumento*, definida internamente como:

```
typedef
struct tagRespuestaObtenerPrimerBloqueDocumento {
    [helpstring("Registro")]
}
    IdentificadorBloque Registro;
    [helpstring("Informacion")]
}
    BSTR Informacion;
} RespuestaObtenerPrimerBloqueDocumento;
```

Registro

Informa si se trata de la descarga del último bloque de información ('*BloqueFinal*'), o aún hay más ('*BloqueInformacion*').

Es una de las constantes pertenecientes al enum *IdentificadorBloque*.

		<i>BloqueFinal</i>	0
		<i>BloqueInformacion</i>	1
<i>Informacion</i>	Informa el primer bloque de información de comprobantes en el formato especificado, y dependiendo del tamaño del mismo. Debe enviarse posteriormente el comando fiscal generado por ' <i>ObtenerSiguienteBloqueDocumento()</i> ', tantas veces como sea necesario hasta descargar la información en forma completa.		

ObtenerPrimerBloqueLog()

Permite enviar a la impresora fiscal 2G el comando fiscal de solicitud del primer registro de información interna, generado por las actividades propias del equipo. No se trata de información de interés administrativo, contable o impositivo.

Definido internamente como:

```
ObtenerPrimerBloqueLog([out, retval] RespuestaObtenerPrimerBloqueLog* );
```

Ejemplo:

```
Dim bloque1 As HasarArgentina.RespuestaObtenerPrimerBloqueLog
bloque1 = hasar.ObtenerPrimerBloqueLog()
Debug.Print("Primer bloque =[ " & bloque1.Informacion & " ]")
```

La respuesta fiscal al comando fiscal generado por '*ObtenerPrimerBloqueLog()*' es la estructura *RespuestaObtenerPrimerBloqueLog*, definida internamente como:

```
typedef
struct tagRespuestaObtenerPrimerBloqueLog {
    [helpstring("Registro")]
}
    IdentificadorBloque Registro;
    [helpstring("Informacion")]
}
    BSTR Informacion;
} RespuestaObtenerPrimerBloqueLog;
```

<i>Registro</i>	Informa si se trata de la descarga del último bloque de información (' <i>BloqueFinal</i> '), o aún hay más (' <i>BloqueInformacion</i> ').		
	Es una de las constantes pertenecientes al enum <i>IdentificadorBloque</i> .		
		<i>BloqueFinal</i>	0

	<i>BloqueInformacion</i>	1
<i>Informacion</i>	Informa el primer bloque de información del archivo interno del registro de actividades de la impresora fiscal 2G. En el caso de tratarse del registro final, este campo puede estar vacío.	

ObtenerPrimerBloqueReporteElectronico()

Permite enviar a la impresora fiscal 2G el comando fiscal de solicitud de descarga del primer registro de información correspondiente a los reportes electrónicos para la AFIP que se exigen a los emisores de comprobantes mediante el uso de impresoras fiscales 2G: CTD (cinta testigo digital, o memoria digital), Reporte de Totales (totales por jornada fiscal/comercial), y el Reporte de Duplicados 'A' (incluyendo 'A con leyenda', y 'M', si los hubiere). El primero de los mencionados (CTD) debe conservarlo el contribuyente (dos copias), los otros dos se presentan a la AFIP como DDJJ (declaraciones juradas).

La información descargada se entrega siempre en formato ASCII85.

Definido internamente como:

```
ObtenerPrimerBloqueReporteElectronico(
    [in] DATE FechaInicial,
    [in] DATE FechaFinal,
    [in, optional, defaultvalue(80)] TiposReporteAFIP TipoReporte,
    [out, retval] RespuestaObtenerPrimerBloqueReporteElectronico* );
```

<i>FechaInicial</i>	Permite especificar la fecha de inicio del rango de descarga de información a reportar.				
<i>FechaFinal</i>	Permite especificar la fecha final del rango de descarga de información a reportar.				
<i>TipoReporte</i>	Permite especificar si la descarga es completa (incluye la CTD, los duplicados 'A' y los totales por jornada fiscal/comercial), o sólo la memoria fiscal (totales por jornada fiscal/comercial). Debe ser alguna de las constantes pertenecientes al enum <i>TiposReporteAFIP</i> . <table border="1" data-bbox="606 1456 1053 1568" style="margin-left: auto; margin-right: auto;"> <tr> <td><i>ReporteAFIPMemoriaFiscal</i></td> <td align="center">78</td> </tr> <tr> <td><i>ReporteAFIPCompleto</i></td> <td align="center">80</td> </tr> </table>	<i>ReporteAFIPMemoriaFiscal</i>	78	<i>ReporteAFIPCompleto</i>	80
<i>ReporteAFIPMemoriaFiscal</i>	78				
<i>ReporteAFIPCompleto</i>	80				

Ejemplo:

```
Dim bloque1 As HasarArgentina.RespuestaObtenerPrimerBloqueReporteElectronico
bloque1 = hasar.ObtenerPrimerBloqueReporteElectronico("08/08/2018", _
    "08/08/2018", HasarArgentina.TiposReporteAFIP.ReporteAFIPCompleto)
Debug.Print("Primer bloque =[ " & bloque1.Informacion & " ]")
```

La respuesta fiscal al comando fiscal generado por 'ObtenerPrimerBloqueReporteElectronico()' es la estructura

Grupo HASAR – Compañía HASAR SAIC – OCX 2G

Manual del usuario - Rev 004

RespuestaObtenerPrimerBloqueReporteElectronico, definida internamente como:

```
typedef [uuid(B5483E03-C9F5-4A02-A301-2C33E4638ACB), versio
struct tagRespuestaObtenerPrimerBloqueReporteElectronico {
    [helpstring("Registro")
]
    IdentificadorBloque Registro;
    [helpstring("Informacion")
]
    BSTR Informacion;
} RespuestaObtenerPrimerBloqueReporteElectronico;
```

<i>Registro</i>	<p>Informa si se trata de la descarga del último bloque de información ('<i>BloqueFinal</i>'), o aún hay más ('<i>BloqueInformacion</i>').</p> <p>Es una de las constantes pertenecientes al enum <i>IdentificadorBloque</i>.</p> <table style="margin-left: auto; margin-right: auto;"> <tr> <td style="border-right: 1px solid black;"><i>BloqueFinal</i></td> <td style="text-align: center;">0</td> </tr> <tr> <td style="border-right: 1px solid black;"><i>BloqueInformacion</i></td> <td style="text-align: center;">1</td> </tr> </table>	<i>BloqueFinal</i>	0	<i>BloqueInformacion</i>	1
<i>BloqueFinal</i>	0				
<i>BloqueInformacion</i>	1				
<i>Informacion</i>	<p>Informa el primer bloque de información en formato ASCII85, correspondiente a los reportes a descargar. En el caso de tratarse del registro final de información, el contenido del campo de información podría estar vacío. El contenido del bloque debe grabarse en un archivo de texto.</p>				

ObtenerRangoFechasPorZetas()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta sobre el rango de fechas asociados a un determinado rango de números de *Informes Diarios de Cierre* (cierres 'Z').

Definido internamente como:

```
ObtenerRangoFechasPorZetas(
    [in] long ZetaInicial,
    [in] long ZetaFinal,
    [out, retval] RespuestaObtenerRangoFechasPorZetas* );
```

<i>ZetaInicial</i>	<p>Permite especificar el número del primer <i>Informe Diario de Cierre</i> (cierre 'Z') del rango del cual se quiere conocer su fecha inicial. Debe ser menor o igual a <i>ZetaFinal</i>.</p>
<i>ZetaFinal</i>	<p>Permite especificar el número del último <i>Informe Diario de Cierre</i> (cierre 'Z') del rango del cual se quiere conocer su fecha final. Debe ser mayor o igual a <i>ZetaInicial</i>.</p>

Ejemplo:

```
Dim resp1 As HasarArgentina.RespuestaObtenerRangoFechasPorZetas
```

```
resp1 = hasar.ObtenerRangoFechasPorZetas(1, 3650)  
Debug.Print("Fecha Inicial =[" & resp1.FechaInicial & "]")  
Debug.Print("Fecha Final   =[" & resp1.FechaFinal & "]")
```

La respuesta fiscal al comando fiscal generado por 'ObtenerRangoFechasPorZetas()' es la estructura *RespuestaObtenerRangoFechasPorZetas*, definida internamente como:

```
typedef  
struct tagRespuestaObtenerRangoFechasPorZetas {  
    [helpstring("FechaInicial")  
    ]  
    DATE FechaInicial;  
    [helpstring("FechaFinal")  
    ]  
    DATE FechaFinal;  
} RespuestaObtenerRangoFechasPorZetas;
```

<i>FechaInicial</i>	Informa la fecha inicial del rango de <i>Informes Diarios de Cierre</i> (cierres 'Z') solicitado.
<i>FechaFinal</i>	Informa la fecha final del rango de <i>Informes Diarios de Cierre</i> (cierres 'Z') solicitado.

ObtenerRangoZetasPorFechas()

Permite enviar a la impresora fiscal 2G el comando fiscal de consulta sobre el rango de números de *Informes Diarios de Cierre* (cierres 'Z') correspondientes a un determinado rango de fechas.

Definido internamente como:

```
ObtenerRangoZetasPorFechas (  
    [in] DATE FechaInicial,  
    [in] DATE FechaFinal,  
    [out, retval] RespuestaObtenerRangoZetasPorFechas* );
```

<i>FechaInicial</i>	Permite especificar la fecha del primer <i>Informe Diario de Cierre</i> (cierre 'Z') del rango del cual se quiere conocer su número inicial. Debe ser anterior o igual a <i>FechaFinal</i> .
<i>FechaFinal</i>	Permite especificar la fecha del último <i>Informe Diario de Cierre</i> (cierre 'Z') del rango del cual se quiere conocer su número final. Debe ser posterior o igual a <i>FechaInicial</i> .

Ejemplo:

```
Dim resp2 As HasarArgentina.RespuestaObtenerRangoZetasPorFechas
```

```
resp2 = hasar.ObtenerRangoZetasPorFechas("30/04/2019", "04/05/2019")  
Debug.Print("Zeta Inicial =[ " & resp2.ZetaInicial & " ]")  
Debug.Print("Zeta Final   =[ " & resp2.ZetaFinal & " ]")
```

La respuesta fiscal al comando fiscal generado por 'ObtenerRangoZetasPorfechas()' es la estructura *RespuestaObtenerRangoZetasPorfechas*, definida internamente como:

```
typedef  
struct tagRespuestaObtenerRangoZetasPorFechas {  
    [helpstring("ZetaInicial")  
    ]  
    long ZetaInicial;  
    [helpstring("ZetaFinal")  
    ]  
    long ZetaFinal;  
} RespuestaObtenerRangoZetasPorFechas;
```

<i>ZetaInicial</i>	Informa el número del primer <i>Informe Diario de Cierre</i> (cierre 'Z') que contiene el rango de fechas solicitado.
<i>ZetaFinal</i>	Informa el número del último <i>Informe Diario de Cierre</i> (cierre 'Z') que contiene el rango de fechas solicitado.

ObtenerRevision()

Permite obtener el número que identifica a la revisión de software del OCX 2G. No se genera, ni se envía comando fiscal alguno a la impresora fiscal 2G. El dato obtenido permite determinar la compatibilidad de manejo de la impresora fiscal 2G con la que va a dialogar.

Definido internamente como:

```
ObtenerRevision([out, retval] short* );
```

Ejemplo:

```
Debug.Print("OCX 2G Revisión = [ " & hasar.ObtenerRevision & " ]")
```

ObtenerSiguienteBloqueAuditoria()

Permite enviar a la impresora fiscal 2G el comando fiscal de solicitud del siguiente registro de información almacenada en la CTD (cinta testigo digital, o memoria de auditoría).

Definido internamente como:

```
ObtenerSiguienteBloqueAuditoria([out, retval] RespuestaObtenerSiguienteBloqueAuditoria* );
```

Ejemplo:

```
Dim bloque2 As HasarArgentina.RespuestaObtenerSiguienteBloqueAuditoria
bloque2 = hasar.ObtenerSiguienteBloqueAuditoria()
Debug.Print("Sig. bloque =[" & bloque2.Informacion & "]")
```

La respuesta fiscal al comando fiscal generado por 'ObtenerSiguienteBloqueAuditoria()' es la estructura *RespuestaObtenerSiguienteBloqueAuditoria*, definida internamente como:

```
typedef
struct tagRespuestaObtenerSiguienteBloqueAuditoria {
    [helpstring("Registro")]
}
    IdentificadorBloque Registro;
    [helpstring("Informacion")]
}
    BSTR Informacion;
} RespuestaObtenerSiguienteBloqueAuditoria;
```

<i>Registro</i>	<p>Informa si se trata de la descarga del último bloque de información ('<i>BloqueFinal</i>'), o aún hay más ('<i>BloqueInformacion</i>').</p> <p>Es una de las constantes pertenecientes al enum <i>IdentificadorBloque</i>.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td><i>BloqueFinal</i></td> <td>0</td> </tr> <tr> <td><i>BloqueInformacion</i></td> <td>1</td> </tr> </table>	<i>BloqueFinal</i>	0	<i>BloqueInformacion</i>	1
<i>BloqueFinal</i>	0				
<i>BloqueInformacion</i>	1				
<i>Informacion</i>	<p>Informa el siguiente bloque de información de la CTD (cinta testigo digital, o memoria de auditoría) en el formato solicitado originalmente mediante el comando fiscal generado por '<i>ObtenerPrimerBloqueAuditoria()</i>'. En el caso de tratarse del registro final, puede no contener información.</p>				

ObtenerSiguienteBloqueDocumento()

Permite enviar a la impresora fiscal 2G el comando fiscal de solicitud del siguiente registro de información almacenada en la CTD (cinta testigo digital, o memoria de auditoría), correspondiente a determinado tipo de comprobante.

Definido internamente como:

```
ObtenerSiguienteBloqueDocumento([out, retval] RespuestaObtenerSiguienteBloqueDocumento* );
```

Ejemplo:

```
Dim bloque2 As HasarArgentina.RespuestaObtenerSiguienteBloqueDocumento

bloque2 = hasar.ObtenerSiguienteBloqueDocumento()
Debug.Print("Sig. bloque =[" & bloque2.Informacion & "]")
```

La respuesta fiscal al comando fiscal generado por 'ObtenerSiguienteBloqueDocumento()' es la estructura *RespuestaObtenerSiguienteBloqueDocumento*, definida internamente como:

```
typedef
struct tagRespuestaObtenerSiguienteBloqueDocumento {
    [helpstring("Registro")]
}
    IdentificadorBloque Registro;
    [helpstring("Informacion")]
}
    BSTR Informacion;
} RespuestaObtenerSiguienteBloqueDocumento;
```

<i>Registro</i>	<p>Informa si se trata de la descarga del último bloque de información ('<i>BloqueFinal</i>'), o aún hay más ('<i>BloqueInformacion</i>').</p> <p>Es una de las constantes pertenecientes al enum <i>IdentificadorBloque</i>.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;"><i>BloqueFinal</i></td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;"><i>BloqueInformacion</i></td> <td style="text-align: center;">1</td> </tr> </table>	<i>BloqueFinal</i>	0	<i>BloqueInformacion</i>	1
<i>BloqueFinal</i>	0				
<i>BloqueInformacion</i>	1				
<i>Informacion</i>	<p>Informa el siguiente bloque de información correspondiente a la descarga de comprobantes, en el formato solicitado originalmente mediante el comando fiscal generado por '<i>ObtenerPrimerBloqueDocumento()</i>'. En el caso de tratarse del registro final, este campo podría estar vacío.</p>				

ObtenerSiguienteBloqueLog()

Permite enviar a la impresora fiscal 2G el comando fiscal de solicitud del siguiente registro de información interna, generada por las actividades propias del equipo. No se trata de información de interés administrativo, contable o impositivo.

Definido internamente como:

```
ObtenerSiguienteBloqueLog([out, retval] RespuestaObtenerSiguienteBloqueLog* );
```

Ejemplo:

```
Dim bloque2 As HasarArgentina.RespuestaObtenerSiguienteBloqueLog
```

```
bloque2 = hasar.ObtenerSiguieteBloqueLog()
Debug.Print("Sig. bloque  =[" & bloque2.Informacion & "]")
```

La respuesta fiscal al comando fiscal generado por 'ObtenerSiguieteBloqueLog()' es la estructura *RespuestaObtenerSiguieteBloqueLog*, definida internamente como:

```
typedef
struct tagRespuestaObtenerSiguieteBloqueLog {
    [helpstring("Registro")]
}
    IdentificadorBloque Registro;
    [helpstring("Informacion")]
}
    BSTR Informacion;
} RespuestaObtenerSiguieteBloqueLog;
```

<i>Registro</i>	<p>Informa si se trata de la descarga del último bloque de información ('<i>BloqueFinal</i>'), o aún hay más ('<i>BloqueInformacion</i>').</p> <p>Es una de las constantes pertenecientes al enum <i>IdentificadorBloque</i>.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td><i>BloqueFinal</i></td> <td>0</td> </tr> <tr> <td><i>BloqueInformacion</i></td> <td>1</td> </tr> </table>	<i>BloqueFinal</i>	0	<i>BloqueInformacion</i>	1
<i>BloqueFinal</i>	0				
<i>BloqueInformacion</i>	1				
<i>Informacion</i>	<p>Informa el siguiente bloque de información correspondiente al archivo interno del registro de actividades de la impresora fiscal 2G. En el caso de tratarse del registro final, este campo podría encontrarse vacío.</p>				

ObtenerSiguieteBloqueReporteElectronico()

Permite enviar a la impresora fiscal 2G el comando fiscal de solicitud de descarga del siguiente registro de información correspondiente a los reportes electrónicos para la AFIP, que se exigen a los emisores de comprobantes mediante el uso de impresoras fiscales 2G: CTD (cnta testigo digital, o memoria de auditoria), Reporte de Totales (totales por jornada fiscal/comercial), y el Reporte de Duplicados 'A' (incluyendo 'A con leyenda' y 'M'). El primero de los reportes (CTD) el contribuyente debe conservarlo (dos copias), y los otros dos se presentan a la AFIP como DDJJ (declaraciones juradas).

La información se descarga y se entrega en formato ASCII85.

Definido internamente como:

```
ObtenerSiguieteBloqueReporteElectronico([out, retval] RespuestaObtenerSiguieteBloqueReporteElectronico* );
```

Ejemplo:

```
Dim bloque2 As HasarArgentina.RespuestaObtenerSiguienteBloqueReporteElectronico
```

```
bloque2 = hasar.ObtenerSiguienteBloqueReporteElectronico()  
Debug.Print("Sig. bloque =[" & bloque2.Informacion & "]")
```

La respuesta fiscal al comando fiscal generado por 'ObtenerSiguienteBloqueReporteElectronico()' es la estructura *RespuestaObtenerSiguienteBloqueReporteElectronico*, definida internamente como:

```
typedef  
struct tagRespuestaObtenerSiguienteBloqueReporteElectronico {  
    [helpstring("Registro")  
]  
    IdentificadorBloque Registro;  
    [helpstring("Informacion")  
]  
    BSTR Informacion;  
} RespuestaObtenerSiguienteBloqueReporteElectronico;
```

<i>Registro</i>	<p>Informa si se trata de la descarga del último bloque de información ('<i>BloqueFinal</i>'), o aún hay más ('<i>BloqueInformacion</i>').</p> <p>Es una de las constantes pertenecientes al enum <i>IdentificadorBloque</i>.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;"><i>BloqueFinal</i></td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;"><i>BloqueInformacion</i></td> <td style="text-align: center;">1</td> </tr> </table>	<i>BloqueFinal</i>	0	<i>BloqueInformacion</i>	1
<i>BloqueFinal</i>	0				
<i>BloqueInformacion</i>	1				
<i>Informacion</i>	<p>Informa el siguiente bloque de información, en formato ASCII85, correspondiente a los reportes que se exigen a los emisores de comprobantes. En el caso de tratarse del registro final de información, el contenido del campo de información podría estar vacío.</p>				

ObtenerUltimoEstadoFiscal()

Permite conocer el estado fiscal de la impresora fiscal 2G, reportado en la respuesta fiscal del último comando fiscal enviado.

Definido internamente como:

```
ObtenerUltimoEstadoFiscal([out, retval] EstadoFiscal* );
```

Ejemplo:

```
Dim resp As HasarArgentina.EstadoFiscal
```

```
resp = hasar.ObtenerUltimoEstadoFiscal()  
Debug.Print("Estado fiscal = " & hasar.ObtenerCampoRespuesta("Estado/Fiscal"))  
Debug.Print("Documento Abierto ? =[" & resp.DocumentoAbierto & "]")
```

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

```
Debug.Print("Documento Fiscal Abierto ?=[" & resp.DocumentoFiscalAbierto & "]")
Debug.Print("Error Aritmético ?=[" & resp.ErrorAritmetico & "]")
Debug.Print("Error Ejecución ?=[" & resp.ErrorEjecucion & "]")
Debug.Print("Error Estado ?=[" & resp.ErrorEstado & "]")
Debug.Print("Error General ?=[" & resp.ErrorGeneral & "]")
Debug.Print("Error Memoria Auditoría ?=[" & resp.ErrorMemoriaAuditoria & "]")
Debug.Print("Error Memoria Fiscal ?=[" & resp.ErrorMemoriaFiscal & "]")
Debug.Print("Error Memoria de Trabajo ?=[" & resp.ErrorMemoriaTrabajo & "]")
Debug.Print("Error de Parámetro ?=[" & resp.ErrorParametro & "]")
Debug.Print("Memoria Fiscal Casi Llena ?=[" & resp.MemoriaFiscalCasiLlena & "]")
Debug.Print("Memoria Fiscal Inicializada ?=[" & resp.MemoriaFiscalInicializada & "]")
Debug.Print("Memoria Fiscal Llena ?=[" & resp.MemoriaFiscalLlena & "]")
```

'*ObtenerUltimoEstadoFiscal()*' consulta la respuesta fiscal del último comando fiscal enviado a la impresora fiscal 2G. Entrega la estructura *EstadoFiscal*, definida internamente como:

```
typedef
struct tagEstadoFiscal {
    [helpstring("ErrorMemoriaFiscal")
]
    VARIANT_BOOL ErrorMemoriaFiscal;
    [helpstring("ErrorMemoriaTrabajo")
]
    VARIANT_BOOL ErrorMemoriaTrabajo;
    [helpstring("ErrorMemoriaAuditoria")
]
    VARIANT_BOOL ErrorMemoriaAuditoria;
    [helpstring("ErrorGeneral")
]
    VARIANT_BOOL ErrorGeneral;
    [helpstring("ErrorParametro")
]
    VARIANT_BOOL ErrorParametro;
    [helpstring("ErrorEstado")
]
    VARIANT_BOOL ErrorEstado;
    [helpstring("ErrorAritmetico")
]
    VARIANT_BOOL ErrorAritmetico;
    [helpstring("MemoriaFiscalLlena")
]
    VARIANT_BOOL MemoriaFiscalLlena;
    [helpstring("MemoriaFiscalCasiLlena")
]
    VARIANT_BOOL MemoriaFiscalCasiLlena;
    [helpstring("MemoriaFiscalInicializada")
]
    VARIANT_BOOL MemoriaFiscalInicializada;
    [helpstring("DocumentoFiscalAbierto")
]
    VARIANT_BOOL DocumentoFiscalAbierto;
    [helpstring("DocumentoAbierto")
]
    VARIANT_BOOL DocumentoAbierto;
    [helpstring("ErrorEjecucion")
]
    VARIANT_BOOL ErrorEjecucion;
} EstadoFiscal;
```

<i>ErrorMemoriaFiscal</i>	Informa si se ha producido un error durante el acceso a la memoria fiscal.
<i>ErrorMemoriaTrabajo</i>	Informa si se ha producido un error durante el acceso a la memoria de trabajo.
<i>ErrorMemoriaAuditoria</i>	Informa si se ha producido un error durante el acceso a la CTD (cinta testigo digital, o memoria de auditoría).
<i>ErrorGeneral</i>	Informa si se ha producido un error de tipo general.
<i>ErrorParametro</i>	Informa si se ha producido un error de configuración.
<i>ErrorEstado</i>	Informa si se ha producido un error de estado.
<i>ErrorAritmetico</i>	Informa si se ha producido un error aritmético.
<i>MemoriaFiscalLlena</i>	Informa si se ha agotado la capacidad de almacenamiento de la memoria fiscal.
<i>MemoriaFiscalCasiLlena</i>	Informa si la capacidad de almacenamiento de la memoria fiscal está a punto de agotarse.
<i>MemoriaFiscalInicializada</i>	Informa si la impresora fiscal 2G está operando en modo fiscal, o en modo entrenamiento.
<i>DocumentoFiscalAbierto</i>	Informa si la impresora fiscal 2G tiene un DF (documento fiscal) abierto.
<i>DocumentoAbierto</i>	Informa si la impresora fiscal 2G tiene un documento abierto (de cualquier tipo).
<i>ErrorEjecucion</i>	Informa si ha producido un error al intentar ejecutar un comando fiscal.

ObtenerUltimoEstadoImpresora()

Permite conocer el estado de la impresora estándar, que forma parte de la impresora fiscal 2G; estado reportado en la respuesta fiscal del último comando fiscal enviado.

Definido internamente como:

```
ObtenerUltimoEstadoImpresora([out, retval] EstadoImpresora* );
```

Ejemplo:

```
Dim resp As HasarArgentina.EstadoImpresora
```

Grupo HASAR – Compañía HASAR SAIC – OCX 2G Manual del usuario - Rev 004

```

resp = hasar.ObtenerUltimoEstadoImpresora()
Debug.Print("Cajón Dinero Abierto ?=[" & resp.CajonAbierto & "]")
Debug.Print("Impresora en Error ?=[" & resp.ErrorImpresora & "]")
Debug.Print("Falta Papel J. ?=[" & resp.FaltaPapelJournal & "]")
Debug.Print("Falta Papel T. ?=[" & resp.FaltaPapelReceipt & "]")
Debug.Print("Impresora Ocupada ?=[" & resp.ImpresoraOcupada & "]")
Debug.Print("Impresora Offline ?=[" & resp.ImpresoraOffLine & "]")
Debug.Print("Tapa Abierta ?=[" & resp.TapaAbierta & "]")

```

'ObtenerUltimoEstadoImpresora()' consulta la respuesta fiscal del último comando fiscal enviado a la impresora fiscal 2G. Entrega la estructura *EstadoImpresora*, definida internamente como:

```

typedef
struct tagEstadoImpresora {
    [helpstring("ImpresoraOcupada")
]
    VARIANT_BOOL ImpresoraOcupada;
    [helpstring("ErrorImpresora")
]
    VARIANT_BOOL ErrorImpresora;
    [helpstring("ImpresoraOffLine")
]
    VARIANT_BOOL ImpresoraOffLine;
    [helpstring("FaltaPapelJournal")
]
    VARIANT_BOOL FaltaPapelJournal;
    [helpstring("FaltaPapelReceipt")
]
    VARIANT_BOOL FaltaPapelReceipt;
    [helpstring("TapaAbierta")
]
    VARIANT_BOOL TapaAbierta;
    [helpstring("CajonAbierto")
]
    VARIANT_BOOL CajonAbierto;
    [helpstring("OrLogico")
]
    VARIANT_BOOL OrLogico;
} EstadoImpresora;

```

<i>ImpresoraOcupada</i>	Informa si la impresora, como tal, se encuentra en estado Busy.
<i>ErrorImpresora</i>	Informa si la impresora, como tal, se encuentra en estado de error.
<i>ImpresoraOffLine</i>	Informa si la impresora, como tal, se encuentra fuera de línea.
<i>FaltaPapelJournal</i>	Informa si la impresora fiscal 2G reporta que le falta papel (rollo testigo).
<i>FaltaPapelReceipt</i>	Informa si la impresora fiscal 2G reporta que le falta papel (rollo comprobantes).

<i>TapaAbierta</i>	Informa si la impresora fiscal 2G reporta que la tapa está mal cerrada (o abierta).
<i>CajonAbierto</i>	Informa si la impresora fiscal 2G reporta que el cajón de dinero se encuentra abierto.
<i>OrLogico</i>	Avisa si al menos uno de los campos anteriores retorna el valor 'true' (verdadero).

ObtenerVersionProtocolo()

Permite obtener el número que identifica a la versión de protocolo que maneja el OCX 2G. No se genera, ni se envía comando fiscal alguno a la impresora fiscal 2G. Es importante para determinar compatibilidad entre el OCX 2G y la impresora fiscal 2G que se desea manejar.

Para que el OCX 2G y el firmware fiscal de la impresora fiscal 2G resulten compatibles, la versión de protocolo del OCX 2G debe ser mayor o igual a la que reporta la impresora fiscal 2G.

Definido internamente como:

```
ObtenerVersionProtocolo([out, retval] short* );
```

Ejemplo:

```
Debug.Print("OCX 2G Protocolo = [" & hasar.ObtenerVersionProtocolo & "]")
```

OperacionCajero()

De aplicación y uso interno por parte de las registradoras fiscales 2G y los facturadores web incorporados a las impresoras fiscales 2G.

No disponible en los modelos de impresoras fiscales 2G: SMH/PT-1000F v01.00 y SMH/PT-250F v01.00.

Permite enviar a la impresora fiscal 2G el comando fiscal que registra una entrada o una salida de un cajero y lo asienta cronológicamente en la CTD (cinta testigo digital, o memoria de auditoría) para su posterior consulta.

Definido internamente como:

```
OperacionCajero(  
    [in] long NumeroCajero,  
    [in, optional, defaultvalue("")] BSTR NombreCajero,  
    [in, optional, defaultvalue(69)] OperacionesCajero Operacion);
```

<i>NumeroCajero</i>	Permite especificar el número de identificación del cajero que se presenta a, o abandona, las tareas de facturación.
---------------------	--

<i>NombreCajero</i>	Permite especificar el nombre y apellido del cajero que se presenta a, o abandona, las tareas de facturación.				
<i>Operacion</i>	Permite especificar si el cajero se presenta a, o abandona, las tareas de facturación. Debe ser alguna de las constantes pertenecientes al enum <i>OperacionesCajero</i> . <table><tr><td><i>Entrada</i></td><td>69</td></tr><tr><td><i>Salida</i></td><td>83</td></tr></table>	<i>Entrada</i>	69	<i>Salida</i>	83
<i>Entrada</i>	69				
<i>Salida</i>	83				

Ejemplo:

```
hasar.OperacionCajero(1234, "Cajero 1234", HasarArgentina.OperacionesCajero.Entrada)
```

PedirReimpresion()

Permite enviar a la impresora fiscal 2G el comando fiscal de solicitud de impresión de una copia fiel del último comprobante emitido (cerrado inmediatamente antes de esta solicitud).

En el encabezado del ejemplar de copia podrá verse impresa la leyenda 'REIMPRESION', entendiéndose por reimpresión a la impresión de un ejemplar de copia fiel del comprobante original. Este comando fiscal se puede enviar, consecutivamente, cuantas veces resulte necesario.

No confundir el uso del comando fiscal generado por '*PedirReimpresión()*' con el uso del comando fiscal generado por '*CopiarComprobante()*'.

Definido internamente como:

```
PedirReimpresion([out, retval] RespuestaPedirReimpresion* );
```

Ejemplo:

```
Dim resp As HasarArgentina.RespuestaPedirReimpresion  
resp = hasar.PedirReimpresion()  
Debug.Print("Copia Nro.: [" & resp.NumeroDeCopia & "]")
```

La respuesta fiscal al comando fiscal generado por '*PedirReimpresion()*' es la estructura *RespuestaPedirReimpresion*, definida internamente como:

```
typedef [uuid(CC86CDF0-A98F-4D88-9C2E-
struct tagRespuestaPedirReimpresion {
    [helpstring("NumeroDeCopia")
    ]
    long NumeroDeCopia;
} RespuestaPedirReimpresion;
```

<i>NumeroDeCopia</i>	<p>Informa el número de copia de la reimpresión solicitada.</p> <p>Se informa siempre '1' -sin importar cuantas veces se haya enviado el comando fiscal- de haberse emitido como último comprobante previo al comando fiscal de impresión de un <i>Informe Diario de Cierre</i> (cierre 'Z').</p>
----------------------	---

ReportarZetasPorFecha()

Permite enviar a la impresora fiscal 2G el comando fiscal de impresión del *Informe de Auditoría*, para los *Informes Diarios de Cierre* (cierre 'Z') comprendidos en el rango de fechas indicado.

Definido internamente como:

```
ReportarZetasPorFecha(
    [in] DATE FechaInicial,
    [in] DATE FechaFinal,
    [in] TipoReporteAuditoria Reporte);
```

<i>FechaInicial</i>	Permite especificar la fecha de inicio del rango de búsqueda de jornadas fiscales/comerciales a informar.				
<i>FechaFinal</i>	Permite especificar la fecha final del rango de búsqueda de jornadas fiscales/comerciales a informar.				
<i>Reporte</i>	<p>Permite especificar el tipo de <i>Informe de Auditoría</i> que se desea imprimir. Debe ser alguna de las constantes pertenecientes al enum <i>TipoReporteAuditoria</i>.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td><i>ReporteAuditoriaDiscriminado</i></td> <td>68</td> </tr> <tr> <td><i>ReporteAuditoriaGlobal</i></td> <td>84</td> </tr> </table>	<i>ReporteAuditoriaDiscriminado</i>	68	<i>ReporteAuditoriaGlobal</i>	84
<i>ReporteAuditoriaDiscriminado</i>	68				
<i>ReporteAuditoriaGlobal</i>	84				

Si en la memoria fiscal de la impresora fiscal 2G existen registros correspondientes al rango de fechas solicitado, se informan los datos entre ellas. Si una o ambas fechas no corresponden a registros existentes, se selecciona(n) la(s) fecha(s) más cercana(s) a la(s) solicitada(s) que sí cuenten con registros, siempre dentro del período solicitado.

Ejemplo:

```
hasar.ReportarZetasPorFecha("21/01/2019", "28/01/2019", _
HasarArgentina.TipoReporteAuditoria.ReporteAuditoriaGlobal)
```

ReportarZetasPorNumeroZeta()

Permite enviar a la impresora fiscal 2G el comando fiscal de impresión del *Informe de Auditoría*, para los *Informes Diarios de Cierre* (cierres 'Z') comprendidos en el rango de números de jornadas fiscales/comerciales que se ha indicado.

Definido internamente como:

```
ReportarZetasPorNumeroZeta(
    [in] long ZetaInicial,
    [in] long ZetaFinal,
    [in] TipoReporteAuditoria Reporte);
```

<i>ZetaInicial</i>	Permite especificar el número inicial del rango de búsqueda de jornadas fiscales/comerciales a informar.				
<i>ZetaFinal</i>	Permite especificar el número final del rango de búsqueda de jornadas fiscales/comerciales a informar.				
<i>Reporte</i>	Permite especificar el tipo de <i>Informe de Auditoría</i> que se desea imprimir. Debe ser alguna de las constantes pertenecientes al enum <i>TipoReporteAuditoria</i> . <table border="1" data-bbox="798 1276 1260 1366"> <tr> <td><i>ReporteAuditoriaDiscriminado</i></td> <td>68</td> </tr> <tr> <td><i>ReporteAuditoriaGlobal</i></td> <td>84</td> </tr> </table>	<i>ReporteAuditoriaDiscriminado</i>	68	<i>ReporteAuditoriaGlobal</i>	84
<i>ReporteAuditoriaDiscriminado</i>	68				
<i>ReporteAuditoriaGlobal</i>	84				

Si en la memoria fiscal de la impresora fiscal 2G existen registros correspondientes al rango de *Informes Diarios de Cierre* (cierres 'Z') solicitado, se informan los datos entre ellos. Si uno o ambos números de *Informes Diarios de Cierre* no corresponden a registros existentes, se selecciona(n) el(los) *Informe(s) Diarios(s) de Cierre* más cercano(s) al(los) solicitado(s) que sí cuenten con registros, siempre dentro del período solicitado.

Ejemplo:

```
hasar.ReportarZetasPorNumeroZeta(1, 100, _
HasarArgentina.TipoReporteAuditoria.ReporteAuditoriaGlobal)
```

Agosto 2019, Rev. 004 – Copyright (c) 2013/2019 – **Compañía HASAR SAIC**

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

_ **Compañía 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**.

Marcos Sastre 2214 [B1618CSD] Ricardo Rojas | Tigre | Buenos Aires | Argentina
Tel.: [54.11] 4117.8900 | Fax.: [54.11] 4117.8998 | www.grupohasar.com