# PROTOCOLO BINARIO
# NUEVA GAMA
# Julio 2008
# Ver. 13

Balanza Ver.: K.00G

**1. PROTOCOLO BINARIO ..... 1**

1.1 DESCRIPCIÓN DE UN MENSAJE DE PROTOCOLO BINARIO ..... 1
*1.1.1 Cabecera del mensaje ..... 1*
*1.1.2 Cuerpo del mensaje ..... 2*
*1.1.3 Fin del mensaje ..... 2*
*1.1.4 Campo CRC ..... 2*
*1.1.5 Rutina de ajuste del mensaje al protocolo binario ..... 3*

**2. ACCESO AL MOTOR BASE DE DATOS (IDF:000001) ..... 4**

2.1 ESTRUCTURA DE UN MENSAJE DE ACCESO AL MOTOR ..... 4
*2.1.1 Campo función - respuesta ..... 5*
*2.1.2 Campo método ..... 6*
*2.1.3 Campo funmask ..... 6*
*2.1.4 Campo Key – Llave de acceso principal ..... 7*
*2.1.5 Campo rKey – Llave de acceso extendido o Llave de retorno ..... 7*
*2.1.6 Suma de llaves ..... 7*
*2.1.7 Campo NDATA ..... 7*
*2.1.8 Campo datos ..... 8*
2.2 EJEMPLOS ..... 9
*2.2.1 Programación de Elementos de una base de datos: ..... 9*
2.2.1.1 Programación de todos los Items: ..... 9
2.2.1.2 Programación de un conjunto de Items: ..... 12
2.2.1.3 Programación de una base de datos indexada: ..... 14
2.2.1.4 Ejemplo de Programación de un artículo completo: ..... 15
*2.2.2 Mensaje de Respuesta a una programación: ..... 17*
*2.2.3 Consulta de Elementos de una base de datos: ..... 18*
2.2.3.1 Consulta de todos los Items RKEY=KEYALL: ..... 18
2.2.3.2 Consulta de un conjunto de Items: ..... 21
2.2.3.3 Consulta de totales de artículos. ..... 23
2.2.3.4 Consulta de una Base de Datos Autoindexada ..... 24
2.2.3.5 Consultar Siguiente Elemento ..... 27
2.3 BASES DE DATOS ..... 29
*2.3.1 Clientes ..... 30*
*2.3.2 Países: ..... 31*
*2.3.3 Cambios de precio ..... 32*
*2.3.4 Ingredientes (Fuera de uso): ..... 33*
*2.3.5 Taras codificadas: ..... 34*
*2.3.6 Mercancías (Autoindexada) ..... 34*
*2.3.7 Ivas (Autoindexada): ..... 34*
*2.3.8 Artículos + Totales ..... 35*
*2.3.9 Descriptivos (Encadenada): ..... 46*
*2.3.10 Textos: cabeceras, leyendas, textos libres y publicidad (Autoindexada): ..... 46*
*2.3.11 Códigos de barras (Autoindexada): ..... 47*
*2.3.12 Vendedores + Totales: ..... 48*
*2.3.13 Teclas de vendedores: ..... 52*
*2.3.14 Pedidos: ..... 52*
*2.3.15 Vacuno (autoindexada): ..... 53*
*2.3.16 Datos envasado de artículo (Encadenada) ..... 55*

**3. PETICION DE TIQUETS ..... 56**

3.1 PETICION DE FORMATO (IDF:000003) ..... 56
*3.1.1 Formato de la pregunta ..... 56*
*3.1.2 Formato de la respuesta ..... 56*
3.2 PETICION DE TICKETS (IDF:000002) ..... 58
*3.2.1 Formato de la pregunta ..... 58*
*3.2.2 Formato de la respuesta. ..... 60*
3.3 RESPUESTA ERROR (IDF:000100) ..... 61

**4. ESTADO DEL SISTEMA (IDF:000004) ..... 62**

4.1 MENSAJE DE PETICION DE ESTADO DEL SISTEMA ..... 62
4.2 RESPUESTA A LA PETICION DE ESTADO DEL SISTEMA ..... 62
4.3 MENSAJE DE RESPUESTA ERROR (IDF:000100) ..... 62

**5. MENSAJES DE CONTROL DE IDIOMAS (IDF:000005) ..... 63**

**6. CONFIGURACION DE BALANZAS (IDF:000008) ..... 63**

**7. ANEXOS ..... 64**

7.1 ANEXO 1: STRING CODIGO DE BARRAS ..... 64
7.2 ANEXO 2: RUTINAS CONVERSIÓN FECHA ..... 65

# 1. PROTOCOLO BINARIO

## 1.1 DESCRIPCIÓN DE UN MENSAJE DE PROTOCOLO BINARIO

Un mensaje de protocolo binario consta de 3 partes : la cabecera del mensaje, el cuerpo del mensaje (datos), y el fin de mensaje.

### 1.1.1 Cabecera del mensaje

| Campo | Longitud | Descripción | | |
|---|---|---|---|---|
| STX | 1 byte | 2d, 02h | | |
| IDG | 1 byte | Identificador de protocolo binario<br>126d, 7Eh → Datos sin compresión<br>127d, 7Fh → Datos con compresión LZW ***(no implementado)*** | | |
| DST: Flags | 4 bits | Flag | Descripción | |
| | | 0x0<br>0x1<br>0x2 | Por Defecto<br>No backup *(retorna error si la balanza destino es la de backup)*<br>Siguiente *(busca la siguiente balanza conectada a la red)* | |
| DST: Type | 4 bits | Id | Descripción | Contenido de Dst |
| | | 0<br>1<br>2<br>3<br>4<br>5 | Equipo conectado<br>Mostrador<br>Balanza<br>Sistema<br>Master de sistema<br>Backup | NULL<br>Número de mostrador<br>Número de balanza<br>NULL<br>NULL<br>NULL |
| DST: Dest | 1 byte | Destino (0,1-63) *(binario)* *(0 Por Defecto)* | | |
| XCHG CR | 1 byte | Sustituto del CR *(binario)* | | |
| LON | 2 bytes | Longitud de los datos *(binario)* | | |
| IDF | 3 bytes | Identificador de función *(binario)* | | |

**Destino**: Si en este campo se indica 0 la comunicación se realiza sobre la balanza conectada al canal serie.

Si se indica un número diferente de 0 la comunicación se realiza con la balanza indicada, a través de la balanza conectada al canal serie. Es decir la balanza conectada re-envía la orden a la balanza indicada en este campo.

**Campo XCHG CR**

Campo indicador del byte empleado para sustituir el carácter de control 0x0D *(EOM, fin de mensaje)*. El carácter de control EOM solo debe aparecer al final del mensaje, por tanto debe buscarse entre los campos STX y CRC ambos incluidos. El valor de este campo no puede ser ni 0 *(NULL)* ni 0x0D *(EOM)*. El byte de sustitución será aplicado entre los campos **LON** y **DATOS** ambos incluidos.

### 1.1.2 Cuerpo del mensaje

El cuerpo del mensaje depende del Identificador de función del mensaje, según sea un mensaje de Acceso a Bases de Datos o a otros servicios de la Balanza.

El cuerpo del mensaje se describe detalladamente el los siguientes capítulos.

### 1.1.3 Fin del mensaje

| Campo | Longitud | Descripción | |
|---|---|---|---|
| CRC | 2 bytes | CRC de todo el mensaje | *(binario)* |
| EOM | 1 byte | 13d, 0Dh | *(evento de final de mensaje)* |

### 1.1.4 Campo CRC

El CRC se calcula desde el primer byte **(STX)** hasta el final del campo **DATOS** (longitud del mensaje menos 3 bytes). Se debe realizar sobre el mensaje original, sin sustituir los caracteres 0x0D (EOM).

Para el cálculo del CRC se tendrán en cuenta los siguientes puntos:

- El campo XCHG CR deberá ser 0 (NULL).
- El CRC no incluirá los campos CRC ni EOM.
- Si alguno de los bytes del resultado (o ambos) corresponde al carácter EOM, se sustituirá por el carácter 0x0E.

*Rutina en C para el cálculo del CRC de 16 bits:*

```
ushort Calc_CRC (uchar *Data, short Size)
{
    ushort r=0, a, b;
    short t;

    for (t=0; t<Size; t++)
    {
        r+=(t*Data[t])+t;
    }

    a=r>>8;   if (a==0x0D) a=0x0E;
    b=r&0xFF; if (b==0x0D) b=0x0E;

    a<<=8;

    return a|b;
}
```

### 1.1.5 Rutina de ajuste del mensaje al protocolo binario

Esta rutina calcula el CRC, busca un carácter que no aparezca en el mensaje y lo pone en el campo XCHCR, y se encarga de recorrer el mensaje y sustituir todos los valores 0x0D=CR por el valor del campo XCHCR.

```
void AdjBinary(char *Data, short Size)
{
    char   DataKey[256];
    short  t;
    ushort Crc, v;
    uchar  Xchg_CR=0;

    Data[1]=0x7E;     // IDG: Identificador de protocolo binario standard
    Data[4]=NULL;     // XCHG CR: Sustituto del carácter 0x0D

    Crc=Calc_CRC ((uchar*)Data, (short)(Size-3));    // Calcular CRC

    v=(uchar)(Crc>>8);   Data[Size-2]=v;    // Monta la parte alta del CRC en el mensaje
    v=(uchar)(Crc&0xFF); Data[Size-3]=v;    // Monta la parte baja del CRC en el mensaje

    memset (DataKey, 0, sizeof(DataKey));  // Tabla para localizar el campo XCHG CR

    for (t=0; t<Size-1; t++)
    {
        DataKey[Data[t]]=1;    // Marca los bytes usados en el mensaje
    }

    for (t=0; t<256; t++)     // Busca el carácter XCHG CR
    {
        if (!DataKey[t] && t!=0x0D)
        {
            Xchg_CR=(uchar)t;      // Asigna el carácter encontrado al campo XCHG CR
            Data[4]=Xchg_CR;       // Almacena el campo en el array de datos

            for (t=0; t<Size-3; t++)
            {
                // Sustituye el carácter EOM por el campo XCHG CR en todo el mensaje
                if (Data[t]==0x0D) Data[t]=Xchg_CR;
            }

            return;
        }
    }

    // Error de coherencia del mensaje!
}
```

# 2. ACCESO AL MOTOR BASE DE DATOS (IDF:000001)

## 2.1 ESTRUCTURA DE UN MENSAJE DE ACCESO AL MOTOR

| Campo | Longitud | Descripción |
|---|---|---|
| FUN – RSP | 1 byte | Función o respuesta a función |
| MET | 2 bytes | Método |
| FUNMASK | 1 byte | Siempre 0 |
| DBID | 1 byte | Identificador de la base de datos |
| KEY | 1 byte | Llave principal de acceso |
| RKEY | 1 byte | Llave extendida de acceso |

***RKEY!=0xFF***

| Campo | Lon | Descripción |
|---|---|---|
| NDTA | 1 byte | Nº de bloques de datos |
| DTA | ..... | Bloques de datos |

***RKEY==0xFF***

| Campo | Lon | Descripción |
|---|---|---|
| NDTA | 1 byte | Nº de bloques de datos |
| NITEMS | 1 byte | Nº de ítems de datos |
| ITEMS | NITEMS | Id's a los ítems de datos |
| DTA | ..... | Bloques de datos |

**Bloque de datos** ***(campo DTA)***

| Campo | Lon | Descripción |
|---|---|---|
| Indice | 2 bytes | Indice directo al contenido de la base de datos<br>*Solo si la base de datos es autoindexada* |
| Datos | ..... | Datos almacenados consecutivamente según la estructura descrita en el apartado ***"campo datos".***<br>La longitud dependerá de la llave utilizada siendo variable si esta es 0xFF. |

### 2.1.1 Campo función - respuesta

Este campo se utiliza para diferentes propósitos según si es un mensaje de Función ó de Respuesta. También, si es un respuesta de Error se indica en este campo el código del mismo.

Veamos las diferentes Funciones y respuestas.

**Campo como Función:**

| Función | Valor | Descripción |
|---|---|---|
| CONSULTAR | 0x00 | Realiza una consulta |
| PROGRAMAR | 0x03 | Inserta o modifica una entrada |
| BORRAR | 0x06 | Marca una entrada como borrada |

**Campo como Respuesta:**

| Función | Valor | Descripción |
|---|---|---|
| CONSULTAR_OK | 0x01 | Respuesta OK a la función CONSULTAR |
| CONSULTAR_KO | 0x02 | Respuesta KO a la función CONSULTAR |
| PROGRAMAR_OK | 0x04 | Respuesta OK a la función PROGRAMAR |
| PROGRAMAR_KO | 0x05 | Respuesta KO a la función PROGRAMAR |
| BORRAR_OK | 0x07 | Respuesta OK a la función BORRAR |
| BORRAR_KO | 0x08 | Respuesta KO a la función BORRAR |
| BORRAR TODO_OK | 0x0A | Respuesta OK a la función BORRAR_TODO |
| BORRAR TODO_KO | 0x0B | Respuesta KO a la función BORRAR_TODO |
| ACUMULAR_OK | 0x0D | Respuesta OK a la función ACUMULAR |
| ACUMULAR_KO | 0x0E | Respuesta KO a la función ACUMULAR |
| MARCAR_OK | 0x10 | Respuesta OK a la función MARCAR |
| MARCAR_KO | 0x11 | Respuesta KO a la función MARCAR |
| DESMARCAR_OK | 0x13 | Respuesta OK a la función DESMARCAR |
| DESMARCAR_KO | 0x14 | Respuesta KO a la función DESMARCAR |
| SUMAR_OK | 0x16 | Respuesta OK a la función SUMAR |
| SUMAR_KO | 0x17 | Respuesta KO a la función SUMAR |
| RESTAR_OK | 0x19 | Respuesta OK a la función RESTAR |
| RESTAR_KO | 0x1A | Respuesta KO a la función RESTAR |

**Campo como Código de Error:**

| Códigos de error | Valor | Descripción |
|---|---|---|
| ERR_DST | 0x64 | Destino incorrecto |
| ERR_ERROR | 0x65 | Error no identificado |
| ERR_MSG_KO | 0x66 | Función no reconocida |
| ERR_DATA | 0x67 | Datos erróneos |
| ERR_IDF | 0x68 | Identificador de función desconocido |
| ERR_TIMEOUT | 0x69 | Timeout |

### 2.1.2 Campo método

El campo METODO sirve para posicionar-nos dentro de la base de datos sobre el registro que queremos trabajar. Para ello combinaremos el campo Método (Condición que se debe cumplir) con el campo KEY (Items que deben cumplir la condición).

Veamos los distintos valores que puede tomar este campo:

| Método | Valor | Descripción |
|---|---|---|
| NULO | 0x0000 | Método nulo |
| Métodos de búsqueda de primer nivel *(combinables con las de segundo nivel)* | | |
| IGUAL | 0x0001 | Búsqueda por igual que... |
| MAYOR | 0x0002 | Búsqueda por mayor que.... |
| MAYORIGUAL | 0x0003 | Búsqueda por mayor o igual que... |
| MENOR | 0x0004 | Búsqueda por menor que... |
| MENORIGUAL | 0x0005 | Búsqueda por menor o igual que... |
| PRIMERO | 0x0006 | Búsqueda por el primero o el menor |
| ULTIMO | 0x0007 | Búsqueda por el ultimo o el mayor |
| PARECIDO | 0x0008 | Búsqueda por el más parecido |
| RKEY_ZERO | 0x0009 | Busca entradas con la llave extendida con campos nulos |
| RKEY_NOTZERO | 0x000A | Busca entradas con la llave entendida con campos no nulos |
| Métodos de búsqueda de segundo nivel *(combinables con las de primer nivel)* | | |
| L2_ZERO | 0x0010 | Busca entradas con la llave extendida con campos nulos |
| L2_NOTZERO | 0x0020 | Busca entradas con la llave entendida con campos no nulos |
| L2_IGUAL | 0x0030 | Búsqueda por igual según la llave extendida |
| L2_NOIGUAL | 0x0040 | Búsqueda por diferente a la llave extendida |
| Flags añadidos a los métodos de búsqueda | | |
| GROUP | 0x0200 | Procesa entradas hasta fin de condición |
| MARCADO | 0x0400 | Procesa entradas marcadas |
| NO_MARCADO | 0x0800 | Procesa entradas no marcadas |

### 2.1.3 Campo funmask

Está previsto para usos futuros, en esta versión de protocolo debe ser siempre 0.

| Método | Valor | Descripción |
|---|---|---|
| Acceso directo | 0x80 | Efectúa la operación deseada directamente sobre el motor de bases de datos, evitando al servidor y perdiendo por tanto las propiedades de red del sistema. |
| Reservado | 0x40 | Debe ser 0 |
| Reservado | 0x20 | Debe ser 0 |
| Reservado | 0x10 | Debe ser 0 |
| Reservado | 0x08 | Debe ser 0 |
| Reservado | 0x04 | Debe ser 0 |
| Reservado | 0x02 | Debe ser 0 |
| Reservado | 0x01 | Debe ser 0 |

### 2.1.4 Campo Key – Llave de acceso principal

Identificador de la llave de acceso principal. Esta llave de acceso define normalmente los campos de la condición principal. En el Capitulo 2.3 se describen detalladamente las diferentes claves definidas en cada Base de Datos.

### 2.1.5 Campo rKey – Llave de acceso extendido o Llave de retorno

Identificador de la llave de acceso extendida. Los identificadores se muestran a partir del apartado **"Bases de datos"** en el campo **Id**. Esta llave puede ser usada como complemento de la búsqueda principal, modificación parcial de entradas de la base de datos, consulta de campos protegidos, etc. . El uso mas común es en programaciones este campo contiene la Llave de Acceso extendida: Llave con los ítems de la Base de Datos que queremos modificar, y en consulta la Llave de Retorno: Llave con los Items que queremos consultar.

### 2.1.6 Suma de llaves

Las llaves se combinan formando una única llave, eliminando del resultado los campos repetidos. La suma se efectúa copiando los campos de la primera llave y a continuación añadiendo los campos de la segunda teniendo en cuenta que estos no estén repetidos, tal y como se muestra en la siguiente tabla:

**Ejemplo:**

Suma de las llaves *ID_ART_SEARCH_PLUMOS* y *ID_ART_UPDATE_STDPLU* de la base de datos de artículos:

| Llave principal | Llave extendida | Llave suma |
|---|---|---|
| 2 → Código del mostrador | 0 → Código | 2 → Código del mostrador |
| 1 → Número del PLU | 1 → Número del PLU | 1 → Número del PLU |
| | 2 → Código del mostrador | 0 → Código |
| | 3 → Familia | 3 → Familia |
| | 6 → Forma de venta | 6 → Forma de venta |
| | 4 → Precio | 4 → Precio |
| | 5 → Descriptivo | 5 → Descriptivo |
| | 9 → Tara | 9 → Tara |

### 2.1.7 Campo NDATA

El valor del campo NDATA está fijado siempre a '1' ya que de momento solo podemos acceder a una sola posición de la Base de Datos.

### 2.1.8 Campo datos

Los datos son almacenados secuencialmente siguiendo el patrón obtenido por las llaves de acceso, el cual viene determinado por el valor de la llave extendida o la suma de sus contenidos. La estructura de cada uno de los campos viene determinada por el tipo de elemento al que este asociado en el esqueleto de la base de datos, definida con detalle en el capitulo 2.3.

**Veamos el tamaño de los datos según el tipo de datos:**

| Tipo | Longitud | Descripción |
|---|---|---|
| BOOL | 1 byte | Campo booleano |
| BIT | (1–8) 1 byte<br>(9-16) 2 bytes<br>(17-24) 3 bytes<br>(25-32) 4 bytes | Campo de bits con signo |
| UBIT | (1–8) 1 byte<br>(9-16) 2 bytes<br>(17-24) 3 bytes<br>(25-32) 4 bytes | Campo de bits sin signo |
| CHAR o BYTE | 1 byte | Campo de 1 byte con signo |
| UCHAR o UBYTE | 1 byte | Campo de 1 byte sin signo |
| SHORT o WORD | 2 bytes | Campo de 2 bytes con signo |
| USHORT o UWORD | 2 bytes | Campo de 2 bytes sin signo |
| TRIBYTE | 3 bytes | Campo de 3 bytes con signo |
| UTRIBYTE | 3 bytes | Campo de 3 bytes sin signo |
| LONG o DWORD | 4 bytes | Campo de 4 bytes con signo |
| ULONG o UDWORD | 4 bytes | Campo de 4 bytes sin signo |
| STRING<br>MEMO | 2 bytes + StrSize | Campo Memo o String<br>Los dos primeros bytes indican la longitud del string, seguidos del bloque de datos. Se envia todo el string tal como esta en la base de datos, con nulos detras hasta llenar el tamaño del string. |
| POINTER | La del campo apuntado | Campo apuntador a otro campo mediante un LINK |

## 2.2 Ejemplos

### 2.2.1 Programación de Elementos de una base de datos:

Lo primero que tenemos que hacer para programar una posición de la base de datos, es situarnos sobre la posición deseada. Para ello usamos el campo KEY o Llave principal de Acceso, que combinado con el campo METODO nos dará la posición que queremos programar. La mayoría de las bases de datos tienen una clave principal de un solo Item, que suele ser un código (Código de cliente, Código de Articulo, Código de vendedor…etc.), de manera que si queremos programar una posición pondremos en el campo KEY la clave que contenga solo el Item Código, y en el campo MET el método IGUAL (0x01).

Si queremos insertar un elemento nuevo en la Base de Datos siempre debemos usar la Llave Extendida RKEY=0xFF=KEYALL, indicando que se deben programar todos los ítems, mientras que para modificar una posición ya existente podemos usar como RKEY una Llave que contenga únicamente los Items que se desean modificar.

#### 2.2.1.1 Programación de todos los Items:

Para programar todos los Items de una posición de la base de datos pondremos en el campo RKEY o Llave Extendida el valor 0xFF (KEYALL). Esto será necesario siempre que se programa un elemento nuevo en la Base de Datos.

No programa los datos protegidos.

A continuación pondremos en el campo NITEMS el numero de Items de la base de datos.

**Tabla con las bases de datos, su identificador (independiente versión) y el nº. de items (según versión programa) de cada una de ellas.**

| BASE DE DATOS | IBD | NITEMS |
|---|---|---|
| Clientes | 1 | 2 |
| Países | 2 | 2 |
| Cambios de Precio | 3 | 5 |
| Ingredientes (fuera de uso) | 4 | 2 |
| Publicidad (fuera de uso) | 5 | 1 |
| Taras Codificadas | 6 | 2 |
| Mercancías | 7 | 4 |
| Ivas | 8 | 1 |
| Artículos + Totales | 9 | 37 * |
| Descriptivos | 10 | 1 |
| Textos | 11 | 2 |
| Códigos de Barras | 12 | 1 |
| Vendedores + Totales | 13 | 24 * |
| Teclas de Vendedor | 14 | 3 |
| No utilizada | 15 | |
| Pedidos | 16 | 4 |
| PlaceHolder Campos (reservada) | 17 | |
| Formatos (reservada) | 18 | |
| Envasado | 19 | |
| Vacuno | 20 | 10 |

**El caso de Artículos no los programaremos todos, solo lo referente al articulo (15 o 25).*

*Artículos : 15 referentes al Articulo + 10 Descriptivos + 12 referentes a Totales (protegidos).*
**El Caso de los vendedores es similar al de los artículos. Tenemos 3 ítems referentes al vendedor + 21 referentes a Totales (protegidos).*

Seguidamente pondremos los identificadores de los Items que queremos programar, en la mayoría de las bases de datos los pondremos todos.

**EJEMPLO:**

**Pondremos como ejemplo la Base de Datos de Pedidos, que tiene 4 Items:**

| Campo | Id | Modo | Tipo | Tamaño |
|---|---|---|---|---|
| Código | 0 | Normal | ULONG | 4 bytes |
| Pedido | 1 | Normal | LONG | 4 bytes |
| Forma de venta | 2 | Normal | UBIT | 1 bit |
| Descriptivo | 3 | Normal | STRING | 20 bytes |

(Ver capitulo 2.3.15)

El Item = 1 correspondiente al Código del Articulo será la Clave Principal de Acceso, y será la Clave 1 (ID_PEDIDO_INSERT) de esta Base de Datos, que contiene solo el Código de Articulo.

Como programaremos todos los elementos la Clave Extendida será 0xFF.

NOTA: Normalmente antes de programar un pedido de un articulo, lo consultaremos, para conocer la Forma de Venta y el Descriptivo de ese articulo, para que sea igual en la Base de Datos de Pedidos que en la de Articulos.

**El mensaje que enviaremos para programar el pedido será el siguiente**

| Campo | Contenido | Descripción |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Identificador del Protocolo Binario |
| DST | 0x0000 | Al equipo conectado |
| XCHG CR | ??? | Se calcula al final, será un byte que no aparezca en el mensaje |
| LON | ??? | Se calcula al final, será la longitud total del mensaje |
| IDF | 0x010000 | Id del protocolo de Acceso al Motor de Bases de Datos |
| | | |
| FUN – RSP | 0x03 | Función PROGRAMAR |
| MET | 0x0100 | Método IGUAL |
| FUNMASK | 0x00 | |
| DBID | 0x10 | Identificador de la base de datos PEDIDOS = 16 |
| KEY | 0x01 | Llave principal de acceso ID_PEDIDO_INSERT |
| RKEY | 0xFF | Llave extendida de acceso |
| NDATA | 0x1 | Programamos una sola posición de la Base de Datos |
| NITEMS | 0x04 | 4 Items |
| ITEM | 0x00 | ITEM Código de Articulo |
| ITEM | 0x01 | ITEM cantidad de Pedido |
| ITEM | 0x02 | ITEM Forma de Venta |
| ITEM | 0x03 | ITEM Descriptivo |
| DATO | 0x01000000 | Dato correspondiente al Código del Articulo (4Bytes) |
| DATO | 0xD0070000 | Dato correspondiente al Pedido=2000 (4Bytes) |
| DATO | 0x00 | Dato correspondiente a la Forma de Venta (1Byte) |
| DATO | 0x1400+”Patatas…..” | Dato correspondiente al Item Descriptivo (2+20Bytes) |
| | | |

| | | |
|---|---|---|
| CRC | ??? | CRC Calculado según se ha descrito en el capitulo 1.2.2 |
| EOM | 0x0D | EOM |

Si queremos programar la base de datos de artículos podemos poner la RKEY 0xFF y según si queremos programar el Articulo o algún total, pondremos el conjunto de ítems que queremos. También lo podemos hacer poniendo en la RKEY una llave que contenga los ítems que queremos programar, tal como se describe en el siguiente punto.

Lo mismo pasa con los vendedores.

#### 2.2.1.2 Programación de un conjunto de Items:

Cada Base de Datos tiene un conjunto de claves, donde cada clave contiene un conjunto diferente de ítems de esa Base de Datos. La mayoría de las Bases de Datos tienen pocos ítems (como mucho los 5 de la BD de Cambios de Precio),de manera que nos resultara fácil programar todos los ítems a la vez, aunque también ofrecemos claves para programar solo un conjunto de ítems, según nos convenga.

Sin embargo en el caso de los Artículos o de los Vendedores veremos que nos serán muy útiles las claves que solo contengan los items que queremos programar.

Al poner en la RKEY la Llave Extendida con los items que queremos programar, ya no es necesario indicar el numero de Items ni los Items que vamos a Introducir a continuación, ya que esta información va implícita en Llave Extendida, de manera que a continuación de la RKEY pondremos los datos. Veamos un Ejemplo:

**EJEMPLO:**

Por Ejemplo nos puede interesar programar un articulo, y para ello no necesitaríamos programar los totales del articulo. Para este propósito tenemos la Llave ID_ART_UPDATE_ART definida con los items correspondientes al articulo *(Código, PLU, Mostrador, Familia, Fventa, Iva, Precio, Descriptivo, Tara, Caducidad, Consumo Preferente, Etiqueta, N.Preselección, CodigoBarras, Formato CodigoBarras)*, entre muchas otras. (Ver Capitulo 2.3.9)

**Supongamos que queremos programar el siguiente articulo:**

| Campo | Contenido |
|---|---|
| Código | 1234 |
| Número de PLU | 1234 |
| Código de mostrador | 1 |
| Familia | 20 |
| Precio | 100 |
| Descriptivo | “Patatas” |
| Forma de venta | 0 NO PESADO |
| Caducidad | 0 |
| Consumo preferente | 0 |
| Tara | 0 |
| Número de etiqueta | 0xFF POR DEFECTO |
| Tipo de IVA | 0 |
| Formato del código de barras | 0 |
| Código de barras | “” |
| Número de paquete preseleccionado | 0 |

El orden de los datos debe ser el mismo que el descrito en la Clave que usamos como clave Extendida, que no es el mismo que el orden de definición de los Items.

Como Clave Principal de Acceso utilizaremos la Clave ID_ART_INSERT_COD que es la clave 1 . (Ver Capitulo 2.3.9)

**De manera que el mensaje para programar este articulo sería el siguiente:**

| Campo | Contenido | Descripción |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Identificador del Protocolo Binario |
| DST | 0x0000 | Al equipo conectado |
| XCHG CR | ??? | Se calcula al final, será un byte que no aparezca en el mensaje |
| LON | ??? | Se calcula al final, será la longitud total del mensaje |
| IDF | 0x010000 | Id del protocolo de Acceso al Motor de Bases de Datos |
| | | |
| FUN – RSP | 0x03 | Función PROGRAMAR |
| MET | 0x0100 | Método IGUAL |
| FUNMASK | 0x00 | |
| DBID | 0x09 | Identificador de la base de datos ARTICULOS |
| KEY | 0x01 | Llave principal de acceso ID_ART_INSERT_COD (1) |
| RKEY | 0x15 | Llave extendida de acceso ID_ART_UPDATE_ART (21) |
| NDATA | 0x1 | Programamos una sola posición de la Base de Datos |
| DATO | 0xD2040000 | Dato correspondiente al Item Código Articulo (4Bytes) |
| DATO | 0D204 | Dato correspondiente al Item PLU (2Bytes) |
| DATO | 0x01 | Dato correspondiente al Item Mostrador (1Byte) |
| DATO | 0x04 | Dato correspondiente al Item Familia (1Byte) |
| DATO | 0x00 | Dato correspondiente al Item FormaVenta (1Byte) |
| DATO | 0x00 | Dato correspondiente al Item Tipo Iva (1Byte) |
| DATO | 0x640000 | Dato correspondiente al Item Precio (3Bytes) |
| DATO | 0x1400+”Patatas…..” | Dato correspondiente al Item Descriptivo (2+20Bytes) |
| DATO | 0x0000 | Dato correspondiente al Item Tara (2Bytes) |
| DATO | 0x00 | Dato correspondiente al Item Caducidad (1Byte) |
| DATO | 0x00 | Dato correspondiente al Item Consumo Preferente(1Byte) |
| DATO | 0xFF | Dato correspondiente al Item Etiqueta =Por Defecto (1Byte) |
| DATO | 0x00 | Dato correspondiente al Item Preselección(1Bytes) |
| DATO | 0x0D00+”…” | Dato correspondiente al Item Código Barras(2+13Bytes) |
| DATO | 0x00 | Dato correspondiente al Item Formato Código Barras (1Byte) |
| | | |
| CRC | ??? | CRC Calculado según se ha descrito en el capitulo 1.2.2 |
| EOM | 0x0D | EOM |

#### 2.2.1.3 Programación de una base de datos indexada:

La programación de una Base de Datos Autoindexada se hace exactamente igual que las descritas en el primer ejemplo, con la peculiaridad de que estas Bases de Datos no tienen Claves de Acceso, solo tienen definida una clave que es la 0x01, por eso en la clave principal de Acceso siempre se pone un 0x01, y en la Clave extendida se pone siempre 0xFF, de manera que al igual que en el primer ejemplo debemos especificar el numero de Items y los identificadores de los Items que se van a Programar.

Inmediatamente después de los identificadores de los items, y antes de introducir los datos pondremos dos bytes correspondientes al índice de la base de datos, que nos apuntara directamente al Numero de Elemento a programar.

Veamos como Ejemplo como se programaría el Código de Barras para el modo envasado en Red 1.

**EJEMPLO:**

En la Base de Datos de Códigos de Barras Tenemos un único Item: el string con el codigo de barras. Supongamos que queremos programar el Código de Barras para el modo envasado en Red 1, este se corresponde con el índice 6 de esa Base de Datos. (Ver Capitulo 2.3.12).

Supongamos que el código de barras que queremos programar son 4 digitos para el peso 'W', 4 digitos para el precio 'Z', y 4 digitos para el importe 'I'. El string del codigo de barras resultante seria el siguiente: 'WWWWZZZZIIII' (Ver anexo 1).

**El mensaje que enviaremos para programar este Código de Barras será el siguiente**

| Campo | Contenido | Descripción |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Identificador del Protocolo Binario |
| DST | 0x0000 | Al equipo conectado |
| XCHG CR | ??? | Se calcula al final, será un byte que no aparezca en el mensaje |
| LON | ??? | Se calcula al final, será la longitud total del mensaje |
| IDF | 0x010000 | Id del protocolo de Acceso al Motor de Bases de Datos |
| | | |
| FUN – RSP | 0x03 | Función PROGRAMAR |
| MET | 0x0100 | Método IGUAL |
| FUNMASK | 0x00 | |
| DBID | 0x0C | Identificador de la base de datos COD. BARRAS = 12 |
| KEY | 0x01 | Llave principal de acceso AUTOINDEXADA |
| RKEY | 0xFF | Llave extendida de acceso |
| NDATA | 0x1 | Programamos una sola posición de la Base de Datos |
| NITEMS | 0x01 | 1 Item |
| ITEM | 0x00 | ITEM Código de Barras |
| INDICE | 0x06 | INDICE = 6 (Séptima posición de la Base de Datos) |
| DATO | 0x0D00+"WWWWZZZZIIII" | Dato correspondiente al Código de Barras (2+13Bytes) |
| | | |
| CRC | ??? | CRC Calculado según se ha descrito en el capitulo 1.2.2 |
| EOM | 0x0D | EOM |

#### 2.2.1.4 Ejemplo de Programación de un artículo completo:

***Datos del artículo:***

| Campo | Contenido |
|---|---|
| Código | 1234 |
| Número de PLU | 1234 |
| Código de mostrador | 1 |
| Familia | 20 |
| Precio | 100 |
| Descriptivo | “Patatas” |
| Forma de venta | 0 |
| Caducidad | 0 |
| Consumo preferente | 0 |
| Tara | 0 |
| Número de etiqueta | 0xFF |
| Tipo de IVA | 0 |
| Formato del código de barras | 0 |
| Código de barras | “” |
| Número de paquete preseleccionado | 0 |
| Descriptivo 1 | “Bolsa de 5Kg” |
| Descriptivo 2 | “Blancas” |
| Descriptivo 3 | “” |
| Descriptivo 4 | “” |
| Descriptivo 5 | “” |
| Descriptivo 6 | “” |
| Descriptivo 7 | “” |
| Descriptivo 8 | “” |
| Descriptivo 9 | “” |
| Descriptivo 10 | “” |

**Mensaje de programación de un artículo:**

| | Campo | Dato | Descripción |
|---|---|---|---|
| | STX | 0x02 | STX |
| | IDG | 0x7E | Identificador de protocolo binario |
| | DST | 0x0000 | Destino *(al equipo conectado)* |
| | XCHG CR | ??? | Sustituto del CR |
| | LON | ??? | Longitud de los datos |
| | IDF | 0x010000 | Id del protocolo de acceso al motor de bases de datos |
| | | | |
| | FUN | 0x03 | Función programar |
| | MET | 0x0100 | Igual |
| | FUNMASK | 0x00 | Funciones especiales |
| | DBID | 0x09 | Base de datos de artículos |
| | KEY | 0x01 | Llave de inserción *(contiene el código)* |
| | RKEY | 0xFF | Llave extendida... Acceso a todos los campos |
| | NDTA | 0x01 | 1 bloque de datos |
| **?** | NITEMS | 0x19 ó 0x09 | Número de ítems |
| | ID ITEMS | 0x00 | Identificador del campo “código” |
| | ID ITEMS | 0x01 | Identificador del campo “Número de PLU” |
| | ID ITEMS | 0x02 | Identificador del campo “Código de mostrador” |
| | ID ITEMS | 0x03 | Identificador del campo “Familia” |
| | ID ITEMS | 0x04 | Identificador del campo “Precio” |
| | ID ITEMS | 0x05 | Identificador del campo “Descriptivo” |

**Mensaje de programación de un artículo (continuación):**

| | | | |
|---|---|---|---|
| | ID ITEMS | 0x06 | Identificador del campo “Forma de venta” |
| | ID ITEMS | 0x07 | Identificador del campo “Caducidad” |
| | ID ITEMS | 0x08 | Identificador del campo “Consumo preferente” |
| | ID ITEMS | 0x09 | Identificador del campo “Tara” |
| | ID ITEMS | 0x0A | Identificador del campo “Número de etiqueta” |
| | ID ITEMS | 0x0B | Identificador del campo “Tipo de IVA” |
| | ID ITEMS | 0x0C | Identificador del campo “Formato del código de barras” |
| | ID ITEMS | 0x0D | Identificador del campo “Código de barras” |
| | ID ITEMS | 0x0E | Identificador del campo “Nº de paquete preseleccionado” |
| | ID ITEMS | 0x25 | Identificador del campo “Descriptivo 1” |
| | ID ITEMS | 0x26 | Identificador del campo “Descriptivo 2” |
| | ID ITEMS | 0x27 | Identificador del campo “Descriptivo 3” |
| | ID ITEMS | 0x28 | Identificador del campo “Descriptivo 4” |
| | ID ITEMS | 0x29 | Identificador del campo “Descriptivo 5” |
| | ID ITEMS | 0x2A | Identificador del campo “Descriptivo 6” |
| | ID ITEMS | 0x2B | Identificador del campo “Descriptivo 7” |
| | ID ITEMS | 0x2C | Identificador del campo “Descriptivo 8” |
| | ID ITEMS | 0x2D | Identificador del campo “Descriptivo 9” |
| | ID ITEMS | 0x2E | Identificador del campo “Descriptivo 10” |
| | ITEMS | 0xD2040000 | Código |
| | ITEMS | 0xD204 | Número de PLU |
| | ITEMS | 0x01 | Código de mostrador |
| | ITEMS | 0x14 | Familia |
| | ITEMS | 0x640000 | Precio |
| **X** | ITEMS | 0x1400 + “Patatas...” | Descriptivo |
| | ITEMS | 0x00 | Forma de venta |
| | ITEMS | 0x0000 | Caducidad |
| | ITEMS | 0x0000 | Consumo preferente |
| | ITEMS | 0x0000 | Tara |
| | ITEMS | 0xFF | Número de etiqueta |
| | ITEMS | 0x00 | Tipo de IVA |
| | ITEMS | 0x00 | Formato del código de barras |
| **X** | ITEMS | 0x0D00 + “...” | Código de barras |
| | ITEMS | 0x00 | Número de paquete preseleccionado |
| **X** | ITEMS | 0x1400 + “Bolsa de 5Kg...” | Descriptivo 1 |
| **X** | ITEMS | 0x1400 + “Blancas...” | Descriptivo 2 |
| **X** | ITEMS | 0x1400 + “...” | Descriptivo 3 |
| **X** | ITEMS | 0x1400 + “...” | Descriptivo 4 |
| **X** | ITEMS | 0x1400 + “...” | Descriptivo 5 |
| **X** | ITEMS | 0x1400 + “...” | Descriptivo 6 |
| **X** | ITEMS | 0x1400 + “...” | Descriptivo 7 |
| **X** | ITEMS | 0x1400 + “...” | Descriptivo 8 |
| **X** | ITEMS | 0x1400 + “...” | Descriptivo 9 |
| **X** | ITEMS | 0x1400 + “...” | Descriptivo 10 |
| | | | |
| | CRC | ??? | CRC |
| | EOM | 0x0D | EOM |

***Leyenda de la tabla anterior:***

| | |
|---|---|
| | Elementos no necesarios para la programación |
| **?** | El contenido de este campo depende de los elementos en el mensaje |
| **X** | Campos de tipo string<br>· Los 2 primeros bytes indican la longitud de la cadena posterior<br>· Los puntos suspensivos indican que debe completarse la cadena hasta la longitud indicada |

### 2.2.2 Mensaje de Respuesta a una programación:

El mensaje de respuesta a una programación nos indica en el campo RESP el resultado de la programación, que suele ser un 0x04 *(Programación OK)*, o un 0x05 *(Programación NOK)*, a no ser que el mensaje de programación estuviera mal estructurado y nos responda con otro tipo de error.

Los campos MET y FUNMASK no tienen significado alguno, y suelen estar a 0.

El campo DBIBD tiene el Identificador de la Base de datos que hemos programado.

En el caso que la programación no fuera correcta los valores de los otros campos no tiene ningún significado. Pero en el caso de que si fuera correcta nos responde en el campo KEY la Clave de Inserción que hemos utilizado, que es el mismo valor que hemos puesto

en el mensaje de programación. Lo mismo pasa con el Campo RKEY.

El campo NDATA contendrá un 0x00 indicando que no hay datos.

**El mensaje de respuesta del ejemplo anterior sería el siguiente:**

| Campo | Dato | Descripción |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x07E | Identificador de protocolo binario |
| DST | 0x0000 | Destino |
| XCHG CR | ??? | Sustituto del CR |
| LON | ??? | Longitud de los datos |
| IDF | 0x010000 | Id del protocolo de acceso al motor de bases de datos |
| | | |
| RESP | 0x04 | Programación OK |
| MET | 0x0000 | NULO |
| FUNMASK | 0x00 | NULO |
| DBID | 0x09 | Base de datos de artículos |
| KEY | 0x01 | Llave de inserción |
| RKEY | 0xFF | Llave extendida... Acceso a todos los campos |
| NDTA | 0x00 | 0 bloques de datos |
| | | |
| CRC | ??? | CRC |
| EOM | 0x0D | EOM |

### 2.2.3 Consulta de Elementos de una base de datos:

La consulta de elementos de una base de datos se hace exactamente igual que la programación. Es decir con el campo KEY y el campo MET seleccionaremos que posición de la Base de Datos queremos consultar. Con el campo RKEY Llave de Retorno, seleccionaremos que ítems de la Base de Datos queremos consultar.

#### 2.2.3.1 Consulta de todos los Items RKEY=KEYALL:

Al igual que en la programación, si queremos consultar todos los ítems de una posición de la Base de Datos, pondremos en el campo RKEY el valor 0xFF indicando que queremos consultar todos los ítems de la posición seleccionada *(0xFF=KEYALL=Todos)*.

A continuación pondremos el numero de ítems y los identificadores de los ítems pertenecientes a la Llave principal de Acceso, seguidos de los datos, (solo será necesario poner los datos correspondientes a los ítems que pertenecen a la Llave principal de Acceso).

**NOTA:** La consulta con KEY_ALL nos responde los elementos que físicamente se encuentran en la Base de Datos. Los elementos que son Links a otras Bases de Datos, como por ejemplos los Descriptivos del Articulo se deben consultar con una RKEY que contenga estos elementos (Ejemplo: ID_ART_UPDATE_DESCR).

Los elementos guardados en la Base de Datos como “PROTEGIDOS”, tampoco se consultan con KEYALL.

Veamos un ejemplo:

**Ejemplo:**

Supongamos que queremos consultar los pedidos del articulo 2. La base de datos de pedido tiene cuatro Items: Codigo Articulo, Pedido, FormaVenta y Descriptivo *(Ver capitulo 2.3.15)*.

Usaremos como Llave Principal de Acceso la Llave ID_PEDIDO_INSERT (0x01), que contiene únicamente el Código de Articulo.

**El mensaje que enviaremos será el siguiente:**

| Campo | Contenido | Descripción |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Identificador del Protocolo Binario |
| DST | 0x0000 | Al equipo conectado |
| XCHG CR | ??? | Se calcula al final, será un byte que no aparezca en el mensaje |
| LON | ??? | Se calcula al final, será la longitud total del mensaje |
| IDF | 0x010000 | Id del protocolo de Acceso al Motor de Bases de Datos |
| | | |
| FUN – RSP | 0x00 | Función CONSULTAR |
| MET | 0x0100 | Método IGUAL |
| FUNMASK | 0x00 | |
| DBID | 0x10 | Identificador de la base de datos PEDIDOS = 16 |
| KEY | 0x01 | Llave principal de acceso ID_PEDIDO_INSER |
| RKEY | 0xFF | Llave extendida de acceso KEYALL |
| NDATA | 0x1 | Consultamos una sola posición de la Base de Datos |
| NITEMS | 0x01 | 1 Items |
| ITEM | 0x00 | ITEM Código de Articulo |
| DATO | 0x02000000 | Dato correspondiente al Código de Articulo consultado (4Bytes) |
| | | |
| CRC | ??? | CRC Calculado según se ha descrito en el capitulo 1.2.2 |
| EOM | 0x0D | EOM |

**Respuesta:**

Al realizar una consulta con la Llave de Retorno KEYALL, obtendremos como resultado un mensaje con todos los ítems de la Base de Datos. Para saber que Items son estos nos da el campo NITEMS, y la lista de Items con los Identificadores de los Items que nos contesta, de manera que conociendo estos Identificadores sabemos el orden de respuesta de los Items y sabemos el tamaño de cada Item para recoger los Datos. Los Items no tienen porque seguir ningún orden, por eso la recogida de los datos no es trivial. Si la Consulta se realiza con una Llave de retorno diferente a KEYALL, entonces no nos da el numero de ítems ni la lista de Identificadores de los Items, pero si que contesta todos los datos correspondientes a los items pertenecientes a esa Llave (aún que sean nulos), y en el orden especificado por la Llave, lo que nos facilita mucho la recogida de resultados.

**Una posible respuesta a la consulta anterior sería la siguiente:**

| Campo | Contenido | Descripción |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Identificador del Protocolo Binario |
| DST | 0x0000 | Al equipo conectado |
| XCHG CR | ??? | Se calcula al final, será un byte que no aparezca en el mensaje |
| LON | ??? | Se calcula al final, será la longitud total del mensaje |
| IDF | 0x010000 | Id del protocolo de Acceso al Motor de Bases de Datos |
| | | |
| FUN – RSP | 0x01 | Respuesta CONSULTA_OK |
| MET | 0x0100 | Método IGUAL |
| FUNMASK | 0x00 | |
| DBID | 0x10 | Identificador de la base de datos PEDIDOS = 16 |
| KEY | 0x01 | Llave principal de acceso ID_PEDIDO_INSER |
| NDATA | 0x1 | Nos devuelve una sola posición de la Base de Datos |
| RKEY | 0xFF | Llave extendida de acceso KEYALL |
| NITEMS | 0x04 | 4 Items |
| ITEM | 0x00 | ITEM Código de Articulo |
| ITEM | 0x01 | ITEM Pedido |
| ITEM | 0x02 | ITEM Forma de Venta |
| ITEM | 0x03 | ITEM Descriptivo del Articulo |
| DATO | 0x02000000 | Dato correspondiente al Código de Articulo consultado (4Bytes) |
| DATO | 0x0A000000 | Dato correspondiente al Pedido (4Bytes) |
| DATO | 0x00 | Dato correspondiente a la Forma de Venta (Pesado) (1Bytes) |
| DATO | 0x1400+"Patatas…" | Dato correspondiente al Descriptivo del Articulo (2+20Bytes) |
| | | |
| CRC | ??? | CRC Calculado según se ha descrito en el capitulo 1.2.2 |
| EOM | 0x0D | EOM |

Interpretación del resultado:

Código Articulo: 2
Pedido: 10
Forma de Venta: Pesado
Descriptivo: Patatas

#### 2.2.3.2 Consulta de un conjunto de Items:

Al igual que en la programación, si queremos consultar un conjunto de ítems de una posición de la Base de Datos, pondremos en el campo RKEY el valor correspondiente a una Llave que contenga exclusivamente los Items que queremos consultar.

A continuación pondremos el numero de ítems y los identificadores de los ítems pertenecientes a la Llave principal de Acceso (campo KEY), seguidos de los datos correspondientes a estos ítems.

Veamos un ejemplo:

### Ejemplo:

Supongamos que queremos consultar únicamente el Nombre del Vendedor con código 2, de la Base de Datos de Vendedores. Para ello utilizaremos como Llave principal de Acceso (KEY), la Llave ID_INSERT_VND (0x01), que contiene únicamente el Código del vendedor, y como llave de retorno (RKEY) la llave ID_UPDATE_VENDEDOR (0x0B), que contiene el Código del vendedor (Item 1), y el nombre del Vendedor (Item 0) por este orden concreto. *(Ver Capitulo 2.3.13)*

**El mensaje que enviaremos será el siguiente:**

| Campo | Contenido | Descripción |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Identificador del Protocolo Binario |
| DST | 0x0000 | Al equipo conectado |
| XCHG CR | ??? | Se calcula al final, será un byte que no aparezca en el mensaje |
| LON | ??? | Se calcula al final, será la longitud total del mensaje |
| IDF | 0x010000 | Id del protocolo de Acceso al Motor de Bases de Datos |
| | | |
| FUN – RSP | 0x00 | Función CONSULTAR |
| MET | 0x0100 | Método IGUAL |
| FUNMASK | 0x00 | |
| DBID | 0x13 | Identificador de la base de datos VENDEDORES = 13 |
| KEY | 0x01 | Llave principal de acceso ID_ INSERT_VND |
| RKEY | 0x0B | Llave extendida de acceso ID_UPDATE_VENDEDOR |
| NDATA | 0x1 | Consultamos una sola posición de la Base de Datos |
| DATO | 0x02000000 | Dato correspondiente al Código de Vendedor consultado (4 Bytes) |
| | | |
| CRC | ??? | CRC Calculado según se ha descrito en el capitulo 1.2.2 |
| EOM | 0x0D | EOM |

### Respuesta:

**El mensaje de respuesta de la consulta anterior podría ser este:**

| Campo | Contenido | Descripción |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Identificador del Protocolo Binario |
| DST | 0x0000 | Al equipo conectado |
| XCHG CR | ??? | Se calcula al final, será un byte que no aparezca en el mensaje |
| LON | ??? | Se calcula al final, será la longitud total del mensaje |
| IDF | 0x010000 | Id del protocolo de Acceso al Motor de Bases de Datos |
| | | |

| | | |
|---|---|---|
| FUN – RSP | 0x01 | Respuesta CONSULTA_OK |
| MET | 0x0100 | Método IGUAL |
| FUNMASK | 0x00 | |
| DBID | 0x13 | Identificador de la base de datos VENDEDORES = 13 |
| KEY | 0x01 | Llave principal de acceso ID_ INSERT_VND |
| RKEY | 0x0B | Llave extendida de acceso ID_UPDATE_VENDEDOR |
| NDATA | 0x1 | Nos devuelve una sola posición de la Base de Datos |
| DATO | 0x02000000 | Dato correspondiente al Código de Vendedor consultado (4 Bytes) |
| DATO | 0x02+"Pedro Díaz" | Dato correspondiente al nombre del Vendedor (2+20Bytes) |
| | | |
| CRC | ??? | CRC Calculado según se ha descrito en el capitulo 1.2.2 |
| EOM | 0x0D | EOM |

En este caso la extracción de resultados es mas fácil porque no tenemos el numero de Items ni la Lista de Identificadores de Items, sino que directamente tenemos los datos correspondientes a los Items pertenecientes a la Llave de Retorno.

Interpretación del resultado:

Vendedor: 2
Nombre: Pedro Díaz

#### 2.2.3.3 Consulta de totales de artículos.

Se pretende consultar el siguiente artículo con totales NO nulos. El primer artículo se obtiene solicitando a partir del código 0.

Se piden todos los totales: los totales en red (venta y envasado) se deben pedir a la balanza servidora, los totales locales a cada una de las balanzas.

**El mensaje que enviaremos será el siguiente:**

| Campo | Contenido | Descripción |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Identificador del Protocolo Binario |
| DST | 0x0000 | Al equipo conectado |
| XCHG CR | ??? | Se calcula al final, será un byte que no aparezca en el mensaje |
| LON | ??? | Se calcula al final, será la longitud total del mensaje |
| IDF | 0x010000 | Id del protocolo de Acceso al Motor de Bases de Datos |
| | | |
| FUN – RSP | 0x00 | Función CONSULTAR |
| MET | 0x0022 | Método MAYOR QUE + L2_NOTZERO |
| FUNMASK | 0x00 | |
| DBID | 0x09 | Identificador de la base de datos ARTICULO |
| KEY | 0x01 | Llave principal de acceso ID_ART_INSER_COD |
| RKEY | 0x19 | Llave extendida de acceso ID_ ART_BORRAR_TOTAL (Totales) |
| NDTA | 0x01 | 1 Items |
| DATOS | 0x00 | Dato correspondiente al Código de Articulo consultado (4Bytes) |
| | | |
| CRC | ??? | CRC Calculado según se ha descrito en el capitulo 1.2.2 |
| EOM | 0x0D | EOM |

En las siguientes peticiones en el campo DATO (código de artículo) debe utilizarse el último valor obtenido de la balanza.

**Respuesta:**

**Una posible respuesta a la consulta anterior sería la siguiente:**

| Campo | Contenido | Descripción |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Identificador del Protocolo Binario |
| DST | 0x0000 | Al equipo conectado |
| XCHG CR | ??? | Se calcula al final, será un byte que no aparezca en el mensaje |
| LON | ??? | Se calcula al final, será la longitud total del mensaje |
| IDF | 0x010000 | Id del protocolo de Acceso al Motor de Bases de Datos |
| | | |
| FUN – RSP | 0x00 | Función CONSULTAR |
| MET | 0x0022\| | Método MAYOR QUE + L2_NOTZERO |
| FUNMASK | 0x00 | |
| DBID | 0x09 | Identificador de la base de datos ARTICULO |
| KEY | 0x01 | Llave principal de acceso ID_ART_INSER_COD |
| RKEY | 0x19 | Llave extendida de acceso ID_ ART_BORRAR_TOTAL (Totales) |
| NDTA | 0x01 | 1 Items |
| DATOS | 0x00 | Dato correspondiente al Código de Articulo consultado (4Bytes) |
| DATO | 0x16 | Dato correspondiente al Código de Articulo (4Bytes) |
| DATO | 0x04 | Nº operaciones en modo venta (2Bytes). |
| DATO | 0x00 | Nº unidades en modo venta (4Bytes) |
| DATO | 0x638 | Importe en modo venta (4Bytes). |
| DATO | 0x992 | Peso en modo venta (4Bytes) |
| DATO | 0x00 | Nº operaciones en local (2Bytes). |
| DATO | 0x00 | Nº unidades en local (4Bytes) |
| DATO | 0x00 | Importe en local (4Bytes). |
| DATO | 0x00 | Peso en local (4Bytes) |
| DATO | 0x00 | Nº operaciones en envasado (2Bytes). |
| DATO | 0x00 | Nº unidades en envasado (4Bytes) |
| DATO | 0x00 | Importe en envasado (4Bytes). |
| DATO | 0x00 | Peso en envasado (4Bytes) |
| | | |
| CRC | ??? | CRC Calculado según se ha descrito en el capitulo 1.2.2 |
| EOM | 0x0D | EOM |

Interpretación del resultado:
Código Articulo: 38
Sólo hay totales en venta: 4 operaciones, 2450gr de peso y 1592 Ptas.

El tipo de totales en local lo determina la forma de trabajo de la balanza (ver mensaje ESTADO DEL SISTEMA).

El tren de etiquetaje guarda los totales como “venta”.

#### 2.2.3.4 Consulta de una Base de Datos Autoindexada

La consulta a los elementos de una Base de Datos Autoindexada se hace de manera similar a la programación. Podemos hacer consultas a todos los Items de una posición RKEY=KEYALL, o a un conjunto de Items RKEY!=KEYALL igual que hemos hecho en los ejemplos anteriores, con la diferencia que al ser Autoindexada la

clave principal de Acceso siempre es 1 (KEY=0x01), y debemos especificar el Indice del elemento de la Base de Datos que queremos consultar. Normalmente consultaremos todo de manera que usaremos RKEY=KEYALL, y como no necesitamos pasarle ningún dato pondremos en el campo NITEMS un 0, y seguidamente pondremos el Indice.

Veamos un ejemplo:

**Ejemplo:**

Supongamos que queremos consultar la segunda línea de las cabeceras, que es el Indice 2 de la base de datos de Texto. Esta base de datos es autoindexada y tiene 2 Items el Item=0 que es el texto y el Item=1 que es el Tipo de Letra *(Ver Capitulo 2.3.11)*

**El Mensaje de consulta sería el siguiente:**

| Campo | Contenido | Descripción |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Identificador del Protocolo Binario |
| DST | 0x0000 | Al equipo conectado |
| XCHG CR | ??? | Se calcula al final, será un byte que no aparezca en el mensaje |
| LON | ??? | Se calcula al final, será la longitud total del mensaje |
| IDF | 0x010000 | Id del protocolo de Acceso al Motor de Bases de Datos |
| | | |
| FUN – RSP | 0x00 | Función CONSULTAR |
| MET | 0x0100 | Método IGUAL |
| FUNMASK | 0x00 | |
| DBID | 0x11 | Identificador de la base de datos TEXTOS = 11 |
| KEY | 0x01 | Llave principal de acceso AUTOINDEXADA |
| RKEY | 0xFF | Llave extendida de acceso KEYALL |
| NDATA | 0x1 | Consultamos una sola posición de la Base de Datos |
| NITEMS | 0x00 | 0 Items |
| INDICE | 0x0200 | Indice en la Base de Datos (2ª posición) |
| | | |
| CRC | ??? | CRC Calculado según se ha descrito en el capitulo 1.2.2 |
| EOM | 0x0D | EOM |

**Respuesta:**

**El mensaje de respuesta de la consulta anterior podría ser este:**

| Campo | Contenido | Descripción |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Identificador del Protocolo Binario |
| DST | 0x0000 | Al equipo conectado |
| XCHG CR | ??? | Se calcula al final, será un byte que no aparezca en el mensaje |
| LON | ??? | Se calcula al final, será la longitud total del mensaje |
| IDF | 0x010000 | Id del protocolo de Acceso al Motor de Bases de Datos |
| | | |
| FUN – RSP | 0x01 | Respuesta CONSULTA_OK |
| MET | 0x0100 | Método IGUAL |
| FUNMASK | 0x00 | |
| DBID | 0x11 | Identificador de la base de datos TEXTOS = 11 |
| KEY | 0x01 | Llave principal de acceso AUTOINDEXADA |
| RKEY | 0xFF | Llave extendida de acceso KEYALL |
| NDATA | 0x1 | Nos devuelve una sola posición de la Base de Datos |
| NITEMS | 0x02 | 2 Items |
| ITEM | 0 | Texto |
| ITEM | 1 | Tipo Letra |
| INDICE | 0x0200 | Indice en la Base de Datos (2ª posición) |
| DATO | 0x1900+"Cabecera2" | Dato correspondiente al texto (2+20Bytes) |
| DATO | 0x0A | Dato correspondiente al Tipo de Letra (1Byte) |
| | | |
| CRC | ??? | CRC Calculado según se ha descrito en el capitulo 1.2.2 |
| EOM | 0x0D | EOM |

Interpretación del resultado:

Posición 2 de la BD de Texto:
Texto: Cabecera2
Tipo de Letra: 10

#### 2.2.3.5 Consultar Siguiente Elemento

Supongamos que queremos hacer un listado de una Base de Datos, para ello consultaremos los elementos uno a uno y los iremos listando. Para este propósito, consultamos primero el primer elemento con el campo método MAYORIGUAL, y los sucesivos con el campo método MAYOR.

**Ejemplo:**

Supongamos que queremos consultar el pedido siguiente del articulo 2. La base de datos de pedido tiene cuatro Items: Codigo Articulo, Pedido, FormaVenta y Descriptivo *(Ver capitulo 2.3.15).*

Usaremos como Llave Principal de Acceso la Llave ID_PEDIDO_INSERT (0x01), que contiene únicamente el Código de Articulo.

**El mensaje que enviaremos será el siguiente:**

| Campo | Contenido | Descripción |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Identificador del Protocolo Binario |
| DST | 0x0000 | Al equipo conectado |
| XCHG CR | ??? | Se calcula al final, será un byte que no aparezca en el mensaje |
| LON | ??? | Se calcula al final, será la longitud total del mensaje |
| IDF | 0x010000 | Id del protocolo de Acceso al Motor de Bases de Datos |
| | | |
| FUN – RSP | 0x00 | Función CONSULTAR |
| MET | 0x0200 | Método MAYOR |
| FUNMASK | 0x00 | |
| DBID | 0x10 | Identificador de la base de datos PEDIDOS = 16 |
| KEY | 0x01 | Llave principal de acceso ID_PEDIDO_INSER |
| RKEY | 0xFF | Llave extendida de acceso KEYALL |
| NDATA | 0x1 | Consultamos una sola posición de la Base de Datos |
| NITEMS | 0x01 | 1 Items |
| ITEM | 0x00 | ITEM Código de Articulo |
| DATO | 0x02000000 | Dato correspondiente al Código de Articulo consultado (4Bytes) |
| | | |
| CRC | ??? | CRC Calculado según se ha descrito en el capitulo 1.2.2 |
| EOM | 0x0D | EOM |

**Respuesta:**

**Una posible respuesta a la consulta anterior sería la siguiente:**

| Campo | Contenido | Descripción |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Identificador del Protocolo Binario |
| DST | 0x0000 | Al equipo conectado |
| XCHG CR | ??? | Se calcula al final, será un byte que no aparezca en el mensaje |
| LON | ??? | Se calcula al final, será la longitud total del mensaje |
| IDF | 0x010000 | Id del protocolo de Acceso al Motor de Bases de Datos |
| | | |
| FUN – RSP | 0x01 | Respuesta CONSULTA_OK |
| MET | 0x0200 | Método MAYOR |
| FUNMASK | 0x00 | |
| DBID | 0x10 | Identificador de la base de datos PEDIDOS = 16 |
| KEY | 0x01 | Llave principal de acceso ID_PEDIDO_INSER |
| NDATA | 0x1 | Nos devuelve una sola posición de la Base de Datos |
| RKEY | 0xFF | Llave extendida de acceso KEYALL |
| NITEMS | 0x04 | 4 Items |
| ITEM | 0x00 | ITEM Código de Articulo |
| ITEM | 0x01 | ITEM Pedido |
| ITEM | 0x02 | ITEM Forma de Venta |
| ITEM | 0x03 | ITEM Descriptivo del Articulo |
| DATO | 0x03000000 | Dato correspondiente al Código de Articulo (4Bytes) |
| DATO | 0x0F000000 | Dato correspondiente al Pedido (4Bytes) |
| DATO | 0x00 | Dato correspondiente a la Forma de Venta (Pesado) (1Bytes) |
| DATO | 0x1400+"Pistachos.." | Dato correspondiente al Descriptivo del Articulo (2+20Bytes) |
| | | |
| CRC | ??? | CRC Calculado según se ha descrito en el capitulo 1.2.2 |
| EOM | 0x0D | EOM |

Interpretación del resultado:

Código Articulo: 3
Pedido: 15
Forma de Venta: Pesado
Descriptivo: Pistachos

Si queremos implementar un listado de Pedidos correcto, una vez consultado el siguiente pedido deberemos comprobar que el pedido no sea cero, ya que en este caso no seria necesario imprimirlo en el listado.

## 2.3 Bases de datos

| Base de datos | Id | Modalidad | Tipo de comunicaciones |
|---|---|---|---|
| Clientes | 1 | Normal | Copia de sistema |
| Países | 2 | Normal | Copia de sistema |
| Cambios de precio | 3 | Normal | Copia de sistema |
| Ingredientes | 4 | Normal | Copia de sistema |
| Taras codificadas | 6 | Normal | Copia de sistema |
| Mercancías | 7 | Autoindexada | Master de sistema |
| Ivas | 8 | Autoindexada | Copia de sistema |
| Artículos + Totales | 9 | Normal | Copia de sistema |
| Descriptivos | 10 | Encadenada | Copia de sistema |
| Textos cabeceras y leyendas | 11 | Autoindexada | Local |
| Códigos de barras | 12 | Autoindexada | Local |
| Vendedores + Totales | 13 | Normal | Copia de sistema |
| Teclas de vendedores | 14 | Normal | Copia de sistema |
| Pedidos | 16 | Normal | Copia de sistema |
| Envasado de articulo | 19 | Normal | Copia de sistema |
| Vacuno | 20 | Autoindexada | Copia de sistema |

### 2.3.1 Clientes

Número máximo de elementos por defecto: 100 o 200 (depende de version)
Número de elementos predefinidos por el sistema: 0
Modalidad de trabajo: Copia de sistema
Actualizable y autosincronizada
Modo de acceso: Normal

**Estructura básica del esqueleto:**

| Campo | Id | Modo | Tipo | Rango Accesible | Tamaño |
|---|---|---|---|---|---|
| Código | 0 | Normal | U3BYTE | (0x0 - 0x1869F) | 3 bytes |
| Nombre | 1 | Normal | STRING | | 25 bytes (1) |
| Descriptivo 1 (2) | 6 | Normal | POINTER | | Según apuntado |
| Descriptivo 2 (2) | 7 | Normal | POINTER | | Según apuntado |
| Descriptivo 3 (2) | 8 | Normal | POINTER | | Según apuntado |
| Descriptivo 4 (2) | 9 | Normal | POINTER | | Según apuntado |

Los descriptivos apuntan a la base de datos de descriptivos de articulos.

**Llave de acceso:**

| Llave | Id | Id campo | Campos |
|---|---|---|---|
| IDC_INSERT | 1 | 0 | Código |
| IDC_SEARCH | 2 | 0 | Código |
| IDC_CONSULTA | 3 | 0<br>1 | Código<br>Nombre |
| IDC_CONSNOM (2) | 4 | 0<br>1<br>6<br>7<br>8<br>9 | Código<br>Nombre<br>Descriptivo 1<br>Descriptivo 2<br>Descriptivo 3<br>Descriptivo 4 |
| IDC_PROGNOM (2) | 5 | 0<br>6<br>7<br>8<br>9 | Código<br>Descriptivo 1<br>Descriptivo 2<br>Descriptivo 3<br>Descriptivo 4 |

(1) En version de balanza anterior a C.00 el tamaño era 16 bytes
(2) A partir de version C.00 de balanzas

### 2.3.2 Países:

Número máximo de elementos por defecto: 25
Número de elementos predefinidos por el sistema: 1
Modalidad de trabajo: Copia de sistema
Actualizable
Modo de acceso: Normal

**Estructura básica del esqueleto:**

| Campo | Id | Modo | Tipo | Rango Accesible | Tamaño |
|---|---|---|---|---|---|
| Código | 0 | Normal | ULONG | (0x0 - 0x63) | 4 bytes |
| Nombre | 1 | Normal | STRING | | 21 bytes |

**Llave de acceso :**

| Llave | Id | Id campo | Campos |
|---|---|---|---|
| IDPA_CODIGO | 1 | 0 | Código |
| IDPA_NOMBRE | 2 | 1<br>0 | Nombre<br>Código |

### 2.3.3 Cambios de precio

Número máximo de elementos por defecto: 100
Número de elementos predefinidos por el sistema: 0
Modalidad de trabajo: Copia de sistema
Actualizable y autosincronizada
Modo de acceso: Normal

**Estructura básica del esqueleto:**

| Campo | Id | Modo | Tipo | Rango Accesible | Tamaño |
|---|---|---|---|---|---|
| Código | 0 | Normal | ULONG | (0x0 - 0x0F423F) | 4 bytes |
| Precio | 2 | Normal | ULONG | (0x0 - 0x0F423F) | 4 bytes |
| Tipo de cambio | 1 | Normal | CHAR | (0x0 - 0x3) | 1 byte |
| Fecha inicial | 3 | Normal | ULONG | | 4 bytes |
| Memo | 4 | Normal | MEMO | | 2+8 bytes |

***Valores del campo Tipo de Cambio:***

0x0: Cambio de Precio
0x1: Precio Cambiado Hoy
0x2: Oferta Temporal
0x3: Precio Escalonado

***Valores del campo Memo (depende del valor de campo"Tipo de cambio"):***

| Tipo de cambio | Estructura | Tipo de dato | Tamaño | Rango accesible |
|---|---|---|---|---|
| 0x02 | Fecha final | ULONG | 4 bytes | |
| 0x03 | Peso 1 | SHORT | 2 bytes | Depende Cfg Peso |
| | Precio2 | ULONG | 4 bytes | (0x0 - 0x0F423F) |
| | Peso2 | SHORT | 2 bytes | Depende Cfg Peso |

Nota: el tamaño del campo MEMO siempre es 8.

**Llave de acceso :**

| Llave | Id | Id campo | Campos |
|---|---|---|---|
| ID_PRC_TIPO | 1 | 2<br>0 | Tipo de cambio<br>Código |
| ID_PRC_DATAINI | 2 | 2<br>3<br>0 | Tipo de cambio<br>Fecha inicial<br>Código |
| ID_PRC_CODIGO | 3 | 0 | Código |

### 2.3.4 Ingredientes (Fuera de uso):

Número máximo de elementos por defecto: 250
Número de elementos predefinidos por el sistema: 0
Modalidad de trabajo: Copia de sistema
Actualizable
Modo de acceso: Normal

**Estructura básica del esqueleto:**

| Campo | Id | Modo | Tipo | Rango Accesible | Tamaño |
|---|---|---|---|---|---|
| Código | 0 | Normal | ULONG | (0x0 - 0x0F423F) | 4 bytes |
| Texto | 1 | Normal | STRING | | 16 bytes |

**Llave de acceso:**

| Llave | Id | Id campo | Campos |
|---|---|---|---|
| ID_ING_INSERT | 1 | 0 | Código |
| ID_ING_SEARCH | 2 | 0 | Código |
| ID_ING_CONSULTA | 3 | 0<br>1 | Código<br>Texto |

### 2.3.5 Taras codificadas:

Número máximo de elementos: 10
Número de elementos predefinidos por el sistema: 0
Modalidad de trabajo: Copia de sistema
Actualizable
Modo de acceso: Normal

**Estructura básica del esqueleto:**

| Campo | Id | Modo | Tipo | Tamaño |
|---|---|---|---|---|
| Código | 0 | Normal | CHAR | 1 byte |
| Tara | 1 | Normal | USHORT | 2 bytes |

**Llave de acceso**

| Llave | Id | Id campo | Campos |
|---|---|---|---|
| ID_TN_INSERT | 1 | 0 | Código |
| ID_TN_SEARCH | 2 | 1 | Tara |

### 2.3.6 Mercancías *(Autoindexada)*

Número máximo de elementos por defecto: 30
Número de elementos predefinidos por el sistema: 0
Modalidad de trabajo: Master de sistema
Actualizable
Modo de acceso: Autoindexado

**Estructura básica del esqueleto:**

| Campo | Id | Modo | Tipo | Tamaño |
|---|---|---|---|---|
| Artículo | 0 | Normal | LONG | 4 bytes |
| Peso | 1 | Normal | LONG | 4 bytes |
| Certificado | 2 | Normal | SRTING | 12 bytes |
| Fecha | 3 | Normal | ULONG | 4 bytes |

### 2.3.7 Ivas *(Autoindexada):*

Número máximo de elementos por defecto: 4
Número de elementos predefinidos por el sistema: 1
Modalidad de trabajo: Copia de sistema
Actualizable
Modo de acceso: Autoindexado

**Estructura básica del esqueleto:**

| Campo | Id | Modo | Tipo | Tamaño |
|---|---|---|---|---|
| IVA | 0 | Normal | SHORT | 2 bytes |

### 2.3.8 Artículos + Totales

Número máximo de elementos por defecto: ¿?
Número de elementos predefinidos por el sistema: 1 (artículo 0)
Modalidad de trabajo: Copia de sistema
Actualizable y autosincronizada
Modo de acceso: Normal

**Estructura básica del esqueleto:**

| Campo | Id | Modo | Tipo | Rango Accesible | Tamaño |
|---|---|---|---|---|---|
| Código | 0 | Normal | ULONG | (0x0 - 0xF423F) | 4 bytes |
| Número de PLU | 1 | Normal | USHORT | (0x0 - 0x270F) | 2 bytes |
| Código de mostrador | 2 | Normal | UCHAR | (0x0 - 0x63) | 1 byte |
| Familia | 3 | Normal | UCHAR | (0x0 - 0xFF) | 1 byte |
| Precio | 4 | Normal | UTRIBYTE | (0x0 - 0xF423F) | 3 bytes |
| Signo del precio | 72 | Normal | UBIT | 0/1 | 1 bit |
| Descriptivo | 5 | Normal | STRING | | 25(20*) bytes |
| Forma de venta | 6 | Normal | UBIT | (0x0=U, 0x1=W) | 1 bit |
| Caducidad (1) | 7 | Normal | POINTER | | *Según apuntado* |
| Consumo preferente (1) | 8 | Normal | POINTER | | *Según apuntado* |
| Tara(1) | 9 | Normal | POINTER | | *Según apuntado* |
| Número de etiqueta (1) | 10 | Normal | POINTER | | *Según apuntado* |
| Tipo de IVA | 11 | Normal | UBIT | (0x0 - 0x4) | 3 bits |
| Formato del código de barras | 12 | Normal | UBIT | (0x0 ➔ EAN13) | 2 bits |
| Código de barras | 13 | Normal | STRING | Ver Anexo 1 | 12 bytes |
| Nº paquete preseleccionado(1) | 14 | Normal | POINTER | | *Según apuntado* |
| Merma (porcentaje)(2) | 71 | Normal | POINTER | | *Según apuntado* |
| Descuento (3) | 73 | Normal | USHORT | (0x0 – 0x270F) | 2 bytes |
| Inventario | 47 | Protegido | ULONG | (0x0 - 0xF423F) | 4 bytes |
| Trazabilidad ECI. Ficha 1 (5) | 114 | Normal | ULONG | (0x0 - 0xF423F) | 4 bytes |
| Trazabilidad ECI. Ficha 2 (5) | 115 | Normal | ULONG | (0x0 - 0xF423F) | 4 bytes |
| Trazabilidad ECI. Ficha 3 (5) | 116 | Normal | ULONG | (0x0 - 0xF423F) | 4 bytes |
| Mixto (6) | 117 | Normal | UBIT | 0/1 (1=mixto) | 1 bit |
| Peso (6) | 118 | Normal | POINTER | | *Según apuntado* |
| Merma (7) | 114 | Normal | LONG | | *4 bytes* |
| Devolución (7) | 115 | Normal | LONG | | *4 bytes* |
| General (7) | 116 | Normal | LONG | | *4 bytes* |
| Totales en modo venta | | | | | |
| Número de operaciones | 15 | Protegido | USHORT | | 2 bytes |
| Número de unidades | 16 | Protegido | ULONG | | 4 bytes |
| Importe | 17 | Protegido | LONG | | 4 bytes |
| Peso | 18 | Protegido | ULONG | | 4 bytes |
| Totales en local | | | | | |
| Número de operaciones | 19 | Protegido | USHORT | | 2 bytes |
| Número de unidades | 20 | Protegido | ULONG | | 4 bytes |
| Importe | 21 | Protegido | LONG | | 4 bytes |
| Peso | 22 | Protegido | ULONG | | 4 bytes |
| Totales en modo envasado | | | | | |
| Número de operaciones | 23 | Protegido | USHORT | | 2 bytes |
| Número de unidades | 24 | Protegido | ULONG | | 4 bytes |
| Importe | 25 | Protegido | LONG | | 4 bytes |
| Peso | 26 | Protegido | ULONG | | 4 bytes |

Estructura básica del esqueleto (continuación):

| Campo | Id | Modo | Tipo | Rango Accesible | Tamaño |
|---|---|---|---|---|---|
| Descriptivos | | | | | |
| Descriptivo 1 | 37 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 2 | 38 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 3 | 39 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 4 | 40 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 5 | 41 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 6 | 42 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 7 | 43 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 8 | 44 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 9 | 45 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 10 | 46 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 11 | 58 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 12 | 59 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 13 | 60 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 14 | 61 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 15 | 62 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 16 | 63 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 17 | 64 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 18 | 65 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 19 | 66 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 20 (4) | 67 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 21 (4) | 94 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 22 (4) | 95 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 23 (4) | 96 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 24 (4) | 97 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 25 (4) | 98 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 26 (4) | 99 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 27 (4) | 100 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 28 (4) | 101 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 29 (4) | 102 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 30 (4) | 103 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 31 (4) | 104 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 32 (4) | 105 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 33 (4) | 106 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 34 (4) | 107 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 35 (4) | 108 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 36 (4) | 109 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 37 (4) | 110 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 38 (4) | 111 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 39 (4) | 112 | Normal | POINTER | | *Según apuntado* |
| Descriptivo 40 (4) | 113 | Normal | POINTER | | *Según apuntado* |

***Nota: Sólo a partir de versión de tren 8.52 y balanza 9.00:**

- **Id 58..67**
- **Id 5 con 25 caracteres (antes 20)**

Los descriptivos son punteros a la tabla de descriptivos.

Ver notas ( )… en siguiente página.

**(1): A partir de versión B00 de balanza pasan a ser punteros a tabla de datos envasado de artículos**

**(2): A partir de la versión D00 de balanza. Es un puntero a tabla de datos envasado de artículos.**

**(3): A partir de la versión F.00A. Está inactivo excepto en la versión especial ALCAMPO (J0F.00A).**

**(4): Este campo solo existe desde la versión F.00C de balanza**

**(5): Este campo sólo existe desde la versión G.003 de balanza y SOLO en versión especial EL CORTE INGLES (H0G.003).**

**(6): Este campo sólo existe desde la versión G.005 de balanza y está inactivo excepto en versión especial PLU mixto (E0G.005).**

**(7): Este campo sólo existe desde la versión K.00D de balanza y SOLO en versión especial T0K.00D.**

Llaves de acceso :

| Llave | Id | Id campo | Campos |
|---|---|---|---|
| ID_ART_INSERT_COD | 1 | 0 | Código |
| ID_ART_SEARCH_PLUMOS | 2 | 2 | Código del mostrador |
| | | 1 | Número de PLU |
| ID_ART_SEARCH_CODFAM | 3 | 3 | Familia |
| | | 0 | Código |
| ID_ART_UPDATE_STDPLU | 4 | 0 | Código |
| | | 1 | Número de PLU |
| | | 2 | Código de mostrador |
| | | 3 | Familia |
| | | 6 | Forma de venta |
| | | 4 | Precio |
| | | 5 | Descriptivo |
| | | 9 | Tara |
| ID_ART_UPDATE_ENVPLU | 5 | 7 | Caducidad |
| | | 8 | Consumo preferente |
| | | 10 | Número de etiqueta |
| | | 14 | Número de paquete preseleccionado |
| | | 0 | Código |
| ID_ART_UPDATE_CODBAR | 6 | 13 | Código de barras |
| | | 12 | Formato del código de barras |
| | | 0 | Código |
| ID_ART_UPDATE_TIVA | 7 | 11 | Tipo de iva |
| | | 0 | Código |
| ID_ART_UPDATE_PRC | 8 | 4 | Precio |
| | | 0 | Código |
| ID_ART_UPDATE_TOTVNT | 9 | 15 | Nº operaciones en modo venta |
| | | 16 | Nº unidades en modo venta |
| | | 17 | Importe en modo venta |
| | | 18 | Peso en modo venta |
| ID_ART_UPDATE_TOTLOC | 10 | 19 | Nº operaciones en local |
| | | 20 | Nº unidades en local |
| | | 21 | Importe en local |
| | | 22 | Peso en local |
| ID_ART_UPDATE_TOTENV | 11 | 23 | Nº operaciones en modo envasado |
| | | 24 | Nº unidades en modo envasado |
| | | 25 | Importe en modo envasado |
| | | 26 | Peso en modo envasado |
| ID_ART_LISTAR_TOTVNT | 12 | 15 | Nº operaciones en modo venta |
| | | 16 | Nº unidades en modo venta |
| | | 17 | Importe en modo venta |
| | | 18 | Peso en modo venta |
| | | 5 | Descriptivo |
| | | 3 | Familia |
| | | 0 | Código |
| ID_ART LISTAR__TOTLOC | 13 | 19 | Nº operaciones en local |
| | | 20 | Nº unidades en local |
| | | 21 | Importe en local |
| | | 22 | Peso en local |
| | | 5 | Descriptivo |
| | | 3 | Familia |
| | | 0 | Código |

**Llaves de acceso (continuación):**

| Llave | Id | Id campo | Campos |
|---|---|---|---|
| ID_ART_LISTAR _TOTENV | 14 | 23 | Nº operaciones en modo envasado |
| | | 24 | Nº unidades en modo envasado |
| | | 25 | Importe en modo envasado |
| | | 26 | Peso en modo envasado |
| | | 5 | Descriptivo |
| | | 3 | Familia |
| | | 0 | Código |
| ID_ART_CONSUL_TOTAL | 15 | 15 | Nº operaciones en modo venta |
| | | 16 | Nº unidades en modo venta |
| | | 17 | Importe en modo venta |
| | | 18 | Peso en modo venta |
| | | 19 | Nº operaciones en local |
| | | 20 | Nº unidades en local |
| | | 21 | Importe en local |
| | | 22 | Peso en local |
| | | 23 | Nº operaciones en modo envasado |
| | | 24 | Nº unidades en modo envasado |
| | | 25 | Importe en modo envasado |
| | | 26 | Peso en modo envasado |
| | | 0 | Código |
| ID_ART_UPDATE_DESCR0 | 16 | 37 | Descriptivo 1 |
| | | 38 | Descriptivo 2 |
| | | 39 | Descriptivo 3 |
| | | 40 | Descriptivo 4 |
| | | 41 | Descriptivo 5 |
| | | 0 | Código |
| ID_ART_UPDATE_DESCR1 | 17 | 42 | Descriptivo 6 |
| | | 43 | Descriptivo 7 |
| | | 44 | Descriptivo 8 |
| | | 45 | Descriptivo 9 |
| | | 46 | Descriptivo 10 |
| | | 0 | Código |
| ID_ART_UPDATE_DESCR | 18 | 37 | Descriptivo 1 |
| | | 38 | Descriptivo 2 |
| | | 39 | Descriptivo 3 |
| | | 40 | Descriptivo 4 |
| | | 41 | Descriptivo 5 |
| | | 42 | Descriptivo 6 |
| | | 43 | Descriptivo 7 |
| | | 44 | Descriptivo 8 |
| | | 45 | Descriptivo 9 |
| | | 46 | Descriptivo 10 |
| | | 0 | Código |

**Llaves de acceso (continuación):**

| Llave | Id | Id campo | Campos |
|---|---|---|---|
| ID_ART_BORRAR_TOTAL | 19 | 15 | Nº operaciones en modo venta |
| | | 16 | Nº unidades en modo venta |
| | | 17 | Importe en modo venta |
| | | 18 | Peso en modo venta |
| | | 19 | Nº operaciones en local |
| | | 20 | Nº unidades en local |
| | | 21 | Importe en local |
| | | 22 | Peso en local |
| | | 23 | Nº operaciones en modo envasado |
| | | 24 | Nº unidades en modo envasado |
| | | 25 | Importe en modo envasado |
| | | 26 | Peso en modo envasado |
| ID_ART_CLEAR_PRC (No existe para ver. T0K.00D y siguientes). | 20 | 4 | Precio |
| ID_ART_UPDATE_ART | 21 | 0 | Código |
| | | 1 | Número de PLU |
| | | 2 | Código de mostrador |
| | | 3 | Familia |
| | | 6 | Forma de venta |
| | | 11 | Tipo de IVA |
| | | 4 | Precio |
| | | 5 | Descriptivo |
| | | 9 | Tara |
| | | 7 | Caducidad |
| | | 8 | Consumo preferente |
| | | 10 | Número de etiqueta |
| | | 14 | Número de paquete preseleccionado |
| | | 13 | Código de barras |
| | | 12 | Formato del código de barras |
| ID_ART_SEARCH_FAM | 22 | 3 | Familia |
| ID_FAM_LISTAR_TOTVNT | 23 | 15 | Nº operaciones en modo venta |
| | | 16 | Nº unidades en modo venta |
| | | 17 | Importe en modo venta |
| | | 18 | Peso en modo venta |
| ID_FAM LISTAR__TOTLOC | 24 | 19 | Nº operaciones en local |
| | | 20 | Nº unidades en local |
| | | 21 | Importe en local |
| | | 22 | Peso en local |
| ID_FAM_LISTAR _TOTENV | 25 | 23 | Nº operaciones en modo envasado |
| | | 24 | Nº unidades en modo envasado |
| | | 25 | Importe en modo envasado |
| | | 26 | Peso en modo envasado |
| ID_ART_SEARCH_CODBAR | 26 | 13 | Código de barras |
| ID_ART_UPDATE_INVENT | 27 | 47 | Inventario |
| ID_ART_CONSUL_INVENT | 28 | 0 | Código |
| | | 47 | Inventario |
| | | 6 | Forma de venta |
| | | 5 | Descriptivo |

**Llaves de acceso (continuación):**

| Llave | Id | Id campo | Campos |
|---|---|---|---|
| ID_ART_CONSUL_INVENT (ver. T0K.00D y siguientes) | 28 | 0 | Código |
| | | 47 | Inventario |
| | | 114 | Merma |
| | | 115 | Devolución |
| | | 116 | General |
| | | 6 | Forma de venta |
| | | 5 | Descriptivo |
| ID_ART_CONSPROT | 29 | 0 | Código |
| | | 1 | Número de PLU |
| | | 2 | Código de mostrador |
| | | 3 | Familia |
| | | 4 | Precio |
| | | 5 | Descriptivo |
| | | 6 | Forma venta |
| | | 7 | Caducidad |
| | | 8 | Consumo preferente |
| | | 9 | Tara |
| | | 10 | Número de etiqueta |
| | | 11 | Tipo de IVA |
| | | 12 | Formato del código de barras |
| | | 13 | Código de barras |
| | | 14 | Número de paquete preseleccionado |
| | | 37 | Descriptivo 1 |
| | | 38 | Descriptivo 2 |
| | | 39 | Descriptivo 3 |
| | | 40 | Descriptivo 4 |
| | | 41 | Descriptivo 5 |
| | | 42 | Descriptivo 6 |
| | | 43 | Descriptivo 7 |
| | | 44 | Descriptivo 8 |
| | | 45 | Descriptivo 9 |
| | | 46 | Descriptivo 10 |
| ID_ART_SEARCH_MOSCOD | 30 | 2 | Código Mostrador |
| | | 0 | Código Articulo |

**Llaves de acceso (continuación):**

| Llave | Id | Id campo | Campos |
|---|---|---|---|
| ID_ART_UPDATE_DESCR20 | 31 | 37 | Descriptivo 1 |
| | | 38 | Descriptivo 2 |
| | | 39 | Descriptivo 3 |
| | | 40 | Descriptivo 4 |
| | | 41 | Descriptivo 5 |
| | | 42 | Descriptivo 6 |
| | | 43 | Descriptivo 7 |
| | | 44 | Descriptivo 8 |
| | | 45 | Descriptivo 9 |
| | | 46 | Descriptivo 10 |
| | | 58 | Descriptivo 11 |
| | | 59 | Descriptivo 11 |
| | | 60 | Descriptivo 13 |
| | | 61 | Descriptivo 14 |
| | | 62 | Descriptivo 15 |
| | | 63 | Descriptivo 16 |
| | | 64 | Descriptivo 17 |
| | | 65 | Descriptivo 18 |
| | | 66 | Descriptivo 19 |
| | | 67 | Descriptivo 20 |
| | | 0 | Código |

**Llaves de acceso (continuación):**

| Llave | Id | Id campo | Campos |
|---|---|---|---|
| ID_ART_CONSPROTL | 32 | 0 | Código |
| | | 1 | Número de PLU |
| | | 2 | Código de mostrador |
| | | 3 | Familia |
| | | 4 | Precio |
| | | 5 | Descriptivo |
| | | 6 | Forma venta |
| | | 7 | Caducidad |
| | | 8 | Consumo preferente |
| | | 9 | Tara |
| | | 10 | Número de etiqueta |
| | | 11 | Tipo de IVA |
| | | 12 | Formato del código de barras |
| | | 13 | Código de barras |
| | | 14 | Número de paquete preseleccionado |
| | | 37 | Descriptivo 1 |
| | | 38 | Descriptivo 2 |
| | | 39 | Descriptivo 3 |
| | | 40 | Descriptivo 4 |
| | | 41 | Descriptivo 5 |
| | | 42 | Descriptivo 6 |
| | | 43 | Descriptivo 7 |
| | | 44 | Descriptivo 8 |
| | | 45 | Descriptivo 9 |
| | | 46 | Descriptivo 10 |
| | | 58 | Descriptivo 11 |
| | | 59 | Descriptivo 11 |
| | | 60 | Descriptivo 13 |
| | | 61 | Descriptivo 14 |
| | | 62 | Descriptivo 15 |
| | | 63 | Descriptivo 16 |
| | | 64 | Descriptivo 17 |
| | | 65 | Descriptivo 18 |
| | | 66 | Descriptivo 19 |
| | | 67 | Descriptivo 20 |
| ID_ART_UPDATE_ARTM (ver D.000) | 36 | 0 | Código |
| | | 1 | Número de PLU |
| | | 2 | Código de mostrador |
| | | 3 | Familia |
| | | 6 | Forma de venta |
| | | 11 | Tipo de IVA |
| | | 4 | Precio |
| | | 5 | Descriptivo |
| | | 9 | Tara |
| | | 7 | Caducidad |
| | | 8 | Consumo preferente |
| | | 10 | Número de etiqueta |
| | | 14 | Número de paquete preseleccionado |
| | | 13 | Código de barras |
| | | 12 | Formato del código de barras |
| | | 71 | Merma |

**Llaves de acceso (continuación):**

| Llave | Id | Id campo | Campos |
|---|---|---|---|
| ID_ART_UPDATE_ARTMD (ver E0G.005) | 39 | 0 | Código |
| | | 1 | Número de PLU |
| | | 2 | Código de mostrador |
| | | 3 | Familia |
| | | 6 | Forma de venta |
| | | 11 | Tipo de IVA |
| | | 4 | Precio |
| | | 5 | Descriptivo |
| | | 9 | Tara |
| | | 7 | Caducidad |
| | | 8 | Consumo preferente |
| | | 10 | Número de etiqueta |
| | | 14 | Número de paquete preseleccionado |
| | | 13 | Código de barras |
| | | 12 | Formato del código de barras |
| | | 71 | Merma |
| | | 72 | Signo del precio |
| | | 73 | Descuento |
| | | 114 | Trazabilidad ECI. Ficha 1 |
| | | 115 | Trazabilidad ECI. Ficha 2 |
| | | 116 | Trazabilidad ECI. Ficha 3 |
| | | 117 | Mixto |
| | | 118 | Peso (mixto) |

**Llaves de acceso (continuación):**

| Llave | Id | Id campo | Campos |
|---|---|---|---|
| ID_ART_UPDATE_DESCRL | 40 | 37 | Descriptivo 1 |
| | | 38 | Descriptivo 2 |
| | | 39 | Descriptivo 3 |
| | | 40 | Descriptivo 4 |
| | | 41 | Descriptivo 5 |
| | | 42 | Descriptivo 6 |
| | | 43 | Descriptivo 7 |
| | | 44 | Descriptivo 8 |
| | | 45 | Descriptivo 9 |
| | | 46 | Descriptivo 10 |
| | | 58 | Descriptivo 11 |
| | | 59 | Descriptivo 11 |
| | | 60 | Descriptivo 13 |
| | | 61 | Descriptivo 14 |
| | | 62 | Descriptivo 15 |
| | | 63 | Descriptivo 16 |
| | | 64 | Descriptivo 17 |
| | | 65 | Descriptivo 18 |
| | | 66 | Descriptivo 19 |
| | | 67 | Descriptivo 20 |
| | | 94 | Descriptivo 21 |
| | | 95 | Descriptivo 22 |
| | | 96 | Descriptivo 23 |
| | | 97 | Descriptivo 24 |
| | | 98 | Descriptivo 25 |
| | | 99 | Descriptivo 26 |
| | | 100 | Descriptivo 27 |
| | | 101 | Descriptivo 28 |
| | | 102 | Descriptivo 29 |
| | | 103 | Descriptivo 30 |
| | | 104 | Descriptivo 31 |
| | | 105 | Descriptivo 32 |
| | | 106 | Descriptivo 33 |
| | | 107 | Descriptivo 34 |
| | | 108 | Descriptivo 35 |
| | | 109 | Descriptivo 36 |
| | | 110 | Descriptivo 37 |
| | | 111 | Descriptivo 38 |
| | | 112 | Descriptivo 39 |
| | | 113 | Descriptivo 40 |
| | | 0 | Código |
| ID_ART_UPDATE_MERMA (ver. T0K.00D y siguientes) | 42 | 114 | Merma |
| ID_ART_UPDATE_DEVOL (ver. T0K.00D y siguientes) | 43 | 115 | Deolución |
| ID_ART_UPDATE_GENERAL (ver. T0K.00D y siguientes) | 44 | 116 | General |

### 2.3.9 Descriptivos *(Encadenada):*

Número máximo de elementos por defecto: 20000
Número de elementos predefinidos por el sistema: 0
Modalidad de trabajo: Copia de sistema
No Actualizable *(su actualización depende de a que base de datos este conectada)*
Modo de acceso: Encadenado

**A esta base de datos sólo deben acceder otras bases de datos mediante campos de tipo LINK**

**Estructura básica del esqueleto:**

| Campo | Id | Modo | Tipo | Tamaño |
|---|---|---|---|---|
| Descriptivo | 0 | Normal | STRING | 25 (20) bytes |

***Nota: Sólo a partir de versión de tren 8.52 y balanza 9.00:**

- **Id 0 con 25 caracteres (antes 20)**

### 2.3.10 Textos: cabeceras, leyendas, textos libres y publicidad *(Autoindexada):*

Número máximo de elementos por defecto: 34
Número de elementos predefinidos por el sistema: 0
Modalidad de trabajo: Local
Actualizable
Modo de acceso: Autoindexado

Entradas de la 0 a la 4 → Cabeceras
Entradas de la 5 a la 9 → Leyendas
Entradas de la 10 a la 19 → Textos libres
Entradas de la 20 a la 29 → Publicidad Ofertas
Entradas de la 30 a la 33 → Publicidad Continua

**Estructura básica del esqueleto:**

| Campo | Id | Modo | Tipo | Tamaño |
|---|---|---|---|---|
| Texto | 0 | Normal | STRING | 26 bytes |
| Tipo de letra | 1 | Normal | UCHAR | 1 byte |

La balanza tiene 80 tipos de letra diferentes, del 1 al 80, y el valor del byte de tipo de letra guardado en la Base de Datos es el Tipo de letra + 47.

### 2.3.11 Códigos de barras *(Autoindexada):*

Número máximo de elementos por defecto: 14
Número de elementos predefinidos por el sistema: 0
Modalidad de trabajo: Local
Actualizable
Modo de acceso: Autoindexado

**Estructura básica del esqueleto:**

| Campo | Id | Modo | Tipo | Tamaño |
|---|---|---|---|---|
| Código de barras | 0 | Normal | STRING | 13 bytes |

**Código de Barras en cada Posición:**

| Indice | Código de Barras |
|---|---|
| 0 | Código Barras Modo Venta en Red |
| 1 | Código Barras Modo Venta en Local |
| 2 | Código Barras Modo Mixto en Red |
| 3 | Código Barras Modo Mixto en Local |
| 4 | Código Barras Modo Super en Red |
| 5 | Código Barras Modo Super en Local |
| 6 | Código Barras Modo Envasado en Red 1 |
| 7 | Código Barras Modo Envasado en Local 1 |
| 8 | Código Barras Modo Envasado en Red 2 |
| 9 | Código Barras Modo Envasado en Local 2 |
| 10 | Código Barras Modo Total en Red 1 |
| 11 | Código Barras Modo Total en Local 1 |
| 12 | Código Barras Modo Total en Red 2 |
| 13 | Código Barras Modo Total en Local 2 |

El significado de los Bytes del String del Código de Barras se muestra en el ANEXO 1.

### 2.3.12 Vendedores + Totales:

Número máximo de elementos por defecto: 100
Número de elementos predefinidos por el sistema: 1
Modalidad de trabajo: Copia de sistema
Actualizable y autosincronizada
Modo de acceso: normal

**Estructura básica del esqueleto:**

| Campo | Id | Modo | Tipo | Tamaño |
|---|---|---|---|---|
| Nombre | 0 | Normal | STRING | 31 bytes |
| Código | 1 | Normal | ULONG | 4 bytes |
| Cajón | 23 | Normal | LONG | 4 bytes |
| Reembolso | 27 | Normal | LONG | 4 bytes |
| Desembolso | 28 | Normal | LONG | 4 bytes |
| Totales venta | | | | |
| Número de tiquets | 2 | Protegido | ULONG | 4 bytes |
| Número de operaciones | 3 | Protegido | USHORT | 2 bytes |
| Importe | 4 | Protegido | LONG | 4 bytes |
| Número de operaciones negativas | 5 | Protegido | USHORT | 2 bytes |
| Importes negativos | 6 | Protegido | LONG | 4 bytes |
| Número de operaciones canceladas | 7 | Protegido | USHORT | 2 bytes |
| Importes cancelados | 8 | Protegido | LONG | 4 bytes |
| Totales local | | | | |
| Número de tiquets | 9 | Protegido | ULONG | 4 bytes |
| Número de operaciones | 10 | Protegido | USHORT | 2 bytes |
| Importe | 11 | Protegido | LONG | 4 bytes |
| Número de operaciones negativas | 12 | Protegido | USHORT | 2 bytes |
| Importes negativos | 13 | Protegido | LONG | 4 bytes |
| Número de operaciones canceladas | 14 | Protegido | USHORT | 2 bytes |
| Importes cancelados | 15 | Protegido | LONG | 4 bytes |
| Totales envasado | | | | |
| Número de tiquets | 16 | Protegido | ULONG | 4 bytes |
| Número de operaciones | 17 | Protegido | USHORT | 2 bytes |
| Importe | 18 | Protegido | LONG | 4 bytes |
| Número de operaciones negativas | 19 | Protegido | USHORT | 2 bytes |
| Importes negativos | 20 | Protegido | LONG | 4 bytes |
| Número de operaciones canceladas | 21 | Protegido | USHORT | 2 bytes |
| Importes cancelados | 22 | Protegido | LONG | 4 bytes |
| Cajón | | | | |
| Total cajón | 23 | Protegido | LONG | 4 bytes |

**Llaves de acceso:**

| Llave | Id | Id campo | Campos |
|---|---|---|---|
| ID_INSERT_VND | 1 | 1 | Código |
| ID_BORRAR_TOTAL_VND | 2 | 2 | Tiquets en modo venta |
| | | 3 | Operaciones en modo venta |
| | | 4 | Importe en modo venta |
| | | 5 | Operaciones negativas en modo venta |
| | | 6 | Importes negativos en modo venta |
| | | 7 | Operaciones canceladas en modo venta |
| | | 8 | Importes cancelados en modo venta |
| | | 9 | Tiquets en local |
| | | 10 | Operaciones en local |
| | | 11 | Importe en local |
| | | 12 | Operaciones negativas en local |
| | | 13 | Importes negativos en local |
| | | 14 | Operaciones canceladas en local |
| | | 15 | Importes cancelados en local |
| | | 16 | Tiquets en modo envasado |
| | | 17 | Operaciones en modo envasado |
| | | 18 | Importe en modo envasado |
| | | 19 | Operaciones negativas en envasado |
| | | 20 | Importes negativos en envasado |
| | | 21 | Operaciones canceladas en envasado |
| | | 22 | Importes cancelados en envasado |
| ID_UPDATE_TOTVNT_VND | 3 | 2 | Tiquets en modo venta |
| | | 3 | Operaciones en modo venta |
| | | 4 | Importe en modo venta |
| | | 5 | Operaciones negativas en modo venta |
| | | 6 | Importes negativos en modo venta |
| | | 7 | Operaciones canceladas en modo venta |
| | | 8 | Importes cancelados en modo venta |
| ID_UPDATE_TOTLOC_VND | 4 | 9 | Tiquets en local |
| | | 10 | Operaciones en local |
| | | 11 | Importe en local |
| | | 12 | Operaciones negativas en local |
| | | 13 | Importes negativos en local |
| | | 14 | Operaciones canceladas en local |
| | | 15 | Importes cancelados en local |
| ID_UPDATE_TOTENV_VND | 5 | 16 | Tiquets en modo envasado |
| | | 17 | Operaciones en modo envasado |
| | | 18 | Importe en modo envasado |
| | | 19 | Operaciones negativas en envasado |
| | | 20 | Importes negativos en envasado |
| | | 21 | Operaciones canceladas en envasado |
| | | 22 | Importes cancelados en envasado |

**Llaves de acceso (continuación):**

| Llave | Id | Id campo | Campos |
|---|---|---|---|
| ID_LISTAR _TOTVNT_VND | 6 | 2 | Tiquets en modo venta |
| | | 3 | Operaciones en modo venta |
| | | 4 | Importe en modo venta |
| | | 5 | Operaciones negativas en modo venta |
| | | 6 | Importes negativos en modo venta |
| | | 7 | Operaciones canceladas en modo venta |
| | | 8 | Importes cancelados en modo venta |
| | | 1 | Código |
| ID_LISTAR _TOTLOC_VND | 7 | 9 | Tiquets en local |
| | | 10 | Operaciones en local |
| | | 11 | Importe en local |
| | | 12 | Operaciones negativas en local |
| | | 13 | Importes negativos en local |
| | | 14 | Operaciones canceladas en local |
| | | 15 | Importes cancelados en local |
| | | 1 | Código |
| ID_LISTAR _TOTENV_VND | 8 | 16 | Tiquets en modo envasado |
| | | 17 | Operaciones en modo envasado |
| | | 18 | Importe en modo envasado |
| | | 19 | Operaciones negativas en envasado |
| | | 20 | Importes negativos en envasado |
| | | 21 | Operaciones canceladas en envasado |
| | | 22 | Importes cancelados en envasado |
| | | 1 | Código |
| ID_CONSULTAR_TOTAL_VND | 9 | 2 | Tiquets en modo venta |
| | | 3 | Operaciones en modo venta |
| | | 4 | Importe en modo venta |
| | | 5 | Operaciones negativas en modo venta |
| | | 6 | Importes negativos en modo venta |
| | | 7 | Operaciones canceladas en modo venta |
| | | 8 | Importes cancelados en modo venta |
| | | 9 | Tiquets en local |
| | | 10 | Operaciones en local |
| | | 11 | Importe en local |
| | | 12 | Operaciones negativas en local |
| | | 13 | Importes negativos en local |
| | | 14 | Operaciones canceladas en local |
| | | 15 | Importes cancelados en local |
| | | 16 | Tiquets en modo envasado |
| | | 17 | Operaciones en modo envasado |
| | | 18 | Importe en modo envasado |
| | | 19 | Operaciones negativas en envasado |
| | | 20 | Importes negativos en envasado |
| | | 21 | Operaciones canceladas en envasado |
| | | 22 | Importes cancelados en envasado |
| | | 1 | Código |
| ID_UPDATE_CAJON (Hasta ver. K.00E) | 10 | 1 | Código |
| | | 23 | Cajón |
| ID_UPDATE_CAJON (Desde ver. K.00E) | 10 | 1 | Código |
| | | 23 | Cajón |
| | | 27 | Reembolso |
| | | 28 | Desembolso |
| ID_UPDATE_VENDEDOR | 11 | 1 | Código |
| | | 0 | Nombre |

| | | | |
|---|---|---|---|
| ID_BORRAR_TOTALC_VND | 12 | 2 | Tiquets en modo venta |
| | | 3 | Operaciones en modo venta |
| | | 4 | Importe en modo venta |
| | | 5 | Operaciones negativas en modo venta |
| | | 6 | Importes negativos en modo venta |
| | | 7 | Operaciones canceladas en modo venta |
| | | 8 | Importes cancelados en modo venta |
| | | 9 | Tiquets en local |
| | | 10 | Operaciones en local |
| | | 11 | Importe en local |
| | | 12 | Operaciones negativas en local |
| | | 13 | Importes negativos en local |
| | | 14 | Operaciones canceladas en local |
| | | 15 | Importes cancelados en local |
| | | 16 | Tiquets en modo envasado |
| | | 17 | Operaciones en modo envasado |
| | | 18 | Importe en modo envasado |
| | | 19 | Operaciones negativas en envasado |
| | | 20 | Importes negativos en envasado |
| | | 21 | Operaciones canceladas en envasado |
| | | 22 | Importes cancelados en envasado |
| | | 1 | Código |
| | | 23 | Total cajón |
| ID_BORRAR_TOTALP_VND (Desde ver. K.00E) | 16 | 2 | Tiquets en modo venta |
| | | 3 | Operaciones en modo venta |
| | | 4 | Importe en modo venta |
| | | 5 | Operaciones negativas en modo venta |
| | | 6 | Importes negativos en modo venta |
| | | 7 | Operaciones canceladas en modo venta |
| | | 8 | Importes cancelados en modo venta |
| | | 24 | Total Peso |
| | | 9 | Tiquets en local |
| | | 10 | Operaciones en local |
| | | 11 | Importe en local |
| | | 12 | Operaciones negativas en local |
| | | 13 | Importes negativos en local |
| | | 14 | Operaciones canceladas en local |
| | | 15 | Importes cancelados en local |
| | | 25 | Total Peso Local |
| | | 16 | Tiquets en modo envasado |
| | | 17 | Operaciones en modo envasado |
| | | 18 | Importe en modo envasado |
| | | 19 | Operaciones negativas en envasado |
| | | 26 | Total Peso Envasado |
| | | 20 | Importes negativos en envasado |
| | | 21 | Operaciones canceladas en envasado |
| | | 22 | Importes cancelados en envasado |
| | | 23 | Total cajón |
| | | 27 | Total Reembolso |
| | | 28 | Total Desembolso |

### 2.3.13 Teclas de vendedores:

Número máximo de elementos: 140
Número de elementos predefinidos por el sistema: 0
Modalidad de trabajo: Copia de sistema
Actualizable y autosincronizada
Modo de acceso: Normal

**Estructura básica del esqueleto:**

| Campo | Id | Modo | Tipo | Tamaño |
|---|---|---|---|---|
| Código de mostrador | 0 | Normal | CHAR | 1 byte |
| Código de vendedor | 1 | Normal | ULONG | 4 bytes |
| Tecla actual | 2 | Normal | CHAR | 1 byte |

**Llave de acceso:**

| Llave | Id | Id campo | Campos |
|---|---|---|---|
| ID_SEARCH_TECLAMOS | 1 | 0<br>2 | Código de mostrador<br>Tecla actual |
| ID_SEARCH_CODEVND | 2 | 1 | Código de vendedor |
| ID_INSERT_TECLASVND | 3 | 0<br>2<br>1 | Código de mostrador<br>Tecla actual<br>Código de vendedor |
| ID_SEARCH_CLIENTEPM | 4 | 2 | Tecla actual |

### 2.3.14 Pedidos:

Número máximo de elementos por defecto: 100
Número de elementos predefinidos por el sistema: 0
Modalidad de trabajo: Copia de sistema
Modo de acceso: Normal

**Estructura básica del esqueleto:**

| Campo | Id | Modo | Tipo | Tamaño |
|---|---|---|---|---|
| Código | 0 | Normal | ULONG | 4 bytes |
| Pedido | 1 | Normal | LONG | 4 bytes |
| Forma de venta | 2 | Normal | UBIT | 1 bit |
| Descriptivo | 3 | Normal | STRING | 20 bytes |

**Llaves de acceso :**

| Llave | Id | Id campo | Campos |
|---|---|---|---|
| ID_PEDIDO_INSERT | 1 | 0 | Código |
| ID_UPDATE_PEDIDO | 2 | 0<br>1 | Código<br>Pedido |

### 2.3.15 Vacuno *(autoindexada):*

Número máximo de elementos por defecto: 9 o 255 (depende de version)
Número de elementos predefinidos por el sistema: 0
Modalidad de trabajo: Copia de sistema
Actualizable y autosincronizada
Modo de acceso: Autoindexado

**Estructura básica del esqueleto:**

| Campo | Id | Modo | Tipo | Tamaño |
|---|---|---|---|---|
| Identificador del animal (1) | 1 | Normal | STRING | 10 o 20 bytes |
| Codigo pais de nacimiento | 2 | Normal | USHORT | 2 bytes |
| Codigo pais de crianza | 3 | Normal | USHORT | 2 bytes |
| Codigo pais de sacrificio | 4 | Normal | USHORT | 2 bytes |
| Codigo pais de despiece | 5 | Normal | USHORT | 2 bytes |
| Codigo pais de produccion | 6 | Normal | USHORT | 2 bytes |
| RSI de sala de sacrificio(1) | 7 | Normal | STRING | 10 o 20 bytes |
| RSI de sala de despiece(1) | 8 | Normal | STRING | 10 o 20 bytes |
| Categoria del animal | 9 | Normal | CHAR | 1 byte |
| Raza | 10 | Normal | CHAR | 1 byte |
| Fecha de sacrificio (2) | 20 | Normal | ULONG | 4 bytes |
| Edad (2) | 21 | Normal | UCHAR | 1 byte |
| Sexo (2) | 22 | Normal | STRING | 1 byte |
| Peso (2) | 23 | Normal | ULONG<br>Peso en Kg con un decimal | 4 bytes |
| Texto extra (2) | 24 | Normal | STRING | 25 bytes |
| Fecha de nacimiento (3) | 25 | Normal | ULONG | 4 bytes |
| Fecha de despiece (3) | 26 | Normal | ULONG | 4 bytes |
| Lote del animal (3) | 27 | Normal | STRING | 20 bytes |
| Totales en modo venta | | | | |
| Número de operaciones | 11 | Protegido | USHORT | 2 bytes |
| Número de unidades | 12 | Protegido | ULONG | 4 bytes |
| Importe | 13 | Protegido | LONG | 4 bytes |
| Peso | 14 | Protegido | ULONG | 4 bytes |
| Totales en local | | | | |
| Número de operaciones | 15 | Protegido | USHORT | 2 bytes |
| Número de unidades | 16 | Protegido | ULONG | 4 bytes |
| Importe | 17 | Protegido | LONG | 4 bytes |
| Peso | 18 | Protegido | ULONG | 4 bytes |
| Estado de la ficha | | | | |
| Estado de totalización | 19 | Normal | UCHAR<br>0x00:Total borrado en estado 0x03.<br>0x01:Total borrado en estado 0x04.<br>0x02:Sin total.<br>0x03:Con total impreso.<br>0x04:Con total no impreso. | 1 byte |

Nota:
(1): A partir de versión de balanza D.TST pasa de 10 a 20 caracteres máximo.
(2): Solo están a partir de la versión D.TST
(3): Solo a partir de la versión D.000

**Llaves de acceso :**

| Llave | Id | Id campo | Campos |
|---|---|---|---|
| ID_VAC_LISTARTOTVNT | 2 | 11 | Número de operaciones |
| | | 12 | Número de unidades |
| | | 13 | Importe |
| | | 14 | Peso |
| ID_VAC_LISTARTOTLOC | 3 | 15 | Número de operaciones |
| | | 16 | Número de unidades |
| | | 17 | Importe |
| | | 18 | Peso |
| ID_VAC_TOT | 4 | 15 | Número de operaciones |
| | | 16 | Número de unidades |
| | | 17 | Importe |
| | | 18 | Peso |
| | | 11 | Número de operaciones |
| | | 12 | Número de unidades |
| | | 13 | Importe |
| | | 14 | Peso |
| ID_VAC_STATUS | 5 | 19 | Estado totalizacion |
| ID_VAC_BORRAR | 6 | 19 | Estado totalizacion |
| | | 15 | Número de operaciones |
| | | 16 | Número de unidades |
| | | 17 | Importe |
| | | 18 | Peso |
| | | 11 | Número de operaciones |
| | | 12 | Número de unidades |
| | | 13 | Importe |
| | | 14 | Peso |

### *2.3.16* Datos envasado de artículo *(Encadenada)*

Número máximo de elementos por defecto: ¿?
Número de elementos predefinidos por el sistema: 0
Modalidad de trabajo: Copia de sistema
No actualizable (se actualiza con la base de datos conectada)
Modo de acceso: Normal

**Estructura básica del esqueleto:**

| Campo | Id | Modo | Tipo | Rango Accesible | Tamaño |
|---|---|---|---|---|---|
| Caducidad | 0 | Normal | UBIT | (0x0 - 0x03E7) | 10 bits |
| Consumo preferente | 1 | Normal | UBIT | (0x0 - 0x03E7) | 10 bits |
| Tara | 2 | Normal | USHORT | (0x0 - 0x260F) | 2 bytes |
| Número de etiqueta | 3 | Normal | BIT | (0x0 - 0xA)<br>(0xB➔ No Etq)<br>(0xFF➔Defecto) | 6 bits |
| Nº paquete preseleccionado | 4 | Normal | UCHAR | (0x0 - 0x63) | 1 byte |
| Merma, porcentaje a restar del peso indicado (1) | 7 | Normal | UCHAR | (0..99) | 1 byte |
| Peso (2) | 12 | Normal | ULONG | (0..0x3A98)- peso | 4 bytes |

**Esta base de datos solo existe desde la versión de balanzas B00**
**(1): Este campo solo esta desde la versión D00 de balanzas**
**(2): Este campo sólo existe desde la versión G.005 de balanza y está inactivo excepto en versión especial PLU mixto (E0G.005).**

# 3. PETICION DE TIQUETS

Si se solicita, la balanza envía el mayor número de tiquets **completos** que pueda contener un mensaje.

Si un tiquet no cabe completamente en un mensaje, se envía aislado en uno, dos o los mensajes necesarios.

## 3.1 PETICION DE FORMATO (IDF:000003)

### 3.1.1 Formato de la pregunta

Con este mensaje se solicita el formato que utilizará la balanza para responder a la petición de tiquets

No tiene datos de pregunta, la respuesta es la lista ordenada de recepción de identificadores de campos con la longitud de los mismos.

### 3.1.2 Formato de la respuesta

Informa de la estructura del mensaje respuesta a la petición de tiquets. Consiste en una secuencia de identificadores (ID) y tamaño de los datos (lon). Tiene 3 partes separadas por identificadores de grupo según el tipo de información.

Ninguno de los identificadores se envía posteriormente con los datos de los tiques.

El mensaje tiene el siguiente formato.

:

<IDC1><ID1><lon1><ID2><lon2>….<IDn><lon><IDC2><IDn+1><lonn+1>…

IDCi, IDi y lon tienen longitud 1 byte, siendo

<IDCi> identificador de grupo.

<IDi>: identificador del dato.

<lon> longitud del mismo.

**Identificadores de grupo:**

| CAMPO | IDC | DESCRIPCION |
|---|---|---|
| Cabecera de menaje | 50 | Información general de todo el mensaje de respuesta: nº. tiquets enviados, tiquet inicial, etc.<br>Los ID que le acompañan se envían 1 por cada mensaje |
| Cabecera de tiquet | 51 | Los ID que le acompañan forman la cabecera de los tiquets.<br>Se envían 1 por cada tiquet que contiene el mensaje. |
| Línea de tiquet | 52 | Los ID que le acompañan forman cada línea de tiquet.<br>Se envían 1 por cada línea de tiquet que contenga el mensaje. |

**Formato del mensaje y contenido de lon:**

| CAMPO | ID | lon | COMENTARIOS |
|---|---|---|---|
| Cabecera de menaje | 50 | | |
| Número de tiquets | 1 | 1 | Total de tiquets que contiene el mensaje. |
| Línea inicial **(1)** | 2 | 1 | Número de la primera línea. |
| Número líneas **(1)** | 3 | 1 | Sólo es válido si el mensaje contiene parte de un tiquet. |
| Número de tiquet | 4 | 4 | Nº. de tiquet enviado o del primero si se envían varios. |
| Cabecera de tiquet | 51 | | |
| Número de tiquet | 150 | 4 | |
| Vendedor | 151 | 4 | Código del vendedor. |
| Mostrador | 152 | 1 | |
| Fecha y hora | 153 | 4 | En segundos a partir del 1 de Enero de 1980 |
| Nº. de balanza | 154 | 1 | |
| Importe total | 155 | 4 | Punto decimal virtual. |
| Nº. de líneas **(1)** | 156 | 1 | |
| Reservado 1 **(7)** | 157 | 1 | |
| Lín. Canceladas **(1)** | 158 | 1 | Número de líneas canceladas. |
| Tipo de tiquet | 159 | 1 | |
| Estado tiquet **(1)** | 160 | 1 | |
| Cliente | 161 | 4 | |
| Descuento tiquet | 162 | 1 | En % |
| Lote alfanumérico **(4)** | 165 | * | 6 digitos + nulo ó 20 caracteres. ***Puede tener tamaño cero***. |
| Forma de pago **(5) (8)** | 166 | 1 | 0xFF: no se utiliza forma de pago.<br>0x00 :Moneda, 0x01: cheque, 0x02: tarjeta, 0x03: genérico. |
| Valor entregado **(5)** | 167 | 4 | Punto decimal virtual. Cantidad entregada por el comprador. |
| Reservado 2 **(2)** | 199 | 2 | |
| N. tqt interno **(2)** | 164 | 2 | |
| Importe total bruto | 163 | 4 | Punto decimal virtual. Total bruto (sin descuento) |
| Línea de tiquet | 52 | | |
| Nº. de línea **(1)** | 100 | 1 | |
| Código artículo | 101 | 4 | |
| Precio | 102 | 4 | Precio de venta. Punto decimal virtual. |
| Cantidad | 103 | 4 | Peso en gramos ó unidades. |
| Tipo de iva | 149 | 1 | |
| Tipo de línea | 104 | 1 | |
| Descuento en línea **(6)** | 106 | * | En % |
| Ficha vacuno **(3)** | 107 | 1 | Número de ficha de vacuno. |
| Importe | 105 | 4 | Punto decimal virtual. |

(1) Antes de versión 9 lon=2.
(2) Antes de versión 9 estos dos campos estaban juntos: ID 199 lon=6
(3) A partir de versión B de balanzas
(4) Tamaño nulo excepto versiones especiales. A partir de b10C.004 (CAPRABO) 6 dígitos más nulo. A partir de b30F.00D 20 caracteres alfanuméricos.
(5) Sólo versión especial INVICTA, a partir de bF0E.00A. Resto versiones no se envía.
(6) En versión estándar lon=1 (0 a 99%). En versión especial ALCAMPO (bJ0F.00D) lon=2 (0 a 99.99%, con punto decimal virtual).
(7) Eliminado 25/01/06. Antes de versión 9 lon=2.
(8) A partir de versión K.00E se envía la forma de pago para todas las versiones.

## 3.2 PETICION DE TICKETS (IDF:000002)

### 3.2.1 Formato de la pregunta

| CAMPO | Bytes | VALORES/DESCRIPCION |
|---|---|---|
| que | | 1: Consulta igual numero<br>2: Mayor<br>3: Mayor o igual<br>4: Menor<br>5: Menor o igual |
| tipo numeración | 1 | 0: Numeración externa según configuración del sistema<br>1: Numeración interna, sin tipo de tiquet en orden de realización<br>2: Numeración indice sin tipo de tiquet (red, local, rollo, envasado…) en orden de la base de datos |
| numero tiquet | 4 | Número de tiquet primero solicitado según el valor del campo [que] |
| cantidad de tiquets | 1 | Número de tiquets solicitado. FF los que quepan enteros en la respuesta.<br>Un tiquet que no cabe en un mensaje completo viene solo, nunca al final de una respuesta con varios tiquets hay un tiquet incompleto |
| línea inicial | 2 | Número de línea inicial solicitada, la primera es la línea 1, campo para usar en caso de tiquets que no caven en una sola respuesta |
| pedido | 1 | 0 : No marcar como pedido<br>1 : Marcar como pedido<br>2 : Marcar como confirmado |
| tipo de tiquet | 1 | Tipo de tiquet. Tipos 4..7 solo para peticiones por numeración externa. Tipo 8 solo para numeración índice o interna.<br>0 : Rollo<br>1 : Súper<br>2 : Envasado<br>3 : Autoservicio<br>4 : Rollo local<br>5 : Súper local<br>6 : Envasado local<br>7 : Autoservicio local<br>8 : Global sin tipo |
| Nº ítem de consulta | 4 | En petición por numeración externa: nº balanza, mostrador, vendedor, fecha… según el tipo de numeración del sistema.<br>0 en el resto de casos. |
| Condición de consulta (1) | 1 | Mirar o no el dato de condición del estado. Si bit a 0, no utilizar la condición.<br>bit 0 : Local<br>bit 1 : Encargo<br>bit 2 : Pedido<br>bit 3 : Pedido i confirmado<br>bit 4 : Cerrado<br>bit 5 : Recargado<br>bit 6 : Cancelado |
| Dato de condición (1) | 1 | Estado de la condición. La condición debe cumplirse si bit a 1.<br>Los bits son los mismos de 'Condición de consulta' |

**Formato de la pregunta (continuación)**

| CAMPO | Bytes | VALORES/DESCRIPCION |
|---|---|---|
| Condición de consulta extra (2) | 1 | Otra condición que debe cumplir la consulta del tiquet<br>0: No hay condición<br>1: Igual vendedor<br>2: Igual mostrador<br>3: Igual fecha<br>4: Igual balanza<br>5: Igual cliente |
| Dato de condición extra (2) | 4 | Dato (ítem) que debe cumplir la condición extra |

(1) y (2): estos campos no son necesarios y la balanza solo los utiliza a partir de la versión 9.001.

Si no están los campos 'Condición de consulta' y 'Dato de condición' se considera consulta sin condición y 'tipo de tiquet' a 8 para 'tipo de numeración' interna o índice.

Si no están los campos 'Condición de consulta extra' y 'Dato de condición extra' se considera consulta sin condición extra.

En la petición por numeración externa según la configuración del sistema, son necesarios el tipo de tiquet y el nº de ítem de consulta. En los tiquets en local, la numeración siempre es por balanza.

En la petición por numeración interna sólo son válidas las peticiones por "igual número" y por "mayor que".

En la petición por tipo de numeración "índice", el primer índice es el 0.

Los tiquets en red se deben pedir a la balanza servidora.

La balanza backup guarda sus tiquets en local junto a la copia de los tiquets de red (por orden de llegada). Para pedir los tiquets en local a esta balanza conviene utilizar la petición condicional con el bit de local activado.

### 3.2.2 Formato de la respuesta.

Formato de la respuesta con “y” tiquets numerados a partir de “x”

<Cabecera de mensaje><Cabecera de tiquet x><Línea 1>…<Línea n>…<Cabecera de tiquet x+y><Línea 1>…<Línea n>

Si la balanza no dispone de más tiquets para enviar, el mensaje de respuesta retorna IDF:0000100 (0x000064).

Los datos y tamaño que aparecen en <Cabecera de tiquet> y <Línea> dependen de la versión de programa. Estos datos se deben obtener a partir del mensaje IDF:000003.

**Comentarios y posibles valores que toman los diferentes campos:**

| CAMPO | ID | DESCRIPCION |
|---|---|---|
| Cabecera de menaje | 50 | |
| Número de tiquets | 1 | Número total de tiquets de la respuesta, si es uno puede estar incompleto. |
| Línea inicial | 2 | Número de línea inicial enviada en la respuesta.<br>Las líneas se numeran a partir de 1.<br>Solo tiene sentido si la respuesta tiene un sólo tiquet. |
| Número de líneas | 3 | Número de líneas de la respuesta.<br>Solo tiene sentido si la respuesta tiene un solo tiquet. |
| Número de tiquet | 4 | Numeración índice del primer tiquet consultado. |
| Cabecera de tiquet | 51 | |
| Número de tiquet | 150 | Número de tiquet asignado según numeración del sistema |
| Vendedor | 151 | Código del vendedor que ha realizado el tiquet |
| Mostrador | 152 | Número de mostrador en el que se ha realizado el tiquet |
| Fecha y hora | 153 | Tiempo en que se ha realizado el tiquet, contiene fecha y hora.<br>En segundos a partir del 1 de Enero de 1980 |
| Nº. de balanza | 154 | Número de balanza que ha hecho el total |
| Importe total | 155 | Con punto decimal virtual. |
| Nº. total de líneas | 156 | Número de líneas del tiquet |
| Reservado 1 | 157 | |
| Líneas canceladas | 158 | Número de líneas canceladas |
| Tipo de tiquet | 159 | 1: Tiquet rollo<br>2: Tiquet super<br>3: Tiquet envasado<br>4: Tiquet autoservicio |
| Estado del tiquet | 160 | Cada bit indica el estado del tiquet, 1 es cierto, de izquierda a derecha :<br>bit 0 : Local<br>bit 1 : Encargo<br>bit 2 : Pedido<br>bit 3 : Pedido i confirmado<br>bit 4 : Cerrado<br>bit 5 : Recargado<br>bit 6 : Cancelado |
| Cliente | 161 | Cliente del tiquet (pe. sistema suturno) |
| Forma Pago | 166 | Tipo de Forma de pago (A partir de ver. K.00E). |
| Descuento tiquet | 162 | Porcentaje de descuento aplicado al total del tiquet<br>bit 7: si esta a cero, no hay descuento<br>bit 0..6: porcentaje de descuento |
| Lote alfanumérico | 165 | Lote alfanumérico. Sólo en programa especial b10X.xxx Para etiquetas. |
| Reservado 2 | 199 | |
| Num. tiquet interno | 164 | numeración de tiquet interna de la red de balanzas |

| | | |
|---|---|---|
| Importe total bruto | 163 | Importe sin aplicar descuento (suma de las líneas) |
| Línea de tiquet | 52 | |
| Nº. de línea | 100 | Número de línea |
| Código artículo | 101 | Código de articulo<br>Si es cero es una línea de venta con entrada de precio. |
| Precio | 102 | Precio de venta. Con punto decimal virtual. |
| Cantidad | 103 | Puede ser peso en gramos o unidades según campo [tipo de línea] |
| Reservado 3 | 149 | |
| Tipo de línea | 104 | 4 bits altos :<br>0000 - venta a peso<br>0001 - venta unidades<br>0010 - venta peso negativo<br>0011 - venta unidades negativo<br>4 bits bajos :<br>0000 - línea no cancelada<br>0001 - línea cancelada |
| Descuento línea | 106 | Porcentaje de descuento aplicado a la línea de tiquet<br>bit 7: si esta a cero, no hay descuento<br>bit 0..6: porcentaje de descuento |
| Ficha de vacuno | 107 | Número de ficha de vacuno. |
| Importe | 105 | |

### 3.3 Respuesta error (IDF:000100)

Si hay algún error en el mensaje de preguntas de tiquets, se responde con el idf 0x000100 y en datos hay el código de error.

Los errores pueden ser:

- los descritos en el acceso a motor de base de datos
- 0x02: error consulta.

# 4. ESTADO DEL SISTEMA (IDF:000004)

## 4.1 MENSAJE DE PETICION DE ESTADO DEL SISTEMA

| Campo | Longitud | Descripción |
|---|---|---|
| FUN | 1 byte | Función. |

## 4.2 RESPUESTA A LA PETICION DE ESTADO DEL SISTEMA

| Campo | Longitud | Descripción | |
|---|---|---|---|
| Equipo 0 | byte 1 | Bit | Descripción |
| | | 0 | Estado en la red |
| | | 1 | Master / No master (1/0) |
| | | 2 | Backup / No backup (1/0) |
| | | 3 – 5 | Forma de venta |
| | | 6 | Totales |
| | | 7 | Tiquets |
| | byte 2 | 0 | Actualización artículos |
| | | 1 | Actualización vendedores |
| | | 2-7 | Reservados |
| | byte 3 | 0-7 | Mostrador |
| Equipo 2 ... | | ... | |
| ... Equipo 64 | | ... | |

Siempre se envía la información de 64 equipos (aunque no existan físicamente).

Dentro de un byte los bit se numeran de 0 a 7 correspondiendo 0 al menos significativo (LSB) y 7 al mas significativo (MSB).

Totales y tiquets:

0: No hay. 1: Hay datos.

La balanza backup no diferencia los datos de local o red, aunque no tenga datos en local indica que hay totales y tiquets en red.

Forma de venta:

1: Venta (001 en bits).
2: Super (010).
3: Envasado (011).
4: Autoservicio (100).
5: Mixto (101).

## 4.3 MENSAJE DE RESPUESTA ERROR (idf:000100)

Si hay algun error en el mensaje , se responde con el idf 0x000100 y en datos hay el codigo de error. Los codigos de error son los descritos en el acceso a motor de base de datos.

## 5. MENSAJES DE CONTROL DE IDIOMAS (IDF:000005)

RESERVADO PARA APLICACIONES PROPIAS

## 6. CONFIGURACION DE BALANZAS (IDF:000008)

RESERVADO PARA APLICACIONES PROPIAS

# 7. ANEXOS

## 7.1 ANEXO 1: STRING CODIGO DE BARRAS

En la siguiente tabla se muestra la correspondencia entre los caracteres guardados por la base de datos en los campos correspondientes a códigos de barras y la letra introducida por teclado, así como una breve descripción.

**Significado de los Bytes del String del Código de Barras:**

| DATO | Significado | Carácter |
|---|---|---|
| 0x00 | Nada | ‘ ‘ |
| 0x01 | Overflow (obsoleto) | ‘S’ |
| 0x02 | Peso | ‘W’ |
| 0x03 | Precio | ‘Z’ |
| 0x04 | Importe | ‘I’ |
| 0x05 | Ticket | ‘T’ |
| 0x06 | PLU | ‘P’ |
| 0x07 | Código | ‘C’ |
| 0x08 | Vendedor | ‘V’ |
| 0x09 | N. Línea | ‘A’ |
| 0x0A | N. Lote | ‘L’ |
| 0x0B | N. Orden | ‘0’ |
| 0x0C | Fabricante | ‘F’ |
| 0x0D | País | ‘M’ |
| 0x0E | Cliente | ‘X’ |
| 0x0F | Peso 2 decimales | ‘D’ |
| 0x10 | Balanza | ‘N’ |
| 0x11 | Mostrador | ‘Y’ |
| 0x12 | Importe Nacional | ‘J’ |
| 0x13 | Check Intermedio | ‘K’ |
| 0x14 | Dígito 0 | ‘0’ |
| 0x15 | Dígito 1 | ‘1’ |
| 0x16 | Dígito 2 | ‘2’ |
| 0x17 | Dígito 3 | ‘3’ |
| 0x18 | Dígito 4 | ‘4’ |
| 0x19 | Dígito 5 | ‘5’ |
| 0x1A | Dígito 6 | ‘6’ |
| 0x1B | Dígito 7 | ‘7’ |
| 0x1C | Dígito 8 | ‘8’ |
| 0x1D | Dígito 9 | ‘9’ |
| 0x1E | Unidades | ‘U’ |
| 0x1F | Grupo mostrador | ‘B’ |
| 0x20 | Importe neto | ‘Q’ |
| 0x21 | Peso neto incluyendo merma (bruto-tara) | ‘G’ |
| 0x24 | Familia (1) | ‘R’ |
| 0x25 | Tipo de IVA (1) | ‘S’ |

(1) Solo para la compilación especial T0K.00D, sino el valor será ‘0’.

## 7.2 ANEXO 2: RUTINAS CONVERSIÓN FECHA

*Rutinas auxiliares:*

```
#define EsBisiesto(a)      (!(a&3))
int Dias[12]={31,28,31,30,31,30,31,31,30,31,30,31};

long DiasAnyo (int i)// Días del año.
{
    return 365+EsBisiesto(i);
}

long DiasMes( int anyo, int mes )   // Días del mes
{
    long res = Dias[mes];
    if (mes==1) res += EsBisiesto(anyo);
    return res;
}
```

*Conversión de long a fecha en año, mes, días:*

```
void Long2Fecha (long tptr, short *Dias, short *Mes, short *Anyo)
{
    long  Mdias, ADias;

    Dias = tptr / 86400L + 1;
    Anyo = 1980;
    while( Dias > ( ADias = DiasAnyo( Anyo) ) )
    {
        Dias -= ADias;
        Anyo++;
    }
    Mes = 0;
    while( Dias > ( MDias = DiasMes( Anyo, Mes ) ) )
    {
        Dias -= MDias;
        Mes++;
    }
    Mes++;
}
```

*Conversión de fecha en año, mes, días a long:*

```
long Fecha2Long(short Dia, short Mes, short Anyo)
{
    long Fecha, anyo, mes, dia;
    long i;

    anyo = Anyo%100;
    anyo -= (1980%100);
    if (anyo <0) anyo=Anyo+100-(1980%100);
    Fecha = 0;
    for( i = 0; i < anyo; i++ )
        Fecha += DiasAnyo( 1980 + i );
    mes = Mes-1;
    for( i = 0; i < mes; i++ )
        Fecha += DiasMes( 1980 + anyo, i );
    Fecha += Dia - 1;
    return( Data * 86400L );
}
```