

Contenido General

IMPRESORAS FISCALES HASAR - CICLO VITAL	2
VIDA INTRAUTERINA	2
VIDA FISCAL	2
VIDA DESPUÉS DE LA MUERTE	3
IMPRESORAS FISCALES Y AUTÓMATAS DE ESTADOS	4
RESET DE CAJA.....	6
COMPROBANTES FISCALES INCOMPLETOS.....	7
TRATAMIENTO DE LA RESPUESTA FISCAL.....	8
CONSULTA DE ESTADO INTERMEDIO.....	12
USO DEL COMANDO STATPRN	12
<i>Caso Normal:</i>	12
<i>Caso falta de papel:</i>	12
IMPUESTOS INTERNOS - FACTOR K.....	14
COMO DETERMINARLO	14
COMO SE REFLEJA EN EL PAPEL:	15
<i>Documento fiscal "A"</i>	15
<i>Documento fiscal "B/C"</i>	15
PRECIOS UNITARIOS - MÁS DE DOS DECIMALES	17
EJEMPLO: LA VENTA DE COMBUSTIBLES	17
<i>Cálculo del factor K:</i>	17
REDONDEO DE DECIMALES	19
EJEMPLO: FACTURA "A"	19
OBTENER DISCRIMINACIÓN:	21
<i>Vía el OCX:</i>	21
<i>Vía strings de comandos:</i>	21
IMPRESORAS FISCALES DE PÁGINA COMPLETA	23
CÓMO AUMENTAR LA MEMORIA DISPONIBLE EN DOS BAJO WINDOWS.....	24
IMPRESORAS FISCALES Y REDES	26
NOTA PARA LOS DESARROLLADORES DE SOFTWARE.....	27
PROTOCOLO FISCAL.....	28

Impresoras Fiscales HASAR - Ciclo vital

El ciclo vital de las impresoras fiscales HASAR consta de tres fases:

1. Vida intrauterina,
2. Vida fiscal, y ...
3. Vida después de la muerte.

Vida intrauterina

La impresora fiscal sale de la línea de producción en el llamado “modo entrenamiento”. En este modo, la impresora fiscal opera de manera similar a la forma de hacerlo durante su “vida fiscal”, excepto por las siguientes diferencias:

- a. En todos los documentos que se emiten, tanto fiscales, como no fiscales y no fiscales homologados, los espacios en blanco son reemplazados por signos de interrogación.
- b. Al enviarse el comando de cierre diario (**DailyClose**), los datos correspondientes no son grabados en la memoria fiscal. Por lo tanto, si se solicita un informe de auditoría (**DailyCloseByNumber** o **DailyCloseByDate**), éste tendrá los valores en cero. Lo mismo sucede con los reportes solicitados a través del canal de la autoridad fiscal.
- c. Al emitirse documentos fiscales y/o notas de crédito, no se imprime el logo DGI, ni se imprime – de corresponder – el número de CAI, ni su respectiva fecha de vencimiento.

La impresora fiscal sale del “modo entrenamiento” e ingresa en el “modo fiscal” mediante la operación de “inicialización”. Esta operación sólo puede ser realizada por alguno de los Servicios Técnicos Autorizados de Cía HASAR SAIC.

Vida fiscal

Luego de “inicializada”, la impresora fiscal no mantiene registros de las operaciones realizadas en el “modo entrenamiento” (es decir, no tiene recuerdos prenatales) .

Durante la “vida fiscal”, cada vez que se emite un comando de cierre diario se almacena en memoria fiscal un registro que contiene los datos de las ventas diarias, y otros de interés fiscal para la autoridad competente.

La memoria fiscal, dependiendo del modelo de impresora fiscal, tiene capacidad para almacenar 1850, o 3800, de estos registros (algo más de cinco años en el primer caso, y de diez años en el segundo); cuando se supera el registro 1820, o el 3770, al final del informe impreso cada vez que se ejecuta el comando de cierre diario, se imprime una línea que indica la cantidad de registros que aún quedan sin llenar.

La “vida fiscal” de la impresora fiscal finaliza por una de dos razones:

- a. Porque se han llenado los (1850 / 3800) registros disponibles.
- b. Porque se ha ejecutado el comando de bloqueo de la memoria (**KillEpromFiscal**).
Este comando sólo puede ser emitido por la autoridad fiscal competente.

Vida después de la muerte

Una vez llenos los 1850 (3800) registros de la memoria fiscal, o emitido el comando de bloqueo de la memoria fiscal, ésta no admite que se graben más datos en ella. Desde el punto de vista operativo, se puede retirar y reemplazar por otra con los registros vacíos.

Sin embargo, es posible leer los datos ya almacenados en una memoria fiscal llena, o bloqueada. Por ello, el propietario deberá conservarla durante el tiempo que la reglamentación vigente establezca.

Impresoras Fiscales y Autómatas de Estados

Se llama autómata de estados a una máquina que, a medida que opera, va pasando por sucesivos estados. En el caso de las impresoras fiscales, la máquina de estados llega a un estado a través de un comando determinado, y sale de dicho estado por medio de otro comando.

El estado en que se encuentra la impresora fiscal determina exactamente cuáles operaciones puede realizarse y cuáles no. Dicho en otras palabras, sólo una parte del universo de operaciones que la impresora fiscal puede realizar está disponible en un estado determinado y, para realizar las otras operaciones, deberá cambiarse el estado en que se encuentra.

Desde este punto de vista, las impresoras fiscales HASAR operan como autómatas de estados: de los comandos que el host envíe, sólo se ejecutarán aquellos permitidos en el estado en que se encuentra y se rechazarán los otros.

Por ejemplo, luego de emitirse el comando de apertura de un documento fiscal (**OpenFiscalReceipt**, ver manual de comandos), la impresora sólo ejecutará los correspondientes al estado “Documento fiscal abierto”: se podrá escribir un texto fiscal (**PrintFiscalText**), o vender un producto (**PrintLineItem**), pero se rechazarán comandos como el de apertura de un documento no fiscal (**OpenNonFiscalReceipt**) o el cierre de jornada fiscal (**DailyClose**).

A su vez, muchos estados tienen sus propios subestados. Por ejemplo, el estado “Documento fiscal abierto” tiene un subestado, “Monto cero o negativo / IVA negativo” que no permite cerrar el comprobante fiscal, es decir, no acepta el comando **CloseFiscalReceipt**. Para poder cerrar el comprobante fiscal primero debe venderse un producto de monto positivo e IVA cero, o positivo, mediante el comando **PrintLineItem**. Este comando cambiará el subestado de la impresora a “Monto positivo / IVA cero o positivo”, subestado en el cual se acepta el comando **CloseFiscalReceipt**.

Lo anterior implica que el software de gestión del punto de venta debe saber exactamente el estado en que se encuentra la impresora fiscal en todo momento. Esto es de fundamental importancia para el correcto funcionamiento del sistema. Por ejemplo, de acuerdo con las especificaciones que rigen el funcionamiento de los controladores fiscales, al volver a encenderse la impresora fiscal tras un corte de energía, ésta debe conservar el estado en que se encontraba antes del corte. En ese caso, el programa debe saber cuál es ese estado para, en caso necesario, comunicárselo al operador del punto de venta para que actúe en consecuencia.

Las impresoras fiscales HASAR comunican el estado en que se encuentra mediante dos procedimientos:

1. Como respuesta al comando de consulta de estado (**StatusRequest**). La impresora fiscal responde con varios campos de datos en formato ASCII. El primero representa el status fiscal (comprobante fiscal abierto, comando no válido, etc.), el segundo representa el status de la impresora ((off)online, falta de papel, etc.) y el resto de los campos brinda información auxiliar acerca del estado fiscal general.
2. Como respuesta a cualquier comando, aún los no permitidos en el estado fiscal en que se encuentra, la impresora fiscal responde con los dos primeros campos -- citados en el punto 1 -- y, de corresponder, con más campos de información.

Tal como se indica en el manual de comandos, la respuesta es enviada desde la impresora fiscal al host a través del mismo canal serie por el que se transmiten los comandos. En los apéndices correspondientes de dicho manual se indica cómo deben interpretarse el campo al status fiscal, y el campo de status de la impresora.

También hay una apéndice que explica como debe interpretarse el campo de status auxiliar, en el manual de comandos de las impresoras fiscales que incluyen este campo como parte de la respuesta al comando **StatusRequest**.

Es evidente que el programador del software de gestión del punto de venta debe tener un conocimiento acabado de los comandos permitidos en cada uno de los estados de la impresora fiscal. Sin embargo, al estar el autómata constituido por estados que a su vez se dividen en subestados, la estructura es bastante compleja y difícil de representar en forma esquemática. Para mayores detalles se deberá consultar la descripción de cada comando en el manual correspondiente.

Reset de Caja

En los casos en que, por desincronización entre el host y la impresora fiscal, sea necesario hacer un reset del sistema, puede llevarse la impresora fiscal a un estado conocido ejecutando los siguientes tres comandos, uno a continuación del otro:

Cancel (TotalTender), CloseFiscalReceipt y CloseNonFiscalReceipt.

- **Cancel (TotalTender).** Si se encontraba abierto un comprobante fiscal y aún no se había realizado un pago, este comando cancela el comprobante en curso. El uso de **Cancel** o **TotalTender** depende del modelo de impresora fiscal en uso. Los montos del comprobante fiscal no se acumulan en la memoria fiscal.
- **CloseFiscalReceipt.** Si se encontraba abierto un comprobante fiscal y ya se había realizado un pago, este comando cierra el comprobante fiscal completando, de ser necesario, el pago. Los montos del comprobante fiscal se acumulan en la memoria fiscal.

Nota: Debe tenerse en cuenta que de no implementarse una inteligencia apropiada para el tratamiento de estas situaciones, pueden producirse diferencias entre la impresora fiscal y el software de gestión del punto de venta. Estas diferencias pueden verse reflejadas en la numeración, y/o en los acumulados.

- **CloseNonFiscalReceipt.** Si se encontraba abierto un comprobante no fiscal, este comando lo cierra.

Comprobantes Fiscales Incompletos

En caso de que, luego de cerrarse un comprobante fiscal y/o una nota de crédito, el comprobante NO se imprima totalmente, incluyendo la última línea que contiene el logo fiscal (*CF DGI*), y además NO aparezca en el mismo comprobante un mensaje impreso indicando que el documento se ha cancelado, debe realizarse el siguiente procedimiento:

1. El comprobante debe ser retenido por el emisor (no debe entregarse al comprador).
2. Debe emitirse un nuevo comprobante para documentar la compra, o devolución.
3. Para anular el documento incompleto mencionado en el punto 1., debe emitirse el correspondiente contra documento (nota de crédito o débito) por el monto del comprobante mal impreso.

Tratamiento de la Respuesta Fiscal

Para analizar el status fiscal y el status de impresora, se debe separar en campos al string de respuesta (ver manual de comandos) y examinar los bits de cada uno de dichos campos. Los campos se encuentran separados por el caracter FS (Field Separator = ASCII 28 decimal).

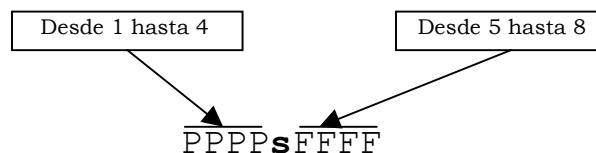
El siguiente ejemplo (en lenguaje Basic) extrae los campos de status de la respuesta y realiza el análisis de sus bits.

El string de respuesta siempre comienza de la siguiente forma: "PPPP**s**FFFF"

donde:

PPPP:	Estado de la impresora (cuatro dígitos hexadecimales).
s :	Separador de campos (ASCII 28 decimal).
FFFF:	Estado fiscal (cuatro dígitos hexadecimales).

Como estas posiciones son las únicas fijas en el string de respuesta, se las puede identificar de la siguiente forma:



Suponiendo que los primeros nueve caracteres del string de respuesta se encuentran almacenados en la variable Respuesta, el uso de la función Left\$() para los primeros cuatro caracteres (status de impresora), y el uso de la función Mid\$() para los cuatro caracteres que siguen al caracter número 5 (separador de campo), permiten extraer los valores de los estados:

```
Dim PrinterStatus as String  
Dim FiscalStatus as String
```

```
PrinterStatus = Left$(Respuesta, 4)  
FiscalStatus = Mid$(Respuesta, 5, 4)
```

Del modo anterior se han separado en dos variables de tipo string el status fiscal y el status de impresora.

Si se hubiera necesitado un campo de longitud variable (como podría ser cualquier otro campo adicional de la respuesta), habría que haber usado la función InStr() para buscar el separador de campos tantas veces como campos hay que saltar para determinar el comienzo del campo buscado. InStr() toma como primer parámetro op-

cional la posición a partir de la cual buscar, por lo que en las sucesivas llamadas hay que pasar la posición encontrada anterior más uno:

```
Dim n as Integer, m as Integer  
Dim PrimerCampo as String, SegundoCampo as String  
  
n = InStr( Respuesta, chr$(28) )  
m = InStr( n + 1, Respuesta, chr$(28) )  
  
PrimerCampo = Left$(Respuesta, n - 1 )  
SegundoCampo = Mid$( Respuesta, n + 1, m - n - 1 )
```

Una vez extraídos los substrings con los estados se presenta otro problema: estos dos valores están en formato hexadecimal, y hay que convertirlos a variables enteras para poder usarlos. Asumiendo que en **PrinterStatus** y en **FiscalStatus** están los valores de los substrings recién extraídos:

```
Dim Printer as Integer  
Dim Fiscal as Integer  
  
Printer = Val( "&H" + PrinterStatus )  
Fiscal = Val( "&H" + FiscalStatus )
```

El string "&H" fuerza a la función **Val** (que convierte un string en un entero) a considerar el string como hexadecimal. Una vez hecho esto, ya estamos preparados para examinar los bits. Hasta aquí se ha logrado que el estado de la impresora se encuentre almacenado, en forma de entero, en la variable **Printer**, y el estado fiscal, también en forma de entero, en la variable **Fiscal**.

Prosiguiendo con el ejemplo, es posible examinar el bit 2 que indica error de impresora, en la variable **Printer** (tercer bit a partir del menos significativo).

```
Const ERRORPRINTER = 4           '// 4 = 0100 en binario  
  
if (( Printer And ERRORPRINTER ) = ERRORPRINTER ) then  
    Print "Error mecánico de impresora"  
end if
```

El examen se realiza mediante una operación "AND" a nivel de bits.

Por ejemplo, si el estado del printer era C084, la comparación efectuada es la siguiente:

C084		1100000010000100
	AND	
0004		0000000000000100
	=	
0004		0000000000000100

Cualquier bit que en la máscara elegida (**ERRORPRINTER**) esté en cero, queda en cero en el resultado, y cualquiera que esté en uno queda en uno, si en el número testeado (C084) ya estaba en uno.

De forma que si el resultado del "AND" es distinto de cero, alguno de los bits "prendidos" de la máscara estaban también "prendidos" en el número chequeado.

Si el resultado es exactamente el de la máscara, entonces exactamente todos los bits de la máscara estaban prendidos. Este tipo de operaciones con máscaras pueden ser combinadas, mediante el operador "OR":

```
Const FISCALMEMFAIL = &H1  
Const WORKINGMEMFAIL = &H2  
Const FISCALMEMFULL = &H80
```

```
Dim ErrorCritico AS Integer
```

```
ErrorCritico = FISCALMEMFULL OR WORKINGMEMFAIL OR FISCALMEMFAIL
```

```
if (( Fiscal and ErrorCritico ) <> 0 ) Then  
    Print "Hay un error fiscal crítico"  
end if
```

Otra forma de analizar los bits de status y mostrar mensajes asosicados, es mediante un array de estados:

Suponiendo que en la variable entera **FiscalStatus** se encuentra el status fiscal devuelto por la impresora fiscal, entonces:

```
Dim Mask As Long  
Dim i As Integer  
Dim FiscalStatusError (16) as String  
  
FiscalStatusError (1) = "Fiscal Memory Fail"  
FiscalStatusError (2) = "Working Memory Fail"  
FiscalStatusError (3) = ""  
FiscalStatusError (4) = "Unrecognized Command"  
FiscalStatusError (5) = "Invalid Field Data"  
FiscalStatusError (6) = "Invalid Command"  
FiscalStatusError (7) = "Total Overflow"  
FiscalStatusError (8) = "Fiscal Memory Full"  
FiscalStatusError (9) = "Fiscal Memory Near Full"  
FiscalStatusError (10) = "Fiscal Terminal Certified"  
FiscalStatusError (11) = "Fiscal Terminal Fiscalized"  
FiscalStatusError (12) = "Date Set Fail"  
FiscalStatusError (13) = "Receipt/Slip Open"  
FiscalStatusError (14) = "Slip Open"  
FiscalStatusError (15) = "Receipt Open"  
FiscalStatusError (16) = ""  
  
Mask = 1
```

```
For i = 1 To 16
    if (( FiscalStatus And Mask ) <> 0 ) then
        Print FiscalStatusError( i )
    end if
    Mask = Mask * 2
Next i
```

Lo que se hace en el ejemplo anterior es "prender" uno a uno los bits de la máscara (**Mask**) para testarlos contra el valor del status fiscal (**FiscalStatus**). **Mask** vale uno en primera instancia, y se va multiplicando por dos en cada iteración, produciendo cada una de las potencias de dos:

1	0001
2	0010
4	0100
8	1000

La variable **Mask** tiene que ser de tipo *long* porque de otra manera provocaría un overflow al sobrepasar la frontera del entero con signo (32767).

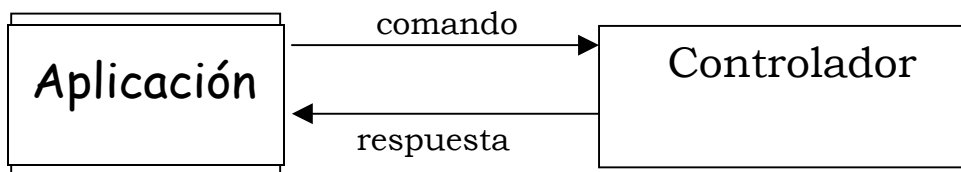
Consulta de Estado Intermedio

Uso del comando StatPrn

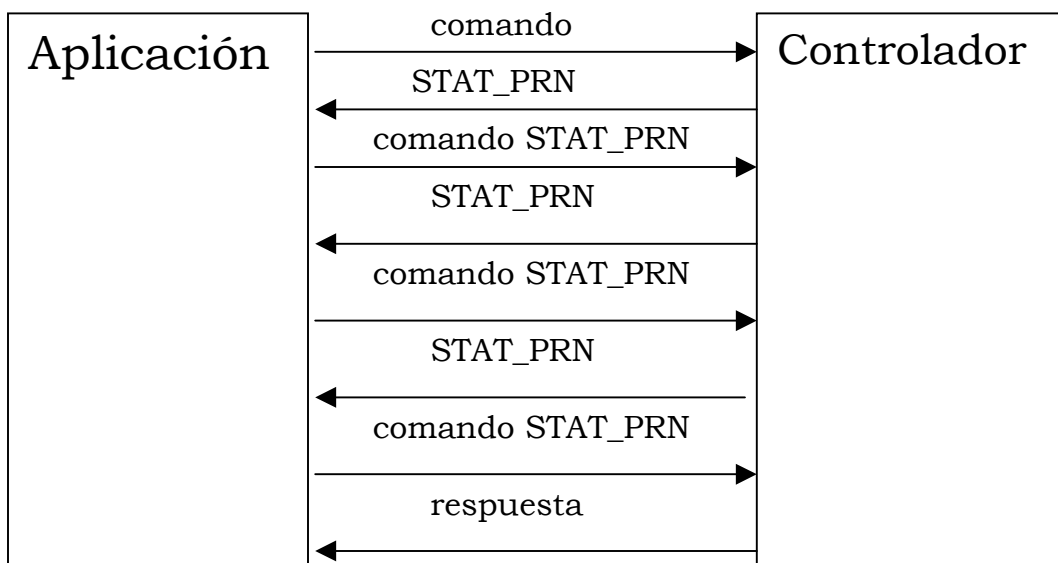
Existen modelos de impresoras fiscales (biprotocolo) en los cuales se dispone de una modalidad de trabajo -- consulta de estado intermedio -- que permite enterarse de errores mecánicos de impresora (enredo de papel, falta de papel, etc.) durante la ejecución de comandos, cosa que en otros modelos (monoprocolo) esta prestación no se halla disponible.

El protocolo que permite la consulta de estado intermedio ha de activarse de alguna forma (consultar en **drivers.pdf** las distintas posibilidades, según el driver en uso) y funciona de la siguiente forma:

Caso Normal:



Caso falta de papel:



Durante el estado de falta de papel, el controlador enviará a través del driver respuestas especiales del tipo STAT_PRN para que la aplicación pueda avisar al usuario de los problemas.

Si la aplicación detecta una respuesta del tipo STAT_PRN ante un comando, puede encuestar a la impresora fiscal con comandos especiales (**StatPrn**) para saber si aún permanece en el estado de error. Cuando la impresora fiscal salga de ese estado, devolverá la respuesta correspondiente al comando que generó el error.

En cada driver, se podrá determinar si la respuesta recibida es especial (STAT_PRN) o no.

El siguiente es un algoritmo posible, escrito en pseudocódigo:

```
Enviar( Comando )

Si ( Respuesta = STAT_PRN ) Entonces
    Hacer
        AvisarUsuario( Respuesta )
        Enviar( StatPrn )
    Hasta que ( Respuesta <> STAT_PRN )
Fin Si

Analizar( Respuesta )
```

La función **Enviar()** envía un comando a la impresora fiscal, **AvisarUsuario()** da un aviso de advertencia al operador, y **Analizar()** analiza la respuesta del comando.

Impuestos Internos - Factor K

Como determinarlo

Para indicar los impuestos internos en aquellos comandos que contengan un campo que lo requiera, su especificación puede ser de tres formas:

- Mediante porcentaje del precio unitario.
- Mediante un monto fijo.
- Mediante un factor K (valor entre 0 y 1)

Las dos primeras formas no están disponibles en todos los modelos de impresoras fiscales, en tanto que la especificación del factor K si se encuentra disponible en todos los modelos homologados por Cía. HASAR SAIC.

En el ejemplo que se verá se asume que la especificación de impuestos internos se hará mediante factor K.

Supongamos la venta de un ítem cuyo precio base (PB) es \$100.00, está afectado por una alícuota de IVA (%iva) del 21%, y un monto fijo de impuesto interno (\$ii) por unidad de \$10.01. Supongamos también, que el producto se vende por precio base.

Dado que el precio final (PF) se construye como:

$$PF = PB * (1 + \$iva + \$ii)$$

Donde:

$$\begin{aligned} PB &= 100.00 \\ \$iva &= PB * \%iva = 100.00 * (21 / 100) = 21.00 \\ \$ii &= 10.01 \end{aligned}$$

Entonces:

$$PF = 100.00 + 21.00 + 10.01 = 131.01$$

Para que lo anterior quede plasmado en el papel se debe pasar primero de un valor en monto fijo, a un porcentaje (%ii).

Luego:

$$\%ii = \$ii / PB = 10.01 / 100.00 = 0.1001$$

Como segundo paso convertimos el porcentaje de impuestos internos en el factor K, mediante la siguiente expresión:

$$\text{FactorK} = 1 / (1 + \%ii) = 1 / (1 + 0.1001) = 0.9090$$

En consecuencia, 0.9090 es el valor a pasar en el campo de impuestos internos del comando (por ejemplo, **PrintLineItem**).

Si el impuesto interno es fijo (no debe afectarse por descuentos ni recargos), debe especificarse precedido por el signo de suma: +0.9090

Como se refleja en el papel:

Documento fiscal "A"

El comando de impresión de ítem (**PrintLineitem**) correspondiente al ejemplo usado para determinar el valor del factor K permite ver (si es lo único vendido), la siguiente impresión:

1.000/100.0000	(21.00)	
Producto....	[90.90]	100.00
NETO SIN IVA		100.00
IVA 21.00 %		21.00
CONCEPTOS NO GRAVADOS		10.01
TOTAL	\$	131.01

Como puede apreciarse, el valor entre corchetes dividido 100 es el factor K determinado por cálculo. No cumple otra finalidad que reportarnos el factor K que la impresora fiscal recibió por comando.

Documento fiscal "B/C"

El comando de impresión de ítem (**PrintLineitem**) correspondiente al ejemplo usado para determinar el valor del factor K permite ver (si es lo único vendido), la siguiente impresión -- se usa exactamente el mismo comando que en los documentos fiscales "A" --:

Producto....	(21.00)	
	[16.03]	131.01
TOTAL	\$	131.01

En este tipo de documentos, el valor entre corchetes permite calcular el monto de IVA para el producto en cuestión, considerando su precio final:

$$\text{\$iva} = (16.03 / 100) * 131.01 = 21.00$$

O sea que del precio final del ítem, \$21.00 corresponden a IVA.

Precios Unitarios - Más de dos decimales

Para ciertos rubros comerciales existe la necesidad de indicar precios unitarios con una precisión inferior al centavo. Dado que en determinados modelos de impresoras fiscales HASAR no es posible especificar montos con una precisión mayor a dos dígitos decimales, *Cía. HASAR saic* presentó diferentes soluciones a la AFIP con el objeto de poder manejar esta situación bajo las circunstancias ya mencionadas. Este organismo recomienda que todos los contribuyentes afectados apliquen el siguiente método de solución:

- 1) El programa aplicativo deberá calcular el precio total de la venta para el ítem vendido, y enviar a la impresora fiscal HASAR dicho monto como precio unitario, especificando cantidad uno.
- 2) En el campo de descripción del producto se deberá indicar la cantidad real vendida por el precio unitario, incluyendo a continuación los detalles descriptivos del ítem vendido.

En resumen: una venta de varias unidades de un mismo ítem puede ser concretada como venta de un único lote.

Ejemplo: La venta de combustibles

Utilizar el comando **PrintLineItem** del modo que se indica a continuación:

Cantidad	=	1	
Descripción	=	10Lts/\$0.9770 N. Super	
		(se supone precio por litro al público de \$0.9770 y una venta de 10 Lts)	
Monto	=	9.77	
IVA	=	21.00	
Factor K	=	0.9529	(ver cálculo)
Precio	=	T	(precio total)

Cálculo del factor K:

$$\text{Precio Base} = (\text{Precio Final} - (\text{Monto Fijo de Impuesto})) / (1 + (\text{IVA} / 100))$$

Porcentaje de Impuesto = (Monto Fijo de Impuesto / Precio Base por Litro) * 100

Factor K = $1 / (1 + (\text{Porcentaje de Impuesto} / 100))$

Ejemplo: (asumiendo **\$0.0383** de Monto Fijo de Impuesto)

Precio Base = $(0.9770 - 0.0383) / (1 + 0.21) = \mathbf{0.7757}$

Porcentaje de Impuesto = $(0.0383 / 0.7757) * 100 = \mathbf{4.9374}$

Factor K = $1 / (1 + (4.9374 / 100)) = 1 / (1 + 0.049374) = 1 / 1.049374 = \mathbf{0.9529}$

Tener presente que en el Factor K se pueden utilizar hasta un máximo de 8 dígitos decimales al enviar el comando **PrintLineItem**.

Redondeo de decimales

Si, para evitar consultas constantes a la impresora fiscal, el software del punto de venta lleva un registro interno de las operaciones realizadas, es imprescindible que este registro guarde absoluta coherencia con las operaciones de la impresora fiscal. Para ello debe respetarse el algoritmo usado por la impresora fiscal para redondear decimales:

Todos los montos son redondeados a dos decimales teniendo en cuenta la tercera posición decimal. Si ésta es 5 o menor, se trunca directamente luego del segundo decimal. Si es 6 o mayor, previo al truncado se aumenta en uno la segunda posición decimal.

Ejemplos: 5.325 se redondea a 5.32
 5.326 se redondea a 5.33.

Para mayor seguridad, antes de cerrar un comprobante fiscal puede consultarse el monto acumulado a la impresora fiscal, mediante el comando **Subtotal**.

Una vez cerrado el documento, pueden consultarse los montos de IVA, percepciones e impuestos internos del comprobante recién cerrado, mediante los comandos **SendFirstIVA** y **NextIVATransmission**.

El proceso de truncado (y redondeo, de corresponder) descrito se realiza cada vez que es necesario imprimir un monto en el comprobante; y debido a exigencias de la normativa vigente.

Ejemplo: Factura "A"

Suponer la venta de un ítem cuyo precio final es \$100.00- , afectado por una alícuota de IVA del 19% y un 10% de impuesto interno.

Luego la factura impresa tendrá este aspecto:

Cant	Desc	P. Unit.	Alíc. IVA	Precio Neto
1.000	Producto	77.5194	(19.00)	77.52

Notar que el Precio Neto es llevado a dos decimales.

Luego, la discriminación de la factura se vería impresa como:

Subtotal	Alic. IVA	IVAConceptos No Gravados
77.52	(19.00)	14.737.75

TOTAL \$ 100.00

Ahora, si se examina las cuentas puede verse que:

Precio Neto = $100 / (1 + 0.19 + 0.10) = 100 / 1.29 = 77.51937984496$

Como este campo se imprime con cuatro decimales, entonces:

Precio Neto = 77.5194

Siendo el anterior, el valor es el que se imprime en la comuna de "Precio Unitario" en la factura.

Luego:

Monto de IVA = $77.51937984496 * 0.19 = 14.72868217054$

Como este valor se muestra en la columna "IVA" al imprimir la discriminación, se verá: 14.73

Monto de Impuesto Interno = $77.51937984496 * 0.10 = 7.751937984496$

Como este valor se muestra en la columna "Conceptos No Gravados" al imprimir la discriminación, se verá: 7.75

El mismo criterio de los ítems se emplea al realizar descuentos, recargo, pagos, etc.. Las cuentas, internamente, se hacen con todos los decimales y, al momento de imprimir montos, se procede a truncar y redondear a dos dígitos. Esto significa que el total se calcula con todos los decimales pero se trunca y redondea también en el momento de su impresión.

Finalmente, la suma de valores impresos debe coincidir con el total impreso en la factura. De no haber coincidencia es cuando aparece el **Ajuste por B.I** -- ajuste por base imponible --.

Algo análogo sucede con la discriminación. Si la suma de valores impresos en ella no coincide con total impreso, es cuando aparece el **Ajuste por Redondeo**.

Como puede verse, no hay misterios en las cuentas.

Los párrafos anteriores explican como hace las cuentas la impresora fiscal. Ahora veremos como el software del punto de venta puede pedirle a la impresora fiscal la discriminación calculada, para volcarlos al propio sistema; y no tener diferencias entre el software del punto de venta y la impresora fiscal.

Obtener discriminación:

Vía el OCX:

Consulta **activex.pdf** y el manual de comandos de la impresora fiscal para mayor información.

```
.....
HASAR1.AbrirComprobante Fiscal( .... )
.....
HASAR1.Subtotal( )
GuardarInfoSubtotal( HASAR1.Respuesta( 0 ) ) '// Guardar subtotal para comparar después
HASAR1.CerrarComprobanteFiscal( )
.....
HASAR1.PedirPrimerIva( ) '// Se empieza a pedir la discriminación

While ( HASAR1.Respuesta( 1 ) < > "0" ) '// Se pide el resto de la discriminación
    GuardarDiscrim( HASAR1.Respuesta( 0 ) )
    HASAR1.PedirSiguienteIva( )
Wend

SumarInfoDiscrim( ) '// Suma la discriminación del comprobante
n = CompararDiscSubt( ) '// n es la dif. entre el subtotal y la suma anterior
                        '// Si es n < > 0 , entonces el valor de n es el
                        '// Ajuste BI
```

Las rutinas **GuardarInfoSubtotal()**, **GuardarDiscrim()**, **SumarInfoDiscrim()** y **CompararDiscSubt()** son responsabilidad del desarrollador del software para puntos de venta.

Vía strings de comandos:

Consulta el manual de comandos de la impresora fiscal para mayor información.

```
.....
OpenFiscalReceipt
.....
Subtotal
GuardarInfoSubtotal( CampoResp( 0 ) ) // Guardar subtotal para comparar después.
CloseFiscalReceipt
.....
SendFirstIva // Se empieza a pedir la discriminación.
```

```
While ( CampoResp( 1 ) < > "0" )           // Se pide el resto de la discriminación.  
    GuardarDiscrim( CampoResp( 0 ) )  
    NextIvaTransmission  
Wend  
  
SumarInfoGuardada( )           // Suma la discriminación del comprobante  
n = CompararDiscSubt( )       // n es la dif. entre el subtotal y la suma anterior  
                                // Si es n < > 0 , entonces el valor de n es el Ajuste  
                                // BI
```

Las rutinas **GuardarInfoSubtotal()**, **GuardarDiscrim()**, **SumarInfoDiscrim()** y **CompararDiscSubt()** son responsabilidad del desarrollador del software para puntos de venta.

La rutina **CampoResp()** es también responsabilidad del desarrollador, y es equivalente en uso a la propiedad **Respuesta()** del OCX.

Luego del Cierre Diario "Z" se puede hacer algo similar, para que el software del punto de venta no sume todos los comprobantes emitidos (pedir la discriminación del Cierre Diario "Z" no está disponible en todo los modelos de impresoras fiscales) .

Impresoras Fiscales de Página Completa

A diferencia de las impresoras fiscales emisoras de tiques, las impresoras fiscales de página completa no tienen capacidad para responder si un texto se ha logrado imprimir. Es decir, el programa de control fiscal de la impresora fiscal (por indicación del software de gestión del punto de venta) envía una orden de impresión al circuito controlador de impresión, pero no recibe ningún aviso una vez que el texto es efectivamente impreso.

Lo anterior puede originar la siguiente situación: el software de gestión del punto de venta envía a la impresora fiscal el comando de cierre de documento fiscal (factura, recibo fiscal, nota de débito, etc.) que se encuentra abierto. El programa de control fiscal transmite este comando al circuito controlador de impresión y se comienza a imprimir el cierre del documento. Al no tener posibilidad de recibir una respuesta, el programa de control fiscal da por descontado que el documento se ha cerrado (impreso correctamente) y acumula los montos correspondientes en la memoria fiscal. Sin embargo, si antes de terminar la impresión (por ejemplo, antes de imprimirse el logo fiscal), se produce un corte de luz el documento emitido no es legalmente válido, pero los montos respectivos igual se han acumulado. En estas condiciones, el usuario debe retener la factura y emitir una nota de crédito.

Si bien la emisión de la nota de crédito no es responsabilidad obligatoria del software de gestión del punto de venta, se solicita a los desarrolladores de software que incluyan una instrucción similar a la siguiente en los manuales de sus productos:

“En caso de que, luego de cerrarse un comprobante fiscal (factura "A", "B/C", recibo fiscal "A", "B/C", o nota de débito "A", "B/C"), éste no se imprima totalmente, incluyendo la última línea que contiene el logo fiscal, y además no aparezca en el mismo un mensaje indicando que el documento se ha cancelado, debe realizarse el siguiente procedimiento:

1. El comprobante debe ser retenido por el emisor (no debe entregarse al comprador).
2. Debe emitirse un nuevo comprobante fiscal para documentar la compra.
3. Para anular el comprobante incompleto mencionado en el punto 1, debe emitirse una nota de crédito por el monto de dicho documento.”

Cómo aumentar la memoria disponible en DOS bajo Windows

Por defecto y bajo Windows, DOS no utiliza el área de memoria superior. Utilizando dicha área pueden cargarse drivers y residentes en memoria superior, aumentando la cantidad de memoria convencional disponible.

Para ello debe editarse el archivo **config.sys** y colocar al principio del mismo las siguientes líneas:

```
device=c:\windows\himem.sys
device=c:\windows\emm386.exe ram
dos=high
dos=umb
```

Esto hace que aparezca una cierta cantidad de memoria disponible en el área entre 640 Kb y 1 Mb, habitualmente del orden de 90 Kb. Esta memoria está agrupada en uno o más bloques (UMBs) que pueden usarse para cargar drivers y/o residentes.

Si se desea ganar 64 Kb adicionales en el área superior, la línea del **emm386** puede escribirse así:

```
device=c:\windows\emm386.exe ram noems
```

Esto deshabilita el mecanismo de EMS (memoria extendida) y recupera la ventana (frame) de 64 Kb que utiliza, con lo cual la memoria superior disponible subirá a unos 154 Kb. Sin embargo, los programas de DOS que utilicen EMS no funcionarán.

Otra manera un poco más suave de ganar esos 64 Kb es la siguiente:

```
device=c:\windows\emm386.exe ram frame=none
```

Esto mantiene activo el sistema de EMS pero anula su ventana (frame). Algunos programas que utilicen EMS pero no necesiten la ventana (no son muchos) pueden seguir funcionando.

Para cargar un driver en memoria superior, en **config.sys** se debe reemplazar la correspondiente línea:

```
“device=...”
```

por:

```
“devicehigh=...”
```

Obviamente esto sólo puede hacerse después de haberse ejecutado las líneas que se indicaban más arriba.

Para cargar un residente en memoria superior, en **autoexec.bat** debe anteponerse al nombre del programa residente el comando **loadhigh** o su abreviatura **lh**.

Por ejemplo, si antes decía “**lptfis.exe**” ahora debe decir “**loadhigh lptfis.exe**”.

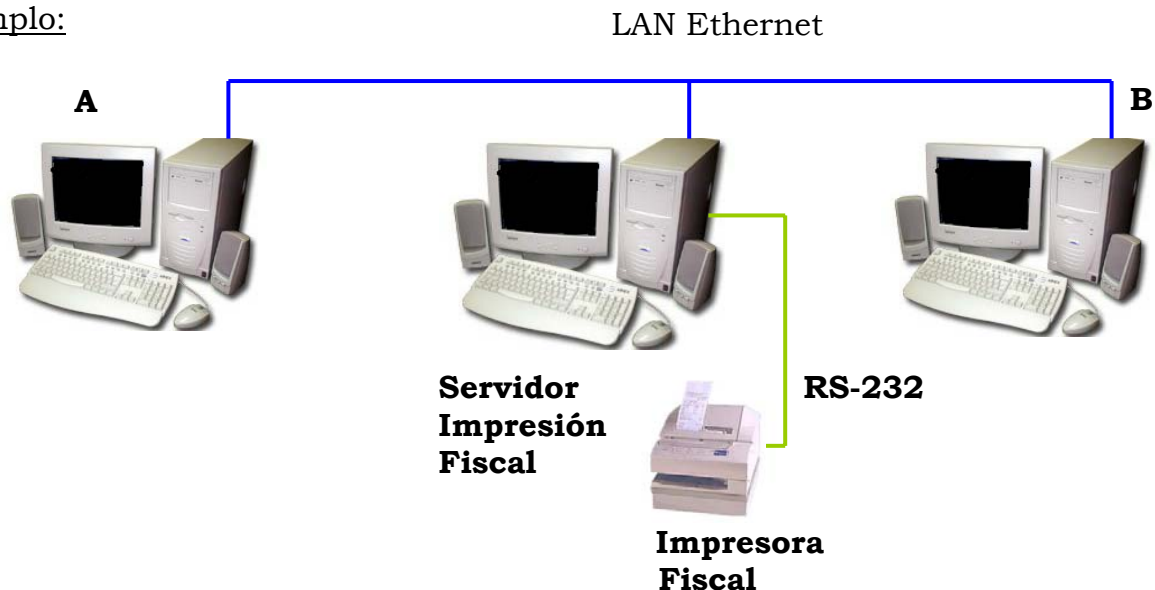
Impresoras Fiscales y Redes

Ante reiteradas consultas sobre la incorporación y uso de las impresoras fiscales en redes, se hace necesario aclarar lo siguiente:

- Desde el punto de vista técnico es viable la incorporación y uso de impresoras fiscales sobre redes.
- Las restricciones que existen son de tipo legal y se desprenden de las Resoluciones Fiscales AFIP, en vigencia, que regulan el uso de las impresoras fiscales.

En consecuencia, la implementación y uso de las impresoras fiscales sobre una red será posible siempre y cuando **“se garantice la concomitancia entre la captura de datos y su impresión, a través de la red”**. Es decir, si existen varios puestos de facturación y una sola impresora fiscal en la red, entonces, mientras un puesto factura el resto debe estar inhibido para esa misma tarea.

Ejemplo:



Para el caso del esquema anterior, si el puesto de facturación **A** está emitiendo un comprobante fiscal en la impresora fiscal, en el puesto de facturación **B** debe inhibirse la emisión de comprobantes fiscales. Ello, debido a que la obligación de respetar la **concomitancia** impide armar toda la factura en forma previa, y luego enviarla a impresión.

Para los casos donde toda la facturación se realice en un momento determinado (del día, de la semana o del mes) se deberá solicitar directamente a la AFIP la autorización escrita correspondiente para la implementación de la impresión diferida (no concomitante).

Para los casos donde el rubro comercial se halla claramente especificado en la Resolución Fiscal, autorizándoseles el uso de la impresión diferida (no concomitante), pueden incorporar en la red las impresoras fiscales obviando los comentarios de este documento, dado que técnicamente ello es posible.

Nota para los Desarrolladores de Software

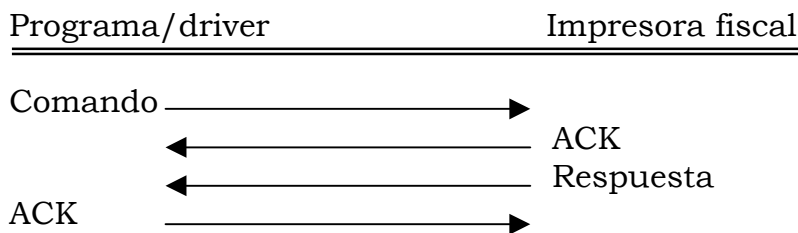
Como tales, deben tener presente que firmarán una declaración jurada (formulario **445/E**) donde se deja plena constancia de que el software utilizado para facturar cumple con las normas fiscales en vigencia (léase Resoluciones Fiscales AFIP). En consecuencia, se convierten en posibles acreedores de una sanción por parte de la AFIP, si correspondiese la aplicación de la misma.

Protocolo Fiscal

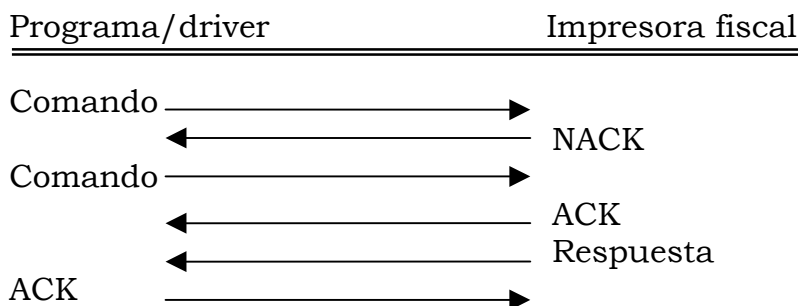
Cía HASAR no da soporte al desarrollo de drivers desarrollados por los usuarios; si en cambio da soporte al uso y comportamiento de drivers/librerías/utilitarios distribuidos por *Cía. HASAR*, y sobre el uso y comportamiento de comandos fiscales y la impresora fiscal.

Sin embargo, si el usuario opta por el desarrollo de un driver propio, el protocolo fiscal se describe más ampliamente en el manual **seri.....pdf** correspondiente a la impresora fiscal en uso. Este documento es solo un acercamiento a dicho protocolo.

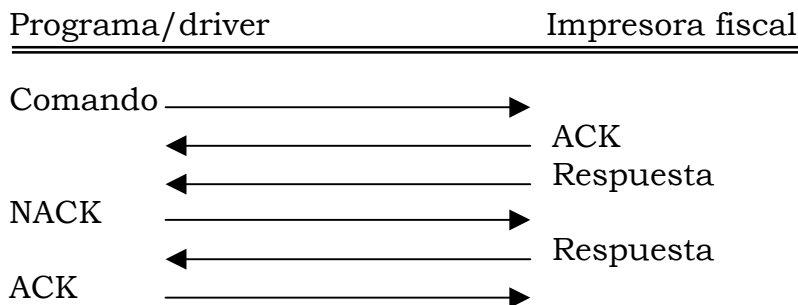
Caso 1: General e ideal



Caso 2: La impresora fiscal no entendió el string de comando

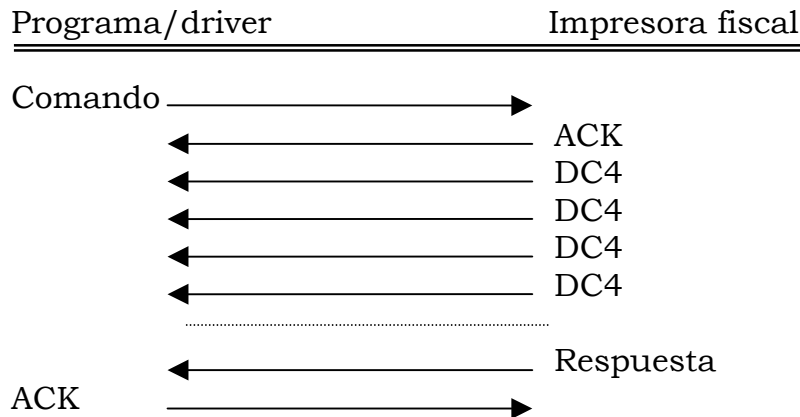


Caso 3: El programa/driver no entendió la respuesta



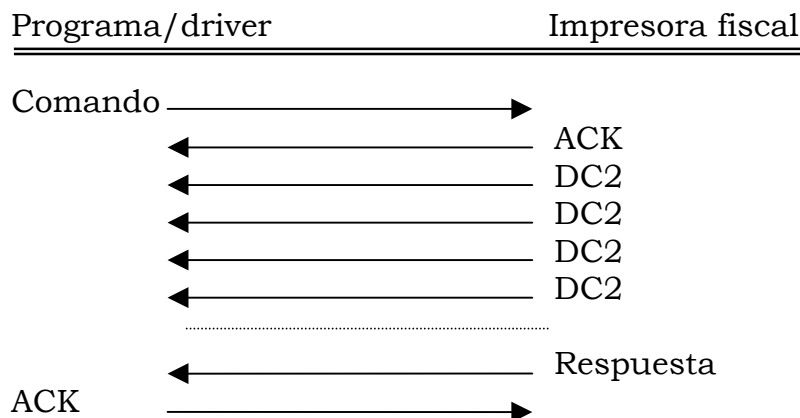
Caso 4: Falta papel

Los caracteres DC4 siguen siendo enviados hasta que se normaliza la situación.



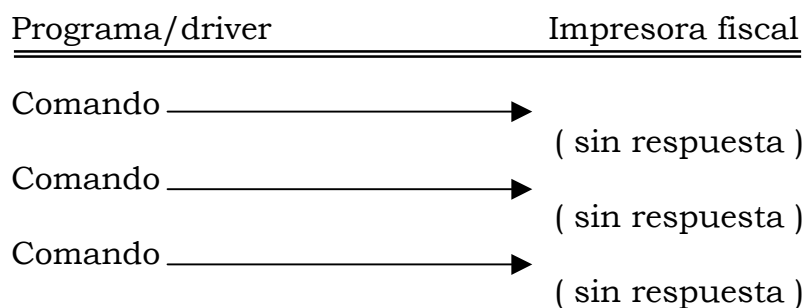
Caso 5: Keep Alive

Los caracteres DC2 indican al programa que espere, que aún no terminó de procesar el comando recibido.



Caso 6: Timeout

La cantidad de reintentos queda a su criterio.



Formato del paquete que viaja:

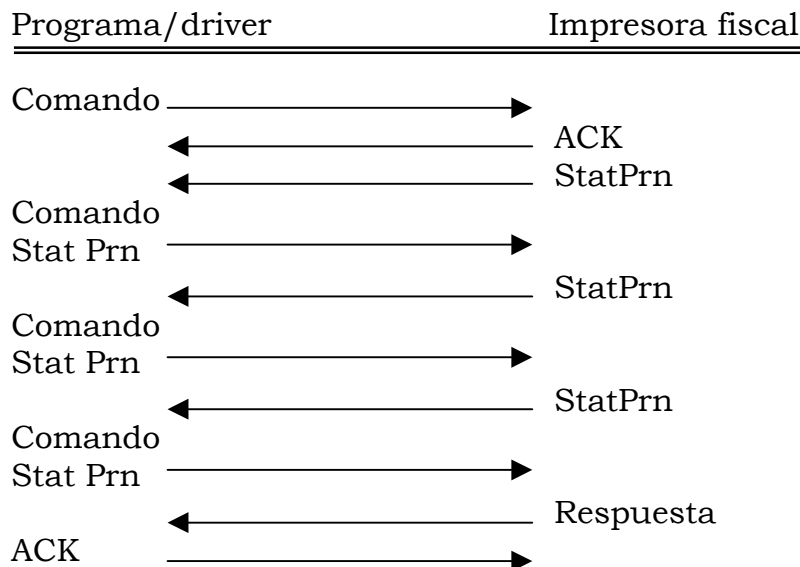
STX	Nro.	ESC	Comando/Respuesta	ETX	Checksum
-----	------	-----	-------------------	-----	----------

Con el Nro. de secuencia del paquete considerar que:

- No debe coincidir con el número del último paquete respondido por la impresora fiscal (sino ésta entrega la respuesta del comando anterior).
- Al llegar a 7fh se debe comenzar en 20h otra vez.
- Si la respuesta coincide en nro de paquete, con el comando enviado, se debe verificar que el caracter de ID de comando (incluido en la respuesta) coincida con el caracter de ID del comando enviado. De lo contrario, incrementar nro de paquete y reenviar comando.

El caracter ESC, previo al string de comando solamente se usa (opcional) para que el protocolo permita consultas de estado intermedio, que no son posibles en el protocolo sin ESC.

Ejemplo:



En el zip de drivers fiscales se incluyen los fuentes de la librería para lenguaje “C”, que pueden tomarse como referencia.