



ANEXO II.B

PLIEGO DE PRESCRIPCIONES TÉCNICAS PARTICULARES QUE HA DE REGIR EL CONTRATO DE SUMINISTRO E INSTALACION DE SISTEMAS DE BILLETAJE PARA LA FLOTA INTERURBANA DE AUTOBUSES DE LA RED DE CONSORCIOS DE TRANSPORTE DE ANDALUCIA MEDIANTE PROCEDIMIENTO ABIERTO.



*Red de Consorcios de Transporte
de Andalucía*



Librería de transportes

Documento de especificaciones técnicas

Referencia CTAN-RU-LT-DET

Versión 0.1

02/10/17



Índice

1. GLOSARIO DE TÉRMINOS UTILIZADOS	1
2. OBJETIVO	2
3. INTRODUCCIÓN A LA TARJETA MIFARE DESFIRE EV2	4
3.1. IDENTIFICADOR ÚNICO.....	4
3.2. ORGANIZACIÓN DE LA MEMORIA	5
3.2.1. Aplicaciones.....	5
3.2.2. Archivos	5
3.3. SEGURIDAD.....	7
3.3.1. Claves en las aplicaciones.....	7
3.3.2. Seguridad en los archivos.....	7
3.4. RESUMEN DE COMANDOS DE LA TARJETA.....	9
4. DEFINICIÓN GENERAL DE LA API.....	11
4.1. CONVENIO DE INTERCAMBIO DE INFORMACIÓN.....	11
4.1.1. Convenio de orden MSB y LSB.....	11
4.1.2. Convenio de organización del contenido.....	12
4.1.3. Convenio de espacio reservado para los arrays data_in y data_out.....	13
4.1.4. Registro TLV de Cabecera obligatorio en data_in y data_out.....	13
4.1.5. Posibles registros TLV en data_out para cualquier función.....	14
4.2. RESUMEN DE LAS FUNCIONES DISPONIBLES EN LA LIBRERÍA (API).....	16
4.3. OPERATIVA.....	17
4.4. REDES SOPORTADAS	18
5. FUNCIONES DE LA API.....	20
5.1. VERSION	20
5.1.1. Descripción y uso.	20
5.1.2. Sintaxis.....	20
5.1.3. Entradas.....	20
5.1.4. Salidas.....	20
5.2. INIT.....	22
5.2.1. Descripción y uso.	22
5.2.2. Sintaxis.....	22
5.2.3. Entradas.....	22
5.2.4. Salidas.....	29
5.3. PROCESS.....	30
5.3.1. Descripción y uso.	30
5.3.2. Sintaxis.....	30
5.3.3. Entradas.....	30
5.3.4. Salidas.....	38
5.3.5. Relación de Acciones <-> Respuestas.....	47
5.4. LASTERROR.....	49
5.4.1. Descripción y uso.	49
5.4.2. Sintaxis.....	49
5.4.3. Entradas.....	49
5.4.4. Salidas.....	49
5.5. GETINFO	50
5.5.1. Descripción y uso.	50
5.5.2. Sintaxis.....	50
5.5.3. Entradas.....	50
5.5.4. Salidas.....	50
5.6. RESUMEN DE TIPOS EN "DATA_IN".....	55
5.7. RESUMEN DE TIPOS EN "DATA_OUT"	55
6. TABLA ASCII DE CARACTERES DE TEXTO.....	57
7. TABLA DE CARACTERES DE TRATAMIENTO ESPECIAL	58



Histórico

Fecha	Versión	Descripción
02/10/2017	0.1	Documento inicial



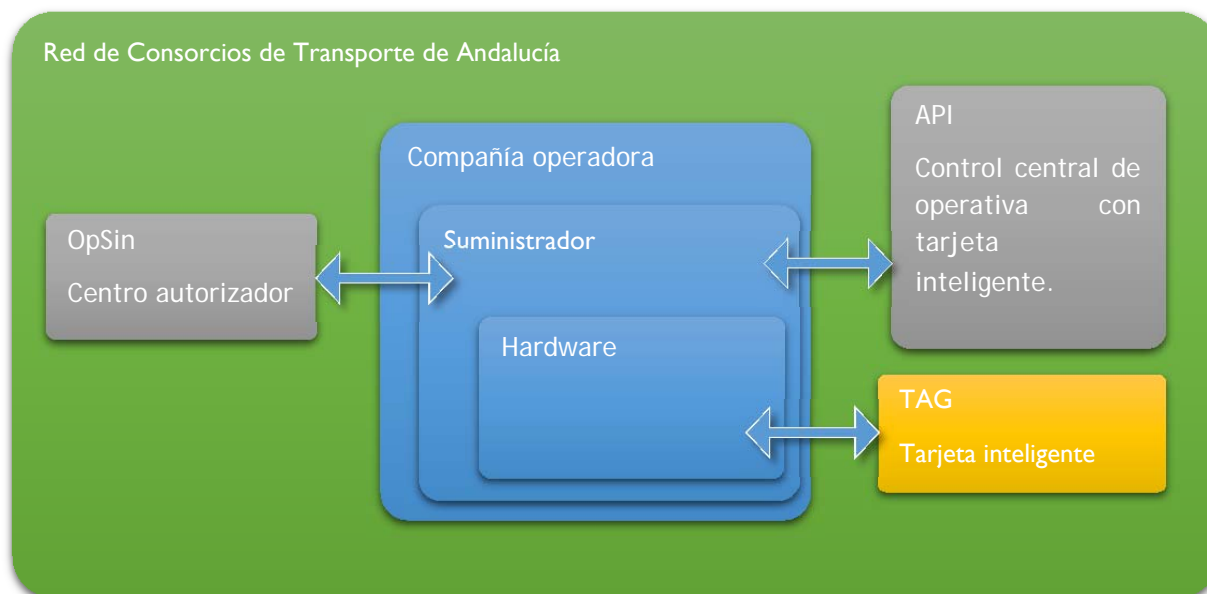
1. GLOSARIO DE TÉRMINOS UTILIZADOS

Termino	Descripción
Actuar con la tarjeta	Cualquier procedimiento de operación de venta, carga, recarga, cancelación, inspección, etc... que puede ser efectuada en un TAG
AID	Identificador de una aplicación de la tarjeta Desfire EV2.
AMK	Application Master Key o Clave maestra de la aplicación. Indica la clave principal de una aplicación con AID ≠ 00 00 00
API	Interface para la Programación de Aplicaciones. Se tratará de funciones externas que ofrece funciones necesarias para el desarrollo de un determinado software.
CID	Código de identificación de la acción a ejecutar. Cada registro/acción retornado en "data_out" de la función "Process()" dispone de un identificador numérico único. Este valor es tomado como referencia para los resultados de la acción.
FID	Identificador de un archivo, contenido dentro de una aplicación de la tarjeta Desfire EV2.
Hardware	Hace referencia a cualquier dispositivo utilizado para realizar las actuaciones con las tarjetas.
HOST o Software suministrador	Programa de la empresa suministradora que requiere la utilización de la librería para realizar transacciones con la tarjeta Desfire EV2
LSB	Posición menos significativa en un array de información
MK	Master Key o clave maestra. Indica la clave principal de la tarjeta, perteneciente a la aplicación AID = 00 00 00.
Modulo Centralizador	Conjunto de archivos que conforman la API
MSB	Posición más significativa en un array de información
Notaciones numéricas	En este documento se representaran varias notaciones numéricas que serán diferenciadas con la siguiente norma: Decimales: Representaremos el numero tal cual. Ejemplo 137. Hexadecimales : representado en notación "0x", ejemplo 0x123F Binarios: representados con una "b" al final, ejemplo 010011b
RAW	Información digital en bruto
Suministrador	Empresa suministradora de los dispositivos de venta, recarga, cancelación, inspección a las diferentes compañías anexadas a la operatividad del Consorcio de Transportes de Andalucía y que requieran del uso de la API.
TAG	Indica la tarjeta de proximidad DesFire EV2.
TLV	Indicativo de estructura de datos Tipo Longitud Valor.
UID	Identificador único de la tarjeta Desfire EV2.
Viajero o Usuario	Persona física propietaria de la tarjeta y por lo tanto al que se debe ofrecer toda la información necesaria de la transacción efectuada sobre su tarjeta.

2. OBJETIVO

Los requerimientos establecidos para la migración tecnológica de la tarjeta de la Red de Consorcios de Transportes de Andalucía desde su actual soporte MiFare Classic al nuevo soporte Desfire EV2, para cualquier operativa asociada (venta, recarga, uso, etc...), establecen que cualquier actuación sobre las tarjetas se realicen de forma exclusiva y centralizada por una única librería independiente y de uso obligado por cualquier sistema de Suministrador que requiera la interacción con la tarjeta de transporte.

Este marco de actuación queda reflejado en el siguiente esquema:



Cualquier software del suministrador/es de la solución de determinado operador de transporte, deberá utilizar la librería “*módulo centralizador*” para interactuar con la nueva tarjeta de transporte del Consorcio basada en tecnología DESFIRE EV2.

Mediante llamadas a las distintas funciones del API de esta librería, el software del suministrador será capaz de realizar operaciones de Venta, Recarga y Cancelación sobre las tarjetas de los usuarios.

Cuando hacemos referencia en este documento a *módulo centralizador*, se hace referencia al conjunto de archivos que conforman la librería que contiene las distintas operaciones contenidas en su API.

El uso del API de la librería establece un marco de trabajo único para cualquier suministrador de soluciones, asegurando los siguientes objetivos:

- **Seguridad.** Al existir un único *módulo centralizador* para las operaciones con la tarjeta, todas las operaciones que requieran autenticación y/o el uso de determinadas credenciales de acceso son procesadas/calculadas por dicho módulo y solo conocidas por él. Además también incluye todos los procedimientos de control lógico a realizar en las diferentes operativas de funcionamiento, venta, recarga, cancelación e inspección.
- **Flexibilidad.** El *módulo centralizador* estará preparado y desarrollado para poder ser utilizado en los entornos de desarrollo más actuales, ya que adopta la norma ISO (ISO/IEC 9899:1990) permitiendo la portabilidad entre plataformas y arquitecturas, aunque, para ello sería necesario disponer del módulo definitivo (“release”) en el formato de salida requerido por el suministrador del operador en cuestión.



De forma inicial, se dispondrá de las librerías en los siguientes formatos

Sistema Operativo	Formato
Windows 32 bits	Librería dinámica DLL nativa
Linux	Librería .lib con cabecera de prototipo .h

- **Centralización.** El uso del *módulo centralizador* se basará en archivos de librerías de enlace directo (no será necesario ningún tipo de instalación adicional) de esta forma todos los sistemas que hagan uso de ella dispondrán de acceso a posibles actualizaciones de la misma. En caso de ser necesario una modificación/ampliación de la librería se podría utilizar el método “reemplazar y listo” para continuar con su funcionamiento.

Existirá disponible un método que permita conocer la versión de la librería en uso.

- **Independencia del hardware.** Esto permite mayor facilidad en su integración ya que el *módulo centralizador* es totalmente independiente del hardware utilizado, ya que este es responsabilidad de funcionamiento y uso por parte del Suministrador del mismo.

En este documento veremos como todas las funciones de la API para operar con las tarjetas de proximidad están basadas en órdenes básicas o primitivas.

3. INTRODUCCIÓN A LA TARJETA MIFARE DESFIRE EV2

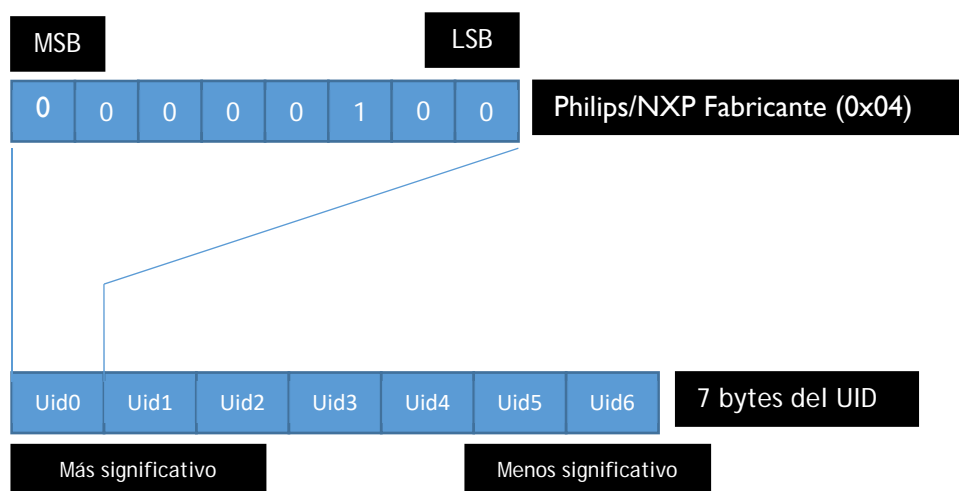
Este apartado repasa brevemente, sin entrar en profundidad, los detalles de la tarjeta Desfire EV2, destacando los puntos que tienen más relación con la librería.

La 'MiFare Desfire EV2' es una tarjeta de memoria que para la transmisión y recepción de datos con un lector utiliza una comunicación por radio frecuencia sin necesidad de utilizar baterías. Dentro del estándar ISO 14443 A y cumple con las 4 primeras normas de dicho estándar.

Existen varios modelos según su capacidad de almacenamiento, 2 kbytes, 4 kbytes y 8 kbytes. El tratamiento de todas ellas es idéntico, ya que solo cambia su tamaño de almacenamiento. En este proyecto serán utilizadas las de 4 kbytes.

3.1. IDENTIFICADOR ÚNICO

Dispone de un identificador "único" UID de 7 bytes. El cual dispone en su byte más significativo MSB el código de fabricante de la tarjeta, que en el caso de Philips/NXP su código es 0x04. De esta forma se establece la norma de lectura del UID para su control y representación, tanto en decimal como en hexadecimal.



Un ejemplo de UID podría ser Hexadecimal **0x042A59DAE82280**, Decimal **1172465319944832**

Este UID es establecido por el fabricante de la tarjeta y no es posible cambiarlo por ningún método.



3.2. ORGANIZACIÓN DE LA MEMORIA

La tarjeta tiene la capacidad de almacenar hasta 28 aplicaciones (AID) y en cada una de ellas hasta 16 archivos (FID).

Se debe tener en cuenta, como detalle, que el borrado de aplicaciones y/o archivos es “lógico”, es decir no se borra físicamente en la tarjeta el espacio ocupado por dicho elemento, por lo que existen métodos que permiten recuperar dicho espacio ocupado.

3.2.1. Aplicaciones

La representación de una aplicación, AID, está compuesta por 3 bytes de propósito general, o sea, cuando es creada una nueva aplicación, el valor establecido en estos 3 bytes puede ser cualquiera, con la excepción de que dicho AID no sea el Hexadecimal **00 00 00** y que no esté ya creada.

La AID hexadecimal **00 00 00** es especial, ya que se trata por especificaciones de la propia estructura de la tarjeta Desfire EV2 de la aplicación “master”. Esta aplicación **NO** puede contener Archivos y sus propiedades de acceso son diferentes al resto.

El resto de aplicaciones en hexadecimal **00 00 01** a la **FF FF FF**, pueden contener hasta 16 archivos como máximo y cada uno de ellos pueden ser de diferentes tipos de organización de datos.

Para poder crear/borrar aplicaciones, es necesario tener seleccionada la aplicación **00 00 00**.

Para poder crear/borrar archivos, es necesario tener seleccionada la aplicación contenedora de dicho archivo, como se ha comentado, la aplicación **00 00 00** **NO** permite contener archivos.

Los derechos de acceso de la Aplicaciones son detalladas más adelante.

3.2.2. Archivos

La representación de un archivo, FID, solo la compone un byte de propósito general, o sea, puede tener el valor desde **00** a **FF**. De forma independiente y común para cada uno de los archivos, se dispone de dos parámetros de relevante importancia, establecidos durante la creación:

- Derechos de acceso según las claves establecidas en la aplicación contenedora, explicados más adelante
- Método de cifrado para la codificación de las comunicaciones, que pueden ser:
 - Comunicación en plano.
 - Comunicación en plano securizada por DES/3DES MAC.
 - Comunicación cifrada en DES/3DES.

Los archivos pueden ser de 5 tipos diferentes, cada uno de ellos dispone de ciertas propiedades de control que indicamos a continuación:

- Archivo de datos estándar, código 0x00.

Se trata de un archivo de almacenamiento de información estándar con un tamaño en bytes fijo, especificado en su creación. Se debe tener en cuenta que la organización de estos archivos internamente consumen bloques de 32 bytes (un archivo de, por ejemplo 1 byte, ocupara en la tarjeta 32 bytes).

- Archivo de datos con backup, código 0x01.

Se trata de un archivo de almacenamiento exactamente igual que el anterior, pero con la peculiaridad de que dispone de un duplicado o backup, por lo tanto debe tenerse en consideración que ocupará el doble de espacio en la tarjeta.

Durante toda su operativa, las transacciones se realizarán en el archivo duplicado, siendo necesario realizar una operación de “*commit transaction*” para que la información sea trasladada al archivo original.

La tarjeta solo permite disponer de 8 archivos de este tipo como máximo y obligatoriamente deben estar numerados con un FID comprendido entre **00** y **07**.

- Archivo de dato de valor (siempre con backup), código 0x02.

Permite controlar un valor numérico de 32 bits con signo. En su operativa se pueden establecer el valor mínimo, el valor máximo y la posibilidad de operativa a crédito activada o desactivada.

Este tipo de archivos siempre está respaldado por backup, por lo que será necesario la utilización de la operación “*commit transaction*” para que la información sea trasladada al archivo original.

- Archivo de datos con registros lineales (siempre con backup), código 0x03.

Permite crear un archivo que permite el almacenamiento de múltiples registros del mismo tamaño en bytes. Será necesario indicar el tamaño del registro en bytes y la cantidad máxima de registros almacenados. El tamaño total del archivo será calculado multiplicando el tamaño del registro por la cantidad de ellos.

Para poder acceder a los datos, tanto en lectura como en escritura, será necesario indicar la posición del registro a la que se desea acceder.

Este tipo de archivos siempre está respaldado por backup, por lo que será necesario la utilización de la operación “*commit transaction*” para que la información sea trasladada al archivo original.

- Archivo de datos con registros lineales cíclicos (siempre con backup), código 0x04.

Este tipo de archivos en su definición es igual al anterior, permitiendo almacenar múltiples registros de un tamaño específico, pero su funcionamiento es controlado de forma cíclica, o sea, una vez el archivo tiene lleno completamente todos sus registros, comenzará a sobrescribir el más antiguo. Este funcionamiento lo realiza automáticamente la tarjeta.

Este tipo de archivos siempre está respaldado por backup, por lo que será necesario la utilización de la operación “*commit transaction*” para que la información sea trasladada al archivo original.



3.3. SEGURIDAD

3.3.1. Claves en las aplicaciones

La aplicación master AID = 00 00 00 SOLO dispone de 1 clave denominada MASTER KEY (MK), ubicada única y exclusivamente en la posición 0. Las propiedades de esta clave pueden ser:

- Permiso de cambiar la Master Key sin la autenticación con la Master Key (si/no).
- Permiso de listar las aplicaciones sin autenticación con Master Key (si/no).
- Permiso de crear y borrar aplicaciones sin la autenticación con la Master Key (si/no).
- Permiso de cambiar estos parámetros sin la autenticación con la Master Key (si/no).

En cualquier otra aplicación AID \neq 00 00 00 se puede disponer de hasta 14 claves configurables, donde la clave 0 es denominada como la “Application Master Key” (AMK):

Esta cantidad de claves se establece en el momento de la creación de la aplicación, para poder crear aplicaciones será necesario estar seleccionada la aplicación AID = 00 00 00 y tener los privilegios adecuados. Las propiedades de este conjunto de claves pueden ser:

- Es necesario la “Application Master Key” para cambiar cualquier clave.
- Es necesario la autenticación con la clave “N” (1...13) para cambiar cualquier clave.
- Es necesario la autenticación con la clave “N” (1...13) para cambiar la clave “N”.
- Todas las claves están congeladas.
- Permiso de cambiar estos parámetros sin la autenticación de la “Application Master Key”, (si/no, en este caso, estos parámetros se consideran congelados).
- Permiso de crear y borrar archivos sin necesidad de autenticación con la “Application Master Key” (si/no).
- Permiso para listar y leer propiedades de archivos sin necesidad de autenticación con la “Application Master Key” (si/no).
- Permiso de cambiar la “Application Master Key” sin la autenticación con la propia “Application Master Key” (si/no, en este caso la “Application Master Key” está congelada).

3.3.2. Seguridad en los archivos.

El acceso a los datos almacenados en los archivos está protegido individualmente por cada aplicación, en la cual, como se ha visto en el punto anterior, puede contener hasta 14 claves diferentes (0...13).

También es posible indicar como índice de clave los valores 14 y 15, que indican:

- 14 = Acceso libre, no es necesario ninguna clave para la realizar la operación.
- 15 = Acceso denegado, no es posible realizar la operación (operación congelada).

Por cada archivo será necesario indicar que clave (de la 0 a la 15) tiene derecho a realizar cada una de las siguientes operaciones:

- Lectura, leer valor y decrementos.
- Escritura, leer valor, decrementos y crédito limitado.



- Lectura/Escritura, leer valor, decrementos, crédito limitado e incrementar valor.
- Configuración de estos derechos de acceso.

Por ejemplo, para una AID concreta, además de su “Application Master Key (AMK)” (índice = 0), se ha añadido una nueva clave con el índice 1 la cual se desea utilizar para “solo lectura” en el archivo 00, por lo tanto, en la creación del archivo 00 es necesario indicar los siguientes derechos de acceso:

Derecho de acceso	Índice de clave
Lectura, leer valor y decrementos.	1
Escritura, leer valor, decrementos y crédito limitado.	0
Lectura/Escritura, leer valor, decrementos, crédito limitado e incrementar valor.	0
Configuración de estos derechos de acceso.	0

Otro aspecto que es definido en la creación de los archivos es el nivel de seguridad para las comunicaciones, que pueden ser acorde a las siguientes:

- Comunicaciones en plano.
- Comunicaciones DES/3DES MAC
- Comunicaciones completamente encriptadas DES/3DES



3.4. RESUMEN DE COMANDOS DE LA TARJETA

Detallamos a continuación una tabla con un resumen general de las acciones de operaciones que podrán ser utilizados de forma general con la tarjeta Desfire EV2. Referencia para todos los registros de órdenes que se explicaran más adelante en este documento.

Se debe tener en cuenta que el detalle de estas acciones cumple con la norma establecida como estándar por Philips Semiconductors en sus especificaciones de producto MF3 IC D40.

Ámbito	Comando	Detalles
Seguridad	0x0A	Autenticar
	0x54	Cambiar propiedades de seguridad de la clave maestra
	0x45	Leer propiedades de seguridad de la clave maestra
	0xC4	Cambiar claves
	0x64	Leer versión de claves
Nivel de tarjeta	0xCA	Crear aplicación
	0xDA	Borrar aplicación
	0x6A	Listar aplicaciones existentes.
	0x5A	Seleccionar una aplicación.
	0xFC	Formatear completamente la memoria
	0x60	Leer versión de fabricación
Nivel de aplicación	0x6F	Listar archivos de la aplicación seleccionada
	0xF5	Leer información de un archivo específico
	0x5F	Cambiar los parámetros de acceso de un archivo
	0xCD	Crear archivo de datos estándar.
	0xCB	Crear archivo de datos estándar con backup
	0xCC	Crear archivo de dato de valor.
	0xC1	Crear archivo de registros lineales.
	0xC0	Crear archivo de registros lineales cíclico.
	0xDF	Borrar archivo.
Nivel de archivos	0xBD	Leer datos de archivo estándar o con backup
	0x3D	Escribir datos en archivo estándar o con backup
	0x6C	Leer valor de archivo de valor.
	0x0C	Incremento del valor en archivo de valor.
	0xDC	Decremento del valor en archivo de valor.



	0x1C	Incremento del valor limitado por el crédito establecido.
	0x3B	Escribir datos en un registro en un archivo de registros lineales o cíclicos.
	0xBB	Leer datos de registros de un archivo de registros lineales o cíclicos
	0xEB	Borra completamente un archivo de registros lineales o cíclicos.
	0xC7	Confirmar transacción. Valida todas las escrituras previamente realizadas en archivos de datos estándar con backup, de valor y registros.
	0xA7	Invaldar transacción, anula todas las escrituras previamente realizadas en archivos de datos estándar con backup, de valor y registros.

4. DEFINICIÓN GENERAL DE LA API

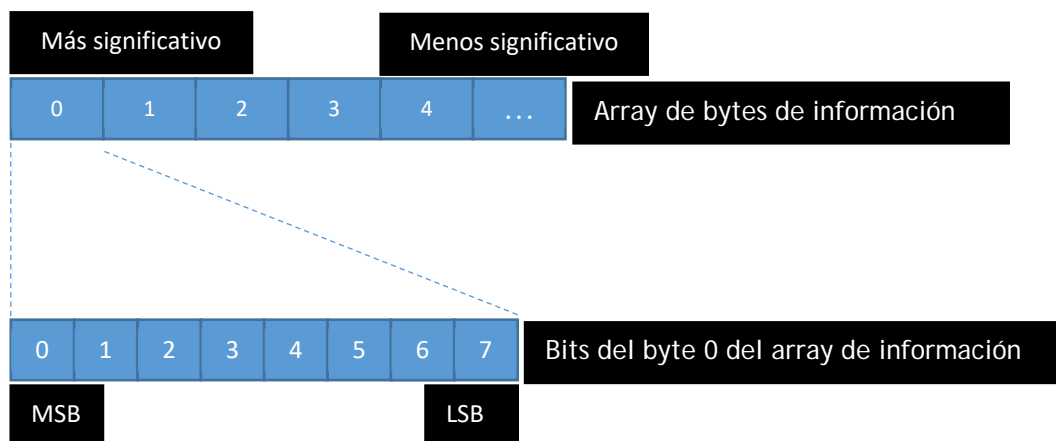
Debido a los requerimiento de seguridad en el uso de la nueva tarjeta, la manera de utilizar la librería por el software del suministrador establece una operativa donde el elemento central es la propia librería, siendo esta la encargada de centralizar y securizar todo el proceso de una determinada transacción: Venta, Recarga y Cancelación. El suministrador hace las funciones de intermediario entre su hardware (lector físico de tarjetas sin contacto) y la librería, abstrayéndose completamente de la operativa interna.

4.1. CONVENIO DE INTERCAMBIO DE INFORMACIÓN.

4.1.1. Convenio de orden MSB y LSB

Como normal general, es importante dejar claro la referencia del posicionamiento del byte y del bit más significativo y menos significativo en caso de los array's de datos.

El bit/byte Más significativo estará situado en el extremo izquierdo y el menos significativo en el extremo derecho. Queda esta información mucho más clara y detallada con la siguiente representación gráfica:



Ejemplo 1: si se desea transmitir en 1 solo byte el valor 137 en decimal, 0x89 en hexadecimal, quedaría representado de la siguiente forma:

Bytes	0
	0x89

Bits	0	1	2	3	4	5	6	7
	1	0	0	0	1	0	0	1

$$137 = 0x89 = 10001001b$$

Ejemplo 2: si se desea transmitir en 2 bytes el valor 30000 en decimal 0x7530 en hexadecimal, quedaría representado de la siguiente forma:



Bytes	0	1
	0x75	0x30

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	1	1	1	0	1	0	1	0	0	1	1	0	0	0	0

$$30000 = 0x7530 = 111010100110000b$$

4.1.2. Convenio de organización del contenido

Se define un convenio para el intercambio de la información establecido en todas las funciones de llamada a la API que hacen referencia a un puntero a datos ***data_in** o ***data_out**, estos array's de información deben cumplir con dicho convenio en caso contrario la información será rechazada.

La información será tratada en formato TLV (Tipo/Longitud/Valor), cada registro de información deberá cumplir dicha norma. En todas las tramas, el primer byte indica T (Tipo), los 2 siguientes indica L (Longitud) y el resto tendrá una longitud de "L" bytes que indica el V (Valor), es decir L indica la cantidad de bytes del registro **sin incluir ni T ni L**.

Bytes	0	1	2	3	.	.	.	N
Campo	T	L	V					

En un mismo array de datos "data_in" o "data_out" pueden ir tantos registros TLV como sea necesario, pero siempre serán iniciados por un registro TLV denominado registro de cabecera, el cual además de información general de operatividad, indica la cantidad de registros TLV que preceden.

```
<TLV de cabecera> <registro TLV 1> [... <registro TLV N>]
```

Por ejemplo, si es llamada a la función "Version()", esta devolverá en su array "data_out" la siguiente información estructurada:

```
<TLV de cabecera> <TLV de versión de núcleo> <TLV de versión de
validación> <TLV de versión de inspección> <TLV de comprobación
de datos>
```


4.1.3. Convenio de espacio reservado para los arrays data_in y data_out

La adopción de la norma ISO (ISO/IEC 9899:1990), implica que los “punteros” de datos de intercambio NO dispone de método para determinar el tamaño de los mismos, por lo tanto se establece, por el ámbito de uso, que el responsable de asignar el tamaño adecuado para el intercambio de la información en los arrays “data_in” y “data_out” sea HOST y **tengan siempre una longitud de 1024 bytes (1 Kb)**.

La primera función de la librería será establecer a 0 el espacio reservado del array “data_out” mediante “set” directo a memoria.

4.1.4. Registro TLV de Cabecera obligatorio en data_in y data_out.

Detallamos ahora la sintaxis que debe cumplir el registro TLV de cabecera en todos los casos:

Byte	Campo	Detalles
0	T	0xF1
1...2	L	Longitud en bytes del registro, sin incluir ni T ni L. valor 12.
3	F1_VER	Versión del registro de cabecera. 1 = versión 1.
4...5	F1_CNT	Contador de la secuencia del proceso. Solo necesario para la función “Init()” y “Process()”. Este contador se inicia a 0 cuando se inicia el proceso con la función de la API “Init()” y será incrementado en 1 de forma balanceada por el HOST y la librería, en las llamadas consecutivas a la función “Process()”. La librería será la encargada de detectar que este valor tiene su secuencialidad correcta respecto al anterior, en caso de error producirá un retorno =2 (error) con 06_CERR = 1 = Error del campo CNT en cabecera. En la cabecera de la primera llamada de la función “Init()” → CNT=0 En la cabecera de la siguiente llamada a “Process()”, en data_in, CNT=1 y en data_out, CNT=2 En la cabecera de la siguiente llamada a “Process()”, en data_in, CNT=3 y en data_out, CNT=4 ...
6..7	F1_FIRMA	Código de seguridad Se trata de un código de control obtenido a partir de un algoritmo realizado con todos los registros que forman el “RAW” complete de la información. Se detalla en un punto más Adelante.
8..9	F1_NREG	Nº de registros TLV contenidos a continuación de este registro de cabecera, o sea, sin contar el propio registro de cabecera.
10	F1_HH	Hora del sistema
11	F1_MM	Minuto del sistema
12	F1_SS	Segundos del sistema
13...14	F1_RFU	Reservados para futuro uso



4.1.5. Posibles registros TLV en data_out para cualquier función.

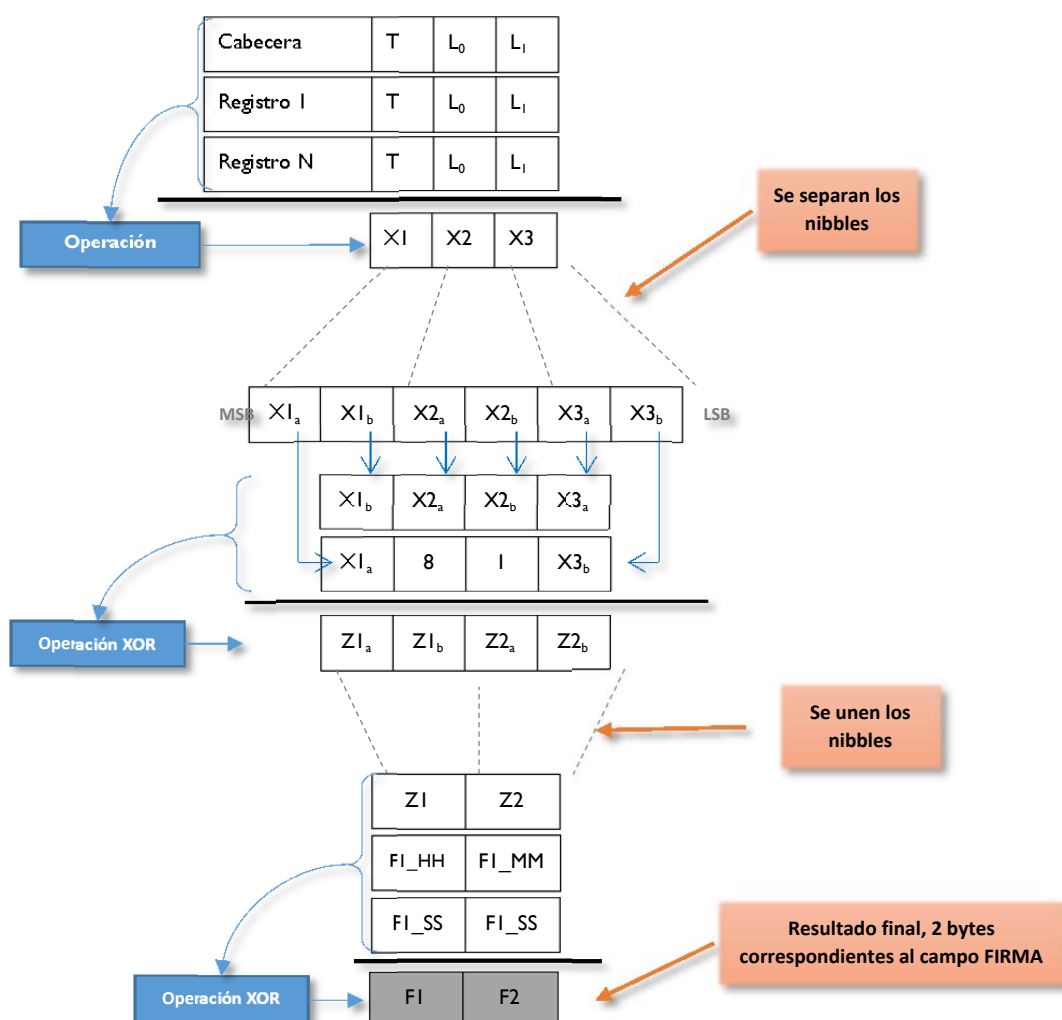
- Tiempo de ejecución de la función:

Byte	Campo	Detalles
0	T	0xF2
1...2	L	Longitud = 4
3...6	F2_TC	Tiempo en milisegundos consumido por la ejecución completa de la función

4.1.5.1. Cálculo de la firma de seguridad

En el registro de cabecera hay un campo llamado **FIRMA** que debe ser calculado “antes” de realizar el envío de la información hacia la librería, de igual forma, la librería realizará el cálculo del campo FIRMA antes de producir la respuesta.

El cálculo es el resultado de efectuar el OR exclusivos “XOR” de una forma concreta a los bytes T y L de **TODO**s los registros que conforman el RAW completo de información, incluido el registro de cabecera, y para añadir variabilidad se incluirá también los campos FI_HH, FI_MM y FI_SS. El cálculo se indica en el siguiente esquema:





4.2. RESUMEN DE LAS FUNCIONES DISPONIBLES EN LA LIBRERÍA (API)

En esta tabla muestra un resumen de todas las funciones que dispone la API.

Función	Detalles
Version()	Función genérica que puede ser llamada en cualquier momento deseado que retorna las diferentes versiones de la API.
Init()	Función de inicio de operaciones, es obligatorio su llamada antes de iniciar el ciclo de operaciones con la función "Process()".
Process()	Función de proceso principal, permite la ejecución paso a paso de la operativa con una tarjeta.
LastError()	Función que retorna el último error interno producido en la función "Init()" o "Process()", en caso de se haya producido algún error. Este error es reseteado en la llamada a la función "Init()".
GetInfo()	Función que retorna la información visual o impresa que es necesario producir al viajero una vez finalizado el proceso completo con la tarjeta (retorno de "Process()" = 1), esta información es reseteada en la llamada de "Init()".

4.3. OPERATIVA

La manera en la cual será utilizada la librería será simple, la llamada a las funciones de la API tienen que ser ejecutadas en un determinado orden “Init() → Process() → Process() → ... → GetInfo()”, y el software del suministrador es el encargado de hacer efectivas las acciones recibidas por la función “Process()” durante el ciclo de llamadas.

La función principal “Process()”, opera como un *controlador* de operaciones según la información recibida por el HOST que utiliza la librería en el parámetro de entrada “data_in”. “Process()” siempre devuelve 2 valores:

- El valor de retorno indicando “continuar/finalizado/error”
- Un ‘array’ de registros TLV denominado “data_out” con la información de acciones a tomar por el software del suministrador (HOST).

Cualquier transacción (VENTA, RECARGA, CANCELACIÓN, ETC ...) con una nueva tarjeta colocada en el lector o dispensador (PICC) contactless, es iniciado siempre por el software del suministrador (HOST).

La primera llamada a la librería siempre debe ser siempre la función “Init()”:

1. Se indica que se desea hacer (vender, validar, etc...).
2. Parámetros necesarios dependiendo del tipo de transacción con la tarjeta: Venta / Recarga / Cancelación, etc.

Tras la llamada a “Init()”, el HOST realizará un ciclo de llamadas consecutivas a la función “Process()”, esta retornará al HOST (en data_out) la lista de acción o acciones que debe realizar con la tarjeta presente (son responsabilidad del host de hacerlas efectiva físicamente) y, una vez finalizada la ejecución, informar de nuevo a la API con la función “Process()” de todos los resultados obtenidos de la ejecución de las acciones de forma individual. Este ciclo continuará hasta que la función “Process()” indique fin o error.

Independientemente de la operativa a realizar, en la primera llamada a la función “Process()” se espera que el HOST informe de los datos generales del TAG presente, entre ellos el UID del mismo, mediante el registro TLV “0xB0”

Todas las acciones recibidas en los registros data_out de la función “Process()” deben tener un resultado/respuesta concreta que indicar en los registros data_in de la siguiente llamada a la función “Process()” (si este indica continuar).

Si durante la ejecución de alguna acción recibida de la función “Process()”, ya sea referente a la tarjeta o al lector, produce algún error, **el HOST debe interrumpir la ejecución de las acciones** e informar a la función “Process()” del resultado/respuesta de las acciones ejecutadas hasta el momento (incluida la que produjo el error).

```
graph TD
    Start([Load Library]) --> WaitTag[Esperamos detección TAG]
    WaitTag --> IsPresent{¿Presente?}
    IsPresent -- SI --> Init[Init()]
    IsPresent -- NO --> WaitTag
    Init --> HostInfoIn[HOST informa en 'data_in' con F1_CNT=0 y registro 0xB0 a "Process()"]
    HostInfoIn --> Process[Process()]
    Process --> IsError{¿Error?}
    IsError -- SI --> GetInfo[GetInfo()]
    IsError -- NO --> IsFin{¿Fin?}
    GetInfo --> ProduceInfo[Producir información al viajero]
    ProduceInfo --> WaitRetTag[Esperamos retirada de TAG]
    WaitRetTag --> WaitTag
    IsFin -- SI --> HostExecOut1[HOST ejecuta 'data_out' (Si hubiere)]
    HostExecOut1 --> GetInfo
    IsFin -- NO --> HostExecOut2[HOST ejecuta 'data_out']
    HostExecOut2 --> HostInfoOut[HOST informa con 'data_in' y F1_CNT=F1_CNT+1 a "Process()"]
    HostInfoOut --> Process
```

El diagrama de flujo describe el proceso de procesamiento de datos de un lector de tarjetas de acceso. Comienza con la carga de la biblioteca (Load Library), seguida de la espera de la detección de la TAG. Si la TAG está presente, se ejecuta la función Init(), lo que genera una solicitud de procesamiento al HOST. El HOST devuelve información a través de 'data_in' (conteniendo F1_CNT=0 y el registro 0xB0), lo que desencadena la ejecución de la función Process().

Desde Process(), se evalúa si hay un error. Si es así, se ejecuta GetInfo() para producir información al viajero y se espera la retirada de la TAG antes de volver a intentar la detección. Si no hay error, se verifica si se ha alcanzado el final del procesamiento (¿Fin?). Si es así, se ejecuta 'data_out' (si está disponible) y se devuelve la información al viajero. Si no se ha alcanzado el final, se ejecuta 'data_out' y se incrementa el contador F1_CNT, lo que genera una nueva solicitud de procesamiento al HOST.

Detallamos los códigos de redes soportadas.

Código de red	Descripción
1	Área de Sevilla
2	Bahía de Cadiz
3	Área de Granada
4	Área de Málaga
5	Área de Gibraltar



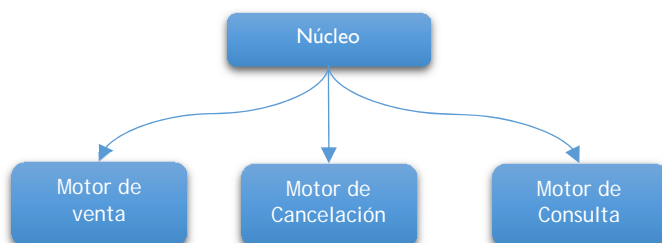
6	Almería
7	Jaén
8	Córdoba
9	Huelva

5. FUNCIONES DE LA API

5.1. VERSION

5.1.1. Descripción y uso.

Es importante disponer de las versiones operativas de los diferentes módulos de proceso que contiene la API, en su conjunto la API dispone de 4 versiones diferenciadas



Esta función puede ser llamada en cualquier momento sin que interrumpa o interfiera en cualquier otra llamada.

5.1.2. Sintaxis.

```
void Version( unsigned char *data_out )
```

5.1.3. Entradas.

Ninguna.

5.1.4. Salidas.

```
unsigned char *data_out
```

Array de información con los registros TLV del detalle de las versiones de los diferentes módulos además de información adicional para el programador.

Los registros son los siguientes:

- Versión del núcleo

Byte	Campo	Descripción
0	T	0x01
1...2	L	longitud = 4
3	01_V1	Valor de la versión mayor
4	01_V2	Valor de la versión menor.
5...6	01_BUILD	Valor de la compilación

- Versión del módulo de venta.



Byte	Campo	Descripción
0	T	0x02
1...2	L	longitud = 2
3	02_V1	Valor de la versión mayor
4	02_V2	Valor de la versión menor.

- Versión del módulo de validación

Byte	Campo	Descripción
0	T	0x03
1...2	L	longitud = 2
3	03_V1	Valor de la versión mayor
4	03_V2	Valor de la versión menor.

- Versión del módulo de inspección.

Byte	Campo	Descripción
0	T	0x04
1...2	L	longitud = 2
3	04_V1	Valor de la versión mayor
4	04_V2	Valor de la versión menor.

- Datos de comprobación para el desarrollador.

Este registro siempre contendrá la misma información y podrá ser utilizada para determinar la correcta interpretación de los valores.

Byte	Campo	Descripción
0	T	0x05
1...2	L	longitud = 7
3	05_TEST1	Valor de test de 1 byte = 100000011b = 0x83 = 131
4...5	05_TEST2	Valor de test de 2 bytes = 10000011 11100001b = 0x83E1 = 33761
6...9	05_TEST3	Valor de test de 4 bytes = 00000001 00000011 00001111 00001111 = 0x103070F = 16975631

5.2. INIT

5.2.1. Descripción y uso.

Esta función marca el inicio de la con un TAG, inicializa todo el entorno de control de la librería y establece los valores de inicio necesarios según la operativa indicada.

Es de obligatoria llamada antes de comenzar a operar con un nuevo TAG presente y solo debe realizarse 1 vez esta llamada al inicio para toda la operativa con dicho TAG.

La llamada a esta función provocará el “reset” interno de las variables que utiliza la librería para operar, una llamada en medio de un ciclo provocará un error en la transacción.

5.2.2. Sintaxis.

```
void Init(  const unsigned int id_operativa,
           const unsigned char * data_in,
           unsigned char * data_out )
```

5.2.3. Entradas.

```
const unsigned int id_operativa
```

Código que indica el tipo de operativa que se desea realizar.

Código	Operativa	Gestionado por el módulo
1	Compra	Venta
2	Recarga	Venta
3	Particularización (no es posible su anulación)	Venta
4	Consulta información	Inspección
5	Devolución	Venta
6	Sustitución	Venta
7	Regularización de saldo	Venta
8	Anulación	Venta
10	Cancelación	Cancelación
11	Anulación de la última cancelación	Cancelación



const unsigned char *data_in

Array de registros TLV con la información de los parámetros de inicio necesarios para realizar la operativa indicada.

- **Registro de datos de situación actual (módulo de cancelación).**

Parámetros básicos de la situación lógica del equipo que va a efectuar la operativa.

Aplicable a la red : Todas.

	Código	Operativa	
Necesario para	1	Compra	
	2	Recarga	
	3	Particularización (no es posible su anulación)	
	4	Consulta información	
	5	Devolución	
	6	Sustitución	
	7	Regularización de saldo	
	8	Anulación	
	10	Cancelación	•
	11	Anulación de la última cancelación	•

Byte	Campo	Descripción												
0	T	0xC0												
1...2	L	Longitud = 12												
3	C0_RED	Código de identificador de la red de integración. * Ver redes soportadas.												
4	C0_OPERADOR	Código de identificación operador												
5	C0_ZONA	Código de la zona actual												
6	C0_SZONA	Código de la subzona actual												
7...8	C0_LINEA	Código de la línea												
9...10	C0_PARADA	Código de la parada												
11	C0_SENTIDO	Código del sentido: <table><tr><th>Para sistemas cerrados</th><th>Para sistemas abiertos</th></tr><tr><td>0 = Ida</td><td>0 = Entrada</td></tr><tr><td>1 = vuelta</td><td>1 = Salida</td></tr><tr><td>2 = Circular ida</td><td></td></tr><tr><td>3 = Circular vuelta</td><td></td></tr><tr><td>9 = sin sentido</td><td></td></tr></table>	Para sistemas cerrados	Para sistemas abiertos	0 = Ida	0 = Entrada	1 = vuelta	1 = Salida	2 = Circular ida		3 = Circular vuelta		9 = sin sentido	
Para sistemas cerrados	Para sistemas abiertos													
0 = Ida	0 = Entrada													
1 = vuelta	1 = Salida													
2 = Circular ida														
3 = Circular vuelta														
9 = sin sentido														
12	C0_SISTEMA	Sistema 0 = Sistema abierto. 1 = Sistema cerrado												
13...14	C0_COMERCIO	Número del comercio												



- Registro de datos de situación actual (módulo de ventas).**

Parámetros básicos de la situación lógica del equipo que va a efectuar la operativa.

Aplicable a la red: Todas.

	Código	Operativa	
Necesario para	1	Compra	•
	2	Recarga	•
	3	Particularización (no es posible su anulación)	•
	4	Consulta información	
	5	Devolución	•
	6	Sustitución	•
	7	Regularización de saldo	•
	8	Anulación	•
	10	Cancelación	
	11	Anulación de la última cancelación	

Byte	Campo	Descripción
0	T	0xC1
1...2	L	Longitud = 6
3	C1_RED	Código de identificador de la red de integración. * Ver redes soportadas.
4	C1_OPERADOR	Código de identificación operador
5	C1_ZONA	Código de la zona actual
6	C1_SZONA	Código de la subzona actual
7...8	C1_COMERCIO	Número del comercio

- Registro de fecha y hora de operación.**

Indica la fecha y hora “efectiva” de la operativa a realizar. Lo normal será indicar en estos campos la fecha y hora actual del sistema que realiza la operación.

Aplicable a la red: Todas.

	Código	Operativa	
Necesario para	1	Compra	•
	2	Recarga	•
	3	Particularización (no es posible su anulación)	•
	4	Consulta información	
	5	Devolución	•
	6	Sustitución	•
	7	Regularización de saldo	•
	8	Anulación	•
	10	Cancelación	•
	11	Anulación de la última cancelación	•

Byte	Campo	Descripción
0	T	0xC5
1...2	L	Longitud = 6
3	C5_DIA	Día actual (1..31)



4	C5_MES	Mes actual (1..12)
5	C5_ANO	Año actual (17 = 2017)
6	C5_HORA	Hora actual (0..23)
7	C5_MINUTO	Minuto actual (0..59)
8	C5_SEGUNDO	Segundo actual (0..59)

- **Registro de parámetros auxiliares para la operativa de cancelación o anulación de la cancelación.**

Parámetros adicionales necesarios indicar en la operativa de cancelación o anulación.

Aplicable a la red: Todas.

Necesario para	Código	Operativa	
	1	Compra	
	2	Recarga	
	3	Particularización (no es posible su anulación)	
	4	Consulta información	
	5	Devolución	
	6	Sustitución	
	7	Regularización de saldo	
	8	Anulación	
	10	Cancelación	•
	11	Anulación de la última cancelación	•

Byte	Campo	Descripción
0	T	0xC6
1...2	L	Longitud = 2
3	C6_MIN_PB	Minutos para considerar passback también utilizado como tiempo de acompañante
4	C6_MIN_ANUL	Minutos para permitir anulación

- **Registro de tarificación dependiendo de saltos ubicados en la tarjeta.**

Parámetros de la tarifa a aplicar en aquellas cancelaciones cuyo precio depende del número de saltos configurados en la propia tarjeta.

Aplicable a la red: (I-Área de Sevilla).

Necesario para	Código	Operativa	
	1	Compra	
	2	Recarga	
	3	Particularización (no es posible su anulación)	
	4	Consulta información	
	5	Devolución	
	6	Sustitución	
	7	Regularización de saldo	
	8	Anulación	
	10	Cancelación	•
	11	Anulación de la última cancelación	

Byte	Campo	Descripción
------	-------	-------------



0	T	0xC7
1...2	L	Longitud = N los bytes 3 al 7 se repiten por cada salto/tarifa a indicar
3	C7_SALTOS	Nº de saltos Valor entre 0 y 15
4...5	C7_BASE	Precio en céntimos de euro de tarifa base
6...7	C7_ADICIO	Precio en céntimos de euro para tarifa adicional.
...		

- Registro de tarificación origen/destino.**

Parámetros de la tarifa a aplicar en aquellas cancelaciones cuyo precio es obtenido según los saltos zonales producidos entre zona de origen (obtenida de los parámetros de situación actual y la zona destino indicada en este propio registro.

Aplicable a la red: Todas.

Necesario para	Código	Operativa	
	1	Compra	
	2	Recarga	
	3	Particularización (no es posible su anulación)	
	4	Consulta información	
	5	Devolución	
	6	Sustitución	
	7	Regularización de saldo	
	8	Anulación	
	10	Cancelación	•
	11	Anulación de la última cancelación	

Byte	Campo	Descripción
0	T	0xC8
1...2	L	Longitud = 7
3	C8_ZDEST	Código de zona destino
4...5	C8_BASE	Precio en céntimos de euro de tarifa base
6...7	C8_ADICIO	Precio en céntimos de euro para tarifa adicional.
8...9	C8_PENALIZ	Precio en céntimos de euro de penalización por ruptura de transbordo.

La zona de origen es tomada del registro de situación actual.

- Registro de parámetros de transbordo.**

Información necesaria para determinar, durante el proceso de cancelación, si se ha producido transbordo o no.



Aplicable a la red: Todas.

Necesario para	Código	Operativa	
	1	Compra	
	2	Recarga	
	3	Particularización (no es posible su anulación)	
	4	Consulta información	
	5	Devolución	
	6	Sustitución	
	7	Regularización de saldo	
	8	Anulación	
	10	Cancelación	•
	11	Anulación de la última cancelación	

Byte	Campo	Descripción
0	T	0xC9
1...2	L	Longitud = N Los bytes 6 al 15 se repiten por cada salto a indicar.
3	C9_PET2	Porcentaje de control de descuento en transbordos para la segunda etapa
4	C9_PET3	Porcentaje de control de descuento en transbordos para la tercera y posteriores etapas.
5	C9_MGRA	minutos de gracia para transbordos
6	C9_SAL	Numero de saltos (entre 0 y 15)
7...8	C9_TG	Tiempo en minutos de tiempo máximo de transbordo global
9	C9_NG	Número máximo de transbordos globales
10...11	C9_TI	Tiempo en minutos de tiempo de transbordo interno
12	C9_NI	Número máximo de transbordos internos
13	C9_RIEI	Aplicar restricción IEI (Interno Externo Interno)
14	C9_TP	Tipo de tiempo de prevalencia (0=global/1=interno)
15	C9_NP	Numero de prevalencia (0=global/1=interno)
...		

- Registro de tipo de información a solicitar.**

Indica que tipo de información se debe obtener en la operativa de Consulta de Información. En el apartado de detalles de la función “GetInfo” se detalla cada uno de los campos que estarán disponibles como resultado en la ejecución de esta función.

Se debe tener en cuenta, que cuanto más información es solicitada, será necesario realizar más operativas de acceso y lectura a la tarjeta, lo que conllevará más tiempo de proceso.

Aplicable a la red: Todas.

Necesario para	Código	Operativa	
	1	Compra	
	2	Recarga	
	3	Particularización (no es posible su anulación)	
	4	Consulta información	•



5	Devolución	
6	Sustitución	
7	Regularización de saldo	
8	Anulación	
10	Cancelación	
11	Anulación de la última cancelación	

Byte	Campo	Descripción
0	T	0xCA
1...2	L	Longitud = 1
3	CA_TIPO	<p>Tipo de información a solicitar</p> <p>1 = Información básica de la tarjeta.</p> <p>Se comprende los datos de titularidad de la tarjeta e identificación y saldos del título.</p> <p>2 = Información general de la tarjeta.</p> <p>Además de la información anterior, se añade el estado del viaje actual.</p> <p>3 = Información completa de la tarjeta.</p> <p>Además de la información anterior, se añade todos los registros históricos de uso, de cargas y de reservas.</p>

- Registro de importe.**

Indica que importe debemos utilizar para realizar la operativa de carga/recarga y regularización del saldo.

Aplicable a la red: Todas.

Necesario para	Código	Operativa	
	1	Compra	•
	2	Recarga	•
	3	Particularización (no es posible su anulación)	
	4	Consulta información	
	5	Devolución	
	6	Sustitución	
	7	Regularización de saldo	•
	8	Anulación	
	10	Cancelación	
	11	Anulación de la última cancelación	

Byte	Campo	Descripción
0	T	0xCB
1...2	L	Longitud = 5
3	CB_SIGNO	<p>Signo del importe.</p> <p>Solo tiene sentido para la operativa de regularización de saldo.</p> <p>0 = Positivo</p> <p>1 = Negativo.</p>
4...7	CB_IMPTE	Importe representado en céntimos de euro.

- Registro de cantidad de pasajeros para anular.**

Indica un valor de cantidad de pasajeros necesaria para la operativa de anulación.



Aplicable a la red: Todas.

Necesario para	Código	Operativa	
	1	Compra	
	2	Recarga	
	3	Particularización (no es posible su anulación)	
	4	Consulta información	
	5	Devolución	
	6	Sustitución	
	7	Regularización de saldo	
	8	Anulación	
	10	Cancelación	
	11	Anulación de la última cancelación	•

Byte	Campo	Descripción
0	T	0xCC
1...2	L	Longitud = 1
3	CC_VIAJEROS	Cantidad de pasajeros

5.2.4. Salidas.

unsigned char *data_out

La función “Init” solo ofrece información general.

5.3. PROCESS

5.3.1. Descripción y uso.

Función que centraliza todas las operaciones a realizar con la tarjeta presente según la operativa indicada en la función “Init()”, que obligatoriamente deber haber sido llamada con anterioridad.

La primera llamada a la función “Process()”, la que inicia el ciclo, necesita solo el registro de información del TAG presente, registro de Tipo **0xB0**. A partir de dicho momento el HOST debe atender al código de retorno proporcionado, para la continuación o interrupción del ciclo.

- En caso de retorno = **2** erróneo, el HOST debe obtener el LastError y el GetInfo para determinar el motivo del error y la información a mostrar al viajero.
- En caso de retorno = **0**, el HOST debe “ejecutar” las acciones recibidas en el array “data_out” y conformar el array “data_in” para realizar una nueva llamada a la función “Process()”¹ (ciclo).
- En caso de retorno = **1**, el HOST debe “ejecutar” las acciones recibidas en el array “data_out”, en caso de que los hubiera. Si no fuera necesario ejecutar ninguna acción, el array “data_out” solo contendrá el registro de cabecera con el campo FI_NREG a 0. Tras ello, obtener los datos con el uso de la función “GetInfo()” para ofrecer la información al viajero.
- Es necesario que la ejecución de las acciones recibidas se realicen “**en el mismo orden**” que están establecidas en el array “data_out”.
- En caso de que la ejecución de alguna acción produzca un error, el HOST NO debe continuar con la ejecución de las acciones y solo informar de nuevo la función “Process()” con las que ya hubiere ejecutado, por supuesto, incluyendo la que produjo el error.
- Se entiende pues, que mientras la función “Process()” retorne el valor 0 (correcto y continuar), el HOST debe permanecer en el ciclo de llamadas.

5.3.2. Sintaxis.

```
unsigned int Process(          const unsigned char *data_in,
                               unsigned char *data_out )
```

5.3.3. Entradas.

```
const unsigned char *data_in
```

Registros TLV con la información según el ciclo actual del proceso.

En la primera llamada solo contendrá el registro con la información de la tarjeta presente (0xB0) y en las llamadas restantes contendrá la información de resultado requerida por las

¹ Información más detallada en el punto “relación de acciones ↔ respuestas”



acciones obtenidas en el array “data_out” retornado de la función “Process()” del ciclo anterior.

- **Información del TAG presente.**

Registro especial de obligado uso y único (junto con el registro de cabecera) en la primera llamada a la función “Process()”.

Tras la llamada a la función “Init()”, la primera llamada a la función “Process()” debe ser con este registro, el cual contiene la información e identificación de la tarjeta TAG que va a ser tratada.

Byte	Campo	Descripción
0	T	0xB0
1...2	L	Longitud = 10
3...9	BO_UID_TAG	7 bytes correspondientes al UID del TAG presente
10...11	BO_ATQA	2 bytes de la respuesta del request efectuado a la tarjeta
12	BO_SAK	1 byte de la respuesta de aceptación del comando select efectuado a la tarjeta

- **Información de resultado en la ejecución de la acción.**

Utilizado como respuesta de las acciones 0x10 (autenticar), 0x14 (seleccionar aplicación), 0x19 (grabar archivo), 0x1B (incrementar valor), 0x1C (decrementar valor), 0x1D (incrementar valor crédito), 0x1E (grabar registro), 0x20 (vaciar registros), 0x21 (confirmar transacciones) y 0x22 (invalidar transacciones), además de en cualquier caso de error producido en la acción.

Byte	Campo	Descripción
0	T	0xB1
1...2	L	Longitud=4
3	B1_CID	Código de identificación de la acción.
4	B1_UNIDAD	Código de la unidad que ha producido el error 0 = Sin unidad, solo para el caso de correcto. 1 = Lector 2 = TAG 30 = Centro autorizador Consorcial
5...6	B1_RES	Código de resultado producido, según tablas adjuntas.

- De la unidad Lector y TAG

B1_RES	Estado	Descripción
0x0000	CORRECTO	Acción realizada correctamente.



0x000C	NO_CHANGES	No se han realizado cambios, no es necesario CommitTransaction / AbortTransaction
0x000E	OUT_OF_EEPROM_ERROR	No hay suficiente NV-Memory para completar el comando
0x001C	ILLEGAL_COMMAND_CODE	Código de comando no soportado
0x001E	INTEGRITY_ERROR	CRC o MAC No coinciden los datos, los bytes no son válidos.
0x0040	NO_SUCH_KEY	No es válido el ID de clave especificado.
0x007E	LENGTH_ERROR	Longitud de comando no válido.
0x009D	PERMISSION_DENIED	La configuración actual no admite el comando solicitado
0x009E	PARAMETER_ERROR	Valor del parámetro no válido.
0x00A0	APPLICATION_NOT_FOUND	Aplicación no existe en el PICC
0x00A1	APPL_INTEGRITY_ERROR	Error irreparable en la aplicación, esta será deshabilitada.
0x00AE	AUTHENTICATION_ERROR	La autenticación actual no permite el comando solicitado
0x00AF	ADDITIONAL_FRAME	Additional data frame is expected to be sent
0x00BE	BOUNDARY_ERROR	La lectura o escritura excede los límites del archivo.
0x00C1	PICC_INTEGRITY_ERROR	Error irreparable con el PICC, este será deshabilitado.
0x00CA	COMMAND_ABORTED	Previous Command was not fully completed Not all Frames were requested or provided by the PCD
0x00CD	PICC_DISABLED_ERROR	El PICC está deshabilitado por un error irreparable
0x00CE	COUNT_ERROR	El número de aplicaciones está limitado a 28, imposible crear más.
0x00DE	DUPLICATE_ERROR	La creación del archivo o aplicación ha fallado porque ya existe
0x00EE	EEPROM_ERROR	No se puede completar la operación de escritura en la NV-Memory debido a pérdida de alimentación, se ha activado el mecanismo de backup/rollback interno *
0x00F0	FILE_NOT_FOUND	El archivo no existe
0x00F1	FILE_INTEGRITY_ERROR	Error irreparable con el archivo, este será deshabilitado *
0xFFnn	GENERAL_ERROR	Error general con código indicado en nn

- De la unidad Centro Autorizador Consorcial.

En este caso, código correcto no existe, puesto que las respuestas correctas del centro autorizador son transmitidas con un TLV diferente a este.

BI_RES	Estado	Descripción
0x0001	SIN_COMUNICACIÓN	No ha sido posible conectar con el centro autorizador.
0x0002	SIN_RESPUESTA	Se ha establecido la conexión pero no se recibe respuesta en el tiempo establecido.
0x0003	SIN_RESPUESTA_LOGICA	Se ha recibido respuesta, pero no cumple el formato adecuado.

- Información de las propiedades de la clave maestra de la aplicación actualmente seleccionada.**

Utilizado como respuesta de la acción 0x45 (Propiedades de la clave maestra).

Byte	Campo	Descripción								
0	T	0xB2								
1...2	L	Longitud = 3								
3	B2_CID	Código de identificación de la acción.								
4	B2_KS	<p>1 byte con la información de las propiedades de la clave maestra según las especificaciones técnicas de la tarjeta DESFIRE EV2.</p> <p>En caso de AID = 00 00 00</p> <table><tr><td>bit7</td><td>bit6</td><td>bit5</td><td>bit4</td><td>bit3</td><td>bit2</td><td>bit1</td><td>bit0</td></tr></table>	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0			



		<table><tr><td>RFU</td><td>flag que indica que la MK es necesaria para modificar estas propiedades.</td><td>flag que indica que es necesaria la MK para borrar/crear aplicaciones</td><td>flag que indica que es necesaria la MK para listar aplicaciones</td><td>flag que indica que es necesaria la MK para cambiar la MK (=0 indica MK congelada)</td></tr></table>	RFU	flag que indica que la MK es necesaria para modificar estas propiedades.	flag que indica que es necesaria la MK para borrar/crear aplicaciones	flag que indica que es necesaria la MK para listar aplicaciones	flag que indica que es necesaria la MK para cambiar la MK (=0 indica MK congelada)											
RFU	flag que indica que la MK es necesaria para modificar estas propiedades.	flag que indica que es necesaria la MK para borrar/crear aplicaciones	flag que indica que es necesaria la MK para listar aplicaciones	flag que indica que es necesaria la MK para cambiar la MK (=0 indica MK congelada)														
		<p>En caso de AID ≠ 00 00 00</p> <table><tr><th>bit7</th><th>bit6</th><th>bit5</th><th>bit4</th><th>bit3</th><th>bit2</th><th>bit1</th><th>bit0</th></tr><tr><td colspan="4">0x0 = APK necesaria para cambiar la APK 0x1 ... 0xD = La clave de aplicación especificada es necesaria para cambiar las claves. 0xE = Autenticación con la misma clave es necesaria para cambiar la clave. 0xF =Todas las claves, excepto la APK están congeladas.</td><td>flag que indica que la APK es necesaria para modificar estas propiedades.</td><td>flag que indica que es necesaria la APK para borrar/crear archivos</td><td>flag que indica que es necesaria la APK para listar archivos</td><td>flag que indica que es necesaria la APK para cambiar la APK (=0 indica APK congelada)</td></tr></table>	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	0x0 = APK necesaria para cambiar la APK 0x1 ... 0xD = La clave de aplicación especificada es necesaria para cambiar las claves. 0xE = Autenticación con la misma clave es necesaria para cambiar la clave. 0xF =Todas las claves, excepto la APK están congeladas.				flag que indica que la APK es necesaria para modificar estas propiedades.	flag que indica que es necesaria la APK para borrar/crear archivos	flag que indica que es necesaria la APK para listar archivos	flag que indica que es necesaria la APK para cambiar la APK (=0 indica APK congelada)
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0											
0x0 = APK necesaria para cambiar la APK 0x1 ... 0xD = La clave de aplicación especificada es necesaria para cambiar las claves. 0xE = Autenticación con la misma clave es necesaria para cambiar la clave. 0xF =Todas las claves, excepto la APK están congeladas.				flag que indica que la APK es necesaria para modificar estas propiedades.	flag que indica que es necesaria la APK para borrar/crear archivos	flag que indica que es necesaria la APK para listar archivos	flag que indica que es necesaria la APK para cambiar la APK (=0 indica APK congelada)											
5	B2_NK	1 byte con el número de claves que la aplicación tiene alojadas. En caso de que la aplicación actualmente seleccionada sea la AID = 00 00 00, este valor será 1.																

- Información de la versión de claves solicitada.**

Utilizado como respuesta de la acción 0x64 (Leer versión de claves).

Byte	Campo	Descripción
0	T	0xB3
1...2	L	Longitud = 2
3	B1_CID	Código de identificación de la acción.



4	B3_KV	1 byte indicando la versión de claves
---	-------	---------------------------------------

- Lista de aplicaciones existentes en la tarjeta.**

Utilizado como respuesta de la acción 0x6A (Listar aplicaciones).

Para la operativa de esta acción la aplicación AID = 00 00 00 debe estar seleccionada previamente.

Byte	Campo	Descripción
0	T	0xB4
1...2	L	Longitud. Depende de la cantidad de AIDs contenidas en el registro. En caso de no existir ninguna aplicación en la tarjeta, se indicara en la longitud un valor de 0. En esta lista NO debe incluirse la aplicación maestra AID = 00 00 00, ya que se considera que esta siempre existe en la tarjeta.
3	B4_CID	Código de identificación de la acción.
4...6	B4_AID ₀ ... B4_AID _N	3 bytes por cada identificación de aplicación AID obtenida de la tarjeta
...		

- Información de la versión de fabricación de la tarjeta.**

Utilizado como respuesta de la acción 0x60 (Información de la tarjeta).

Informará con toda la información de las especificaciones de la tarjeta presente.

Byte	Campo	Descripción	
0	T	0xB5	
1...2	L	Longitud = 29	
3	B5_CID	Código de identificación de la acción.	
4	B5_HVID	Información hardware	código del vendedor
5	B5_HTI		Código del tipo
6	B5_HST		Código del subtipo
7	B5_HMAV		Código de la versión Mayor
8	B5_HMIV		Código de la versión menor
9	B5_HSZ		Código del tamaño de almacenamiento
10	B5_HCP		Código del tipo de protocolo de comunicaciones
11	B5_SVID	Información del software	código del vendedor
12	B5_STI		Código del tipo
13	B5_SST		Código del subtipo



14	B5_SMAV		Código de la versión Mayor
15	B5_SMIV		Código de la versión menor
16	B5_SSZ		Código del tamaño de almacenamiento
17	B5_SCP		Código del tipo de protocolo de comunicaciones
18...24	B5_UID	Identificador único de la tarjeta	
25...29	B5_BNO	Número de lote de producción	
30	B5_CWP	Código de la semana de producción.	
31	B5_YP	Código del año de producción.	

- **Lista de archivos existentes en la aplicación actualmente seleccionada.**

Utilizado como respuesta de la acción 0x6F (Listar archivos).

Byte	Campo	Descripción
0	T	0xB6
1...2	L	Longitud Depende de la cantidad de archivos FIDs contenidos en la aplicación actualmente seleccionada. En caso de no existir ningún archivo en la aplicación, se indicara en la longitud un valor de 1. Esta acción no es posible realizarse si la aplicación seleccionada en la tarjeta es la AID = 00 00 00, ya que esta NO puede contener archivos.
3	B6_CID	Código de identificación de la acción.
4	FID ₀ ... FID _N	1 byte por cada identificación de archivo FID obtenido de la aplicación.
...		

- **Propiedades de un archivo.**

Utilizado como respuesta de la acción 0xF5 (Información de archivo).

La información suministrada en este registro depende del tipo de archivo al cual se refiera, nos podemos encontrar con 5 diferentes agrupados en 3 tipos, datos estándar (archivos de tipo 0x00 y 0x01), valor (de tipo 0x02) o registros (de tipo 0x03 y 0x04).

- **Información de archivo de datos estándar (0x00) o de backup (0x01)**

Byte	Campo	Descripción
0	T	0xB7
1...2	L	Longitud = 9
3	B7_CID	Código de identificación de la acción. 0xB7: Propiedades del archivo.
4	B7_FID	Código de identificación del archivo



5	B7_FT	<p>Puede ser</p> <p>0x00 = Archivo estándar</p> <p>0x01 = Archivo estándar de backup.</p>								
6	B7_CC	<p>Configuración de las comunicaciones.</p> <p>0x00 = Comunicaciones en plano.</p> <p>0x01 = Comunicaciones en plano aseguradas con DES/3DES por MAC</p> <p>0x02 = Comunicaciones aseguradas con cifrado DES/3DES completo</p>								
7...8	B7_ACC	<p>Condiciones o derechos de acceso.</p> <p>Indica con que numero de clave de la aplicación donde está ubicado el archivo se puede hacer la acción requerida.</p> <table><tr><td>bit15 ... bit12</td><td>bit11 ... bit8</td><td>bit7 ... bit4</td><td>bit3 ... bit0</td></tr><tr><td>lectura</td><td>Escritura</td><td>Lectura/Escritura</td><td>Cambiar condiciones de acceso</td></tr></table> <p>En cada una de las agrupaciones de 4 bits arriba detalladas, se establece el valor del número de clave (número de clave que debe estar alojada en la aplicación contenedora del archivo), con la cual es necesario realizar la “autenticación” en la aplicación para poder tener el acceso que indica la agrupación en concreto.</p> <p>Existe 3 valores, además del número de clave que puede ser posible indicar:</p> <p>0 = APK, indica que es necesario la clave maestra de la aplicación.</p> <p>0xE = Acceso libre, no es necesario ninguna autenticación.</p> <p>0xF = Acceso congelado o sin acceso.</p>	bit15 ... bit12	bit11 ... bit8	bit7 ... bit4	bit3 ... bit0	lectura	Escritura	Lectura/Escritura	Cambiar condiciones de acceso
bit15 ... bit12	bit11 ... bit8	bit7 ... bit4	bit3 ... bit0							
lectura	Escritura	Lectura/Escritura	Cambiar condiciones de acceso							
9...11	B7_TAM	Tamaño del archivo								

○ **Información de archivo de valor (0x02).**

Byte	Campo	Descripción
0	T	0xB7
1...2	L	Longitud = 19
3	B7_CID	Código de identificación de la acción.
4	B7_FID	Código de identificación del archivo
5	B7_FT	0x02 = Archivo de tipo valor.
6	B7_CC	Configuración de las comunicaciones. (ver archive standard)
7...8	B7_ACC	Condiciones o derechos de acceso. (ver archive standard)
9...12	B7_VAL_MIN	Valor mínimo permitido.
13...16	B7_VAL_MAX	Valor máximo permitido.
17...20	B7_LIM_CRE	Valor del límite del crédito permitido.
21	B7_CRE	Flag que indica si el crédito limitado está permitido =1 o no =0.



- **Información de archivo de datos registros linear (0x03) o cíclicos (0x04).**

Byte	Campo	Descripción
0	T	0xB7
1...2	L	Longitud = 15
3	B7_CID	Código de identificación de la acción.
4	B7_FID	Código de identificación del archivo
5	B7_FT	Puede ser 0x03 = Archivo de registros lineales. 0x04 = Archivo de registros cíclicos.
6	B7_CONF	Configuración de las comunicaciones. (ver archive standard)
7...8	B7_ACCESS	Condiciones o derechos de acceso. (ver archive standard)
9...11	B7_TAM_REG	Tamaño del registro (1 solo registro)
12...14	B7_MAX_REG	Cantidad máxima de registros que admite.
15...17	B7_CUR_REG	Cantidad actual de registros almacenados. Este número es igual al máximo número de registros que pueden ser leídos del archivo.

- **Datos de lectura de un archivo estándar.**

Utilizado como respuesta de la acción 0xBD (Leer archivo).

Byte	Campo	Descripción
0	T	0xB8
1...2	L	Longitud a especificar dependerá de la longitud del campo info
3	B8_CID	Código de identificación de la acción.
4...6	B8_AID	3 bytes del Identificador de la aplicación contenedora del archivo
7	B8_FID	1 byte del Identificador del archivo
8...N	B8_INFO	N bytes correspondientes a la información leída del archivo.

- **Datos de lectura de un archivo de valor.**

Utilizado como respuesta de la acción 0x6C (Leer valor).

Byte	Campo	Descripción
0	T	0xB9
1...2	L	Longitud = 9
3	B9_CID	Código de identificación de la acción.
4...6	B9_AID	3 bytes del Identificador de la aplicación contenedora del archivo



7	B9_FID	1 byte del Identificador del archivo
8...11	B9_VALOR	4 bytes con el valor del archivo

- Datos de lectura del registro de un archivo de registros.**

Utilizado como respuesta de la acción 0xBB (Leer registro)

Byte	Campo	Descripción
0	T	0xBA
1...2	L	Longitud depende de la longitud del valor
3	BA_CID	Código de identificación de la acción.
4...6	BA_AID	3 bytes del Identificador de la aplicación contenedora del archivo.
7	BA_FID	1 byte del Identificador del archivo.
8	BA_REG	1 byte indicando el registro leído, entre 0 y número existente de registros -1, siendo 0 el último almacenado cronológicamente.
9...N	BA_VALOR	N bytes correspondientes a la información leída del registro del archivo.

- Datos de respuesta de petición del Centro Autorizador del Consorcio.**

Utilizado como respuesta de la acción 0x30 (Operativa con el Centro Autorizador)

Byte	Campo	Descripción
0	T	0x30
1...2	L	=N
3	BA_CID	Código de identificación de la acción.

Pendiente de definición

5.3.4. Salidas.

unsigned int

Código de retorno de la función.

Código	Detalles
0	Correcto y continuar
1	Correcto y finalizar



2	Error de proceso
---	------------------

unsigned char *data_out

En caso de error “código de retorno = 2”

En caso de que el código de retorno de la función sea **2=Error**, Es indicativo que se ha producido un error de SISTEMA (por ejemplo algún parámetro erróneo, o falta algún parámetro esencial).

El contenido de *data_out incluirá el array TLV con la información del error producido.

El campo FI_CNT en el registro de cabecera no será gestionado y tendrá un valor 0.

No debe confundirse este registro de error con, por ejemplo una *validación de rechazo*, esta última NO será informada como error, sino como *l=correcto y finalizar*, ya que la función “GetInfo()” será la encargada de informar del resultado de la operativa con la tarjeta.

- **Información de error producido.**

Byte	Campo	Descripción
0	T	0x99
1...2	L	longitud = 3
3	99_MOD	1 byte con el código del módulo que produjo el error
4...5	99_CERR	2 bytes con el código del error producido, es dependiente del módulo Ver capítulo 8 para la relación de errores.

Referencias a la tabla de errores en el documento “Tabla de códigos de error de la función Process”

En caso de correcto, “código de retorno = 0 o 1”

En cualquier otro caso, el contenido de *data_out incluye el array de registros (uno por cada acción) con las acciones que son necesario ejecutar por el HOST.

En caso de no existir ninguna acción a ejecutar, solo contendrá el registro de cabecera con el campo N_REG=0.

Las acciones recibidas deben ser ejecutadas por el HOST en el estricto orden en el que se han recibido en el array. Cada acción está identificado por un valor único y secuencial CID que identifica dicha acción en caso de producirse algún error en su ejecución (ver registro 0xB1 en “data_in”).



Si el HOST, durante la ejecución de las acciones, se produjera un error, debe interrumpir el proceso e informar a la librería mediante la llamada a la función "Process()" con el registro 0xB1.

- **Autenticación.**

Byte	Campo	Descripción
0	T	0x10
1...2	L	Longitud = N.
3	10_CID	Código de identificación de la acción
4	10_KN	Índice de la clave en la tarjeta (0..13)
5	10_KT	Tipo de encriptación de la clave. 0 = DES/3DES (Clave OAKEY de 8 bytes) 1 = 3KDES (Clave OAKEY de 24 bytes) 2 = AES (Clave OAKEY de 16 bytes)
6...N	10_KEY	La clave, su tamaño variara dependiendo del tipo de encriptación OA_KT.

- **Obtener la configuración de la APK de la aplicación actualmente seleccionada.**

Byte	Campo	Descripción
0	T	0x11
1...2	L	Longitud = 1
3	11_CID	Código de identificación de la acción

- **Leer versión de una clave de la aplicación actualmente seleccionada.**

Byte	Campo	Descripción
0	T	0x12
1...2	L	Longitud = 2
3	12_CID	Código de identificación de la acción
4	12_KN	1 byte que especifica el número de clave de la cual se solicita la versión. En caso de que la aplicación actualmente seleccionada sea la AID= 00 00 00, este valor debe ser =0, en caso contrario, AID ≠ 00 00 00, el valor debe ser entre 0 y el máximo de claves alojadas en dicha aplicación.

- **Listar aplicaciones existentes en la tarjeta.**

Byte	Campo	Descripción
------	-------	-------------



0	T	0x13
1...2	L	Longitud = 1
3	13_CID	Código de identificación de la acción

- **Selección de una aplicación.**

Byte	Campo	Descripción
0	T	0x14
1...2	L	Longitud = 4
3	14_CID	Código de identificación de la acción
4...6	14_AID	3 bytes del código de identificación de la aplicación a seleccionar

- **Obtener información de fabricación de la tarjeta.**

Byte	Campo	Descripción
0	T	0x15
1...2	L	Longitud = 1
3	15_CID	Código de identificación de la acción

- **Listar archivos de la aplicación actualmente seleccionada.**

Byte	Campo	Descripción
0	T	0x16
1...2	L	Longitud = 1
3	16_CID	Código de identificación de la acción

- **Obtener información de un archivo de la aplicación actualmente seleccionada.**

Byte	Campo	Descripción
0	T	0x17
1...2	L	Longitud = N
3	17_CID	Código de identificación de la acción
4	17_FID	Código de identificación del archivo del que se solicita la información.

- **Lectura de archivo estándar de datos.**



Byte	Campo	Descripción
0	T	0x18
1...2	L	Longitud = 9
3	18_CID	Código de identificación de la acción
4	18_FID	1 byte de código de identificación del archivo a leer.
5	18_CS	Cifrado de la comunicación 0 = Comunicación plana 1 = Cifrado por DES/3DES MACing 3 = Cifrado completo DES/3DES
6...8	18_POS	Posición de inicio de la lectura.
9...11	18_LONG	Longitud de datos que se desean leer.

- **Escritura en archivo estándar de datos.**

Byte	Campo	Descripción
0	T	0x19
1...2	L	Longitud = N, depende del tamaño de la información a escribir.
3	19_CID	Código de identificación de la acción
4	19_FID	1 byte de código de identificación del archivo a escribir
5	19_CS	Cifrado de la comunicación 0 = Comunicación plana 1 = Cifrado por DES/3DES MACing 3 = Cifrado completo DES/3DES
6...8	19_POS	Posición de inicio de la escritura.
9...N	19_INFO	N bytes de la información a grabar en el archivo.

- **Lectura de archivo de valor.**

Byte	Campo	Descripción
0	T	0x1A
1...2	L	Longitud = 3
3	1A_CID	Código de identificación de la acción
4	1A_FID	1 byte de código de identificación del archivo a leer.
5	1A_CS	Cifrado de la comunicación 0 = Comunicación plana 1 = Cifrado por DES/3DES MACing



		3 = Cifrado completo DES/3DES
--	--	-------------------------------

- Incremento (Credit) de archivo de valor.**

Byte	Campo	Descripción
0	T	0x1B
1...2	L	Longitud = 7
3	1B_CID	Código de identificación de la acción
4	1B_FID	1 byte de código de identificación del archivo a incrementar el valor.
5	1B_CS	Cifrado de la comunicación 0 = Comunicación plana 1 = Cifrado por DES/3DES MACing 3 = Cifrado completo DES/3DES
6...9	1B_VALOR	Valor de incremento.

- Decremento (Debit) de archivo de valor.**

Byte	Campo	Descripción
0	T	0x1C
1...2	L	Longitud = 7
3	1C_CID	Código de identificación de la acción
4	1C_FID	1 byte de código de identificación del archivo a incrementar el valor.
5	1C_CS	Cifrado de la comunicación 0 = Comunicación plana 1 = Cifrado por DES/3DES MACing 3 = Cifrado completo DES/3DES
6...9	1C_VALOR	Valor de decremento.

- Incremento limitado por el crédito establecido en archivo de valor.**

Byte	Campo	Descripción
0	T	0x1D
1...2	L	Longitud = 7
3	1D_CID	Código de identificación de la acción
4	1D_FID	1 byte de código de identificación del archivo a incrementar el valor.
5	1D_CS	Cifrado de la comunicación 0 = Comunicación plana



		1 = Cifrado por DES/3DES MACing 3 = Cifrado completo DES/3DES
6...9	1D_VALOR	Valor de incremento.

- **Escritura de datos en archivo de registros.**

Byte	Campo	Descripción
0	T	0x1E
1...2	L	Longitud = 9
3	1E_CID	Código de identificación de la acción
4	1E_FID	1 byte de código de identificación del archivo
5	1E_CS	Cifrado de la comunicación 0 = Comunicación plana 1 = Cifrado por DES/3DES MACing 3 = Cifrado completo DES/3DES
6...8	1E_POS	Posición de inicio en bytes de inicio de la grabación de los datos dentro del propio registro. (entre 0 y 3B_DATOS-1)
9...11	1E_LNG	Longitud de información a escribir (entre 1 y 3B_DATOS - 3B_POS)
12...N	1E_DATOS	Información a escribir.

- **Lectura de archivo de registros.**

Byte	Campo	Descripción
0	T	0x1F
1...2	L	Longitud = 9
3	1F_CID	Código de identificación de la acción
4	1F_FID	1 byte de código de identificación del archivo
5	1F_CS	Cifrado de la comunicación 0 = Comunicación plana 1 = Cifrado por DES/3DES MACing 3 = Cifrado completo DES/3DES
6...8	1F_REGP	Registro de inicio que desea ser leído, entre 0 y número de registros existentes -1, siendo 0 el último registro almacenado.
9...11	1F_REGN	Cantidad de registros a leer, comenzando siempre por el ultimo insertado y en orden cronológico. En caso de indicar 0, deberá ser leído todos los registros existentes.

- **Vaciado/borrado completo de archivo de registros.**

Byte	Campo	Descripción
------	-------	-------------



0	T	0x20
1...2	L	Longitud = 3
3	20_CID	Código de identificación de la acción
4	20_FID	1 byte de código de identificación del archivo que se desea vaciar.
5	20_CS	Cifrado de la comunicación 0 = Comunicación plana 1 = Cifrado por DES/3DES MACing 3 = Cifrado completo DES/3DES

- **Confirmación de transacciones.**

Repercutirá en todas las transacciones efectuadas en los archivos que dispongan de backup (estándar con backup, valor y registros).

Byte	Campo	Descripción
0	T	0x21
1...2	L	Longitud = 1
3	21_CID	Código de identificación de la acción

- **Invalidación de todas las transacciones.**

Repercutirá a todas las efectuadas hasta el momento en los archivos que dispongan de backup (estándar con backup, valor y registros).

Byte	Campo	Descripción
0	T	0x22
1...2	L	Longitud = 1
3	22_CID	Código de identificación de la acción

- **Solicitud de operativa con el Centro Autorizador.**

En determinados casos de necesidad por la operativa en ejecución, será necesario que el HOST efectúe comunicaciones (conexión, envío y recepción) con el Centro Autorizador Consorcial *OpSin*. En esta acción, la librería suministrara una relación de datos que serán necesarios para efectuar la comunicación.

Sera responsabilidad del HOST realizar los mecanismos parametrizados para llevar a cabo correctamente dicha conexión, envío y recepción de datos de respuesta del Centro Autorizador.

Byte	Campo	Descripción
0	T	0x3

Pendiente de definición



1...2	L	Longitud = N
3	30_CID	Código de identificación de la acción
4...10	30_UID	Identificador único de la tarjeta
11	30_TPET	Código del tipo de petición a realizar 1 = Venta 2 = Recarga
12...N	30_DAT	Datos de la solicitud

5.3.5. Relación de Acciones <-> Respuestas.

Como se ha detallado, la función "Process()" envía en su "data_out" las acciones a realizar por el HOST y espera en su próxima llamada que en "data_in" se especifique el resultado de todas las ordenes realizadas.

Esto indica que cada acción solicitada tiene específicamente una respuesta concreta que ofrecer en caso de éxito, o una respuesta estándar en caso de error.

Aquí indicamos una tabla de correspondencia entre las acciones que pueden solicitar la función y las respuestas esperadas.

Acción	Descripción	Respuesta Éxito	Respuesta Fracaso
0x10	Autenticar	0xB1	0xB1
0x11	Propiedades de la aplicación	0xB2	0xB1
0x12	Leer versión claves	0xB3	0xB1
0x13	Listar aplicaciones	0xB4	0xB1
0x14	Seleccionar aplicación	0xB1	0xB1
0x15	Información de fábrica	0xB5	0xB1
0x16	Listar archivos	0xB6	0xB1
0x17	Información archivo	0xB7	0xB1
0x18	Leer archivo	0xB8	0xB1
0x19	Grabar archivo	0xB1	0xB1
0x1A	Leer valor	0xB9	0xB1
0x1B	Incrementar valor	0xB1	0xB1
0x1C	Decrementar valor	0xB1	0xB1
0x1D	Incrementar valor crédito	0xB1	0xB1
0x1E	Grabar registro	0xB1	0xB1
0x1F	Leer registro	0xBA	0xB1



0x20	Vaciar registros	0xB1	0xB1
0x21	Confirmar (commit)	0xB1	0xB1
0x22	Invalidar (rollback)	0xB1	0xB1
0x30	Operativa con el Centro Autorizador	0xBB	0xB1



5.4. LASTERROR

5.4.1. Descripción y uso.

Su cometido es informar del ultimo error registrado por la librería, básicamente debe ser llamada cada vez que la función “Process()” devuelva un resultado = 2 (error).

5.4.2. Sintaxis.

```
void LastError( unsigned char *data_out )
```

5.4.3. Entradas.

Ninguna.

5.4.4. Salidas.

```
unsigned char *data_out
```

*Hacer referencia al data_out en caso de error de la funcion “Process()”



5.5. GETINFO

5.5.1. Descripción y uso.

Una vez finalizado todo el proceso con la tarjeta, ya sea con éxito o sin él, esta función ofrece todos los datos necesarios para producir la información, tanto visual como impresa, que debe ofrecerse al usuario.

5.5.2. Sintaxis.

```
void GetInfo( unsigned char * data_out )
```

5.5.3. Entradas.

Ninguna

5.5.4. Salidas.

```
unsigned char *data_out
```

Array de registros TLV con los diferentes datos necesarios para su uso en la información al viajero.

- Resultado de operativa efectuada.

Byte	Campo	Descripción
0	T	0x07
1...2	L	longitud = 2
3...4	07_CRO	Código del resultado de la última operación efectuada. Ver tabla a continuación

Ámbito	Código 07_CRO	Descripción
General	0x0000	No hay operación efectuada
	0x0030	ERROR_CABECERA
	0x0031	ERROR_VARIABLES
	0x0032	ERROR_MONDERO
	0x0033	ERROR_HISTORICO
	0x0034	TARJETA_ANULADA
	0x0035	TARJETA_NO_OPERATIVA
	0x0036	ERROR_VERSION
	0x0037	ERROR_MODALIDAD_PRUEBAS
	0x0038	BLOQUES_VAR_ERRONEOS
	0x0039	HISTORICO_ERRONEO
	0x003A	ERROR_PUNTERO_VIAJES
	0x003B	ERROR_MAXIMO_MOVIMIENTOS



	0x003C	ERROR_PUNTERO_VARIABLES
	0x003D	ERROR_MATRIZ_RECUPERACION
	0x0100	Error en venta
Módulo de venta	0x0101	Venta correctamente efectuada
	0x0201	Recarga correctamente efectuada
	0x0300	Error en Validación
Módulo de validación	0x0301	CORRECTO_TEMPORAL
	0x0302	CORRECTO_TEMPORAL_TRANSBORDO
	0x030A	MONEDERO_CANCELACION_1
	0x030B	MONEDERO_CANCELACION_N
	0x030C	MONEDERO_TRANSBORDO_1
	0x030D	MONEDERO_TRANSBORDO_N
	0x0314	ANUL_CORRECTA
	0x0369	MENSUAL_CADUCADA
	0x036A	MENSUAL_PASSBACK
	0x036C	SIN_SALDO
	0x036D	ERROR_TIPO_BONO
	0x0370	SOBREPASO_VIAJEROS
	0x0377	ERROR_ID_CONSORCIO
	0x0378	ERROR_FECHA_LIMITE
	0x0379	PRECIO_9999
	0x037A	ERROR_SALTOS
	0x037B	OTRO_CONSORCIO
	0x037C	FORZAR_ACOMP_NO_PERMITIDO
	0x037D	ANUL_ERROR_CANTIDAD
	0x037E	ANUL_SIN_USO
	0x037F	ANUL_SOBREPASO_VIAJEROS
	0x0380	ANUL_FUERA_DE_AMBITO
	0x0381	ANUL_VIAJE_YA_ANULADO
Módulo de inspección	0x0400	Error en inspección
	0x0401	Inspección correctamente efectuada

- Información de campo numérico.

Byte	Campo	Descripción
0	T	0x08
1...2	L	longitud = 9
3...6	08_CC	Código de campo *Tabla de códigos de campo detallada a continuación.
7	08_SGN	Signo del valor 0 = positivo 1 = negativo
8...11	08_VALOR	Valor del campo

TABLA DE CODIGOS DE CAMPOS

La columna **C6_Tipo** indica que el campo será entregado en caso de solicitar dicho tipo de información. En el caso de solicitar información de la tarjeta.



La columna **Opcional** indica que dicho campo NO es esencial y por lo tanto no será devuelto por la función si no existe.

Ámbito	Código HEX	Descripción	C6_Tipo = 1	C6_Tipo = 2	C6_Tipo = 3	Opcional
Identificación de tarjeta	08 01 03 00	Propietario de la tarjeta	*	*	*	
	08 01 04 00	Responsable de numeración	*	*	*	
	08 01 05 00	Número de tarjeta	*	*	*	
	08 01 06 00	Dígito de control	*	*	*	
	08 01 07 00	Modelo	*	*	*	
	08 01 08 00	Mes de emisión	*	*	*	
	08 01 0C 00	Año de emisión	*	*	*	
	08 01 0F 00	Tipo de documento	*	*	*	
	08 01 10 00	Tipo de asistencia	*	*	*	

Identificación del título	08 10 30 00	Propietario del título	*	*	*	
	08 10 31 00	Número del título	*	*	*	
	08 10 32 00	Clase del título	*	*	*	
	08 10 34 00	Características 1 de tarifa	*	*	*	
	08 10 35 00	Características 2 de tarifa	*	*	*	
	08 10 36 00	Operatividad	*	*	*	
	08 10 37 00	Passback	*	*	*	
	08 10 38 00	Tipo de sistema de tarifas	*	*	*	
	08 11 3B 00	Clave aleatoria	*	*	*	
	08 13 30 00	Regularización económica	Propietario del título	*	*	*
	08 13 31 00		Identificación del título	*	*	*
	08 13 3D 00		Cantidad	*	*	*
	08 13 3E 00		Signo	*	*	*
	08 15 30 00	Restricción de uso	Propietario del título	*	*	*
	08 15 31 00		Identificación del título	*	*	*
	08 15 42 00		Uso	*	*	*
	08 16 30 00	Fechas de validez	Propietario del título	*	*	*
	08 16 31 00		Identificación del título	*	*	*
	08 16 4A 00		Año inicio validez	*	*	*
	08 16 4B 00		Mes inicio de validez	*	*	*
	08 16 4C 00		Día inicio de validez	*	*	*
	08 16 4D 00		Año fin de validez	*	*	*
	08 16 4E 00		Mes fin de validez	*	*	*
	08 16 4F 00		Día fin de validez	*	*	*
	08 18 30 00	Limitación horaria	Propietario del título	*	*	*
	08 18 31 00		Identificación del título	*	*	*
	08 18 52 00		Hora de inicio	*	*	*
	08 18 53 00		Hora final	*	*	*
	08 18 55 00		Lunes	*	*	*
	08 18 56 00		Martes	*	*	*
	08 18 57 00		Miércoles	*	*	*
	08 18 58 00		Jueves	*	*	*
	08 18 59 00		Viernes	*	*	*
	08 18 5A 00		Sábado	*	*	*
	08 18 5B 00		Domingo	*	*	*

Monedero	01 1F 00 00	Saldo en céntimos de euro	*	*	*	
----------	-------------	---------------------------	---	---	---	--

Ultimo Movimiento aa 02 = Ultimo movimiento 05 = Última reserva	aa 20 60 00	Movimiento sin reserva	Número de línea	*	*	*
	aa 20 61 00		Flota	*	*	*
	aa 20 4A 00		Año	*	*	*
	aa 20 4B 00		Mes	*	*	*
	aa 20 4C 00		Día	*	*	*
	aa 20 62 00		Hora	*	*	*
	aa 20 63 00		Minuto	*	*	*
	aa 20 64 00		Sentido	*	*	*
	aa 20 65 00		Consortorio	*	*	*
	aa 20 66 00		Etapas	*	*	*
	aa 20 67 00		Control sistemas cerrados	*	*	*
	aa 21 60 00	Movimiento con reserva	Número de línea	*	*	*
	aa 21 61 00		Flota	*	*	*
	aa 21 4A 00		Año	*	*	*
	aa 21 4B 00		Mes	*	*	*
	aa 21 4C 00		Día	*	*	*
	aa 21 62 00		Hora	*	*	*
	aa 21 63 00		Minuto	*	*	*
	aa 21 64 00		Sentido	*	*	*
	aa 21 65 00		Consortorio	*	*	*
	aa 21 66 00		Etapas	*	*	*
	aa 21 67 00		Control sistemas cerrados	*	*	*
	aa 22 30 00	Importe y viajeros	Propietario del título	*	*	*
	aa 22 31 00		Identificación del título	*	*	*
	aa 22 3D 00		Cantidad	*	*	*
	aa 22 6A 00		Número de viajeros que viajan juntos	*	*	*



			utilizando el mismo título				
	aa 22 68 00		Anulaciones de viajeros		*	*	*
	aa 24 70 00	Zonas y saltos	Zona		*	*	*
	aa 24 71 00		Subzona		*	*	*
	aa 24 72 00		Salto zona		*	*	*
	aa 24 73 00		Salto subzona		*	*	*
	aa 28 75 00	Zonas origen / destino	Zona origen		*	*	*
	aa 28 76 00		Zona destino		*	*	*
	aa 29 77 00	Cont. Urbano / interurbanos	Número de etapas urbanas efectuadas		*	*	*
	aa 29 78 00		Número de etapas interurbanas efectuadas		*	*	*
	aa 2A 77 00	Minutos urbano / interurbano	Minutos límite para poder efectuar transbordo urbano		*	*	*
	aa 2A 78 00		Minutos límite para poder efectuar transbordo interurbano		*	*	*

Históricos aa 03 Histórico de movimientos 06 = Historico de reservas nn Los campos de movimientos se encuentran organizados en registros el valor numérico de nn representa el valor del registro en el que está ubicado el campo.	aa 20 60 nn	Movimiento sin reserva	Número de línea			*	*
	aa 20 61 nn		Flota			*	*
	aa 20 4A nn		Año			*	*
	aa 20 4B nn		Mes			*	*
	aa 20 4C nn		Día			*	*
	aa 20 62 nn		Hora			*	*
	aa 20 63 nn		Minuto			*	*
	aa 20 64 nn		Sentido			*	*
	aa 20 65 nn		Consorcio			*	*
	aa 20 66 nn		Etapas			*	*
	aa 20 67 nn		Control sistemas cerrados			*	*
	aa 21 60 nn	Movimiento con reserva	Número de línea			*	*
	aa 21 61 nn		Flota			*	*
	aa 21 4A nn		Año			*	*
	aa 21 4B nn		Mes			*	*
	aa 21 4C nn		Día			*	*
	aa 21 62 nn		Hora			*	*
	aa 21 63 nn		Minuto			*	*
	aa 21 64 nn		Sentido			*	*
	aa 21 65 nn		Consorcio			*	*
	aa 21 66 nn		Etapas			*	*
	aa 21 67 nn		Control sistemas cerrados			*	*
	aa 22 30 nn	Importe y viajeros	Propietario del título			*	*
	aa 22 31 nn		Identificación del título			*	*
	aa 22 3D nn		Cantidad			*	*
	aa 22 6A nn		Número de viajeros que viajan juntos utilizando el mismo título			*	*
	aa 22 6B nn		Anulaciones de viajeros			*	*
	aa 24 70 nn	Zonas y saltos	Zona			*	*
	aa 24 71 nn		Subzona			*	*
	aa 24 72 nn		Salto zona			*	*
	aa 24 73 nn		Salto subzona			*	*
	aa 28 75 nn		Zona origen			*	*
	aa 28 76 nn	Zonas origen / destino	Zona destino			*	*
	aa 29 77 nn	Cont. Urbano / interurbanos	Número de etapas urbanas efectuadas			*	*
	aa 29 78 nn		Número de etapas interurbanas efectuadas			*	*
	aa 2A 77 nn	Minutos urbano / interurbano	Minutos límite para poder efectuar transbordo urbano			*	*
	aa 2A 78 nn		Minutos límite para poder efectuar transbordo interurbano			*	*

Log de Cargas nn Los campos de log de cargas se encuentran organizados en registros el valor numérico de nn representa el valor del registro en el que está ubicado el campo.	04 40 90 nn	Movimiento PA	ID comercio que realiza el movimiento			*	*
	04 40 91 nn		Red de integración del que escribe el movimiento			*	*
	04 40 4A nn		Año			*	*
	04 40 4B nn		Mes			*	*
	04 40 4C nn		Día			*	*
	04 40 62 nn		Hora			*	*
	04 40 63 nn		Minuto			*	*
	04 40 92 nn		Segundo			*	*
	04 10 30 nn	Identificación del título	Propietario del título			*	*
	04 10 31 nn		Número del título			*	*
	04 10 32 nn		Clase del título			*	*
	04 10 34 nn		Características 1 de tarifa			*	*
	04 10 35 nn		Características 2 de tarifa			*	*
	04 10 36 nn		Operatividad			*	*
	04 10 37 nn		Passback			*	*
	04 10 38 nn		Tipo de sistema de tarifas			*	*
	04 15 30 nn	Restricciones de uso	Propietario del título			*	*
	04 15 31 nn		Número del título			*	*
	04 15 42 nn		Uso			*	*
	04 16 30 nn	Fechas de validez	Propietario del título			*	*
	04 16 31 nn		Número del título			*	*
	04 16 4A nn		Año			*	*
	04 16 4B nn		Mes			*	*
	04 16 4C nn		Día			*	*
	04 16 4D nn		Año finalización			*	*
	04 16 4E nn		Mes finalización			*	*
	04 16 4F nn		Día finalización			*	*



	04 18 30 nn	Limitación horaria	Propietario del título			•	•
	04 18 31 nn		Número del título			•	•
	04 18 52 nn		Hora de inicio			•	•
	04 18 53 nn		Hora final			•	•
	04 18 55 nn		Lunes			•	•
	04 18 56 nn		Martes			•	•
	04 18 57 nn		Miércoles			•	•
	04 18 58 nn		Jueves			•	•
	04 18 59 nn		Viernes			•	•
	04 18 5A nn		Sábado			•	•
	04 18 5B nn		Domingo			•	•
	04 42 30 nn	Importe	Propietario del título			•	•
	04 42 31 nn		Número del título			•	•
	04 42 3D nn		Cantidad			•	•
	04 42 3E nn	Fianza	Signo			•	•
	04 44 3D nn		Cantidad			•	•
	04 44 3E nn		Signo			•	•
	04 45 30 nn	Regularización	Propietario del título			•	•
	04 45 31 nn		Número del título			•	•
	04 45 3D nn		Cantidad			•	•
	04 46 30 nn	Cambio operatividad	Propietario del título			•	•
	04 46 31 nn		Número del título			•	•
	04 46 36 nn		Operatividad			•	•
	04 47 30 nn	Nuevo trayecto habitual	Propietario del título			•	•
	04 47 31 nn		Número del título			•	•
	04 47 34 nn		ST-1			•	•
	04 47 35 nn	Nuevos saltos habituales	ST-2			•	•
	04 48 30 nn		Propietario del título			•	•
	04 48 31 nn		Número del título			•	•
	04 48 34 nn		ST-1			•	•
	04 48 35 nn		ST-2			•	•

- Información de campo de texto.

Byte	Campo	Descripción
0	T	0x09
1...2	L	longitud = Depende de la longitud del valor
4	09_CC	Código de campo *Tabla de códigos de campo detallada a continuación.
5...N	09_VALOR	Valor del campo Cada byte representara en formato ASCII un carácter. ²

TABLA DE CODIGOS DE CAMPOS

La columna **C6_Tipo** indica que el campo será entregado en caso de solicitar dicho tipo de información. En el caso de solicitar información de la tarjeta.

La columna **Opcional** indica que dicho campo NO es esencial y por lo tanto no será devuelto por la función si no existe.

Ámbito	Código HEX	Descripción	C6_Tipo = 1	C6_Tipo = 2	C6_Tipo = 3	Opcional
Identificación de tarjeta	08 01 0D 00	Nombre del usuario	•	•	•	
	08 01 0E 00	Número del documento	•	•	•	

² Ver tabla ASCII de caracteres de texto



5.6. RESUMEN DE TIPOS EN "DATA_IN".

Función	TI	Detalles
Init()	0xF1	Cabecera.
	0xC0	Registro de datos de situación actual (módulo de validación)
	0xC1	Parámetros de situación (módulo de ventas)
	0xC5	Registro de fecha y hora de operación.
	0xC6	Registro de parámetros auxiliares para el módulo de validación.
	0xC7	Registro de tarificación relacionado con saltos en la tarjeta.
	0xC8	Registro de tarificación origen/destino.
	0xC9	Registro de parámetros de transbordo.
	0xCA	Registro de tipo de información a solicitar en la consulta.
	0xCB	Registro de importe
	0xCC	Registro de pasajeros a anular
Process()	0xB0	Información general del TAG presente.
	0xB1	Resultado de una operación efectuada.
	0xB2	Información con las propiedades de la clave maestra.
	0xB3	Información de la versión de claves.
	0xB4	Lista de aplicaciones existentes.
	0xB5	Información de la versión de fabricación.
	0xB6	Lista de archivos existentes en la aplicación actualmente seleccionada.
	0xB7	Información de un archivo específico.
	0xB8	Información de lectura de archivo estándar.
	0xB9	Información de lectura de archivo de valor.
	0xBA	Información de lectura de archivo de registro.

5.7. RESUMEN DE TIPOS EN "DATA_OUT"

Función	TI	Detalles
*	0xF1	Cabecera.
	0xF2	Tiempo de consumo de la función ejecutada
Version()	0x01	Versión del núcleo.
	0x02	Versión del módulo de venta.
	0x03	Versión del módulo de validación.



	0x04	Versión del módulo de inspección.
	0x05	Información al desarrollador.
Process()	0x99	Información de error de sistema.
	0x10	Autenticación.
	0x11	Propiedades de la clave maestra.
	0x12	Leer versión de una clave.
	0x13	Listar aplicaciones existentes en la tarjeta.
	0x14	Seleccionar una aplicación.
	0x15	Obtener información de fabricación.
	0x16	Listar archivos de la aplicación actualmente seleccionada.
	0x17	Obtener información de un archivo.
	0x18	Leer información de un archivo estándar de datos.
	0x19	Escribir información en un archivo estándar de datos.
	0x1A	Leer información de un archivo de valor.
	0x1B	Incremento de valor de un archivo de valor.
	0x1C	Decremento de valor de un archivo de valor.
	0x1D	Incremento del valor limitado por el crédito establecido.
	0x1E	Escribir información de un registro en un archivo de registros.
	0x1F	Leer información de un registro de un archivo de registros.
	0x20	Borrar completamente un archivo de registros.
	0x21	Confirmar transacción.
	0x22	Invaldar transacción.
	0x30	Operativa con el centro autorizador
LastError()	0x06	Información del último error de sistema.
GetInfo()	0x07	Detalles de la Información al viajero.



6. TABLA ASCII DE CARACTERES DE TEXTO.

Car	Dec	Hex	Car	Dec	Hex	Car	Dec	Hex	Car	Dec	Hex	Car	Dec	Hex
Espacio	32	0x20	Q	81	0x51	é	130	0x82		179	0xB3	Σ	228	0xE4
!	33	0x21	R	82	0x52	â	131	0x83	¡	180	0xB4	σ	229	0xE5
”	34	0x22	S	83	0x53	ä	132	0x84	¢	181	0xB5	μ	230	0xE6
#	35	0x23	T	84	0x54	à	133	0x85	£	182	0xB6	τ	231	0xE7
\$	36	0x24	U	85	0x55	å	134	0x86	¥	183	0xB7	ρ	232	0xE8
%	37	0x25	V	86	0x56	ç	135	0x87	¥	184	0xB8		233	0xE9
&	38	0x26	W	87	0x57	ê	136	0x88	¥	185	0xB9	Ω	234	0xEA
‘	39	0x27	X	88	0x58	ë	137	0x89	¥	186	0xBA	δ	235	0xEB
(40	0x28	Y	89	0x59	è	138	0x8A	¥	187	0xBB	∞	236	0xEC
)	41	0x29	Z	90	0x5A	ì	139	0x8B	¥	188	0xBC	ø	237	0xED
*	42	0x2A	[91	0x5B	í	140	0x8C	¥	189	0xBD	ε	238	0xEE
+	43	0x2B	\	92	0x5C	î	141	0x8D	¥	190	0xBE	η	239	0xEF
,	44	0x2C]	93	0x5D	Ā	142	0x8E	¥	191	0xBF	■	240	0xF0
-	45	0x2D	^	94	0x5E	Ă	143	0x8F	¥	192	0xC0	±	241	0xF1
.	46	0x2E	_	95	0x5F	É	144	0x90	¥	193	0xC1	≥	242	0xF2
/	47	0x2F	`	96	0x60	æ	145	0x91	¥	194	0xC2	≤	243	0xF3
0	48	0x30	a	97	0x61	Æ	146	0x92	¥	195	0xC3	[244	0xF4
1	49	0x31	b	98	0x62	ō	147	0x93	¥	196	0xC4]	245	0xF5
2	50	0x32	c	99	0x63	ö	148	0x94	¥	197	0xC5	÷	246	0xF6
3	51	0x33	d	100	0x64	ō	149	0x95	¥	198	0xC6	≈	247	0xF7
4	52	0x34	e	101	0x65	û	150	0x96	¥	199	0xC7	°	248	0xF8
5	53	0x35	f	102	0x66	ü	151	0x97	¥	200	0xC8	”	249	0xF9
6	54	0x36	g	103	0x67	ÿ	152	0x98	¥	201	0xC9	•	250	0xFA
7	55	0x37	h	104	0x68	Ŏ	153	0x99	¥	202	0xCA	√	251	0xFB
8	56	0x38	i	105	0x69	Ŭ	154	0x9A	¥	203	0xCB		252	0xFC
9	57	0x39	j	106	0x6A	φ	155	0x9B	¥	204	0xCC	²	253	0xFD
:	58	0x3A	k	107	0x6B	£	156	0x9C	¥	205	0xCD	•	254	0xFE
;	59	0x3B	l	108	0x6C	¥	157	0x9D	¥	206	0xCE		255	0xFF
<	60	0x3C	m	109	0x6D	℔	158	0x9E	¥	207	0xCF			
=	61	0x3D	n	110	0x6E	f	159	0x9F	¥	208	0xD0			
>	62	0x3E	o	111	0x6F	á	160	0xA0	¥	209	0xD1			
?	63	0x3F	p	112	0x70	í	161	0xA1	¥	210	0xD2			
@	64	0x40	q	113	0x71	ó	162	0xA2	¥	211	0xD3			
A	65	0x41	r	114	0x72	û	163	0xA3	¥	212	0xD4			
B	66	0x42	s	115	0x73	ñ	164	0xA4	¥	213	0xD5			
C	67	0x43	t	116	0x74	Ñ	165	0xA5	¥	214	0xD6			
D	68	0x44	u	117	0x75	ª	166	0xA6	¥	215	0xD7			
E	69	0x45	v	118	0x76	º	167	0xA7	¥	216	0xD8			
F	70	0x46	w	119	0x77	¿	168	0xA8	¥	217	0xD9			
G	71	0x47	x	120	0x78	®	169	0xA9	¥	218	0xDA			
H	72	0x48	y	121	0x79	™	170	0xAA	¥	219	0xDB			
I	73	0x49	z	122	0x7A	½	171	0xAB	¥	220	0xDC			
J	74	0x4A	{	123	0x7B	¼	172	0xAC	¥	221	0xDD			
K	75	0x4B		124	0x7C	ı	173	0xAD	¥	222	0xDE			
L	76	0x4C	}	125	0x7D	«	174	0xAE	¥	223	0xDF			
M	77	0x4D	~	126	0x7E	»	175	0xAF	¥	224	0xE0			
N	78	0x4E	delete	127	0x7F	»	176	0xB0	¥	225	0xE1			
O	79	0x4F	€	128	0x80	»	177	0xB1	¥	226	0xE2			
P	80	0x50	ü	129	0x81	»	178	0xB2	¥	227	0xE3			



7. TABLA DE CARACTERES DE TRATAMIENTO ESPECIAL.

Dec	Hex	Ctrl
0	0x00	NUL nulo
1	0x01	SO comienzo cabecera
2	0x02	STX comienzo texto
3	0x03	ETX final texto
4	0x04	EOT fin transmisión
5	0x05	ENQ pregunta
6	0x06	ACK respuesta
7	0x07	BEL sonido
8	0x08	BS retroceso
9	0x09	T tabulador horizontal
10	0x0A	LF avance línea
11	0x0B	VT tabulador vertical
12	0x0C	FF avance pagina
13	0x0D	CR retorno carro
14	0x0E	SO despl. inactivo
15	0x0F	SI despl. activo