



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

**UIT-T**

**T.611**

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

(11/94)

**TERMINALES PARA SERVICIOS DE TELEMÁTICA**

---

**INTERFAZ DE PROGRAMACIÓN  
DE COMUNICACIÓN APPLI/COM  
PARA SERVICIOS FACSIMIL GRUPO 3,  
FACSIMIL GRUPO 4, TELETEX, TÉLEX,  
CORREO ELECTRÓNICO Y TRANSFERENCIA  
DE FICHEROS**

**Recomendación UIT-T T.611**

(Anteriormente «Recomendación del CCITT»)

---

## PREFACIO

El UIT-T (Sector de Normalización de las Telecomunicaciones) es un órgano permanente de la Unión Internacional de Telecomunicaciones (UIT). Este órgano estudia los aspectos técnicos, de explotación y tarifarios y publica Recomendaciones sobre los mismos, con miras a la normalización de las telecomunicaciones en el plano mundial.

La Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen Recomendaciones sobre dichos temas.

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución N.º 1 de la CMNT (Helsinki, 1 al 12 de marzo de 1993).

La Recomendación UIT-T T.611 ha sido revisada por la Comisión de Estudio 8 (1993-1996) del UIT-T y fue aprobada por el procedimiento de la Resolución N.º 1 de la CMNT el 11 de noviembre de 1994.

---

### NOTA

En esta Recomendación, la expresión «Administración» se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

© UIT 1995

Es propiedad. Ninguna parte de esta publicación puede reproducirse o utilizarse, de ninguna forma o por ningún medio, sea éste electrónico o mecánico, de fotocopia o de microfilm, sin previa autorización escrita por parte de la UIT.

## ÍNDICE

		<i>Página</i>
PARTE I – DESCRIPCIÓN GENERAL .....		1
1	Alcance.....	1
2	Definiciones y referencias .....	1
2.1	Definiciones relativas a la interfaz.....	1
2.2	Definiciones relativas a los ficheros .....	2
2.3	Referencias .....	3
2.4	Abreviaturas y acrónimos .....	4
2.5	Sistemas operativos.....	6
2.6	Marcas comerciales registradas .....	6
3	Estructura de esta Recomendación.....	6
3.1	Ampliaciones de esta Recomendación.....	7
4	Principios generales .....	7
4.1	Modelo.....	7
4.2	Intercambio de información.....	8
4.3	Entorno de configuración de interfaz (ICE) .....	10
4.4	Principios de depósito.....	11
5	Comportamiento funcional.....	11
5.1	Clases funcionales y perfiles de servicio .....	11
5.2	Tratamiento de errores .....	12
5.3	Múltiples LA y múltiples CA .....	13
5.4	Medios de identificación.....	13
5.5	Facilidad de despacho de ficheros recibidos (DRF) .....	15
5.6	Control de comunicaciones – Registro de CA .....	15
6	Descripciones de datos de tareas (TDD) .....	23
6.1	Presentación genérica de las TDD .....	23
6.2	Descripción de los elementos de TDD.....	25
6.3	Identificador de código (Code-ID) .....	25
6.4	Codificación basada en texto .....	25
6.5	Tratamiento de documentos.....	35
6.6	Funcionalidad de las TDD .....	36
7	Método de intercambio.....	55
7.1	Visión de conjunto de las funciones del método de intercambio básico.....	55
7.2	Funciones del método de intercambio básico .....	58
7.3	Ejecución de las funciones del método de intercambio básico .....	68
8	Formatos de transferencia .....	70
8.1	Formatos de transferencia ASCII ampliado y ASCII estándar de APPLI/COM .....	72
8.2	Formato de transferencia T.61 de APPLI/COM.....	73
8.3	Formato de transferencia TIFF de APPLI/COM .....	73
8.4	Limitaciones de servicio aplicables a los formatos de transferencia .....	82
9	Entorno de configuración de interfaz (ICE) .....	83
9.1	Presentación del ICE.....	85
9.2	Acceso a la información de ICE .....	85
9.3	ICE maestro .....	85
9.4	Descriptor de CA .....	86
9.5	Componentes del descriptor de CA .....	86

10	Clases funcionales y perfiles de servicio.....	88
10.1	Clase funcional A .....	88
10.2	Clase funcional B.....	88
10.3	Funciones adicionales .....	90
10.4	Perfiles de servicio.....	90
PARTE II – DEPENDENCIAS DE SERVICIOS.....		91
11	Servicio: Telefax del grupo 3.....	91
11.1	Elementos de sintaxis específicos del servicio .....	91
11.2	Codificación basada en texto .....	91
11.3	Funcionalidad adicional.....	97
11.4	Fijaciones del descriptor de CA.....	100
12	Servicio: Telefax del grupo 4.....	102
12.1	Elementos de sintaxis específicos del servicio .....	102
12.2	Codificación basada en texto .....	103
12.3	Funcionalidad adicional.....	107
12.4	Fijaciones del descriptor de CA.....	107
13	Servicio: Teletex .....	107
13.1	Elementos de sintaxis específicos del servicio .....	107
13.2	Codificación basada en texto .....	112
13.3	Funcionalidad adicional.....	115
13.4	Fijaciones del descriptor de CA.....	118
14	Servicio: Télex vía teletex.....	119
14.1	Elementos de sintaxis específicos del servicio .....	119
14.2	Codificación basada en texto .....	120
14.3	Funcionalidad adicional.....	122
14.4	Fijaciones del descriptor de CA.....	123
15	Servicio: Télex .....	125
15.1	Elementos de sintaxis específicos del servicio .....	125
15.2	Codificación basada en texto .....	125
15.3	Funcionalidad adicional.....	127
15.4	Fijaciones del descriptor de CA.....	130
16	Servicio: Correo electrónico.....	131
16.1	Elementos de sintaxis específicos del servicio .....	132
16.2	Codificación basada en texto .....	133
16.3	Mensajería interpersonal.....	142
16.4	Notificación interpersonal.....	153
16.5	Informe de entrega.....	154
16.6	Correspondencia de los elementos de servicio MHS.....	156
16.7	Perfiles de correo electrónico .....	158
16.8	Fijaciones del descriptor de CA.....	159
17	Servicio: transferencia de ficheros .....	159
17.1	Elementos de sintaxis específicos del servicio .....	163
17.2	Codificación basada en texto .....	164
17.3	Funcionalidad adicional.....	167
17.4	Fijaciones del descriptor de CA.....	169

	<i>Página</i>
PARTE III – ESQUEMA DE CODIFICACIÓN BINARIA.....	170
18 Descripción del lenguaje C genérico.....	170
18.1 Codificación binaria de las TDD .....	170
PARTE IV – DEPENDENCIA DE PLATAFORMA.....	194
19 Aspectos que dependen de la realización .....	194
19.1 Correspondencia de tipos de datos TDD codificados en binario .....	194
19.2 Método de intercambio por defecto .....	194
19.3 Aplicación del método de intercambio de primitivas .....	195
Anexo A – Sintaxis para presentación y codificación .....	204
A.1 Sintaxis de estilo BNF .....	204
A.2 Notación del lenguaje C.....	205
Anexo B – Ubicación del entorno de configuración de interfaz (ICE) .....	205
Anexo C – Lista de los códigos de error de APPLI/COM .....	207
Anexo D – Ejemplos de intercambios de TDD .....	209
D.1 Ejemplo de sesión de envío .....	209
D.2 Ejemplo de sesión de recepción.....	210
D.3 Ejemplo de sesión de rastreo .....	212
Anexo E – Ejemplo de entorno de configuración de interfaz (ICE).....	214
Anexo F – Método de intercambio de la versión 1992.....	216
F.1 Funciones del método de intercambio de la versión de 1992 .....	216
Anexo G – Información específica del servicio .....	222
G.1 Renglón de identificación de la llamada del servicio telefax G4 y del servicio teletex.....	222
G.2 Identificador de terminal.....	222
Anexo H – Resumen de los formatos de transferencia y de transmisión.....	223
H.1 Formato de transferencia relacionado con los formatos de transmisión .....	223
H.2 Formatos de transmisión relacionados con el servicio.....	223

## **RESUMEN**

La Recomendación T.611, tal como fue aprobada en 1992, define una interfaz de programación de comunicación denominada «APPLI/COM», que proporciona acceso unificado a los servicios facsímil grupo 3, facsímil grupo 4, télex y teletex.

Esta Recomendación revisada se ha redactado de nuevo para mayor claridad y para incluir los servicios de correo electrónico en general, los sistemas de tratamiento de mensajes descritos en las Recomendaciones de la serie UIT-T X.400 en particular, y los servicios de transferencia de ficheros.

Se ha tratado especialmente de garantizar la compatibilidad de esta Recomendación con la versión de 1992.

**INTERFAZ DE PROGRAMACIÓN DE COMUNICACIÓN APPLI/COM PARA SERVICIOS  
FACSIMIL GRUPO 3, FACSIMIL GRUPO 4, TELETEx, TÉLEX, CORREO  
ELECTRÓNICO Y TRANSFERENCIA DE FICHEROS**

(revisada en 1994)

**PARTE I – DESCRIPCIÓN GENERAL**

**1 Alcance**

En esta Recomendación se define la interfaz de comunicación de programación (PCI, *programming communication interface*) denominado «APPLI/COM».

Los conceptos generales de las interfaces PCI se definen en la Recomendación UIT-T F.581 (1992).

Esta PCI proporciona acceso unificado a diferentes servicios de telecomunicaciones, como el telefax grupo 3 u otros servicios telemáticos.

Esta Recomendación especifica actualmente el acceso a los siguientes servicios telemáticos:

- telefax grupo 3;
- telefax grupo 4;
- teletex;
- télex<sup>1)</sup>;
- servicios de correo electrónico;
- servicios de transferencia de ficheros.

Esta interfaz tiene por objeto el intercambio de mensajes entre dos entidades (la aplicación local y la aplicación de comunicación). Esta Recomendación describe la estructura y el contenido de estos mensajes, así como la manera de intercambiarlos. Se definen dos esquemas de codificación: la codificación basada en texto y la estructurada en binario.

La realización de esta PCI no es una condición necesaria para participar en un servicio telemático. Sin embargo, la utilización de esta interfaz proporcionará compatibilidad binaria entre las entidades mencionadas anteriormente. Por consiguiente, los usuarios finales podrán beneficiarse de la compatibilidad de productos provenientes de diferentes fabricantes, que podrán utilizarse juntos conectándolos simplemente.

Esta Recomendación proporciona un marco para futuras ampliaciones que mantengan las funcionalidades coherentes, y compatibles con las versiones que sigan.

En resumen, esta Recomendación constituye una interfaz de aplicación de programación (API, *application programming interface*) de alto nivel que oculta todas las peculiaridades de la telecomunicación, pero proporciona a los diseñadores de aplicaciones un gran poder de control y supervisión sobre la actividad de telecomunicación.

**2 Definiciones y referencias**

**2.1 Definiciones relativas a la interfaz**

Las siguientes definiciones son aplicables a la interfaz.

**2.1.1 aplicación de comunicación (CA, *communication application*):** Una aplicación de comunicación (CA) es proporcionada por un soporte físico y/o por un soporte lógico que permiten participar en servicios de telecomunicación normalizados. En la cláusula 1 se enumeran los servicios de telecomunicación sustentados por esta Recomendación.

---

<sup>1)</sup> El acceso al servicio teletex no incluye la facilidad de diálogo.

**2.1.2 aplicación local (LA, local application):** Una aplicación local (LA) es una aplicación capaz de generar ficheros o documentos y posiblemente capaz de gestionar comunicaciones de tipo diálogo. Estas LA pueden ser programas de tratamiento de textos, hojas de cálculo, editores gráficos, editores de ficheros, etc. La LA generará mensajes conformes a esta Recomendación.

**2.1.3 descripción de datos de tareas (TDD, task data description):** Una descripción de datos de tareas (TDD) describe la estructura de los mensajes intercambiados entre una LA y una CA (pero no la manera de intercambiarlos efectivamente). Una TDD de petición describe un intercambio originado por una LA hacia una CA. Una TDD de respuesta describe un intercambio originado por una CA hacia una LA.

**2.1.4 método de intercambio (EM, exchange method):** El método de intercambio (EM) describe la manera de intercambiar las TDD entre una CA y una LA. En esta Recomendación se define un método de intercambio genérico que se adaptará al sistema operativo deseado por los medios descritos en esta Recomendación. El método de intercambio genérico admite entornos de red de zona local y red de zona amplia.

**2.1.5 descriptor de CA:** El descriptor de CA es una recopilación de información sobre la utilización de una CA. El descriptor de CA describe las facilidades y características de una CA dada, de manera que cualquier LA que utilice CA sepa cómo actuar. El descriptor de CA es un componente del entorno de configuración de interfaz (ICE).

**2.1.6 entorno de configuración de interfaz (ICE, interface configuration environment):** El entorno de configuración de interfaz (ICE) está compuesto por el ICE principal y descriptores de CA conexos. El ICE principal indica todas las CA que pueden ser alcanzadas desde el interior de una LA dada. La finalidad del ICE es ayudar a la LA en la selección de una CA apropiada en lo que respecta a los requisitos de la LA.

## 2.2 Definiciones relativas a los ficheros

En esta Recomendación se tratan varias clases de ficheros: los intercambiados entre una LA y una CA y los intercambiados a través de una red de telecomunicaciones. En el cuerpo principal de esta Recomendación se utilizan las siguientes definiciones.

**2.2.1 ficheros para transferencia:** Como se muestra en la Figura 1 los ficheros para transferencia son los intercambiados entre la LA y la CA. El formato de estos ficheros (formato de transferencia) se define en la presente Recomendación.

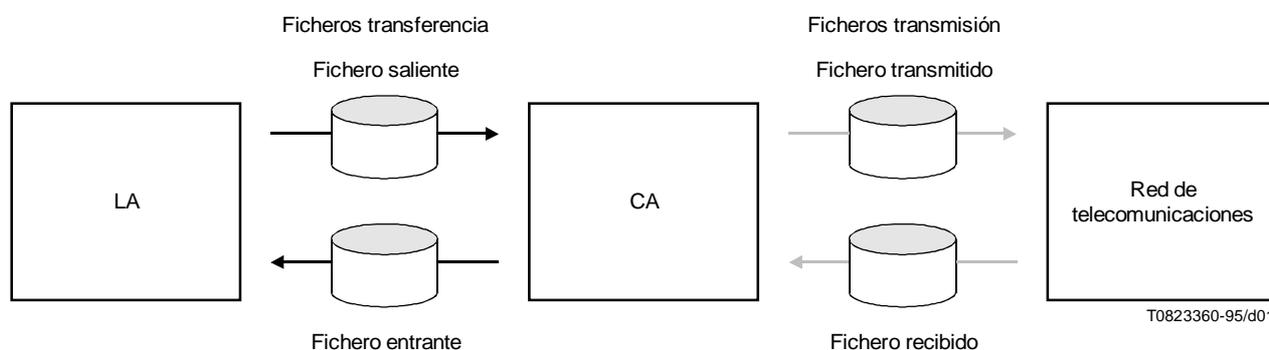


FIGURA 1/T.611  
Distintos tipos de ficheros definidos por APPLI/COM

**2.2.1.1 fichero saliente:** El fichero saliente es un fichero transmitido por la LA a la CA para que ésta lo retransmita a través de una red de telecomunicaciones. El formato de este fichero es uno de los posibles formatos de transferencia.

**2.2.1.2 fichero entrante:** El fichero entrante es un fichero transmitido por la LA a la CA. Generalmente corresponde a ficheros recibidos de la red. El formato de este fichero es uno de los posibles formatos de transferencia.

**2.2.1.3 formato de transferencia:** El formato de transferencia define la estructura de los ficheros para transferencia. Según los servicios de telecomunicaciones utilizados, algunos formatos de transferencia son más adecuados que otros. Los posibles formatos de transferencia se definen en la cláusula 8. Un formato de transferencia es identificado por el ID de conversión. La identificación y selección de un formato de transferencia se describe en 5.4.5.

**2.2.2 ficheros para transmisión:** Los ficheros para transmisión son los intercambiados por la CA a través de la red, según se ilustra en la Figura 1. El formato de estos ficheros, denominado formato de transmisión, es definido intrínsecamente por el servicio de telecomunicaciones utilizado.

**2.2.2.1 fichero transmitido:** Un fichero transmitido es un fichero enviado por la CA a través de una red de telecomunicaciones en un formato adecuado para intercambio mediante el protocolo utilizado en el servicio de telecomunicaciones.

**2.2.2.2 fichero recibido:** Un fichero recibido es un fichero formado por la CA con la información recibida a través de la red de telecomunicaciones en el formato utilizado para intercambio mediante el protocolo empleado en el servicio de telecomunicaciones.

**2.2.2.3 formato de transmisión:** El formato de transmisión define la estructura de los ficheros transmisión de acuerdo con los servicios de telecomunicaciones utilizados. El formato de transmisión es identificado por el identificador de tipo (véase 5.4.6).

## 2.3 Referencias

En esta Recomendación se hace referencia a las siguientes Recomendaciones del UIT-T:

- Recomendación F.59, *Características generales del servicio télex internacional*.
- Recomendación F.60, *Disposiciones relativas a la explotación del servicio télex internacional*.
- Recomendación F.160, *Disposiciones generales relativas a la explotación de los servicios facsímil públicos internacionales*.
- Recomendación F.184, *Disposiciones relativas a la explotación del servicio facsímil internacional entre estaciones de abonado equipadas con aparatos facsímil del grupo 4 (telefax 4)*.
- Recomendación F.200, *Servicio teletex*.
- Recomendación F.581, *Recomendación de servicio sobre interfaces de comunicación de programación*.
- Recomendación S.1, *Alfabeto telegráfico internacional N.º 2*.
- Recomendación T.4, *Normalización de los aparatos facsímil del grupo 3 para transmisión de documentos*.
- Recomendación T.6, *Esquemas de codificación facsímil y funciones de control de codificación para los aparatos facsímil del grupo 4*.
- Recomendación T.30, *Procedimientos de transmisión de documentos por facsímil por la red telefónica general conmutada*.
- Recomendación T.35, *Procedimiento para la asignación de códigos de miembro de la UIT*.
- Recomendación T.50, *Alfabeto internacional de referencia*.
- Recomendación T.51, *Juegos de caracteres codificados del alfabeto latino para los servicios telemáticos*.
- Recomendación T.52, *Juegos de caracteres codificados no latinos para los servicios telemáticos*.
- Recomendación T.61, *Repertorio de caracteres y juegos de caracteres codificados para el servicio teletex internacional*.

NOTA – La Recomendación T.61 se ha remplazado por las Recomendaciones de la serie T.50 en 1993.

- Recomendación T.62, *Procedimientos de control para los servicios teletex y facsímil del grupo 4*.
- Recomendación T.434, *Especificación de la transferencia de ficheros binarios*.
- Recomendación T.500, *Visión de conjunto de las Recomendaciones de la serie T.500*.
- Recomendación T.563, *Características de los terminales para aparatos facsímil del grupo 4*.
- Recomendación T.571, *Características de los terminales para la transferencia de ficheros telemáticos con aparatos facsímil del grupo 4 y teletex*.
- Recomendación X.208, *Especificación de la notación de sintaxis abstracta uno (ASN.1)*.
- Recomendación X.209, *Especificación de las reglas básicas de codificación de la notación de sintaxis abstracta uno (ASN.1)*.
- Recomendación X.400, *Serie de Recomendaciones UIT-T sobre sistema de tratamiento de mensajes*.

Se hace también referencia a las siguientes Normas Internacionales:

- ISO/CEI 639.2, *Code for the presentation of code of languages*.
- ISO/CEI 9735 (1990), *Electronic Data Interchange For Administration Commerce and Transport (EDIFACT) – Application level syntax rules*.

## 2.4 Abreviaturas y acrónimos

API	Interfaz de programación de aplicación ( <i>application programming Interface</i> )
APPLI/COM	Interfaz entre aplicaciones locales y aplicaciones de comunicación ( <i>local applications/communication applications</i> )
ASCII	Códigos de la norma americana para intercambio de información ( <i>American standard codes for information interchange</i> )
ASN.1	Notación de sintaxis abstracta uno ( <i>abstract syntax notation one</i> ) definida por las Recomendaciones UIT-T X.208 y X.209
BFT	Transferencia de ficheros binarios ( <i>binary file transfer</i> ), definida en la Recomendación UIT-T T.434
BNF	Forma Backus Naur ( <i>Backus Naur form</i> )
BTM	Modo transparente básico ( <i>basic transparent mode</i> ), definido en la Recomendación UIT-T T.434
CA	Aplicación de comunicación ( <i>communication application</i> )
CIL	Renglón (o línea) de identificación de la llamada ( <i>call identification line</i> )
CR	Retorno del carro (o retroceso del carro) ( <i>carriage return</i> )
CTL	Documento de control ( <i>control document</i> ); tipo de documento definido en la Recomendación UIT-T T.62
DCX	Fichero PCX multipágina ( <i>multipage PCX file</i> )
DLL	Biblioteca de vinculación dinámica ( <i>dynamic link library</i> )
DRF	Despachar ficheros recibidos ( <i>dispatch received files</i> )
DTM	Modo transparente documento ( <i>document transparent mode</i> ) definido en la Recomendación UIT-T T.434
E-Mail	Correo electrónico ( <i>electronic mail</i> ); término general utilizado para servicios de correo electrónico
EBCDIC	Código EBCDIC ( <i>extended binary coded decimal interchange code</i> )
EDI	Intercambio electrónico de datos ( <i>electronic data interchange</i> ), definido en la Recomendación UIT-T T.571
EM	Método de intercambio ( <i>exchange method</i> )
EMail	Identificador de servicio para los servicios de correo electrónico ( <i>service-ID for E-Mail services</i> )

ESC	Escape ( <i>escape</i> )
FC	Clase funcional ( <i>functional class</i> )
FCA	Clase funcional A ( <i>functional class A</i> )
FCB	Clase funcional B ( <i>functional class B</i> )
FF	Nueva página ( <i>form feed</i> )
FT	Identificador de servicio para los servicios de transferencia de ficheros ( <i>service-ID for fail transfer services</i> )
FX3	Identificador de servicio para los servicios telefax del grupo 3 ( <i>service-ID for telefax group 3 services</i> )
FX4	Identificador de servicio para los servicios telefax del grupo 4 ( <i>service-ID for telefax group 4 services</i> )
ICE	Entorno de configuración de interfaz ( <i>interface configuration environment</i> )
IPM	Mensaje interpersonal ( <i>interpersonal message</i> ); mensaje intercambiado entre usuarios a través de UA (terminología de X.400)
IRA	Alfabeto internacional de referencia ( <i>international reference alphabet</i> ) (Rec. T.50)
IRV	Versión internacional de referencia ( <i>international reference version</i> ) (Rec. T.50)
LA	Aplicación local ( <i>local application</i> )
LAN	Red de zona local ( <i>local area network</i> )
LF	Cambio de renglón (o nueva línea) ( <i>line feed</i> )
MD	Documento de supervisión ( <i>monitor document</i> ); tipo de documento definido en la Recomendación UIT-T T.62
MH	Código Huffman modificado ( <i>modified Huffman code</i> ); definido en la Recomendación UIT-T T.4
MHS	Sistema de tratamiento de mensajes ( <i>message handling system</i> ) definido en las Recomendaciones de la serie X.400
MR	Código Read modificado ( <i>modified Read code</i> ); definido en la Recomendación UIT-T T.4
MS	Almacén de mensajes (dispositivo de almacenamiento de mensajes) ( <i>message store</i> ) (terminología de la Rec. X.400)
MTA	Agente de transferencia de mensajes ( <i>message transfer agent</i> ) (terminología de la Rec. X.400)
MTS	Sistema de transferencia de mensajes ( <i>message transfer system</i> ) (terminología de la Rec. X.400)
Nombre O/R	Nombre originador/recibiente ( <i>originator/recipient name</i> ) (terminología de la Rec. X.400)
OPD	Documento operador ( <i>operator document</i> ); tipo de documento definido en la Recomendación UIT-T T.62
P1	Protocolo de transferencia MTS; protocolo que define el sobre MHS para el intercambio MTA-MTA definido en las Recomendaciones UIT-T X.411 (1988) y X.419 (1988, definición de sintaxis abstracta)
P2	Formato de mensajería interpersonal (IPM), definido en la Recomendación UIT-T X.420 (1984, o como P22 1988)
P3	Protocolo de acceso MTS, definido en la Recomendación UIT-T X.419 (1988)
P7	Protocolo de acceso MS, definido en la Recomendación UIT-T X.413 (1988)
PCI	Interfaz de telecomunicación de programación ( <i>programming communication interface</i> )
PCX	Una variedad de ficheros de información de matrices de puntos utilizados en programas de computadores personales

P <sub>edi</sub>	Formato para el intercambio electrónico de datos de usuario a usuario en la X.400, definido en las Recomendaciones UIT-T F.435 y X.435
RDSI	Red digital de servicios integrados
RTPC	Red telefónica pública conmutada
SP	Espacio ( <i>Space</i> )
TDD	Descripción de datos de tareas ( <i>task data description</i> )
TFT	Transferencia de ficheros telemáticos ( <i>telematic file transfer</i> ) definida en la Recomendación UIT-T T.571
TIFF	Formato de fichero de imagen rotulada ( <i>tagged image file format</i> )
TLX	Identificador de servicio para servicios télex (sin facilidad de diálogo)
TTX	Identificador de servicio para servicios teletex
TX	Identificador de servicio para télex vía servicios teletex
UA	Agente de usuario ( <i>user agent</i> ) (terminología de la Rec. X.400)
WAN	Red de zona amplia ( <i>wide area network</i> )

## 2.5 Sistemas operativos

En esta Recomendación se mencionan las siguientes versiones de sistemas operativos:

MacOS	Todas las versiones
MS-DOS	Versión 3.1 o superior
UNIX	Todas las versiones
Windows	Versión 3.0 o superior
OS/2	Versión 1.0 o superior

En la presente Recomendación el término DOS significa versiones de sistemas operativos compatibles con MS-DOS.

## 2.6 Marcas comerciales registradas

Los nombres indicados a continuación son marcas comerciales registradas, pertenecientes a sus respectivos propietarios:

MacOS	Apple Computer, Inc.
MS-DOS	Microsoft Corporation
UNIX	AT&T
TIFF	Aldus Corporation
Windows	Microsoft Corporation
OS/2	International Business Machines Corporation

## 3 Estructura de esta Recomendación

En esta Recomendación se tratan:

- requisitos de interfaz;
- definiciones;
- principios generales;
- características de aplicación;
- clases de implementación;
- descripción de datos de tareas (TDD);
- método de intercambio;
- ficheros entrantes y salientes;
- control de comunicaciones – registro de CA.

### 3.1 Ampliaciones de esta Recomendación

A fin de tener en cuenta la evolución de las tecnologías de la telecomunicación, esta Recomendación ha sido estructurada de manera que pueda aceptar los nuevos servicios de telecomunicaciones, o las nuevas opciones de los actuales servicios de telecomunicaciones, sin que se altere su organización general.

Además, en esta Recomendación se proporcionan mecanismos de ampliación para enriquecer:

- el método de intercambio entre LA y CA;
- los mensajes (TDD);
- los formatos de transferencia,

al mismo tiempo que se mantiene la compatibilidad con la Recomendación fundamental.

## 4 Principios generales

La interfaz se rige por los siguientes principios:

- Deberá ser independiente del equipo físico del computador (por ejemplo, la interfaz podrá ser soportada por un equipo o un programa de comunicaciones).
- Deberá ser independiente de los sistemas operativos y de los lenguajes de programación (el método de intercambio es la única parte que depende de los sistemas operativos y de los lenguajes de programación). Esta Recomendación proporciona directrices para lograr la compatibilidad en sistemas operativos MS-DOS, OS/2, Windows y Unix. Los demás sistemas operativos se considerarán a petición.
- La descripción formal de los mensajes de interfaz deberá basarse en una descripción «similar a la BNF»<sup>2)</sup> Podrán utilizarse diversos esquemas de codificación para presentar los mensajes de interfaz.
- Deberá estar orientada al depósito de tareas.
- Deberá tener en cuenta la demanda de que pueda haber múltiples aplicaciones activas en el mismo servidor, así como la demanda de aplicaciones LAN/WAN. La interfaz podrá utilizarse cuando estén actuando varias aplicaciones locales y/o varias aplicaciones de comunicación.
- Deberá ser extensible y flexible.

### 4.1 Modelo

La interacción LA-CA se ajusta al modelo cliente-servidor mostrado en la Figura 2.

En este contexto:

- Se considera que la CA es un servidor que proporciona a LA algunos servicios de telecomunicaciones.
- Se considera que LA es el cliente de una CA, y que utiliza los servicios de telecomunicaciones proporcionados por esa CA.

En consecuencia:

- como servidor, la CA tiene que ajustarse a una de las dos clases funcionales descritas en la cláusula 10, y seguir los métodos de intercambio definidos en esta Recomendación;
- como cliente, la LA no está sujeta a ninguna limitación en lo que respecta a la interfaz.

Por consiguiente, la iniciación del intercambio de información entre LA y CA incumbe siempre a la LA. La LA puede optar por no ser perturbada en su trabajo local por posibles eventos provenientes de la CA.

#### 4.1.1 Cometido de la LA

En lo que respecta a la interfaz, pueden distinguirse, en la LA, dos partes funcionales distintas:

- el soporte lógico que genera los ficheros salientes y/o lee los ficheros entrantes;
- el soporte lógico que maneja la comunicación.

---

<sup>2)</sup> BNF Forma Backus Naur (*Backus Naur Form*).

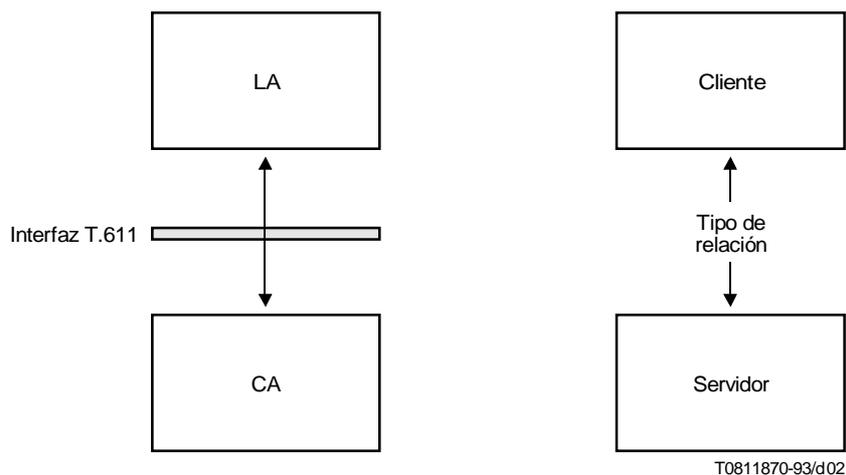


FIGURA 2/T.611  
Relación entre LA y CA según el modelo cliente-servidor

Este último proporciona:

- 1) Diálogos hombre-máquina o procesamiento automático para el envío de ficheros salientes, tratamiento (presentación visual, presentación impresa, salvaguarda) de los ficheros entrantes, supervisión de actividades de la CA, y solicitud de una acción determinada de la gestión del sistema y/o servicio.
- 2) Conversión de documentos desde un formato de transferencia adecuado para la CA.
- 3) Acceso a características facultativas de la CA, indicadas en el descriptor de CA del ICE.

NOTA – La LA, como cliente, no está obligada a utilizar todas las características de la CA.

#### 4.1.2 Cometido de la CA

La CA, como servidor, se encarga de:

- la gestión de comunicaciones;
- la conversión de formatos de ficheros del formato de transferencia al formato de transmisión (y viceversa);
- la gestión de las características de la CA (si las hubiera) descritas en el descriptor de CA del ICE.

## 4.2 Intercambio de información

A través de la interfaz se accede a la siguiente información:

- Ficheros salientes: ficheros o documentos entregados por la LA a la CA para que ésta los retransmita por una red.
- Ficheros entrantes: ficheros o documentos que la CA entrega a la LA después de recibirlos de la red.
- Registros de CA: grupos de información que reflejan los eventos de transmisión y recepción gestionados por la CA.

La información es intercambiada entre las CA y LA por medio de la TDD de petición y de la TDD de respuesta comprendidas en uno de los grupos indicados en el Cuadro 1. Dicho cuadro enumera todos los grupos TDD definidos en esta Recomendación y especifica en particular si la CA genera una TDD de respuesta para una determinada TDD de petición. Las TDD se describen en detalle en la cláusula 6.

CUADRO 1/T.611

**Grupos TDD**

Grupo TDD	¿Respuesta?	Observaciones
Envío (Send)	«Sí», si la petición lo requiere	Pide a la CA que envíe uno o más ficheros a través de la red, mediante un servicio de telecomunicaciones, a un conjunto de recipientes.
Recepción (Receive)	Sí	Pide a la CA que recupere un fichero entrante resultante de un fichero recibido. La TDD de respuesta se rellena con la ubicación y el formato de transferencia del fichero entrante.
Rastreo (Trace)	Sí	Pide a la CA que ejecute alguna acción sobre un registro de CA, o sobre un conjunto de dichos registros, que se encuentran en un estado determinado. La acción que se ejecutará se describe en la petición Trace.
Depósito (Submit)	Sí	Pide a la CA que ejecute una acción como la de convertir un fichero a un formato determinado, o la impresión de un fichero de acuerdo con las reglas propias de un servicio de telecomunicaciones.
Ampliación (o extensión) (Extend)	Sí	Da la posibilidad de ampliaciones a la Recomendación, que pueden realizarse como modificaciones formales de la Recomendación, o como ampliaciones en el plano nacional, o privadas.

El funcionamiento correcto de la interfaz requiere la definición detallada de los datos y la especificación detallada de las interacciones entre la LA y la CA. Pueden coexistir distintos métodos de intercambio, según las necesidades particulares de los diferentes entornos. Se definen dos métodos de intercambio de información (véase la cláusula 7).

- Un método de intercambio de ficheros que puede ser aplicado fácilmente en una amplia gama de computadores y sistemas operativos, pero que proporcionan un caudal de computador más bajo.
- Un método de intercambio de primitivas que emplea mecanismos dependientes del sistema operativo para transportar información. Proporciona mayor caudal, pero menor portabilidad.

En consecuencia, se introduce el concepto de descripción de datos de tareas (TDD, *task data description*) (véase la Figura 3). Una TDD es una estructura abstracta de datos intercambiada entre la LA y la CA en la que se describe una tarea determinada que la CA ha de realizar, o una respuesta de la CA a una de estas tareas.

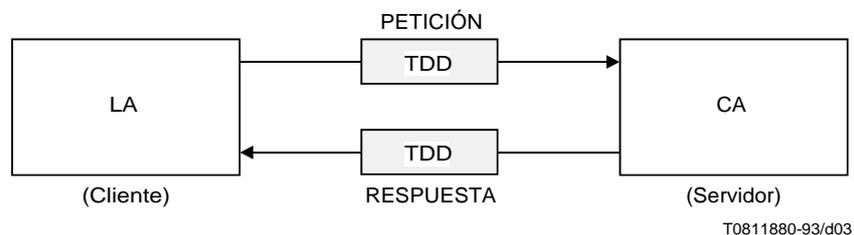


FIGURA 3/T.611  
**Descripción de datos de tareas (TDD)**

Las TDD son independientes del método de intercambio de información utilizado, lo cual tiene por finalidad:

- simplificar los procedimientos de prueba;
- permitir la elección del método de codificación de las TDD.

Una TDD transporta información acerca de la tarea que la LA espera realice la CA, con todos los parámetros apropiados. Dado que la comunicación de esas tareas se ajusta a un modelo «cliente-servidor», se intercambian realmente dos tipos de TDD:

- una TDD de petición generada por la LA y dirigida a la CA, que describe la acción que se ha de efectuar;
- una TDD de respuesta generada por la CA y dirigida a la LA, que describe la respuesta a la anterior petición.

Por construcción, una LA puede enviar muchas TDD de petición sin esperar las TDD de respuesta correspondientes. Algunas peticiones no requieren respuesta.

La CA puede tratar las TDD de petición en cualquier orden.

#### **4.2.1 Codificación de las TDD**

Como se describe con más detalle en 6.1, las TDD se pueden codificar con diferentes esquemas, que están basados en texto o estructurados en binario.

#### **4.2.2 Formato de los datos intercambiados**

Los datos de formatos se intercambian entre las LA y las CA, es decir, el formato de los ficheros salientes y entrantes, se conformará con reglas de codificación y de presentación bien definidas. Estas reglas se denominan formatos de transferencia. Los formatos de transferencia se definen detalladamente en la cláusula 8.

### **4.3 Entorno de configuración de interfaz (ICE)**

El concepto de entorno de configuración de interfaz (ICE) se utiliza para identificar los servicios ofrecidos por la CA dentro de un entorno informático y las características de cada una de las CA. La estructura del ICE tiene dos niveles:

- 1) El ICE principal (Master-ICE) es un fichero accesible universalmente que contiene una lista de todas las CA disponibles. Indica la manera de acceder a la información de configuración detallada para cada CA.
- 2) El descriptor de CA es la fuente de la información de configuración detallada sobre cada CA. El acceso a la información de descriptor de CA puede ser en forma de un fichero o ser generado por métodos dinámicos adaptados al entorno informático, tales como ficheros ejecutables, bibliotecas de vinculación dinámicas (DLL) u otros procedimientos.

Cada CA tiene sus características únicas. El descriptor de CA da acceso a una lista completa de esas características. La combinación del ICE global y de descriptors de CA proporciona un medio normalizado para que las LA accedan a las características originales y/o suplementarias de las CA. En la Figura 4 se muestra la relación entre el ICE principal y los descriptors de CA.

El ICE representa una fuente global para todas las LA conformes a esta Recomendación. El ICE maestro contendrá la siguiente información de encabezamiento para cada CA:

- los servicios soportados por la CA;
- el método o los métodos de intercambio soportados por la CA;
- uno o más métodos de acceso para cada descriptor de CA.

El descriptor de CA contiene, como mínimo, el siguiente conjunto de informaciones de configuración relacionadas con la interfaz:

- el método utilizado para el intercambio de las TDD entre las LA y la CA;
- detalles del método de intercambio (trayectos, memorias tampón, puntos de entrada);
- codificación de las TDD;
- clase funcional;
- servicios de telecomunicaciones sustentados;
- facilidades de la CA.

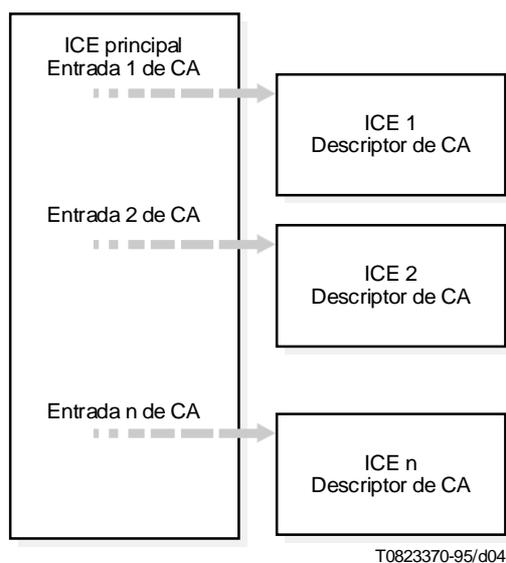


FIGURA 4/T.611  
**Relación entre el ICE principal y los descriptores de CA**

La provisión del descriptor de CA es obligatoria para toda CA. Toda LA puede confiar en la información incluida en el ICE principal o en un descriptor de CA. Las LA no modificarán la información contenida en el ICE principal ni en el descriptor de CA.

El ICE principal es un fichero lógico. La sintaxis y el formato del ICE principal y del descriptor de CA se describen en la cláusula 9. La ubicación del ICE principal en una plataforma dada se describe en el Anexo B.

#### 4.4 Principios de depósito

Dado que las CA y las LA pueden compartir diferentemente las funciones requeridas para ajustarse a un servicio de telecomunicaciones, es posible que algunas de estas funciones se dupliquen o se pierdan. Son ejemplos de estas funciones:

- la comprobación de que el formato de transferencia cumple requisitos específicos del servicio;
- la impresión de un documento de acuerdo con el servicio de telecomunicaciones a través del cual fue (o será) transportado;
- la conversión de formatos de transmisor hacia o desde formatos de transferencia.

El principio de depósito permite a las LA servirse de funciones soportadas por la propia CA. De esta manera, las LA no tienen necesidad de soportar funciones que ya son tratadas por la CA. Con esto se asegura que se sustentan realmente todas las funciones de servicios de telecomunicaciones requeridas para cualquier equipo que cumpla las especificaciones.

## 5 Comportamiento funcional

### 5.1 Clases funcionales y perfiles de servicio

Se definen dos clases funcionales (FC, *functional classes*), denominadas clase funcional A (FCA) y clase funcional B (FCB). La clase funcional B es un superconjunto de la clase funcional A. Para una información más detallada, véase la cláusula 10.

La clase funcional de una CA se indicará explícitamente en la documentación del fabricante y en el campo apropiado del descriptor de CA.

Además, la interfaz proporciona acceso a características adicionales de la CA. Estas características no son esenciales en lo que respecta a las Recomendaciones UIT-T. Una LA puede utilizar cualquiera de estas características adicionales independientemente de su clase funcional, a condición de que la CA la soporte. Las características adicionales soportadas por una CA dada se indicarán explícitamente en el ICE (véase la cláusula 9).

Para facilitar aún más la reducción de las incompatibilidades entre las aplicaciones de diferentes vendedores, esta Recomendación sustenta el concepto de perfiles de servicio. Un perfil de servicio, definido para algunos servicios, agrupa un conjunto determinado de características de servicio que tienen que ser soportadas por las realizaciones que alegan ser compatibles con esta Recomendación (véase la cláusula 10).

## 5.2 Tratamiento de errores

### 5.2.1 Errores simples

La CA no tendrá en cuenta los elementos de sintaxis que estén fuera de contexto (por ejemplo, la especificación de la velocidad de transmisión para el servicio télex) y continuará la ejecución.

### 5.2.2 Rechazo

Los errores sintácticos (por ejemplo, no reconocimiento de elementos sintácticos, ausencia de elementos sintácticos clasificados como obligatorios, elementos sintácticos en conflicto, múltiples apariciones de un mismo elemento sintáctico, parámetros fuera de gama en una TDD) pueden provocar el rechazo de la TDD. Un formato de fichero saliente inválido puede también provocar el rechazo de la correspondiente TDD por la CA. La CA puede también descartar el fichero saliente. Además, la CA puede rechazar la TDD si los ficheros asociados no se transfieren a la CA en un plazo determinado.

### 5.2.3 Gamas de códigos de error

Véase el Cuadro 2.

CUADRO 2/T.611

#### Asignación de códigos de error

Código de error	Significado
0000	Éxito (ausencia de error). Será soportado por todas las CA
0001-4999	Reservado para uso privado de la CA
5000-5999	Errores diversos que no caen en otras categorías
6000-6999	Errores de sintaxis
7000-7999	Errores relativos a los recursos y a las operaciones de entrada/salida (errores relacionados con el sistema operativo y el soporte físico)
8000-8999	Errores de conversión o relacionados con el formato de transferencia
9000-9999	Errores relacionados con el servicio (errores en las señales de servicio y fallos del servicio)

El Anexo C contiene una lista completa de los códigos de error asignados en esta Recomendación.

### 5.3 Múltiples LA y múltiples CA

Gracias a la interfaz, múltiples LA pueden estar conectadas a una o varias CA simultáneamente. Con el fin de controlar el acceso a múltiples CA se ha definido el ICE. A nombre del ICE, la LA seguirá un procedimiento de establecimiento de interfaz en dos pasos durante la fase de inicialización:

- primero, la aplicación local seleccionará una aplicación de comunicación apropiada examinando el entorno de configuración de interfaz (ICE);
- después, la aplicación local se «engachará» («login») a la CA seleccionada.

Una vez que la aplicación local se ha «enganchado» («logged in») la aplicación de comunicación, podrá utilizar libremente cualquier servicio proporcionado por la CA hasta que ponga fin a la sesión de intercambio es decir, se «desenganche», («logs out») de la CA.

#### 5.3.1 Paso 1: Selección de una CA

El primer paso que dará la aplicación local en el procedimiento de establecimiento de la interfaz es acceder al ICE, que proporciona una lista de las CA accesibles desde el sistema (véanse 4.3 y 9).

#### 5.3.2 Paso 2: Enganche de una LA a una CA

Una vez que la aplicación local ha seleccionado una CA apropiada entre las indicadas en el ICE, tiene que registrarse con la CA seleccionada. Esto se denomina proceso de enganche. El proceso de enganche se realiza utilizando un nombre de enganche único y devuelve un identificador de conexión. Este identificador es calculado por la CA.

Para efectuar el enganche, la LA se basará en la información encontrada en el descriptor de CA.

El proceso de «desenganche» (logout) permite el desacoplamiento entre la LA y la CA: lo invoca la LA cuando desea desconectarse de la CA. Como resultado de este proceso de desenganche, la CA descartará el identificador de conexión y se convierte en no válido. Por tanto, la LA ya no podrá tener acceso por medio de dicho identificador de conexión.

Los procesos de enganche y desenganche facilitan la aplicación de esquemas de seguridad, que son especialmente importantes en sistemas multiusuarios. Proporcionan también medios para establecer mecanismos de seguridad entre la LA y la CA.

### 5.4 Medios de identificación

Para atender a los diversos intercambios de información a través de la interfaz es necesario identificar inequívocamente las entidades que comunican y los eventos de comunicación. Los identificadores proporcionan medios para:

- distinguir los diversos eventos de comunicaciones (COM-ID);
- identificar las LA enganchadas a una CA dada (LA-ID);
- identificar las peticiones enviadas por una LA a una CA dada (REQ-ID);
- definir el formato de transferencia que habrá de utilizarse (convert-ID);
- definir el formato de transmisión que habrá de utilizarse (Type-ID).

En las subcláusulas que siguen se describen detalladamente estos identificadores.

#### 5.4.1 Identificación de comunicaciones de CA (COM-ID)

Puesto que una CA puede tratar diferentes peticiones de diferentes LA y/o de la red, es necesario asignar un identificador a cada uno de esos eventos.

El identificador COM-ID es un identificador único proporcionado por la CA y asignado a cada evento de comunicación que se produce en una CA. El COM-ID es un campo del registro de CA (véase 5.6.2).

La CA genera un nuevo COM-ID cuando se genera un registro de CA. Esto sucede cuando:

- 1) una LA emite una petición de envío a la CA;
- 2) una LA emite a la CA una petición de rastreo (Trace), función de reprogramación (Reschedule);
- 3) la CA procesa un fichero recibido.

El COM-ID asegura que una determinada LA pueda recuperar cualquier petición de transmisión programada para envío, incluso si el diálogo LA-CA se terminó por cualquier motivo.

#### **5.4.2 Identificación de las LA en una CA (LA-ID)**

Dado que esta Recomendación permite que múltiples LA, o múltiples instancias de una LA, utilicen simultáneamente una CA dada, las LA o sus instancias tienen que estar identificadas inequívocamente ante esa CA.

A fin de distinguir las diferentes LA, unas de otras, esta Recomendación define el identificador LA-ID como el identificador único que designa una instancia de una determinada LA que comunica con una CA. El LA-ID es un campo del registro de CA (véase 5.6.2).

La CA puede negarse a procesar una TDD de petición que contenga un LA-ID que no conozca. Esto proporciona un medio para controlar el acceso a una CA (véase 5.6.1).

El LA-ID se asigna estáticamente a la LA, o a la instancia de LA. La regla para la asignación del LA-ID está fuera del ámbito de esta Recomendación.

#### **5.4.3 Identificación de peticiones de LA (REQ-ID)**

Esta Recomendación permite a una LA enviar múltiples peticiones a una CA. Puesto que la interfaz permite la entrega de TDD de respuesta a la LA en un orden diferente del que se hicieron las peticiones, es necesario identificar las TDD de petición y las TDD de respuesta correspondientes. Esta Recomendación define por tanto el identificador REQ-ID como la única referencia de petición asignada a cada TDD de petición y a su TDD de respuesta correspondiente.

La LA calcula el identificador REQ-ID por cualquier medio apropiado que garantice que los REQ-ID son únicos para esa LA. El algoritmo que habrá de utilizarse para el cálculo del REQ-ID está fuera del ámbito de esta Recomendación. El REQ-ID es un campo del registro de CA (véase 5.6.2).

Los procedimientos de recuperación que podrían derivarse de la utilización del REQ-ID están fuera del ámbito de esta Recomendación.

#### **5.4.4 Referencia a peticiones de la LA (REQ-REF)**

La referencia a una petición de la LA (REQ-ID) se utiliza en la TDD de rastreo para referenciar anteriores TDD de envío y/o de recepción.

#### **5.4.5 Identificación de formatos de transferencia (ID de conversión)**

El formato de transferencia – es decir, el formato utilizado para transferir un documento entre la LA y la CA – es identificado por el ID de conversión (Convert-ID). Para cada servicio, esta Recomendación define un conjunto obligatorio de diferentes ID de conversión que serán admitidos por la realización de la CA. Véase también el Cuadro H.1 y las subcláusulas específicas del servicio conexo en la Parte II de esta Recomendación.

#### **5.4.6 Identificación de formatos de transmisión (ID de tipo)**

El formato de transmisión – es decir, el formato utilizado para transmitir un documento a través del servicio – se identifica por el identificador de tipo (Type-ID). Cada servicio define su conjunto de formatos de transmisión. Véase también la sección relativa al servicio conexo en la Parte II de esta Recomendación.

## 5.5 Facilidad de despacho de ficheros recibidos (DRF)

Algunos servicios de telecomunicaciones no ofrecen facilidades de subdireccionamiento. En los sistemas en los que la CA puede ser utilizada por diversas LA simultáneamente, es necesario ayudar al encaminamiento de los documentos entrantes a los recibientes previstos. Este proceso se denomina despacho de ficheros recibidos (DRF).

Cuando una CA recibe un fichero de la red, lo asignará a una sola LA recibiente, de modo que únicamente la LA recibiente puede acceder al fichero recibido. La selección de la LA recibiente es un proceso privado de la CA y está fuera del ámbito de esta Recomendación.

Seguidamente, se asigna el LA-ID de la LA recibiente al registro de CA correspondiente al fichero recibido.

Si la CA sustenta la facilidad de despacho de ficheros recibidos, la LA recibiente puede despachar ficheros recibidos a la LA correspondiente por medio de la petición de rastreo:DESPACHO (Trace:DISPATCH) (véase 6.6.3). A continuación, el LA-ID de la LA a la que se despachó el fichero recibido se asigna al registro de CA.

Como esta característica de la CA es facultativa, el soporte de la facilidad DRF debe declararse en el descriptor de CA (véase la cláusula 9).

Si la CA no sustenta la facilidad DRF, la CA asignará (por cualquier medio apropiado) el registro de CA a cualquier LA utilizando el campo LA-ID. Cuando el servicio de telecomunicaciones proporciona subdireccionamiento, la CA asignará automáticamente el registro de CA a la LA recibiente prevista.

Los algoritmos que se utilizarán para despachar las llamadas entrantes a las LA apropiadas están fuera del ámbito de esta Recomendación.

## 5.6 Control de comunicaciones – Registro de CA

Como el método de intercambio entre la LA y la CA se basa en un modelo cliente-servidor, la LA siempre origina TDD de petición dirigidas a la CA. Las TDD de respuesta de la CA no son espontáneas; la LA debe interrogar a la CA para saber si hay TDD de respuesta disponibles.

En la presente Recomendación se proporcionan los medios para que las LA rastreen los eventos de telecomunicaciones que se producen en una CA; sin embargo, las LA no tienen que utilizar esos medios.

A cada evento de comunicaciones de CA corresponde una recopilación de informaciones, como fecha y hora, originador, recibiente(s), servicio de comunicaciones, etc.

El registro de CA es la colección funcional de informaciones mantenida por una CA para procesar una petición de envío o una llamada entrante de la red. Estas informaciones se mantienen en campos distintos, cada uno de los cuales tiene una finalidad especial.

La CA genera un registro de CA cuando se produce uno de los siguientes eventos:

- La CA recibe una petición de envío de una LA. En este caso, si en la petición de envío figuran muchos recibientes, la CA ampliará la lista de recibientes y generará una cantidad de registros de CA igual al número de recibientes en la lista.
- La CA recibe una petición rastreo:REPROGRAMACIÓN de una LA.
- La CA recibe una llamada entrante de la red.

Facultativamente, la CA puede generar un registro de CA en el estado fracasado cuando encuentra condiciones de error que no son consecuencia directa de una petición de envío ni de una llamada entrante.

Los registros de CA son destruidos cuando la LA emite peticiones de rastreo: PURGA o por medios específicos de CA.

NOTA – Un registro de CA sólo puede ser destruido cuando ha llegado a su estado final, es decir el estado «recuperado», «enviado» o «fracasado» (véanse también las Figuras 5 y 6).

El registro de CA es una estructura interna de la CA. El modo de realización del registro de CA está fuera del ámbito de esta Recomendación.

### 5.6.1 Control de acceso del registro de CA

Todas las acciones originadas por las LA sobre registros de CA se realizan por medio de la petición de rastreo. Si su configuración lo permite, una CA puede optar por ocultar alguna información a la LA. Por ejemplo, puede rechazar el acceso a registros de CA «diferidos», originados por otra LA.

Para ayudar a controlar los accesos a una CA, esta Recomendación proporciona un mecanismo para identificar las LA por medio del identificador LA-ID. Por ejemplo, todas las peticiones de rastreo originadas por las LA contienen el identificador LA-ID.

Gracias a este mecanismo, una CA determinada puede optar por limitar o ampliar la utilización de algunas peticiones de control a un solo conjunto de LA. Por ejemplo, la utilización de la petición de rastreo: DESPACHO puede ofrecerse solamente a una LA o a un conjunto de LA diferentes, según sea la configuración del sistema; el acceso a la petición rastreo: PURGA podría también reservarse para una sola LA por razones administrativas.

El fabricante de CA declarará en su documentación cómo habrán de tratarse esos controles de acceso (si los hubiere) y, si fuera procedente, cómo podrían personalizarse los controles de acceso para tener en cuenta las configuraciones de los usuarios. La personalización de las CA deberá realizarse por medios específicos que están fuera del ámbito de esta Recomendación.

### 5.6.2 Campos del registro de CA

El registro de CA contiene una lista mínima de los campos que deberán ser sustentados por todas las CA. Las CA podrán soportar asimismo otros campos, que deberán declararse en el ICE.

El Cuadro 3 muestra la lista mínima de los campos de registro de CA que deberán ser admitidos por cualquier CA en el orden indicado:

CUADRO 3/T.611

#### Lista mínima de los campos del registro de CA

Nombre del campo	Sintaxis	Finalidad
COM-ID	COM-ID	Mantiene el identificador de comunicación único asignado al registro de CA
LA-ID	La-id	Asigna el registro de CA a la LA que lo originó, o a la que está destinado
REQ-ID	Req-id	Mantiene la referencia de la petición
STATE (estado)	State	Indica el estado actual del registro de CA
DIRECTION (sentido)	«Xmit»/«Receive»	Indica si el registro de CA fue generado para transmisión o recepción
NOTA – Los fabricantes de las CA pueden incluir el soporte de campos adicionales, (por ejemplo información de tarificación, sellos de tiempo, direcciones). Véase también 6.6.3.2.		

El registro de CA proporciona a una LA la capacidad de controlar la CA con la que tiene abierta una sesión. Un registro de CA se encuentra, en cualquier momento dado, en uno de los seis estados siguientes (véase el Cuadro 4).

CUADRO 4/T.611

**Estados de registro de CA**

Nombre del estado	Referente a	Significa
«delayed» (diferido)	Transmisión	El (los) documento(s) asociado(s), no ha sido aún procesado por la CA. Está en espera de que la CA lo pase al estado «en envío».
«sending» (en envío)	Transmisión	La CA está procesando el (los) documento(s) asociado(s) para la transmisión, pero aún no ha concluido el proceso (ya sea porque el envío aún se está efectuando o porque la transmisión ha fracasado y la CA efectuará otras tentativas próximamente).
«sent» (enviado)	Transmisión	El (los) documento(s) asociado(s) ha sido transmitido con éxito al recipiente.
«failed» (fracasado)	Transmisión, recepción	El (los) documento(s) asociado(s) no ha podido transmitirse, se han producido errores durante la recepción o se ha producido un error interno en la CA.
«reception» (recepción)	Recepción	El registro describe un documento que la CA está recibiendo en ese momento, o ha sido recibido pero no recuperado por la LA recipiente.
«retrieved» (recuperado)	Recepción	El documento recibido ha sido recuperado por la LA recipiente mediante la petición de recepción.

Las transiciones de un estado a otro están regidas por acciones internas de la CA, procedentes de la red de comunicaciones u originadas en la LA.

Las LA pueden leer los campos de estado del registro de CA por medio de la petición de rastreo:COPIA. Una LA determinada sólo puede tener acceso a los registros cuyo valor de identificador LA-ID coincidan con el suyo (a menos que la CA amplíe los derechos de acceso de ciertas LA). Se asegura así que cada LA sólo pueda consultar los registros de CA que le corresponden.

La única excepción a esta regla es el caso en que la CA no admita la facilidad DRF (véase 5.5), en cuyo caso cualquier LA podrá tener acceso a cualquier registro de CA que se encuentre en el estado recepción.

Seguidamente se describen las transiciones de estado de un registro de CA en relación con los eventos de transmisión y recepción producidos en la CA.

**5.6.3 Proceso de transmisión – Transiciones de estado**

Las transmisiones se inician mediante peticiones de envío. Cuando la CA recibe una petición envío válida de una LA que ha abierto una sesión con ella, crea tantos registros de CA como recipientes figuran en la petición de envío.

Las transmisiones también pueden tener su origen en transmisiones que han fallado anteriormente y han sido reprogramadas por medio de la petición de rastreo:REPROGRAMACIÓN. En algunos casos esto puede evitar la demora de la conversión de los documentos a formatos de transmisión apropiados, ya que esos documentos han sido convertidos en una petición de envío anterior.

La CA asignará un ID-COM a cada nuevo registro de CA. Si la petición envío especifica múltiples recipientes, la CA asigna un COM-ID diferente a cada registro de CA resultante (uno por recipiente). Además, la CA pondrá el campo de estado en el estado «diferido». Seguidamente la CA copiará toda la información de la petición de envío en el registro de CA.

La CA explora a su propio ritmo la lista de los registros de CA que se encuentran en el estado «diferido» y procesa uno de ellos. La elección del registro de CA «diferido» al que procesa en primer término se basa en las fechas y horas programadas especificadas en el campo <tiempo de envío> de la petición de envío (véase el Cuadro 13, subcláusula 6.2). El registro de CA elegido se pasa al estado «en envío».

La CA transmite sucesivamente todos los documentos contenidos en el registro de CA que está enviando. Si la transmisión fracasa por errores de transmisión, la CA puede mantener el registro de CA en el estado «en envío» mientras repite sus tentativas de transmisión.

Si la transmisión finalmente fracasa, el registro de CA se pasa al estado «fracasado». Si la transmisión llega a tener éxito, el registro de CA se pasa al estado «enviado». Entre dos tentativas para el mismo registro de CA, la CA puede elegir otro registro de CA en el estado «diferido» y procesarlo del modo antes descrito. En un momento dado pueden hallarse, por consiguiente, más de un registro de CA en el estado «en envío».

La Figura 5 ilustra estas transiciones de estado.

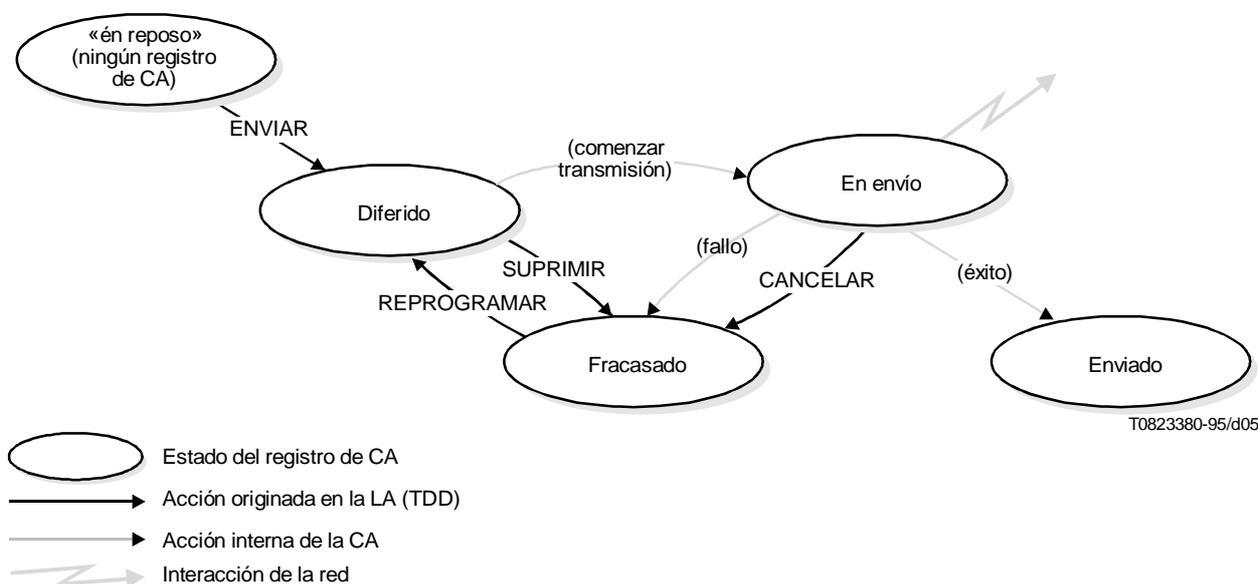


FIGURA 5/T.611  
Transiciones de estado del registro de CA (transmisiones)

#### 5.6.4 Proceso de recepción – Transiciones de estado

Cuando una CA recibe de la red una comunicación entrante, crea un registro de CA. Inmediatamente se asigna a este registro de CA un identificador COM-ID exclusivo y se pone el registro en el estado «recepción». Toda la información relativa a la llamada entrante (originador, fecha y hora, etc.) se copia seguidamente en el registro de CA, en los campos apropiados.

Si se produce un fallo en el curso de la recepción, el registro de CA se pasa inmediatamente al estado «fracasado». En los demás casos, el registro de CA se pasa al estado «recuperado» cuando una LA lo ha recuperado por medio de la petición de recepción.

La petición de rastreo:DESPACHO no tiene efecto sobre el estado del registro de CA, es decir, éste permanece en el estado «recepción»; el registro de CA deja de ser visible para la LA que lo despacha y pasa a ser visible para la LA recibiente.

La Figura 6 ilustra estas transiciones de estado.

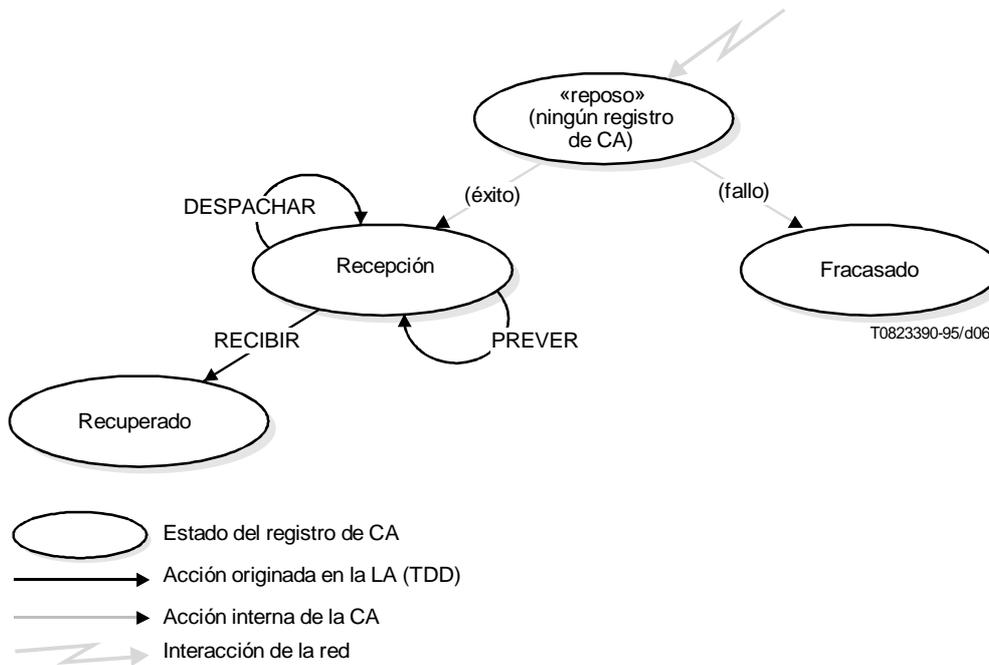


FIGURA 6/T.611  
Transiciones de estado del registro de CA (recepción)

### 5.6.5 Acciones – Convenios de notación

Con miras a la legibilidad, se aplican los siguientes convenios a la notación de las distintas acciones relacionadas con los registros de CA (véase el Cuadro 5).

CUADRO 5/T.611  
Notación de las acciones relativas al registro de CA

Notación	Significa
TRACE:DELETE rq (petición de RASTREO:SUPRIMIR)	Petición «SUPRIMIR» de la TDD RASTREO
Recipient LA (LA recipiente)	El recipiente deseado de la llamada entrante. El recipiente puede determinarse automáticamente (por ejemplo, por un mecanismo de subdireccionamiento), o por despacho manual. El procesamiento automático es un asunto que cae en la esfera privada de la CA, mientras que el despacho manual puede controlarse a través de la interfaz (véase 10.6.1)
Originating LA (LA de origen)	La LA en la que se ha originado el registro de CA (mediante una petición envío)
Transmit (transmisión)	La CA intenta llevar a cabo la acción transmisión del registro de CA.
NOTA – La CA puede permitir que las LA que ella elija se comporten como LA emisora o receptora. Esto puede ser útil en muchas aplicaciones (por ejemplo, para administración).	

## 5.6.6 Acciones – Transmisiones

Esta subcláusula describe todas las acciones que repercuten en los distintos estados de los registros de CA relacionados con las transmisiones.

### 5.6.6.1 Estado diferido

Un registro de CA se encuentra en el estado «diferido» mientras no haya sido procesado por la CA (es decir, no se ha intentado transmitirlo) y la LA de origen no lo haya suprimido. A un registro de CA en el estado «diferido» se le pueden aplicar las siguientes acciones (véase el Cuadro 6).

CUADRO 6/T.611

#### Acciones de transmisión para el estado «diferido»

Acción	Originador	Finalidad	Estado resultante
Trace:DELETE rq (petición de rastreo: SUPRIMIR)	LA de origen	Suprimir una petición de envío. Esta acción tiene efecto en un solo registro de CA.	«fracasado»
Trace:COPY rq (petición de rastreo: COPIAR)	LA de origen	Crear un fichero lógico que contenga una copia de la lista de los registros de CA «diferidos». Esta acción se aplica a todos los registros de CA «diferidos» originados en una misma LA.	«diferido»
Transmit (transmitir)	CA	La CA decide cursar el registro de CA, porque ya corresponde procesarlo.	«en envío»

### 5.6.6.2 Estado en envío

Un registro de CA se encuentra en el estado «en envío» cuando la CA intenta transmitir la información contenida en el mismo.

Si la tentativa fracasa, el registro de CA puede permanecer en el estado «en envío» en espera de la tentativa siguiente; la CA hará el siguiente intento respetando el «intervalo entre tentativas», a menos que se haya alcanzado el «número máximo de tentativas», en cuyo caso el registro de CA se coloca en el estado «fracasado».

A un registro de CA en el estado «en envío» se le pueden aplicar las siguientes acciones (véase el Cuadro 7).

CUADRO 7/T.611

#### Acciones de transmisión para el estado «en envío»

Acción	Originador	Finalidad	Estado resultante
Trace:CANCEL rq (petición de rastreo: ANULAR)	LA de origen	Anular una petición envío. Esta acción afecta a un solo registro CA en cada momento.	«fracasado»
Trace:COPY rq (petición de rastreo: COPIAR)	LA de origen	Crear un fichero lógico que contenga una lista de los registros de CA «en envío».	«en envío»

### 5.6.6.3 Estado enviado

Un registro de CA se encuentra en el estado «enviado» cuando la CA ha logrado transmitir la información contenida en el mismo.

A un registro de CA en el estado «enviado» se le pueden aplicar las siguientes acciones (véase el Cuadro 8).

CUADRO 8/T.611

**Acciones de transmisión para el estado «enviado»**

Acción	Originador	Finalidad	Estado resultante
Trace:PURGE rq (petición de rastreo: PURGA)	LA de origen	Suprimir todos los registros de CA que se encuentran en el estado «enviado»	No aplicable por haberse suprimido (purgado)
Trace:COPY rq (petición de rastreo: COPIA)	LA de origen	Crear un fichero lógico que contenga una copia de lista de los registros de CA «enviados».	«enviado»

**5.6.6.4 Estado fracasado**

Un registro de CA se encuentra en el estado «fracasado» cuando la CA no ha logrado transmitir la información contenida en el mismo, por cualquier razón.

NOTA – El estado «fracasado» también se aplica al proceso de recepción. Para la información correspondiente, véase más adelante.

A un registro de CA en el estado «fracasado» se le pueden aplicar las siguientes acciones (véase el Cuadro 9).

CUADRO 9/T.611

**Acciones de transmisión para el estado «fracasado»**

Acción	Originador	Finalidad	Estado resultante
Trace:PURGE rq (petición de rastreo: PURGA)	LA de origen	Suprimir los registros de CA que se encuentran en el estado «fracasado»	No aplicable por haberse suprimido (purgado)
Trace:RESCHEDULE rq (petición de rastreo: REPROGRAMACIÓN)	LA de origen	Pedir un nuevo procesamiento de una petición fracasada. En él pueden aprovecharse las conversiones anteriores del documento	«diferido»
Trace:COPY rq (petición de rastreo: COPIA)	LA de origen	Crear un fichero lógico que contenga una copia de la lista de los registros de CA «fracasados»	«fracasado»

**5.6.7 Acciones – Recepciones**

Esta subcláusula describe todas las acciones que afectan a los estados de los registros de CA relacionados con la recepción.

**5.6.7.1 Estado recepción**

Un registro de CA se encuentra en el estado «recepción» cuando la CA ha recibido con éxito una llamada entrante de la red, y el registro de CA aún no ha sido despachado.

A un registro de CA en el estado «recepción» se le pueden aplicar las siguientes acciones (véase el Cuadro 10).

CUADRO 10/T.611

**Acciones de recepción para el estado «recepción»**

Acción	Originador	Finalidad	Estado resultante
RECEIVE rq (petición de RECEPCIÓN)	LA recipiente	Recuperar el documento o los documentos de interés para el registro de CA (así como la información correspondiente en la respuesta TDD).	«recuperado»
Trace:PREVIEW rq (petición de rastreo:VISIÓN PREVIA)	Cualquier LA	Recibir una copia del documento o los documentos de interés para el registro de CA. El registro de CA continúa en el estado «recepción»	«recepción»
Trace:DISPATCH rq (petición de rastreo:DESPACHO)	LA recipiente o administrador	Asigna uno o más LA-ID a un registro de CA recibido (véase también la Nota)	«recepción»
Trace:COPY rq (petición de rastreo:copia)	LA recipiente	Construir un fichero lógico que contenga una copia de la lista de los registros de CA en el estado «recepción», y que no hayan sido ya recuperados. Una LA recipiente «ve» solamente los registros de CA que le interesan, es decir, los que concuerdan con el LA-ID.	«recepción»
NOTA – La gestión del administrador es un asunto que cae en la esfera privada de la CA.			

**5.6.7.2 Estado recuperado**

Un registro de CA está en el estado «recuperado» cuando una LA con un LA-ID concordante ha recuperado el registro de CA, por medio de una petición de recepción.

Las siguientes acciones pueden aplicarse a un registro de CA en el estado «recuperado» (véase el Cuadro 11).

CUADRO 11/T.611

**Acciones de recepción para el estado «recuperado»**

Acción	Originador	Finalidad	Estado resultante
Trace:PURGE rq (petición de rastreo:PURGA)	LA recipiente	Suprimir registros de CA que se encuentran en el estado «recuperado»	No es aplicable por haberse suprimido (purgado)
Trace:COPY rq (petición de rastreo:copia)	LA recipiente	Construir un fichero lógico que contenga una copia de la lista de los registros de CA en el estado «recuperado». Una LA recipiente «ve» solamente los registros de CA que le interesan, es decir, los que concuerdan con el LA-ID.	«recuperado»

**5.6.7.3 Estado fracasado**

Un registro de CA está en el estado «fracasado» cuando, por alguna razón, la CA fracasó en la recepción de una llamada entrante de la red.

NOTA – El estado «fracasado» se aplica también al proceso de transmisión. La información pertinente se indicó en 5.6.6.4.

Cuando una CA fracasa en la recepción de una llamada entrante, hace pasar el registro de CA del estado «recepción» al estado «fracasado».

Cuando un registro de CA está en el estado «fracasado», no se le asigna ningún LA-ID. Según sea la configuración, la CA puede (pero no está obligada a) asignar un LA-ID a esos registros «fracasados», por cualquier medio apropiado. La petición de rastreo:REPROGRAMACIÓN (Trace:Reschedule) se aplica a los registros de CA «fracasados».

El Cuadro 12 describe las acciones que pueden aplicarse a los registros de CA en el estado «fracasado»:

CUADRO 12/T.611

**Acciones de recepción para el estado «fracasado»**

Acción	Originador	Finalidad	Estado resultante
Trace:PURGE rq (petición de rastreo:PURGA)	LA recipiente	Suprimir todos los registros de CA que se encuentran en el estado «fracasado»	No es aplicable por haberse suprimido (purgado)
Trace:COPY rq (petición de rastreo:Copia)	LA recipiente	Construir un fichero lógico que contenga una copia de la lista de los registros de CA en el estado «fracasado». Una LA recipiente «ve» solamente los registros de CA que se encuentran en ese estado	«fracasado»

## 6 Descripciones de datos de tareas (TDD)

La funcionalidad ofrecida por la CA a través de la interfaz se invoca por medio de descripciones de datos de tareas (TDD, *task data descriptions*), que se transmiten en el sentido LA a CA como peticiones, y (según la función invocada), en el sentido opuesto como respuestas.

En esta cláusula se describe la semántica, funcionalidad, sintaxis y codificación de las diversas TDD de petición y respuestas.

### 6.1 Presentación genérica de las TDD

Las TDD pueden codificarse según diferentes esquemas de codificación. Algunos de éstos, como por ejemplo los códigos ASCII o EBCDIC, se basan en descripciones de texto. Otros esquemas de codificación no pueden representarse por texto. Un ejemplo de estos otros esquemas de codificación es el de codificación binaria, que se presenta escrito en lenguaje C en la Parte III de esta Recomendación.

Las diversas TDD se describen a continuación, genéricamente, con una sintaxis de estilo BNF. Los detalles de esta sintaxis se explican en el Anexo A.1.

Para facilitar la comprensión, en las subcláusulas que siguen se indicará, junto con la sintaxis genérica en el estilo BNF, la codificación basada en texto. No obstante, el esquema de codificación binaria (definido en la Parte III de esta Recomendación) puede asimismo usarse en las implementaciones.

```

<TDD> :=
    <TDD Header>
    <Send> | <Receive> | <Trace> | <Submit> | <Extend>

<TDD Header> :=
    -- depende de la codificación utilizada. Para la codificación basada en texto,
    -- véase 6.4.3; para el esquema de codificación binaria, véase la cláusula
    -- correspondiente de la Parte III de esta Recomendación

<Send> :=
    <SendTDD> | <SendAckTDD>

<Receive> :=
    <ReceiveTDD>

<Trace> :=
    <DeleteTDD> | <CopyTDD> | <CancelTDD> | <PurgeTDD> |
    <RescheduleTDD> | <DispatchTDD> | <PreviewTDD>

```

<Submit> := <PrintTDD> | <ConvertTDD> | <CheckTDD>

<Extend> := <ExtendTDD> | <NationalTDD> | <PrivateTDD>

<SendTDD> := <SendFunction> <LaId> <ReqId> <Service>  
 [<SendTime>] [<Comment>] [<LastTime>] [<UserKey>]  
 <ServiceDependentKeywordsSend>

<ServiceDependentKeywordsSend> :=  
 -- *definido en la cláusula correspondiente de la Parte II para cada servicio*

<SendAckTDD> := <SendAckFunction> <LaId> <ReqId> <Service> <Error>  
 <Status> [<ComId>] [<SendTime>] [<Comment>] [<LastTime>]  
 [<UserKey>] [<Minor>] [<Warning>]  
 <ServiceDependentKeywordsSendack>

<ServiceDependentKeywordsSendack> :=  
 -- *definido en la cláusula correspondiente de la Parte II para cada servicio*

<ReceiveTDD> := <ReceiveFunction> <LaId> <ReqId> <Error> <Status> [<ComId>]  
 [<ReceiveTime>] [<Service>] [<Delete>] [<Minor>] [<Warning>]  
 <ServiceDependentKeywordsReceive>

<ServiceDependentKeywordsReceive> :=  
 -- *definido en la cláusula correspondiente de la Parte II para cada servicio*

<DeleteTDD> := <DeleteFunction> <LaId> <ReqId> <Error> (<ComId> | <ReqRef>)  
 [<Minor>] [<Warning>]

<CopyTDD> := <CopyFunction> <LaId> <ReqId> <Error> <State> <Target> <Layout>  
 (<ComId> | <ReqRef>) [<Minor>] [<Warning>] [<Previewed>]  
 [<Dispatched>] [<Direction>] [<Time\_Range>] [<Service>]  
 <ServiceDependentKeywordsCopy>

<ServiceDependentKeywordsCopy> :=  
 -- *definido en la cláusula correspondiente de la Parte II para cada servicio*

<CancelTDD> := <CancelFunction> <LaId> <ReqId> <Error> (<ComId> | <ReqRef>)  
 [<Minor>] [<Warning>]

<PurgeTDD> := <PurgeFunction> <LaId> <ReqId> <Error> <State> (<ComId> | <ReqRef>)  
 [<Minor>] [<Warning>]

<RescheduleTDD> := <RescheduleFunction> <LaId> <ReqId> <Error> (<ComId> | <ReqRef>)  
 [<Address>] [<SendTime>] [<LastTime>] [<Minor>] [<Warning>]

<DispatchTDD> := <DispatchFunction> <LaId> <NewLa> {<NewLa>} <ReqId> <Error>  
 <ComId> [<Minor>] [<Warning>]

<PreviewTDD> := <PreviewFunction> <LaId> <ReqId> <Target> <Convert> <Error> <ComId>  
 [<Minor>] [<Warning>]

<PrintTDD> := <PrintFunction> <LaId> <ReqId> <Error> <FileName> <InFormat>  
 [<Printer>] [<Minor>] [<Warning>]

<ConvertTDD> := <ConvertFunction> <LaId> <ReqId> <Error> <FileName> <Target>  
 <InFormat> <OutFormat> [<Minor>] [<Warning>]

<CheckTDD> := <CheckFunction> <LaId> <ReqId> <Error> <FileName> <Check> [<Minor>]  
 [<Warning>]

<ExtendTDD> := <ExtendFunction> <SubFunction> <LaId> <ReqId> <Error> [<Minor>]  
 [Warning] [<ExtendSubFunctionKeywords>]

<ExtendSubFunctionKeywords> :=  
 -- *definido en la cláusula correspondiente de la Parte II para cada servicio*

<NationalTDD> := <NationalFunction> <SubFunction> <LaId> <ReqId> <Error> [<Minor>]  
 [Warning] [<NationalSubFunctionKeywords>]

<NationalSubFunctionKeywords> :=  
 -- *queda en estudio*

<PrivateTDD> := <PrivateFunction> <SubFunction> <LaId> <ReqId> <Error> [<Minor>]  
 [Warning] [<PrivateSubFunctionKeywords>]

<PrivateSubFunctionKeywords> :=  
 -- *queda en estudio*

## 6.2 Descripción de los elementos de TDD

En general, hay dos tipos de elementos de sintaxis en una TDD:

- 1) los elementos de sintaxis cuyos parámetros dependen solamente de la funcionalidad de la TDD;
- 2) los elementos de sintaxis cuyos parámetros dependen del servicio a que se aplica la TDD.

Cuando la TDD se relaciona con el envío o la recepción de ficheros o documentos, algunos elementos de sintaxis pueden tener también parámetros específicos:

- del destinatario (por ejemplo el atributo recipiente primario o de copia para servicios de correo electrónico);
- del propio documento (por ejemplo los formatos de transferencia y de transmisión).

En el Cuadro 13 se explican los elementos de sintaxis, de las TDD, que *no* dependen del servicio. Los elementos de sintaxis de las TDD que son específicos del servicio, del destinatario o del documento se describen en las cláusulas pertinentes de la Parte II de esta Recomendación.

## 6.3 Identificador de código (Code-ID)

Cualquiera que sea la codificación de una TDD, su primer octeto indica el esquema de codificación. Este octeto se denomina el identificador de código (Code-ID). El Cuadro 14 indica los posibles identificadores de códigos asignados por esta Recomendación.

## 6.4 Codificación basada en texto

Esta subcláusula se refiere a las codificaciones en las que el identificador de código está fijado a "A", "B", "I" o "E".

Para la descripción de la codificación basada en texto se utiliza en las subcláusulas siguientes la sintaxis de estilo BNF explicada en el Anexo A.1.

### 6.4.1 Sintaxis y formato

Los elementos de sintaxis indicados en 6.1 se codifican utilizando pares palabra clave/parámetro. La correspondencia entre los pares palabra clave/parámetro con los elementos de sintaxis se muestra en el Cuadro 15. Dado que algunos elementos de sintaxis y la funcionalidad correspondiente dependen del servicio de telecomunicaciones subyacente, sólo se presentan los que no dependen del servicio de telecomunicaciones. Los elementos de sintaxis que dependen del servicio de telecomunicaciones se describen en la cláusula pertinente de la Parte II de esta Recomendación.

**Resumen de los elementos de sintaxis de las TDD que no dependen del servicio**

Elemento de sintaxis	Finalidad
<CancelFunction>	Identifica la función solicitada por la TDD
<Check>	Indica el formato con relación al cual deberá comprobarse un fichero determinado. Sólo se utiliza en la <CheckTDD>
<CheckFunction>	Identifica la función solicitada por la TDD
<ComId>	"ComId" significa identificador de comunicación. Véase 5.4.1
<Comment>	Se utiliza para asociar a un documento un comentario separado, que deberá transmitirse. Este comentario se almacena en el registro de CA, en lugar de transmitirse con el documento a través del servicio de telecomunicaciones
<ConvertFunction>	Identifica la función solicitada por la TDD
<CopyFunction>	Identifica la función solicitada por la TDD
<Delete>	Informa a la CA si un documento recibido deberá o no suprimirse en el almacenamiento interno de la CA, después de que la LA llamante lo haya recuperado correctamente
<DeleteFunction>	Identifica la función solicitada por la TDD
<Direction>	Especifica si el registro de CA pertenece a un documento entrante o saliente
<Dispatched>	Especifica si <DispatchTDD> se ha aplicado al registro de CA en cuestión
<DispatchFunction>	Identifica la función solicitada por la TDD
<Error>	Este elemento de sintaxis retorna el código de error generado por la CA en la respuesta TDD. La LA lo comprobará para determinar si la operación ha tenido éxito
<ExtendFunction>	Identifica la función solicitada por la TDD
<ExtendSubFunction-Keywords>	Véase la cláusula correspondiente de la Parte II de esta Recomendación
<FileName>	Contiene el trayecto del fichero en que se almacena el documento
<InFormat>	Especifica el formato de entrada de un documento. Sólo se utiliza en <Convert-TDD>
<LaId>	LaId significa LA-ID. Véase 5.4.2
<LastTime>	Indica el límite de tiempo hasta el cual una CA deberá tratar de enviar el documento especificado en la TDD correspondiente
<Layout>	Define la organización del fichero de destino de la <CopyTDD>
<Minor>	Este elemento de sintaxis retorna un código de error adicional en la respuesta TDD
<NationalFunction>	Identifica la función solicitada por la TDD
<NationalSubFunction-Keywords>	Queda en estudio
<NewLa>	Contiene el nuevo LA-ID para uso con la <DispatchTDD>
<OutFormat>	Especifica el formato de salida de un documento. Sólo se utiliza en la <Convert-TDD>
<Previewed>	Especifica si la <PreviewTDD> se ha aplicado al registro de CA en cuestión

**Resumen de los elementos de sintaxis de las TDD que no dependen del servicio**

Elemento de sintaxis	Finalidad
<Printer>	ID de la impresora seleccionada para uso con la <PrintTDD>. Depende del sistema operativo empleado. El fabricante de la CA indicará en su documentación cómo controlar las impresoras
<PrintFunction>	Identifica la función solicitada por la TDD
<PrivateFunction>	Identifica la función solicitada por la TDD
<PrivateSubFunction-Keywords>	Queda en estudio
<PurgeFunction>	Identifica la función solicitada por la TDD
<ReceiveFunction>	Identifica la función solicitada por la TDD
<ReqId>	ReqId significa REQ-ID. Véase 5.4.3
<ReceiveTime>	Indica la hora a la que la CA recibió el documento especificado en la correspondiente TDD
<ReqRef>	ReqRef significa REQ-REF. Véase 5.4.4
<RescheduleFunction>	Identifica la función solicitada por la TDD
<SendackFunction>	Identifica la función solicitada por la TDD
<SendFunction>	Identifica la función solicitada por la TDD
<SendTime>	Indica la hora a la que la CA enviará el documento especificado en la correspondiente TDD
<Service>	Especifica el servicio de telecomunicaciones utilizado. Véase también la Parte II de esta Recomendación
<ServiceDependent-KeywordsReceive>	Véase la cláusula apropiada de la Parte II de esta Recomendación
<ServiceDependent-KeywordsSend>	Véase la cláusula apropiada de la Parte II de esta Recomendación
<ServiceDependent-KeywordsSendack>	Véase la cláusula apropiada de la Parte II de esta Recomendación
<State>	Especifica el estado de un registro de CA. Véase también el Cuadro 4 en 5.6.2
<Status>	Retorna la situación ("status") de una transmisión o recepción en la respuesta TDD
<SubFunction>	Identifica la función solicitada por la TDD
<Target>	Se utiliza en la <ConvertTDD> y en la <CopyTDD> para especificar el trayecto del fichero de salida generado por la CA
<TimeRange>	Especifica un intervalo de tiempo para seleccionar determinados registros de CA
<UserKey>	Especifica un intervalo de tiempo para seleccionar determinados registros de CA
<Warning>	Retorna un código adicional de advertencia en la respuesta TDD

CUADRO 14/T.611

## Codificación de las TDD – Identificadores de código

Identificador de código		Representación de TDD
Valor	Presentación en texto	
41 <sub>hex</sub>	"A"	Texto legible, organizado como líneas de pares palabra clave/parámetro. Se utilizará el juego de caracteres ASCII ampliado de APPLI/COM (véase el Cuadro 50)
42 <sub>hex</sub>	"B"	Texto legible, organizado como líneas de pares palabra clave/parámetro. Se utilizará una variante nacional del juego de caracteres definido en la Recomendación UIT-T T.50
43 <sub>hex</sub>	"C"	Esquema de codificación binaria (escrito en el lenguaje de programación C en la Parte III de esta Recomendación)
C5 <sub>hex</sub> <sup>a)</sup>	"E"	Texto legible, organizado como líneas de pares palabra clave/parámetro. Se utilizará el juego de caracteres codificados en EBCDIC
49 <sub>hex</sub>	"I"	Texto legible, organizado como líneas de pares palabra clave/parámetro. Se utilizará el juego de caracteres ASCII estándar de APPLI/COM (véase el Cuadro 53)
50 <sub>hex</sub>	"P"	Se considerará que la presentación y la sintaxis de las TDD están definidas por reglas privadas
Otros valores		Todos los demás valores que no figuren en este cuadro están reservados para normalización futura
<p><sup>a)</sup> Si este identificador de código tiene que aparecer en el ICE, su valor binario se ajustará a la presentación de código elegida para el propio ICE; así, si el identificador de código para el encabezamiento ICE es "I" (ASCII estándar de APPLI/COM), el valor binario de la "E" se codificará como 45<sub>hex</sub>.</p>		

CUADRO 15/T.611

## Correspondencia de los pares palabra clave/parámetro para codificación basada en texto

(Un símbolo ↵ denota el determinante de formato de nueva línea)

Elemento de sintaxis	Par palabra clave/parámetro
<CancelFunction>	"FUNCTION" ":" "Cancel" ↵
<Check>	"CHECK" ":" <Convert-id-parameter> ↵
<CheckFunction>	"FUNCTION" ":" "Check" ↵
<ComId>	"COMID" ":" <Com-id-parameter> ↵
<Comment>	"COMMENT" ":" <Comment-parameter> ↵
<ConvertFunction>	"FUNCTION" ":" "Convert" ↵
<CopyFunction>	"FUNCTION" ":" "Copy" ↵
<Delete>	"DELETE" ":" <Boolean-parameter> ↵
<DeleteFunction>	"FUNCTION" ":" "Delete" ↵
<Direction>	"DIRECTION" ":" <Direction-parameter> ↵
<Dispatched>	"DISPATCHED" ":" <Boolean-parameter> ↵
<DispatchFunction>	"FUNCTION" ":" "Dispatch" ↵
<Error>	"ERROR" ":" <Error-parameter> ↵

CUADRO 15/T.611 (fin)

**Correspondencia de los pares palabra clave/parámetro para codificación basada en texto**

(Un símbolo ↵ denota el determinante de formato de nueva línea)

Elemento de sintaxis	Par palabra clave/parámetro
<ExtendFunction>	"FUNCTION" ":" "Extend" ↵
<FileName>	"FILENAME" ":" <Path-parameter> ↵
<InFormat>	"INFORMAT" ":" <Convert-id-parameter> ↵
<LaId>	"LA-ID" ":" <La-id-parameter> ↵
<LastTime>	"LASTTIME" ":" <Date-time-parameter> ↵
<Layout>	"LAYOUT" ":" <Layout-id-parameter> ↵
<Minor>	"MINOR" ":" <Error-parameter> ↵
<NationalFunction>	"FUNCTION" ":" "National" ↵
<NewLa>	"NEWLA" ":" <La-id-parameter> ↵
<OutFormat>	"OUTFORMAT" ":" <Convert-id-parameter> ↵
<Previewed>	"PREVIEWED" ":" <Boolean-parameter> ↵
<Printer>	"PRINTER" ":" <Printer-id-parameter> ↵
<PrintFunction>	"FUNCTION" ":" "Print" ↵
<PrivateFunction>	"FUNCTION" ":" "Private" ↵
<PurgeFunction>	"FUNCTION" ":" "Purge" ↵
<ReceiveFunction>	"FUNCTION" ":" "Receive" ↵
<ReceiveTime>	"RCVTIME" ":" <Date-time-parameter> ↵
<ReqId>	"REQ-ID" ":" <Req-id-parameter> ↵
<ReqRef>	"REQREF" ":" <Req-id-parameter> ↵
<RescheduleFunction>	"FUNCTION" ":" "Reschedule" ↵
<SendAckFunction>	"FUNCTION" ":" "SendAck" ↵
<SendFunction>	"FUNCTION" ":" "Send" ↵
<SendTime>	"SENDTIME" ":" <Send-time-parameter> ↵
<Service>	"SERVICE" ":" <Service-parameter> ↵
<State>	"STATE" ":" <State-parameter> ↵
<Status>	"STATUS" ":" <Status-parameter> ↵
<SubFunction>	"SUBFUNC" ":" <Subfunc-parameter> ↵
<Target>	"TARGET" ":" <Path-parameter> ↵
<TimeRange>	"RANGE" ":" <Date-time-parameter> "," <Date-time-parameter> ↵
<UserKey>	"USERKEY" ":" <Userkey-parameter> ↵
<Warning>	"WARNING" ":" <Error-parameter> ↵

Para facilitar la lectura de las TDD todos los pares palabra clave/parámetro que aparezcan serán formateados mediante el empleo de determinantes de formato. Se aplican las siguientes reglas:

- los elementos de sintaxis se organizarán como líneas de pares palabra clave/parámetro;
- la palabra clave FUNCTION debe ser siempre la primera palabra clave presentada; si existe una palabra clave SUBFUNC para la TDD, deberá ser la segunda; el orden de todas las demás palabras clave no está definido;

- la línea deberá terminarse por un determinante de formato de "nueva línea" (véase el Cuadro 16);
- el parámetro deberá estar separado de la palabra clave por un carácter ":" ("dos puntos");
- cada línea puede contener un comentario. El comentario deberá introducirse por un carácter ";" y llega hasta el final de la línea. La CA no interpreta los comentarios. Si una declaración ha de contener un carácter punto y coma ";" ha de aplicarse un escape con el carácter barra de fracción invertida "\";
- una línea puede contener determinantes de formato de "espacio blanco" (véase el Cuadro 16);
- las palabras clave son insensibles a la escritura en mayúsculas o minúsculas (las mayúsculas y las minúsculas se tratan igualmente);
- la presentación de los parámetros es insensible a la escritura en mayúsculas o minúsculas, salvo en aquellos casos en que así lo requiera el servicio o el sistema operativo subyacente.

La TDD de petición contiene parámetros de entrada. La TDD de respuesta puede contener parámetros de salida. Para la construcción de la TDD de respuesta, la CA respetará las siguientes reglas<sup>3)</sup>:

- todos los parámetros de entrada de la TDD de petición se devolverán en la TDD de respuesta,
- como mínimo, la CA colocará en la TDD de respuesta todos los parámetros de salida declarados básicos,
- la CA podrá cambiar el orden de los parámetros contenidos en la TDD de petición, salvo que la palabra clave FUNCTION sea la primera palabra clave presentada, seguida por la palabra clave SUBFUNC, si procede,
- la CA podrá cambiar el tipo de letra, mayúscula o minúscula, de los parámetros contenidos en la TDD de petición,
- se pueden suprimir los comentarios contenidos en la TDD de petición.

CUADRO 16/T.611

**Determinantes de formato para la codificación basada en texto**

Determinante de formato	Codificación dependiente del identificador de código			
	"A"	"B"	"E"	"I"
Nueva línea	0D0A <sub>hex</sub> ó 0A <sub>hex</sub>	0D0A <sub>hex</sub> ó 0A <sub>hex</sub>	En estudio	0D0A <sub>hex</sub> ó 0A <sub>hex</sub>
Espacio blanco	20 <sub>hex</sub> ó 09 <sub>hex</sub>	20 <sub>hex</sub> ó 09 <sub>hex</sub>	En estudio	20 <sub>hex</sub> ó 09 <sub>hex</sub>

En la subcláusula 6.6 se indica detalladamente qué parámetros TDD se utilizan como parámetros de entrada y qué otros se utilizan como parámetros de salida.

**6.4.2 Correspondencia de las palabras clave**

**6.4.3 Codificación del encabezamiento de TDD**

En cada TDD (y dentro del ICE) el <TDD Header> (encabezamiento de TDD) es siempre el primer elemento especificado. El <TDD Header> está estructurado como se muestra a continuación (para una explicación de la gramática basada en BNF, véase A.1):

```

<TDD Header> := <Code-ID> <Identification> <Version> <Standard> <Reserved>
<Code-ID> := "A" | "B" | "E" | "I"
-- para la presentación del identificador de código véase también el Cuadro 14.

```

<sup>3)</sup> Estas reglas han cambiado considerablemente en comparación con la versión de 1992 de esta Recomendación. Las reglas definidas en dicha Recomendación son mucho más estrictas.

<Identification> :=        "\*APPLI/COM\*"

<Version> :=                "1994"  
                               -- <Version> designa el año de aprobación de esta Recomendación.

<Standard> :=                "\*UIT-T\*"

<Reserved> :=                STRING (SIZE(0..16))  
                               -- <Reserved> está reservado para ulteriores aplicaciones  
                               -- de esta Recomendación.

NOTA – El elemento de sintaxis <TDD Header> contiene el identificador de código como primer elemento.

#### 6.4.4 Codificación de los parámetros que no dependen del servicio

En esta subcláusula se define la codificación de los parámetros que no dependen del servicio, indicados en el Cuadro 15. Los parámetros que dependen del servicio se mencionan en 6.4.5. Estos parámetros se especifican íntegramente en las cláusulas pertinentes de la Parte II de esta Recomendación.

##### 6.4.4.1 Parámetro booleano

El parámetro Boolean-parameter se codifica con una STRING que toma los valores "YES" o "NO".

*Sintaxis general:*

<Boolean-parameter> :=        "YES"|"NO"

##### 6.4.4.2 Parámetro identificador de comunicación

El parámetro Com-id-parameter se codifica como una STRING que contiene el identificador de comunicación (COM-ID) calculado por la CA. Véase también 5.4.1.

*Sintaxis general:*

<Com-id-parameter> :=         STRING

##### 6.4.4.3 Parámetro de comentario

El parámetro Comment-parameter se codifica como una STRING que contiene el comentario especificado por la LA para el evento de comunicación en cuestión.

*Sintaxis general:*

<Comment-parameter> :=        STRING

##### 6.4.4.4 Parámetro de fecha hora

El parámetro Date-time-parameter se codifica como una STRING que contiene la fecha y la hora según el convenio "YY-MM-DD-HH:MM" (año-mes-día:hora:minutos).

*Sintaxis general:*

<Date-time-parameter> :=        <year> "-" <month> "-" <day> "-" <hours> ":" <minutes>

<year> :=                        <digit> <digit>

<month> :=                       "0" | "1" <digit>

<day> :=                         "0" | ... | "3" <digit>

<hours> :=                       "0" | ... | "2" <digit>

<minutes> :=                     "0" | ... | "5" <digit>

<digit> :=                       "0" | ... | "9"

#### 6.4.4.5 Parámetro de sentido de transmisión

El parámetro Direction-parameter se codifica como una STRING que puede tomar los valores "XMIT" o "RECEIVE".

*Sintaxis general:*

<Direction-parameter> := "XMIT" | "RECEIVE"

#### 6.4.4.6 Parámetro de error

El parámetro Error-parameter se codifica como una STRING estructurada, con la siguiente disposición:

"nnnn"/"Text..."

"nnnn" representa un número de error de 4 dígitos, justificado a la derecha, rellenado a la izquierda con caracteres "0". "0000" = éxito (ausencia de error). El valor del número de error depende del soporte lógico de comunicación utilizado. Los códigos de error son números decimales. Para la asignación de las gamas de códigos de error, véase el Cuadro 2 de 5.2.3.

"Text..." representa un texto escrito en lenguaje ordinario con una longitud de hasta 79 caracteres, que describe el error. Si el campo del parámetro no es lo suficientemente largo para aceptar el texto en escritura ordinaria, se abreviará el texto cuando sea necesario. El texto ordinario para el número de error "0000" es "Success" (éxito).

Los números de error que pueden asignarse se indican en 5.2.3 y en el Anexo C, respectivamente.

*Sintaxis general:*

<Error-parameter> := <digit> <digit> <digit> <digit> / STRING(SIZE(0..79))

<digit> := "0" | ... | "9"

#### 6.4.4.7 Parámetro identificador de aplicación local

El parámetro La-id-parameter contiene la referencia de un LA-ID. Se codifica como una STRING. Este parámetro tiene por finalidad identificar la LA "dueña" de la petición.

*Sintaxis general:*

<La-id-parameter> := STRING

#### 6.4.4.8 Parámetro identificador de disposición

El parámetro Layout-id-parameter se codifica como una NUMERIC-STRING, que puede tomar los valores "0", "1" ó "2".

*Sintaxis general:*

<Layout-id-parameter> := "0" | "1" | "2".

#### 6.4.4.9 Parámetro de trayecto

El parámetro Path-parameter se codifica como un PATH.

*Sintaxis general:*

<Path-parameter> := STRING

-- limitado por los convenios de denominación de fichero  
-- específicos de la plataforma

#### 6.4.4.10 Parámetro identificador de impresora

El parámetro Printer-id-parameter contiene el identificador de una impresora seleccionada, codificado como una STRING. El contenido de la STRING depende del sistema operativo empleado. El fabricante de la CA deberá indicar en su documentación la manera de controlar las impresoras.

*Sintaxis general:*

<Printer-id-parameter> := STRING

#### 6.4.4.11 Parámetro identificador de petición

El parámetro Req-id-parameter contiene la referencia de un identificador de petición [REQ-ID (véase 5.4.3)]. El valor del parámetro se representa como una STRING. Este parámetro tiene por finalidad especificar la relación de una respuesta con una petición precedente. Por consiguiente, el REQ-ID deberá ser único dentro de una LA. Es responsabilidad de la LA asegurar que el REQ-ID sea único.

*Sintaxis general:*

<Req-id-parameter> := STRING

#### 6.4.4.12 Parámetro de tiempo de envío

El parámetro de tiempo de envío Send-time-parameter se codifica como una STRING que contiene la fecha y hora (tiempo) de acuerdo con el convenio "YY-MM-DD-HH:MM" (año-mes-día-hora:minutos) o una indicación específica para tratamiento inmediato o urgente.

*Sintaxis general:*

<Send-id-parameter> := "IMMEDIATE" | "URGENT" | <Date-time-parameter>  
-- El <Date-time-parameter> se describe en 6.4.4.4

#### 6.4.4.13 Parámetro de estado

El parámetro de estado State-parameter especifica el estado del registro de CA. Se codifica como una STRING y puede tomar uno de los valores indicados a continuación:

"delayed" ("diferido")	concierno a la transmisión: la CA todavía no ha procesado el registro.
"sending" ("en envío")	concierno a la transmisión: la CA está procesando el registro para transmisión.
"sent" ("enviado")	concierno a la transmisión: el registro se ha transmitido con éxito al recipiente.
"failed" ("fracasado")	concierno a la transmisión y la recepción: el registro no pudo transmitirse con éxito, o se produjeron errores durante la recepción, o se produjo un error interno en la CA.
"reception" ("recepción")	concierno a la recepción: el registro ha sido recibido, pero todavía no ha sido recuperado.
"retrieved" ("recuperado")	concierno a la recepción: el registro ha sido recuperado.

*Sintaxis general:*

<State-parameter> := "delayed" | "sending" | "sent" | "failed" | "reception" | "retrieved"

#### 6.4.4.14 Parámetro de situación (o "status")

El parámetro de situación (o "status") Status-parameter refleja la situación ("status") de un evento de envío o recepción. Se codifica como una STRING, y puede tomar uno de los siguientes valores:

- "+" positivo;
- "+-" parcialmente negativo;
- "-" negativo;
- "?" desconocido.

*Sintaxis general:*

<Status-parameter> := "+" | "+-" | "-" | "?"

#### 6.4.4.15 Parámetro de subfunción

El parámetro de subfunción Subfunction-parameter se codifica como una STRING que especifica la subfunción que se va a invocar.

*Sintaxis general:*

<Subfunction-parameter> := STRING

#### 6.4.4.16 Parámetro de clave de usuario

El parámetro de clave de usuario Userkey-parameter se codifica como una STRING que contiene una clave de usuario que la LA especifica para el evento de comunicación en cuestión.

*Sintaxis general:*

<Userkey-parameter := STRING

### 6.4.5 Codificación de parámetros específicos del servicio

En esta subcláusula se describen algunos de los parámetros específicos del servicio utilizados de una manera similar para diversos servicios. En las cláusulas pertinentes de la Parte II de esta Recomendación se da una descripción completa de los parámetros mencionados.

#### 6.4.5.1 Parámetro identificador de servicio

El parámetro identificador de servicio Service-id-parameter se incluye en todas las TDD de envío y de recepción. Especifica el servicio que habrá de utilizarse. Se codifica como una STRING y puede tomar uno de los valores siguientes:

"FX3"	para servicios facsímil del grupo 3
"FX4"	para servicios facsímil del grupo 4
"TTX"	para el servicio teletex
"TX"	para el télex a través del servicio teletex (sin posibilidad de diálogo)
"TLX"	para el servicio télex
"EMAIL"	para servicios de correo electrónico
"FT"	para servicios de transferencia de ficheros

*Sintaxis general:*

<Service-id-parameter> := "FX3" | "FX4" | "TTX" | "TX" | "TLX" | "EMAIL" | "FT"

#### 6.4.5.2 Parámetros de fichero de especificaciones de direcciones (File-of-addrspec) y de dirección (Address-parameter)

Estos parámetros aparecen en todas las TDD de emisión y de recepción, y se utilizan de manera similar en todos los servicios.

El parámetro File-of-addrspec se codifica como un PATH (trayecto) que conduce a un fichero que contiene parámetros Addrspec-parameter, que a su vez contiene Address-parameter.

Address-parameter se codifica como STRING (cadena). Para algunos servicios de telecomunicaciones, esta cadena puede limitarse aún más.

*Sintaxis general:*

<File-of-addrspec> := PATH  
-- Trayecto que conduce a un fichero que contiene  
-- una lista de <Addrsec-parameter>

<Addrsec-parameter> := <Address-parameter> <Other-parameters>  
-- El (<Address-parameter>) se describe detalladamente para  
-- cada servicio en la cláusula de la Parte II de esta  
-- Recomendación.

<Address-parameter> := -- Se describe detalladamente en las cláusulas pertinentes de  
-- la Parte II de esta Recomendación.

### 6.4.5.3 Parámetro de identificador de conversión

El parámetro identificador de conversión (Convert-id-parameter) aparece en diversas TDD. Algunas de ellas (TDD de emisión y de recepción) dependen del servicio.

En todos los casos, el parámetro identificador de conversión se codifica como STRING y contiene el identificador de conversión, que especifica el formato de transferencia. Véanse también 5.4.5, cláusula 8 y el Cuadro H.1, así como las cláusulas pertinentes de la Parte II de esta Recomendación, para una información más detallada sobre el identificador de conversión.

*Sintaxis general:*

<Convert-id-parameter> := STRING

### 6.4.5.4 Parámetro identificador de tipo

El parámetro identificador de tipo (Type-id-parameter) aparece en todas las TDD de emisión y de recepción, y se utiliza de una manera similar en todos los servicios.

Este parámetro se codifica como STRING y contiene el identificador de tipo, que especifica el formato de transmisión. Véanse también 5.4.6 y el Cuadro H.2, así como las cláusulas pertinentes de la Parte II de esta Recomendación, para una información más detallada sobre el identificador de tipo.

*Sintaxis general:*

<Type-id-parameter> := STRING

### 6.4.5.5 Parámetro de fichero de especificaciones de ficheros (File-of-filespec) y parámetro de fichero (File-parameter)

Este parámetro aparece en todas las TDD de emisión y de recepción, y se utiliza de una manera similar en todos los servicios. File-of-filespec se codifica como PATH que apunta a un fichero que contiene Filespec-parameters, que a su vez contiene File-parameter.

File-parameter se codifica como PATH, que apunta al fichero transferido.

*Sintaxis general:*

<File-of-filespec> := PATH  
-- Trayecto hasta un fichero que contiene una lista de  
-- <Filespec-parameter>

<Filespec-parameter> := <File-parameter> <Other-parameters>  
-- El (<Filespec-parameter>) se describe detalladamente para  
-- cada servicio en la cláusula pertinente de la Parte II de esta  
-- Recomendación.

<File-parameter> := PATH  
-- Trayecto al fichero transferido.

## 6.5 Tratamiento de documentos

### 6.5.1 Definiciones utilizadas

Algunos servicios de telecomunicaciones ofrecen la posibilidad de transportar un solo fichero o un conjunto de ellos en un solo evento de comunicación. Son útiles las siguientes definiciones:

**6.5.1.1 documento:** Se considera que un documento es una pieza de información perteneciente a un contexto común que se transmitirá (o ha sido transmitido) dentro de una sesión a través de un servicio subyacente de comunicaciones o de telecomunicaciones. Un documento puede comprender varios ficheros.

**6.5.1.2 sesión:** Asociación entre dos sistemas de extremo, que les permite intercambiar datos sin pérdida ni interrupción visible.

**6.5.1.3 fichero:** Un fichero es una colección de datos mantenida por el sistema operativo subyacente.

### **6.5.2 Envío de documentos**

Se admiten dos modalidades de envío de documentos:

- envío de un documento, constituido por un solo fichero, a uno o más destinatarios;
- envío de un documento, constituido por varios ficheros, a uno o varios destinatarios.

Para intercambiar correctamente un documento a través de un servicio de telecomunicaciones es necesario especificar, como mínimo, los tres ítems siguientes:

- el trayecto al fichero (o los trayectos a los ficheros) que constituye(n) el documento;
- el formato de transferencia (Convert-ID) en el que se intercambia el documento entre la LA y la CA;
- el formato de transmisión (Type-ID) en el cual se intercambia el documento a través de la red.

Si el documento consta de un fichero, el trayecto al fichero se puede dar en el elemento de sintaxis <Document>, el formato de transmisión en el elemento de sintaxis <Type>, y el formato de transferencia en el elemento de sintaxis <Convert>. Dado que estos tres elementos de sintaxis dependen del servicio en lo que respecta a los valores, se describen para cada servicio en la Parte II de esta Recomendación.

Si el documento consta de varios ficheros, los tres componentes se reunirán de conformidad con una sintaxis específica del servicio. Como esta sintaxis es específica del servicio, se describe también para cada servicio en la Parte II de esta Recomendación. De acuerdo con esta sintaxis, los componentes reunidos serán puestos en un fichero de indirección ("indirection file") por la CA. El trayecto a este fichero de indirección se especificará entonces en el elemento de sintaxis apropiado de la TDD.

NOTA – El trayecto de un fichero de indirección se especifica siempre precedido por el carácter "@".

### **6.5.3 Recepción de documentos**

Se admiten dos modalidades de recepción de documentos:

- recepción de un documento constituido por un solo fichero;
- recepción de un documento constituido por varios ficheros.

Si el documento recibido consta de varios ficheros, la CA construye un fichero de indirección como se describió en el anterior 6.5.2 y hace preceder el trayecto del fichero por el carácter "@".

Si el documento recibido consta de un solo fichero, el trayecto dado (facultativamente) por la LA se utilizará para almacenar el propio documento. El formato de transmisión se retornará seguidamente en el elemento de sintaxis <Type>, y el formato de transferencia en el elemento de sintaxis <Convert>.

### **6.5.4 Formato del fichero de direccionamiento**

El fichero de direccionamiento ("indirection file") contiene líneas de elementos de sintaxis específicos del servicio, una línea para cada fichero. La presentación de código será conforme al identificador de código (Code-ID) utilizado para las TDD. Para una descripción genérica de los elementos de sintaxis contenidos en el fichero de indirección, véase 6.4.5.5.

NOTA – El fichero de direccionamiento se procesará en el orden de aparición.

## **6.6 Funcionalidad de las TDD**

En esta subcláusula se describe la funcionalidad que es activada por las TDD de petición. Se da información sobre la manera de utilizar los elementos de sintaxis en la petición y en la respuesta. El Cuadro 17 explica los títulos de las columnas de los cuadros relativos a las TDD utilizados en las subcláusulas siguientes.

**Explicación de los títulos de las columnas de los cuadros relativos a las TDD**

Columna	Contenido
Elemento de sintaxis	Indica los elementos de sintaxis utilizados para la TDD
C (Clase)	Representa la clase de la palabra clave. "B" (abreviatura de básico) significa que la palabra clave será soportada por todas las CA; "+" significa que la palabra clave será soportada por la CA únicamente si dicha palabra clave está declarada en el ICE
T (Tipo)	Especifica si la palabra clave es obligatoria ("m") o facultativa ("o"). Cuando la palabra clave es facultativa, la célula "Default" (por defecto) especifica un valor por defecto para el parámetro
I/O (Entrada/Salida)	La columna entrada/salida indica los requisitos relativos al parámetro que sigue a la palabra clave en la petición y en la respuesta  Cuando la columna indica "I", el parámetro contiene un valor de entrada ("input") es decir, la CA interpretará el valor como solicitado por la LA. La CA no modificará el valor y lo retornará en la respuesta  Cuando la columna indica "I/O", el parámetro contiene un valor de entrada ("input") (como cuando se especificó "I") y se espera un valor de salida ("output") (es decir, fijado por la CA en la TDD de respuesta)  Cuando la columna indica "O", el parámetro es un valor de salida ("output") (es decir, fijado por la CA en la TDD de respuesta) y no se fijará ningún valor en la petición (es decir, no se especificará ningún valor de parámetro en la petición). La respuesta contendrá el valor utilizado para ese parámetro por la CA
Palabra clave	Indica el nombre de la palabra clave que se lee en la TDD cuando ésta se codifica como texto legible. La escritura en mayúsculas o minúsculas no es significativa, por ejemplo: "Function" y "funcTion" deberán interpretarse como una misma palabra. Véase también 6.4.1
Parámetro	Indica los posibles valores del parámetro cuando se codifica como texto legible. La correspondencia de los tipos de parámetros se describe en 6.4.4
Por defecto	Indica los valores por defecto del parámetro (cuando son aplicables, es decir cuando la célula "Type" indica "O")
Observaciones	Da una breve explicación de la utilización de las palabras clave y de los parámetros

Se utilizarán las siguientes reglas generales:

- Una TDD dada (de respuesta o de petición) pertenece a una clase funcional. En consecuencia, las palabras clave conexas no dependen de la clase funcional.
- Las palabras clave se subdividen en dos categorías: básicas y aditivas ("+"). Las palabras clave aditivas sólo pueden ser utilizadas por las LA si están declaradas en el ICE (descriptor de CA). Las palabras clave básicas son admitidas por todas las CA.
- La utilización o no utilización de una palabra clave tiene el mismo significado independientemente de que sea básica o aditiva.
- Si una palabra clave está ausente en una TDD, se aplica el valor por defecto del parámetro. Un valor por defecto es siempre un parámetro de entrada, es decir, la clasificación "Input/Output (Entrada/Salida)" es siempre "I".
- Cuando un parámetro se clasifica como "I" (entrada), la CA no modificará su valor en la TDD de respuesta.

El Cuadro 18 explica la utilización de las clasificaciones "Tipo ("Type") y "entrada/salida" ("Input/Output"):

CUADRO 18/T.611

**Utilización de las clasificaciones "tipo" y "entrada/salida"**

Tipo	Entrada/salida	La palabra clave
"m"	"I"	Deberá especificarse en la petición; deberá especificarse también un valor de parámetro. En la respuesta, la palabra clave se especifica también, con su valor de parámetro inalterado
	"O"	Deberá especificarse en la petición con un valor de parámetro vacío. En la respuesta, la palabra clave también está presente, con un valor de parámetro significativo
	"I/O"	Deberá especificarse en la petición; deberá especificarse también el valor del parámetro. En la respuesta, la palabra clave se especifica también con un valor de parámetro que puede ser diferente. El valor de parámetro en la respuesta es significativo
"o"	"I"	En la petición, la palabra clave puede estar o no presente: si está ausente, se aplica el valor por defecto. La respuesta puede incluir la palabra clave solamente si se especificó en la petición; en este caso, el valor del parámetro no se cambia
	"O"	Puede especificarse en la petición (sin ningún valor de parámetro) si se desea obtener un valor de parámetro de respuesta para la palabra clave; en este caso debe especificarse un valor de parámetro en la respuesta. En otro caso no se retornará ningún valor de parámetro de respuesta. No se aplican valores por defecto a la palabra clave
	"I/O"	Puede especificarse en la petición con un valor de parámetro. Si no se especifica en la petición, se aplica el valor de parámetro por defecto. En la respuesta, la palabra clave sólo está presente si lo estaba en la petición; en este caso, el valor de parámetro de respuesta es significativo. Si la petición no especifica la palabra clave, la respuesta no podrá especificarla tampoco

**6.6.1 Envío**

<SendTDD> y <SendAckTDD> se utilizan para enviar uno o más documentos a uno o más recibientes. La respuesta <SendAckTDD> actúa como un acuse de recibo a una petición precedente. La CA no genera respuesta para la petición <SendTDD>.

<SendTDD> se puede utilizar para enviar documentos a cualquier cantidad de recibientes. <SendAckTDD> permite un solo recibiente por petición. <SendAckTDD> está prevista para las LA que sólo aplican la clase funcional A y que, por tanto necesitan un informe de compleción de la CA. <SendTDD> está prevista para las LA que aplican la clase funcional B; por lo que poseen los medios para rastrear peticiones TDD a través de la función Trace. No obstante, ambos tipos de LA pueden utilizar <SendTDD> y <SendAckTDD>.

Si una LA desea enviar documentos a una lista de recibientes, utilizará la <SendTDD> junto con el elemento de sintaxis <AddrList>. La situación ("status") de la transmisión (elemento de sintaxis <Status>) no está disponible a través de esta función, ya que no se genera ninguna TDD de respuesta. No obstante la situación de la transmisión puede obtenerse utilizando la funcionalidad de rastreo: COPIA descrita en 6.6.3.

NOTA – La funcionalidad Trace sólo está disponible a través de las CA conformes con la clase funcional B (FCB).

Se proporcionan dos cuadros: uno para <SendTDD> y el otro para el caso "SendackTDD".

La respuesta <SendAckTDD> se entiende siempre como una situación (status) de compleción, es decir, si la petición tiene éxito, se generará la respuesta cuando haya sido completado el procesamiento de la petición <SendAckTDD>. De esta forma, cada vez que esté disponible la TDD de respuesta, la LA puede basarse definitivamente en el contenido del parámetro status que muestra el resultado de la transmisión. El parámetro status indica una de las cuatro siguientes situaciones:

- 1) documento transmitido completamente, con éxito;
- 2) transmisión fracasada, la CA ha abandonado todos los intentos de transmisión;

- 3) transmisión parcialmente fracasada, se han transmitido algunas partes, pero no pudo realizarse la transmisión completa;
- 4) situación desconocida.

<SendTDD> y <SendAckTDD> pertenecen a las clases funcionales A y B.

Si en una petición <SendAckTDD> se especifican múltiples recibientes, <Status> y <Error> pueden no estar especificados (véanse los Cuadros 19 y 20).

CUADRO 19/T.611

**Elementos de sintaxis de la <SendAckTDD>**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<SendAckFunction>	B	m	I	FUNCTION	"SendAck"	–	La CA generará una respuesta
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqID>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<Service>	B	m	I	SERVICE	<Service-id-parameter>	–	
<Error>	B	m	O	ERROR	<Error-parameter>	–	El error retornado por la CA
<Status>	B	m	O	STATUS	<Status-parameter>	–	Retorna la situación (el "status") de la CA
<ComId>	B	o	O	COMID	<Com-id-parameter>	–	Identificación de la comunicación (calculada por la CA)
<SendTime>	B	o	I	SENDTIME	<Send-time-parameter>	"IMMEDIATE"	La CA procesará efectivamente la petición en el tiempo especificado
<Comment>	+	o	I	COMMENT	<Comment-parameter>	–	
<LastTime>	+	o	I	LASTTIME	<Date-time-parameter>	CA dependent	Límite de tiempo para procesar la petición, expresado como un tiempo absoluto
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<UserKey>	+	o	I	USERKEY	<Userkey-parameter>	–	La petición y la respuesta contendrán el mismo valor de parámetro. La CA no interpretará este parámetro de ninguna manera
<Warning>	+	o	O	WARNING	<Error-parameter>	–	

## Elementos de sintaxis de &lt;SendTDD&gt;

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<SendFunction>	B	m	I	FUNCTION	"Send"	–	La CA NO generará una respuesta
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<Service>	B	m	I	SERVICE	<Service-id-parameter>	–	
<SendTime>	B	o	I	SENDTIME	<Send-time-parameter>	"IMMEDIATE"	La CA procesará efectivamente la petición en el tiempo especificado
<Comment>	+	o	I	COMMENT	STRING	–	
<LastTime>	+	o	I	LASTTIME	<Date-Time-parameter>	CA dependent	Último tiempo para procesar la petición, expresado como un tiempo absoluto
<UserKey>	+	o	I	USERKEY	STRING	–	La CA no interpretará este parámetro de ninguna manera

### 6.6.2 Recepción

Las peticiones <ReceiveTDD> se utilizan para recuperar un documento entrante que ya ha sido recibido por la CA. La <ReceiveTDD> pertenece a las clases funcionales A y B (véase el Cuadro 21).

La petición especifica el servicio de telecomunicaciones, el área de almacenamiento para el fichero entrante y el formato de transferencia deseado. Para algunos servicios, la subdirección del recipiente se puede utilizar también como un selector (véase la cláusula pertinente de la Parte II de esta Recomendación).

Para recuperar (recibir), la LA puede utilizar dos métodos diferentes:

- o bien la LA recupera lo que viene seguidamente, sin ninguna preselección (en este caso la LA tiene que poder tratar adecuadamente el documento recibido, independientemente del formato de transferencia del documento);
- o bien, examina primero una copia de una lista de documentos recibidos utilizando la petición de rastreo <CopyTDD>. Con esa lista la LA puede obtener el COM-ID del documento que desea recuperar. Esto sólo es posible si la CA soporta la clase funcional B y la LA admite la función <CopyTDD>.

El elemento de sintaxis <Convert> se utiliza para almacenar el formato de transferencia efectivo del documento dentro de la TDD de respuesta. Los elementos de sintaxis <Status> y <Error> son necesarios para transportar el código de status y de error del evento recepción. Si la LA conoce el COM-ID del documento que desea recuperar, puede también especificar el elemento de sintaxis <ComId> en la respuesta, para extraer el documento.

El elemento de sintaxis <Delete> *aditivo* controla la supresión del documento dentro de la CA. En funcionamiento normal, <Delete> será verdadero ("true") y la CA suprimirá el documento en su almacenamiento después de haberlo transferido. Así, un documento sólo puede recibirse una vez, con lo que se obliga al registro de CA a pasar al estado "retrieved" (recuperado) descrito en 5.6.7.1.

Si <Delete> se fija a falso, el documento será conservado por la CA y el registro de CA se mantendrá en el estado "reception" (recepción). En consecuencia, en el intento siguiente, el mismo documento podría ser recuperado de nuevo.

NOTA – En la implementación de la CA se podrá optar por efectuar la supresión solo virtualmente y mantener internamente los documentos suprimidos; por ejemplo, se puede hacer que estos documentos estén disponibles para otras LA, o conservarlos para fines de archivo.

CUADRO 21/T.611

**Elementos de sintaxis de <ReceiveTDD>**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<ReceiveFunction>	B	m	I	FUNCTION	"Receive"	–	
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<Error>	B	m	O	ERROR	<Error-parameter>	–	El error retornado por la CA
<Status>	B	m	O	STATUS	<Status-parameter>	–	Retorna el estado de la CA. Cuando no hay documento disponible, la TDD de respuesta especificará el valor "-"
<ComId>	B	o	I/O	COMID	<Com-id-parameter>	–	Identificación de la comunicación (calculada por la CA)
<Service>	B	o	I/O	SERVICE	<Service-id-parameter>	–	Si se utiliza en la petición, se emplea como un criterio de selección
<Delete>	+	o	I	DELETE	<Boolean-parameter>	"Sí"	Si es verdadero ("true"), la CA suprimirá el documento recuperado eliminándolo de su memoria tampón interna
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<ReceiveTime>	+	o	O	RCVTIME	<Date-time-parameter>	–	Tiempo de la recepción del documento por la CA
<Warning>	+	o	O	WARNING	<Error-parameter>	–	

**6.6.3 Rastreo (Trace)**

Las TDD pertenecientes al grupo de rastreo se utilizan para gestionar los registros de CA. La funcionalidad de rastreo (Trace) es específica de la clase funcional B.

Las diversas TDD tienen por finalidad controlar el funcionamiento de una CA. Se definen las siguientes TDD:

- <PurgeTDD> para PURGAR todo registro de CA que se encuentre en cualquiera de los estados "failed" (fracasado), "retrieved" (recuperado) y "sent" (enviado); esta acción es útil para liberar los registros de CA que han pasado a un estado inactivo;
- <CopyTDD> para COPIAR cualquier registro de CA en cualquier estado, pasándolo a un fichero; esta acción es útil para construir lo que se conoce usualmente por diarios ("journals") o cuadernos ("logs");
- <CancelTDD> para CANCELAR cualquier registro de CA en el estado "sending" (en envío); esta acción permite a la LA interrumpir y terminar la transmisión que está realizando la CA<sup>4</sup>);
- <DeleteTDD> para SUPRIMIR cualquier registro de CA en el estado "delayed" (diferido) con el fin de cancelarlo;
- <RescheduleTDD> para REPROGRAMAR cualquier registro de CA en el estado "fracasado" para dar a la CA la oportunidad de retransmitirlo; esta acción facilita la gestión de transmisiones que fracasaron, por ejemplo porque el recipiente estaba ocupado o porque la petición estaba mal hecha;
- <DispatchTDD> para DESPACHAR cualquier registro de CA en el estado "reception" (recepción) para asignarlo a la LA recipiente actual;
- <PreviewTDD> para VER PREVIAMENTE (prever) un documento asociado con un registro de CA en el estado "reception".

Debe señalarse que todas las peticiones se ejecutan en nombre de una LA dada (mediante el LA-ID). Esto limita el alcance de las funciones más allá de los registros de CA asignados a ese LA-ID (salvo las situaciones obvias cuando no se especifica el LA-ID).

Para suprimir una petición anterior que se encuentra en el estado "send" (envío), deberá utilizarse <DeleteTDD>.

Si se conoce, la LA puede especificar el elemento de sintaxis <ComId> en lugar del elemento de sintaxis <ReqRef>.

#### **6.6.3.1 Rastreo:SUPRESIÓN (Trace:DELETE)**

Véase el Cuadro 22

#### **6.6.3.2 Rastreo:COPIA (Trace:COPY)**

Para obtener una copia de registros de CA que se refieren a un estado específico deberá utilizarse <CopyTDD>.

El elemento de sintaxis <State> especifica el estado de los registros de CA que se desea copiar. El elemento de sintaxis <Target> especifica el trayecto que conduce al fichero en el cual la CA generará su salida. Se utiliza <Error> para almacenar el código de error de la operación en la TDD de respuesta. La lista copiada está siempre en el formato implicado por el identificador de código de la TDD solicitante. La disposición de la lista viene dada implícitamente por la entrada RECORD del ICE. El orden de los campos en la lista de destino será el mismo en que se declaran las correspondientes palabras clave en la entrada RECORD.

Si lo conoce, la LA puede especificar el elemento de sintaxis <ComId> en lugar del elemento de sintaxis <ReqRef>.

El elemento de sintaxis facultativo <Layout> puede utilizarse para especificar la disposición, ("layout") de la copia resultante. Los valores que puede tomar este parámetro se explican en el Cuadro 23, y se presentan ejemplos en las Figuras 7 y 8. La representación de los caracteres en la copia es implicado por el <TDD Header> de la correspondiente <CopyTDD> (véase el Cuadro 24).

---

<sup>4</sup>) No se garantiza que la operación de cancelación pueda tener éxito a causa de la naturaleza de la interfaz.

CUADRO 22/T.611

**Elementos de sintaxis de <DeleteTDD>**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<DeleteFunction>	B	m	I	FUNCTION	"Delete"	–	Suprime registros de CA en el estado "diferido"
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<Error>	B	m	O	ERROR	<Error-parameter>	–	El error retornado por la CA
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<Warning>	+	o	O	WARNING	<Error-parameter>	–	
Utilización del elemento de sintaxis <ComId>							
<ComId>	B	m	I	COMID	<Com-id-parameter>	–	Identificación de la comunicación calculada por la CA
Utilización del elemento de sintaxis <ReqRef>							
<ReqRef>	B	m	I	REQREF	<Req-id-parameter>	–	Referencia a un REQ-ID anterior

CUADRO 23/T.611

**Efectos del elemento de sintaxis <Layout>**

Layout-id	Explicación
0	Valor por defecto. La cantidad, disposición y orden de los campos presentados en la copia de destino se definen en el ICE por la entrada RECORD. La entrada RECORD da la longitud de los campos e implica su orden. Los campos de un registro de CA (CA Record) se presentan en una fila, rellenos con SPACE (espacio) en la longitud especificada por la entrada RECORD. No se proporciona una línea de encabezamiento
1	Los campos se presentan fila por fila como valores escritos entre apóstrofes y separados por el carácter "," (coma). La primera fila contiene las palabras clave a que se hace referencia en la columna del campo. Véase también la Figura 7
2	Los campos se presentan fila por fila, separados por caracteres de tabulador. La primera fila del campo contiene las palabras clave a que hace referencia la columna del campo. Véase también la Figura 8

```
"COMID","DIRECTION","LA-ID","REQ-ID","STATE"  
"0001","xmit","Jonny","REQ-1212","delayed"  
"0002","xmit","Jonny","REQ-1213","sent"  
"0003","receive","Jonny","REQ-1214","retrieved"
```

FIGURA 7/T.611

**Efectos del elemento de sintaxis <Layout>**

COMID	DIRECTION	LA-ID	REQ-ID	STATE
0001	xmit	Jonny	REQ-1212	delayed
0002	xmit	Jonny	REQ-1213	sent
0003	receive	Jonny	REQ-1214	retrieved

FIGURA 8/T.611

**Ejemplo de disposición del campo en forma de valores  
separados por tabuladores (Layout-id = 2)**

CUADRO 24/T.611

Elementos de sintaxis de <CopyTDD>

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<CopyFunction>	B	m	I	FUNCTION	"Copy"	–	Copia todos los registros de CA que se encuentran en un estado dado, escribiéndolos en el fichero a que apunta la palabra clave TARGET
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<State>	B	m	I	STATE	<State-parameter>	Todos los estados	Estado del registro de CA
<Target>	B	m	I	TARGET	<Path-parameter>	–	Nombre del fichero de destino
<Error>	B	m	O	ERROR	<Error-parameter>	–	Error retornado por la CA
<Layout>	B	o	I	LAYOUT	<Layout-id-parameter>	"0"	Especifica la disposición ("layout") de la copia resultante
<Direction>	B	o	I	DIRECTION	<Direction-parameter>	Ambos sentidos de transmisión	
<Dispatched>	B	o	I	DISPATCHED	<Boolean-parameter>	Todos los registros de CA	
<Previewed>	B	o	I	PREVIEWED	<Boolean-parameter>	Todos los registros de CA	
<Service>	B	o	I	SERVICE	<Service-id-parameter>	Todos los servicios	
<TimeRange>	B	o	I	RANGE	<Date-time-parameters>	Todos los registros de CA	
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<Warning>	+	o	O	WARNING	<Error-parameter>	–	
Utilización del elemento de sintaxis <ComId>							
<ComId>	B	m	I	COMID	<Com-id-parameter>	–	Identificación de la comunicación calculada por la CA
Utilización del elemento de sintaxis <ReqRef>							
<ReqRef>	B	m	I	REQREF	<Req-id-parameter>	–	Referencia a un REQ-ID anterior

### 6.6.3.3 Trace:CANCEL (rastreo:cancelación)

Para cancelar una anterior petición de "envío" ("send"), la LA utilizará la <CancelTDD> (véase el Cuadro 25).

Si lo conoce, la LA puede también especificar el elemento de sintaxis <ComId> en lugar del elemento de sintaxis <ReqRef>.

NOTA – La cancelación de una petición de envío puede no tener éxito debido a la naturaleza de la interfaz.

CUADRO 25/T.611

#### Elementos de sintaxis de la <CancelTDD>

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<CancelFunction>	B	m	I	FUNCTION	"Cancel"	–	Cancela un registro de CA generado por una anterior petición
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<Error>	B	m	O	ERROR	<Error-parameter>	–	Error retornado por la CA
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<Warning>	+	o	O	WARNING	<Error-parameter>	–	
Utilización del elemento de sintaxis <ComId>							
<ComId>	B	m	I	COMID	<Com-id-parameter>	–	Identificación de la comunicación calculada por la CA
Utilización del elemento de sintaxis <ReqRef>							
<ReqRef>	B	m	I	REQREF	<Req-id-parameter>	–	Referencia a un REQ-ID anterior

#### 6.6.3.4 Trace:PURGE (rastreo:purga)

Para eliminar ("purgar") registros de CA en una CA se utilizará la <PurgeTDD> (véase el Cuadro 26).

El elemento de sintaxis <State> especifica el estado de los registros de CA que habrán de ser purgados (véase 5.6). El elemento de sintaxis <Error> registra el código de error de la operación en la TDD de respuesta.

Si lo conoce, la LA puede también especificar el elemento de sintaxis <Comid> en lugar del elemento de sintaxis <ReqRef>.

CUADRO 26/T.611

#### Elementos de sintaxis de la <PurgeTDD>

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<PurgeFunction>	B	m	I	FUNCTION	"Purge"	–	Elimina ("purga") un registro de CA generado por una anterior petición
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<State>	B	m	I	STATE	<State-parameter>	Todos los estados <sup>a)</sup>	Estado del registro de CA
<Error>	B	m	O	ERROR	<Error-parameter>	–	Error retornado por la CA
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<Warning>	+	o	O	WARNING	<Error-parameter>	–	
Utilización del elemento de sintaxis <ComId>							
<ComId>	B	m	I	COMID	<Com-id-parameter>	–	Identificación de la comunicación calculada por la CA
Utilización del elemento de sintaxis <ReqRef>							
<ReqRef>	B	m	I	REQREF	<Req-id-parameter>	–	Referencia a un REQ-ID anterior
<sup>a)</sup> "Todos los estados" en este caso significa los estados "sent", "failed" y "retrieved". La CA ignorará la aplicación de la funcionalidad "purge" en todos los demás estados.							

### 6.6.3.5 Trace:RESCHEDULE (rastreo:reprogramación)

A fin de "reprogramar" (en el sentido de poner nuevamente en curso) una petición de envío que ha fracasado, deberá utilizarse la <RescheduleTDD>. Si lo conoce, la LA puede especificar también el elemento de sintaxis <ComId> en lugar del elemento de sintaxis <ReqRef> (véase el Cuadro 27).

CUADRO 27/T.611

#### Elementos de sintaxis de la <RescheduleTDD>

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<Reschedule-Function>	B	m	I	FUNCTION	"Reschedule"	–	Se reprograma un registro de CA generado por una anterior petición de envío
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<Error>	B	m	O	ERROR	<Error-parameter>	–	El error retornado por la CA
<Address>	B	o	I	ADDRESS	<Address-parameter>	–	Utilizado por la función "reschedule" solamente; especifica la dirección del nuevo recipiente
<SendTime>	B	o	I	SENDTIME	<Send-time-parameter>	"IMMEDIATE"	Procesa la petición en el tiempo especificado
<LastTime>	+	o	I	LASTTIME	<Date-time-parameter>	Depende de la CA	Límite de tiempo para procesar la petición
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<Warning>	+	o	O	WARNING	<Error-parameter>	–	
Utilización del elemento de sintaxis <ComId>							
<ComId>	B	m	I	COMID	<Com-id-parameter>	–	Identificación de la comunicación calculada por la CA
Utilización del elemento de sintaxis <ReqRef>							
<ReqRef>	B	m	I	REQREF	<Req-id-parameter>	–	Referencia a un REQ-ID anterior

**6.6.3.6 Trace:DISPATCH (rastreo:despacho)**

Para despachar hacia una LA un fichero recibido deberá utilizarse la <DispatchTDD> (véase el Cuadro 28).

El elemento de sintaxis <NewLa>, que puede especificarse repetidamente, contiene el LA-ID de la nueva LA. Cuando el retorno tiene éxito, la nueva LA es la propietaria del documento recibido y puede recuperarlo.

NOTA – La utilización de la función dispatch puede limitarse.

CUADRO 28/T.611

**Elementos de sintaxis de la <DispatchTDD>**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<DispatchFunction>	B	m	I	FUNCTION	"Dispatch"	–	Despacha documentos asociados al registro de CA designado por el COM-ID
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<NewLa>+	B	m	I	NEWLA	<La-id-parameter>	–	Especifica el nombre del nuevo "propietario" del documento. Puede especificarse repetidamente
<Error>	B	m	O	ERROR	<Error-parameter>	–	Error retornado por la CA
<ComId>	B	m	I	COMID	<Com-id-parameter>	–	Identificación de la comunicación calculada por la CA
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<Warning>	+	o	O	WARNING	<Error-parameter>	–	

**6.6.3.7 Trace:PREVIEW (rastreo:visión previa)**

La <PreviewTDD> permite a un administrador de la CA recuperar y despachar los documentos recibidos, al mismo tiempo que los mantiene disponibles (para ulteriores despachos). La función PREVIEW actúa como una función Receive (recepción), con la diferencia de que el registro de CA se mantiene en el estado recepción en lugar de pasar al estado recibido (véase el Cuadro 29).

En el registro de CA deberá también indicarse si el documento fue o no visionado por el administrador, y si el documento ya ha sido despachado. Esto permite a una función Trace:COPY efectuar una búsqueda de los documentos recibidos que no han sido ya visionados o despachados a sus recipientes finales.

Se aplicarán también los siguientes principios:

Cuando la CA genera un registro de CA (llamado registro CA "primario" a los efectos de la explicación) después de recibir un documento entrante, el registro de CA primario se asigna al usuario "administrador". Las banderas del registro de CA primario "viewed" (visionado) y "dispatched" (despachado) se fijan a "no".

Para despachar el documento entrante, el administrador obtiene una visión previa del documento utilizando la función Trace:PREVIEW. La bandera "viewed" del registro de CA primario se fija entonces a "sí". Seguidamente el administrador despacha el documento a uno o varios recibientes. La bandera "dispatched" del registro de CA primario se fija entonces a "sí".

La CA genera internamente una nueva copia del registro de CA primario para cada recibiente, con los siguientes atributos:

- Estado = Recepción
- LA-ID = LoginName del recibiente a que se despachó
- Viewed = "no"
- Dispatched = "no"

Este registro de CA se denomina registro de CA "secundario", para mayor claridad. Ahora, cada registro de CA secundario es propiedad del recibiente a que se despachó. El recibiente puede recuperarlo con la función Trace:COPY usual, y recibirlo con la función Receive. Si el sistema lo permitiera, el usuario podría también despachar, u obtener una visión previa de los documentos recibidos utilizando las mismas llamadas a funciones.

Este mecanismo permite al administrador de la CA establecer una lista de todos los recibientes de un documento dado, observar quienes de ellos lo han recuperado, redespachar un documento a petición, etc.

CUADRO 29/T.611

**Elementos de sintaxis de la <PreviewTDD>**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<PreviewFunction>	B	m	I	FUNCTION	"Preview"	-	Recupera documentos asociados con un registro de CA designado por el COM-ID
<LaId>	B	m	I	LA-ID	<La-id-parameter>	-	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	-	
<ComId>	B	m	I	COMID	<Com-id-parameter>	-	Identificación de la comunicación calculada por la CA
<Convert>	B	m	O	CONVERT	<Convert-id-parameter>	-	Formato de transferencia del fichero de destino
<Target>	B	m	I	TARGET	<Path-parameter>	-	Nombre del fichero de destino
<Error>	B	m	O	ERROR	<Error-parameter>	-	Error retornado por la CA
<Minor>	+	o	O	MINOR	<Error-parameter>	-	
<Warning>	+	o	O	WARNING	<Error-parameter>	-	

**6.6.4 Depósito (Submit)**

La funcionalidad de depósito ("submit") ha sido concebida para que la CA pueda efectuar diversas funciones de uso general.

Tiene por finalidad pedir a la CA que efectúe algunas funciones adicionales que ella puede implementar, como conversiones de formatos de ficheros, o impresión de documentos entrantes. Estas tareas normalmente no las realiza la CA, pero algunos fabricantes de CA pudieran desear admitirlas. Un ejemplo de una situación en que la característica "impresión" ("printing") podría ser útil es la de un "servidor de CA" de una LAN (red de área local). La funcionalidad de depósito soporta las siguientes TDD:

- <PrintTDD> para IMPRIMIR un documento, dado su trayecto y formato;
- <ConvertTDD> para CONVERTIR un documento, dado su trayecto, formato de entrada, formato de salida y nombre de fichero de salida;
- <CheckTDD> para COMPROBAR el formato de transferencia de un documento, dado el trayecto y el modelo de formato de transferencia con respecto al cual se debe comprobar.

#### 6.6.4.1 Submit: PRINT (depósito: impresión)

Véase el Cuadro 30.

CUADRO 30/T.611

#### Elementos de sintaxis de la <PrintTDD>

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<PrintFunction>	B	m	I	FUNCTION	"Print"	-	Depósito de la tarea de comprobación en la CA
<LaId>	B	m	I	LA-ID	<La-id-parameter>	-	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	-	
<FileName>	B	m	I	FILENAME	<Path-parameter>	-	Fichero original
<InFormat>	B	m	I	INFORMAT	<Convert-id-parameter>	-	Formato original
<Error>	B	m	O	ERROR	<Error-parameter>	-	Error retornado por la CA
<Printer>	+	o	I	PRINTER	<Printer-id-parameter>	"STD"	"STD" representa la impresora estándar ("standard printer") configurada en la CA. En el componente "PRINT" del ICE se declaran, para cada CA, otros posibles valores para el Printer-id (véase 9.5)
<Minor>	+	o	O	MINOR	<Error-parameter>	-	
<Warning>	+	o	O	WARNING	<Error-parameter>	-	

### 6.6.4.2 Submit: CONVERT (depósito: conversión)

Véase el Cuadro 31.

CUADRO 31/T.611

#### Elementos de sintaxis de la <ConvertTDD>

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<ConvertFunction>	B	m	I	FUNCTION	"Convert"	–	Depósito de la tarea de comprobación en la CA
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<FileName>	B	m	I	FILENAME	<Path-parameter>	–	Fichero original
<Target>	B	m	I	TARGET	<Path-parameter>	–	Fichero de destino
<InFormat>	B	m	I	INFORMAT	<Convert-id-parameter>	–	Formato original
<OutFormat>	B	m	I	OUTFORMAT	<Convert-id-parameter>	–	Formato de destino
<Error>	B	m	O	ERROR	<Error-parameter>	–	Error retornado por la CA
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<Warning>	+	o	O	WARNING	<Error-parameter>	–	

### 6.6.4.3 Submit: CHECK (depósito: comprobación)

Véase el Cuadro 32.

CUADRO 32/T.611

#### Elementos de sintaxis de la <CheckTDD>

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<DeleteFunction>	B	m	I	FUNCTION	"Check"	–	Depósito de la tarea de comprobación en la CA
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	

CUADRO 32/T.611 (fin)

**Elementos de sintaxis de la <CheckTDD>**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<FileName>	B	m	I	FILENAME	<Path-parameter>	–	Fichero original
<Check>	B	m	I	CHECK	<Convert-id-parameter>	–	Formato con respecto al cual se comprueba
<Error>	B	m	O	ERROR	<Error-parameter>	–	Error retornado por la CA
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<Warning>	+	o	O	WARNING	<Error-parameter>	–	

**6.6.5 Ampliación (o extensión)**

La funcionalidad de extensión (o ampliación) se divide en tres tipos diferentes de TDD:

- <ExtendTDD>, para AMPLIAR la funcionalidad en una base GENERAL;
- <NationalTDD> para ampliar la funcionalidad en una base NACIONAL;
- <PrivateTDD> para ampliar la funcionalidad en una base PRIVADA.

**6.6.5.1 Ampliación: AMPLIACIÓN (Extend: EXTEND)**

<ExtendTDD> tiene por finalidad proporcionar características ampliadas que no son de vital importancia para el funcionamiento de la interfaz, pero cuyo uso se ha difundido mucho, o es necesario. <ExtendTDD> es una facilidad adicional (véase el Cuadro 33).

CUADRO 33/T.611

**Elementos de sintaxis mínimos de la <ExtendTDD>**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<ExtendFunction>	B	m	I	FUNCTION	"Extend"	–	
<SubFunction>	B	m	I	SUBFUNC	<Subfunction-parameter>	–	Para la asignación de valores véase la cláusula pertinente de la Parte II de esta Recomendación
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<Error>	B	m	O	ERROR	<Error-parameter>	–	Error retornado por la CA

### 6.6.5.2 Ampliación: NACIONAL (Extend: NATIONAL)

<NationalTDD> tiene por finalidad proporcionar características nacionales que son propias de cada país. <NationalTDD> es una facilidad adicional (véase el Cuadro 34).

CUADRO 34/T.611

#### Elementos de sintaxis mínimos de la <NationalTDD>

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<NationalFunction>	B	m	I	FUNCTION	"National"	–	
<SubFunction>	B	m	I	SUBFUNC	<Subfunction-parameter>	–	Los valores de parámetros quedan en estudio
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<Error>	B	m	O	ERROR	<Error-parameter>	–	Error retornado por la CA

### 6.6.5.3 Ampliación: PRIVADO (Extend: PRIVATE)

La <PrivateTDD> tiene por finalidad suministrar características privadas que son propias de cada fabricante. La <PrivateTDD> es una facilidad adicional (véase el Cuadro 35).

CUADRO 35/T.611

#### Elementos de sintaxis mínimos de la <PrivateTDD>

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<PrivateFunction>	B	m	I	FUNCTION	"Private"	–	
<SubFunction>	B	m	I	SUBFUNC	<Subfunction-parameter>	–	Los valores de los parámetros quedan en estudio
<LaId>	B	m	I	LA-ID	<La-id-parameter>	–	
<ReqId>	B	m	I	REQ-ID	<Req-id-parameter>	–	
<Error>	B	m	O	ERROR	<Error-parameter>	–	Error retornado por la CA

## 7 Método de intercambio

En esta cláusula se describe la manera de transferir las TDD y los datos correspondientes entre las LA y las CA.

Para transferir las TDD y los datos correspondientes, esta Recomendación define un método de intercambio abstracto entre las LA y las CA. Este método de intercambio puede realizarse por diferentes medios, que dependen en una mayor o menor medida de la plataforma utilizada.

Con el fin de ayudar a los desarrolladores de LA a escribir código portable, las dependencias con respecto a la plataforma se han encapsulado en un nivel funcional constituido por las funciones del método de intercambio básico (funciones BEM, *basic exchange method*).

En la Figura 9 se describe esta situación.

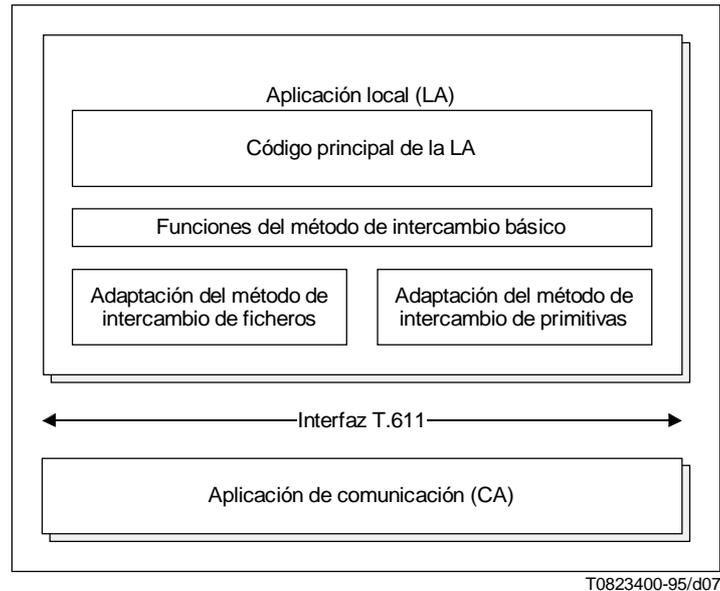


FIGURA 9/T.611

### Utilización de funciones del método de intercambio básico

Las funciones BEM presentan una interfaz funcional, que aísla el código de la LA de los métodos de intercambio reales que habrán de utilizarse.

Los métodos de intercambio reales definidos para las diversas plataformas son de dos tipos: de intercambio de ficheros, o de intercambio de primitivas. Estos dos tipos de método de intercambio se describen detalladamente en 7.3.

NOTA – Dado que sólo se requiere que el intercambio, en binario, de las TDD y los datos correspondientes se ajuste a esta Recomendación, la utilización de las funciones BEM dentro del código de la LA es facultativa. El único requisito que tiene que cumplir la LA es el de ajustarse al método de intercambio real definido para cada plataforma. Por consiguiente, un implementador LA puede decidir interactuar directamente con uno de los métodos de intercambio reales definidos.

#### 7.1 Visión de conjunto de las funciones del método de intercambio básico

NOTA – Las funciones del método de intercambio básico descritas en esta subcláusula replazan al método descrito en la versión 1992 de esta Recomendación. Para una información detallada sobre las diferencias entre ambos, véase el Anexo F, en el cual se ha incluido, por razones de conveniencia, la anterior versión de las funciones del método de intercambio básico.

Son posibles diferentes implementaciones del método de intercambio básico. Sin embargo, todas ellas deberán tener en común las funciones del método de intercambio básico mostradas en el Cuadro 36<sup>5)</sup>.

<sup>5)</sup> Para diferenciar el conjunto de funciones del método de intercambio básico (BEM) definidas en esta Recomendación del conjunto definido en la versión de 1992, los nombres de las funciones del método de intercambio básico (BEM) se escriben comenzando con una "E" (mayúscula).

**Resumen de las funciones del método de intercambio**

Función BEM	Finalidad
ELogin	LA abre un canal de comunicación entre LA y CA
ELogout	LA cierra el canal de comunicación abierto con Elogin ( )
EPutTDD	LA traspa una TDD a una CA
EPutData	LA traspa datos referentes a una TDD determinada
EPollTDD	LA pregunta a CA si está o no disponible una TDD de respuesta
EGetTDD	LA obtiene una TDD de respuesta de la CA
EGetData	LA obtiene datos referentes a una TDD determinada
ESetAlarm	LA establece una alarma dentro de la CA. La CA "despertará" por tanto a la LA cuando se produzcan determinados eventos
EAbortData	LA aborta la transferencia de datos
NOTA – El método de intercambio básico ha sido diseñado para que trabaje con diversas configuraciones, por ejemplo con la LA y la CA formando parte del mismo equipo, o con la CA siendo un servidor de comunicación en una LAN. Se parte de un solo supuesto: el ICE debe ser accesible desde cada LA.	

En general, las funciones del método de intercambio básico se basan en un procedimiento de enganche (ELogin) que devuelve un identificador de conexión. Este identificador de conexión se utiliza entonces en todas las invocaciones subsiguientes de las funciones del método de intercambio básico. El procedimiento de login es comparable a la apertura de un canal de interacción entre una LA y una CA.

**7.1.1 Secuencia de las funciones del método de intercambio básico**

Lo primero que tiene que hacer una LA es enganchar (función ELogin) a la CA solicitada. No puede efectuarse ningún intercambio de TDD antes de concluir el procedimiento de enganche. Por su propia naturaleza, el procedimiento de enganche es un mecanismo síncrono, lo que significa que el enganche requiere una respuesta para que pueda ejecutarse otra acción.

Cuando una LA desea enviar un TDD de petición a una CA, procederá con arreglo a los pasos siguientes:

- construye la TDD de petición (por cualquier medio apropiado);
- invoca la función EPutTDD;
- invoca la función EPutData, si hay que transferir datos correspondientes.

Cuando una LA necesita informarse de posibles eventos que le están destinados, deberá:

- invocar la función EPollTDD.

Cuando una LA necesita recuperar una respuesta TDD (la anterior función EPollTDD indicaba que había disponibles TDD de respuesta), deberá:

- invocar la función EGetTDD;
- invocar la función EGetData, si hay que recuperar datos correspondientes.

Cuando una LA recibe una alarma de la CA (sólo es aplicable si la CA soporta alarmas indicadas en el ICE (véase 9) y si la LA implementa la función de invocación de alarma), deberá:

- invocar la función EPutTDD;
- invocar la función EGetTDD;
- invocar la función EGetData, si hay que recuperar datos correspondientes.

Cuando una LA ya no necesita mantener el diálogo con una CA, deberá desenganchar (función ELogout) de esa CA. La CA conoce, de esta forma, que se ha cortado el trayecto de comunicaciones LA-CA y dejará por tanto utilizar el mecanismo de alarma.

Las funciones EPutTDD, EPollTDD, EGetTDD, ESetAlarm, ECallbackRoutine, ELogin y ELogout son síncronas, lo que significa que la LA sólo podrá continuar la operación cuando dichas funciones hayan sido retornadas. Las funciones y sus parámetros de llamada y de retorno se muestran en 7.2.

La Figura 10 explica el comportamiento antes descrito.

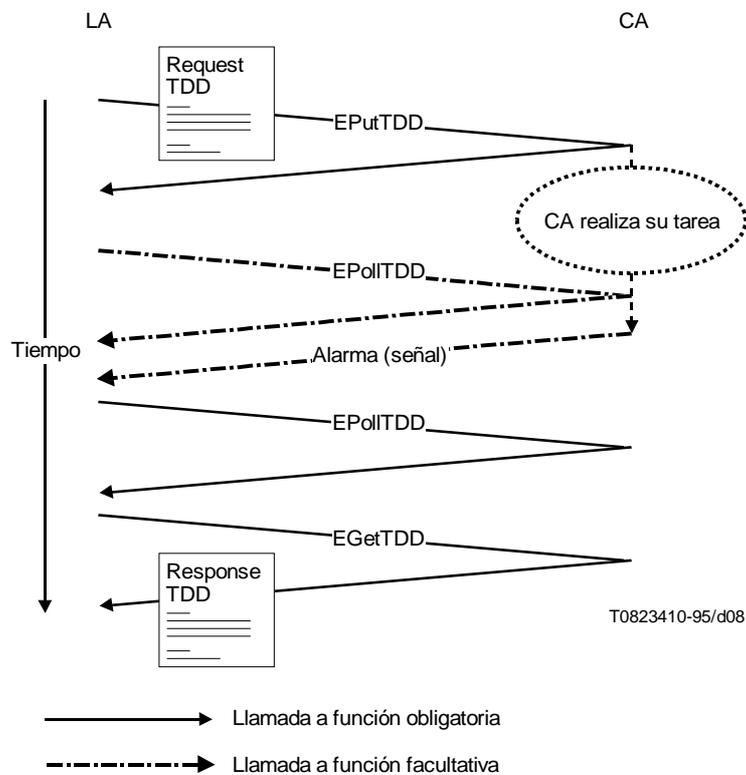


FIGURA 10/T.611

**Ejemplo de secuencias de funciones del método de intercambio básico**

Se supone que la LA ya ha enganchado la CA.  
 No se muestran las funciones de transferencia de datos

Se supone que la LA ya ha enganchado la CA. No se muestran las funciones de transferencia de datos.

**7.1.2 Soporte de alarmas**

Si la CA tiene la característica facultativa de alarmas significa que puede emplear la función ESetAlarm. Esta función permite a la CA despertar a una LA dada cuando se producen determinados eventos de alarmas.

La característica de soporte de alarma se indicará en el ICE (véase la cláusula 9).

### 7.1.3 Identificador de conexión

Con el fin de identificar un trayecto de comunicaciones LA-CA, se define el identificador de conexión (Connection-ID). El identificador de conexión lo calcula la CA al invocarse la petición de Login. La LA utilizará este identificador durante todo el intercambio con la CA, hasta que desenganche.

NOTA – El identificador de conexión (Connection-ID) es diferente del COM-ID: el COM-ID identifica eventos de comunicación que se producen en una CA.

### 7.1.4 Identificador de CA (CA-ID)

El identificador CA-ID identifica a una CA. Este identificador permite que una LA tenga interacciones simultáneas con múltiples LA. Por tanto, el CA-ID es un parámetro utilizado en cada llamada de función del método de intercambio.

## 7.2 Funciones del método de intercambio básico

Las funciones descritas a continuación remplazan o suplementan las funciones del mecanismo de intercambio básico de la versión 1992 de esta Recomendación (véase también el Anexo F). Para diferenciarlas, los nombres de las funciones definidas en esta Recomendación comienzan con la letra mayúscula "E".

Todas las funciones se ejecutan síncronamente, es decir, el llamante no puede continuar la ejecución hasta que la función llamada retorna.

Dado que la aplicación de las funciones BEM depende de la plataforma, se describen de una manera genérica en las subcláusulas que siguen. La implementación de las funciones para las diversas plataformas se describe en la Parte IV de esta Recomendación.

NOTA – El orden en que aparecen los parámetros en la siguiente descripción de las funciones BEM es importante en lo que respecta a la compatibilidad.

Los tipos de datos utilizados para la descripción genérica de las funciones BEM se definen en el Cuadro 37. En las cláusulas apropiadas de la Parte IV de esta Recomendación, estos tipos de datos se hacen corresponder con tipos de datos reales.

La clasificación de sentido "entrada" significa "el parámetro estará presente en la llamada de función", y "salida" significa "parámetro válido al retorno de función".

CUADRO 37/T.611

### Tipos de datos utilizados para la descripción de funciones BEM

Tipo de datos	Explicación
String (cadena)	Representa una cadena de caracteres
Integer (entero)	Representa un número comprendido en la gama mínima de $+2^{15} - (2^{15} - 1)$
Boolean (booleano)	Representa una variable que toma el valor, verdadero ("true") o falso ("false")
Memory address (dirección memoria)	Representa una dirección en la memoria de la LA

### 7.2.1 Función ELogin

La función ELogin será soportada por la CA. La LA deberá invocarla antes de cualquier intercambio LA-CA de TDD de petición y respuesta.

### 7.2.1.1 Finalidad

La función Elogin devuelve a la LA un identificador de conexión que se utilizará durante toda la interacción LA-CA, hasta que la LA se desengancha.

La función Elogin permite además seleccionar una CA según criterios proporcionados por el usuario, y retorna un CA-ID que se utilizará en llamadas subsiguientes de la función BEM para dirigirse a la CA seleccionada.

La selección de una CA se efectúa a través de la especificación de una cadena que contiene palabras clave como "FAX", "ECM" o "EMAIL", separadas por espacios en el parámetro selector. Teniendo en cuenta que esto puede efectuarse por procesos externos, el mecanismo de selección está fuera del ámbito de esta Recomendación.

Si se especifica el parámetro selector, no se especificará el parámetro CA-ID; cuando la llamada de la función retorna, el parámetro CA-ID contiene un identificador de CA válido que se puede utilizar en otras llamadas de función BEM, o bien el parámetro CA-ID está vacío si ninguna CA satisface los criterios de selección. Si el parámetro selector no está especificado en la llamada de función, no se efectuará ninguna selección. En este caso se especificará el parámetro CA-ID.

La función Elogin es el lugar en que una CA puede controlar el acceso de una LA a dicha CA. Esto puede hacerse comprobando el nombre de enganche (Login-name) y la palabra clave dados por la LA. Sin embargo, la medida en que se efectúa el control de los derechos de acceso, es una cuestión que incumbe a la realización de la CA.

### 7.2.1.2 Comportamiento

La CA comprueba los parámetros de la llamada de ELogin. Si concuerdan, genera un identificador de conexión que la LA utilizará subsiguientemente en otras llamadas de funciones BEM. Además, si el parámetro selector especifica criterios, la función retorna el CA-ID de una CA que cumple los criterios. Si el identificador de conexión retornado está puesto a cero, ello significa que la CA fracasó en la conexión con la LA.

### 7.2.1.3 Parámetros

Véase el Cuadro 38.

CUADRO 38/T.611

#### Parámetros de la función ELogin

Parámetro	Tipo de datos	Observaciones	Sentido
Login-name (nombre de login)	String (cadena)	Nombre del usuario LA que conecta con la CA (es diferente del LA-ID)	Entrada
Password (contraseña)	String (cadena)	Una cadena empleada como contraseña	Entrada
Selector	String (cadena)	Una cadena, proporcionada por el usuario, que especifica criterios de conexión	Entrada
CA-ID	Integer (entero)	Indicador de la CA que acepta la conexión	Salida
Connection-ID (identificador de conexión)	Integer (entero)	Indicador (handle) del canal de datos LA-CA. La CA lo retorna si la función ELogin tiene éxito. En caso contrario (es decir, si fracasa la identificación), la CA pone su valor a cero y se da un código de retorno apropiado por medio del parámetro Status (véase más abajo)	Salida
Status (situación)	Integer (entero)	Código de retorno; el valor cero indica éxito en la operación	Salida

## 7.2.2 Función EPutTDD

La función EPutTDD deberá ser soportada por la CA. Puede ser invocada por la LA. La EPutTDD admite la identificación de los ficheros de datos que habrán de transferirse a la CA.

Estos ficheros de datos son por lo general ficheros de listas de direcciones, o documentos, o ambas cosas.

### 7.2.2.1 Finalidad

La función EPutTDD lleva una TDD de petición a la CA y permite a la LA especificar los ficheros que deben transferirse a la CA. La transferencia propiamente dicha de ficheros de datos a la CA se efectúa con la función EPutData.

### 7.2.2.2 Comportamiento

La CA copia la TDD de petición transportada por la función EPutTDD en sus estructuras internas. El resultado se informa inmediatamente a la LA solicitante. Si la LA solicita un identificador de datos especificando una dirección de identificador de datos (Data-ID), la CA retorna el identificador de un grupo de datos como Data-ID.

Si este identificador es NULL, no se transmiten a la CA ficheros de datos (ni memorias tampón). Si el identificador de datos tiene un valor distinto de NULL, la LA efectúa las transferencias subsiguientes de ficheros de datos con la función EPutData. Las transferencias se efectuarán inmediatamente después de la llamada a la función EPutTDD.

NOTA – El Data-ID devuelto puede ser válido sólo durante un tiempo limitado.

### 7.2.2.3 Parámetros

Véase el Cuadro 39.

CUADRO 39/T.611

#### Parámetros de la función EPutTDD

Parámetro	Tipo de datos	Observaciones	Sentido
Connection-ID (identificador de conexión)	Integer (entero)	Indicador (handle) del canal de datos LA-CA retornado por la función ELogin	Entrada
CA-ID	Integer (entero)	Identificador de la CA retornado por la función ELogin	Entrada
TDD-location (ubicación de la TDD)	Memory address (dirección memoria)	Ubicación de la TDD, en la memoria, pasada a la CA. Después del retorno de la función, la TDD de la LA puede suprimirse o utilizarse para otros fines	Entrada
TDD-size (tamaño de TDD)	Integer (entero)	Tamaño de la TDD, en octetos, pasado a la CA	Entrada
Data-ID (identificador de datos)	Integer (entero)	Identificador del grupo de datos, retornado por la CA si se solicita	Entrada/salida
Status (situación)	Integer (entero)	Código de retorno; el valor cero indica éxito en la operación	Salida

## 7.2.3 Función EPutData

Esta función permite la transferencia real de datos (lista de direcciones y/o ficheros de datos) de la LA a la CA. En las distintas implementaciones, los datos pueden ser ficheros o memorias tampón.

### 7.2.3.1 Finalidad

El grupo de ficheros transferidos en una llamada a función está relacionado con una anterior TDD de petición transmitida a la CA por medio de una anterior llamada a la función EPutTDD.

El grupo se identifica por el identificador de datos que la CA retornó como resultado de una anterior llamada de la función EPutTDD.

### 7.2.3.2 Comportamiento

Esta función la invoca la LA después de haber enviado la correspondiente TDD de petición a la CA.

La LA construirá una estructura de datos (Data-Descriptor) que indica el lugar en que se encuentran los ficheros/memorias tampón que se transportarán a la CA. La especificación de la estructura de los datos depende del idioma utilizado, pero la compatibilidad cruzada de las plataformas asegura que la organización binaria no dependa de las plataformas ni de los idiomas.

La LA puede transferir todos los datos de una sola vez o en múltiples llamadas. El parámetro *Next* (siguiente) indica si la llamada a función es la última en el grupo de datos o si seguirán más llamadas a funciones relativas a ese grupo de datos.

### 7.2.3.3 Parámetros

Véase el Cuadro 40.

CUADRO 40/T.611

#### Parámetros de la función EPutData

Parámetro	Tipo de datos	Observaciones	Sentido
Connection-ID (identificador de conexión)	Integer (entero)	Indicador (handle) del canal de datos LA-CA	Entrada
CA-ID	Integer (entero)	Identificador de la CA	Entrada
Data-ID (identificador de datos)	Integer (entero)	Identificador del grupo de datos, retornado por la CA	Entrada
Data-Descriptor (descriptor de datos)	Memory address (dirección memoria)	Dirección memoria de una estructura de datos que especifica los ficheros/memorias tampón que habrán de transportarse a la CA	Entrada
Next (siguiente)	Boolean (booleano)	Indicador que informa si hay más datos para transportar en la siguiente llamada a la función EPutData, o si ya no hay más datos para transportar (fin de grupo de datos)	Entrada
Status (situación)	Integer (entero)	Código de retorno; el valor cero indica éxito en la operación	Salida

### 7.2.4 Función EPollTDD

La función EPollTDD pregunta a la CA cuántas TDD de respuesta están en espera de ser tratadas por la LA solicitante. La EPollTDD devuelve el número de TDD de respuesta pendientes y el tipo y tamaño de la primera TDD de respuesta que será retornada por la próxima llamada a la función EGetTDD.

Además, la función EPollTDD permite seleccionar el tipo de TDD que habrá de devolver la CA.

### 7.2.4.1 Finalidad

La función EPollTDD permite interrogar a la CA para saber cuántas TDD de respuesta están pendientes de recuperación.

### 7.2.4.2 Comportamiento

Esta función permite a la LA seleccionar el tipo de TDD de respuesta que debe obtenerse por interrogación, lo que se hace insertando un determinado parámetro de tipo TDD. Si la LA no especifica ningún tipo de TDD (TDD-type puesto a cero), la función selecciona cualquier tipo de TDD, si está disponible, y devuelve el tipo seleccionado. La TDD seleccionada, en su caso, puede ser entonces recuperada por una llamada subsiguiente a la función EGetTDD.

La función EPollTDD retorna también el tamaño de la TDD de respuesta que se devolverá, si la hubiere. La CA debe atribuir entonces suficiente espacio de almacenamiento para recibir la TDD de respuesta devuelta por la llamada siguiente a la función EGetTDD. El tamaño dado por la CA es una indicación del tamaño mínimo de la memoria tampón que retendrá la TDD de respuesta.

Asimismo, se devuelve la cantidad de TDD de respuesta pendientes del tipo seleccionado. Si la cuenta devuelta es cero, no están pendientes TDD de respuesta del tipo seleccionado.

Cuando la CA tiene muchas TDD de respuesta disponibles, elige la que ha de devolver primero. Esta TDD de respuesta es la TDD que se transmitirá a la LA en la siguiente llamada a la función EGetTDD emitida por la misma LA.

Cuando no hay ninguna TDD de respuesta disponible para la LA solicitante, la cuenta de TDD se pone al valor cero, en cuyo caso el tamaño de TDD devuelto se pone también a cero.

Si la TDD depositada es errónea o desconocida, por ejemplo, si le falta el encabezamiento (<TDD Header>), la función puede devolver la TDD de petición original. En este caso, la CA pondrá a cero el tipo de TDD devuelto y pondrá la situación de retorno a un valor distinto de cero.

Los valores de tipo TDD asignados se muestran en el Cuadro 41.

CUADRO 41/T.611

#### Asignación de tipos de TDD

TDD de respuesta	Tipo de TDD (TDD-type)	TDD de respuesta	Tipo de TDD (TDD-type)
Ningún tipo de TDD <sup>a)</sup>	00 <sub>hex</sub>	DISPATCH Response (respuesta de despacho)	35 <sub>hex</sub>
SENDACK Response (respuesta de acuse de recibo de envío)	10 <sub>hex</sub>	PREVIEW Response (respuesta de visión previa)	36 <sub>hex</sub>
RECEIVE Response (respuesta de recepción)	20 <sub>hex</sub>	PRINT Response (respuesta de impresión)	40 <sub>hex</sub>
COPY Response (respuesta de copia)	30 <sub>hex</sub>	CONVERT Response (respuesta de conversión)	41 <sub>hex</sub>
DELETE Response (respuesta de supresión)	31 <sub>hex</sub>	CHECK Response (respuesta de comprobación)	42 <sub>hex</sub>
CANCEL Response (respuesta de cancelación)	32 <sub>hex</sub>	EXTEND Response (respuesta de ampliación)	50 <sub>hex</sub>
PURGE Response (respuesta de purga)	33 <sub>hex</sub>	NATIONAL Response (respuesta nacional)	60 <sub>hex</sub>
RESCHEDULE Response (respuesta de reprogramación)	34 <sub>hex</sub>	PRIVATE Response (respuesta de privado)	70 <sub>hex</sub>
<sup>a)</sup> Cuando se utiliza como selector (en entrada), 00 <sub>hex</sub> representa "todos los tipos de TDD"; como valor de retorno representa "depositada una TDD desconocida o errónea, o no hay TDD de respuesta disponible".			

### 7.2.4.3 Parámetros

Véase el Cuadro 42.

CUADRO 42/T.611

#### Parámetros de la función EPollTDD

Parámetro	Tipo de datos	Observaciones	Sentido
Connection-ID (identificador de conexión)	Integer (entero)	Indicador (handle) del canal de datos LA-CA retornado por la función Elogin	Entrada
CA-ID	Integer (entero)	Identificador de la CA	Entrada
TDD-type (tipo de datos)	Integer (entero)	Selecciona/retorna el tipo de TDD de respuesta esperada/disponible	Entrada/salida
TDD-size (tamaño de TDD)	Integer (entero)	Tamaño de la siguiente TDD de respuesta lista para ser recuperada	Salida
TDD-count (cuenta de TDD)	Integer (entero)	Cuenta de las TDD de respuesta pendientes. El valor cero indica que no hay respuestas pendientes	Salida
Status (situación)	Integer (entero)	Código de retorno; el valor cero indica éxito en la operación	Salida

### 7.2.5 Función EGetTDD

La función EGetTDD deberá ser soportada por la CA. Puede ser invocada por la LA. EGetTDD permite recuperar una TDD de respuesta y obtener un identificador de grupo de datos para recuperar los ficheros de datos/memorias tampón correspondientes.

#### 7.2.5.1 Finalidad

La función EGetTDD permite a la LA recuperar una TDD de respuesta pasándola a la memoria y obtener un indicador (handle) que señale un grupo de ficheros de datos/memorias tampón. Este indicador es útil para todos los tipos de TDD que comprenden ficheros de datos/memorias tampón, por ejemplo recepción o rastreo:COPIA.

#### 7.2.5.2 Comportamiento

La LA especifica la dirección de la memoria en la que la CA copiará una TDD de respuesta que está disponible para dicha LA.

La CA devolverá a la LA la TDD de respuesta que había sido calificada por la anterior función EPollTDD emitida por la misma LA. La LA tendrá preparada, en sus estructuras internas, una zona de TDD de respuesta del recipiente. La invocación de una función EGetTDD por la LA estará siempre precedida por una llamada a la función EPollTDD. En la llamada, la LA prefijará el parámetro de tamaño de TDD al tamaño de su zona de respuesta. De retorno, el parámetro tamaño de TDD mantendrá el tamaño de la TDD de respuesta suministrado por la CA.

Si la LA invoca dos o más funciones EGetTDD consecutivas (sin una llamada intermedia a la función EPollTDD) la CA puede devolver información inválida.

La función EGetTDD devuelve también el identificador de datos (Data-ID) como un indicador (handle) para recibir los ficheros de datos correspondientes (con una llamada subsiguiente a la función EGetData).

Si el valor del parámetro identificador de datos es distinto de cero, los ficheros de datos deben ser recuperados por una llamada subsiguiente a la función EGetData. Si el valor del parámetro identificador de datos es cero, no hay más datos de la CA para recuperar.

### 7.2.5.3 Parámetros

Véase el Cuadro 43.

CUADRO 43/T.611

#### Parámetros de la función EGetTDD

Parámetro	Tipo de datos	Observaciones	Sentido
Connection-ID (identificador de conexión)	Integer (entero)	Indicador (handle) del canal de datos LA-CA retornado por la función Elogin	Entrada
CA-ID	Integer (entero)	Identificador de la CA	Entrada
TDD-location (ubicación de TDD)	Memory address (dirección memoria)	Especifica el lugar en que se almacenará la TDD de respuesta	Entrada
TDD-size (tamaño de TDD)	Integer (entero)	Tamaño de la memoria tampón reservada para almacenar la TDD de respuesta	Entrada/salida
Data-ID (identificador de datos)	Integer (entero)	Identificador del grupo de datos, si existe	Salida
Status (situación)	Integer (entero)	Código de retorno; el valor cero indica éxito en la operación	Salida

### 7.2.6 Función EGetData

Esta función permite la transferencia de ficheros de datos (o memorias tampón) de la CA a la LA. La invoca la LA.

#### 7.2.6.1 Finalidad

Esta función se utiliza para recuperar los datos de la CA después de recibirse una TDD de respuesta (solicitada por la función EGetTDD) que especifica que existen datos por recuperar (el parámetro identificador de datos tiene un valor distinto de cero).

#### 7.2.6.2 Comportamiento

Inmediatamente después de ejecutada la función EGetTDD, la LA llamará la función EGetData para recibir los datos correspondientes (ficheros o memorias tampón). La CA retorna un descriptor de datos que contiene información sobre los datos recibidos. La CA indica también, en el valor retornado, si la LA deberá o no efectuar más llamadas a la función EGetTDD para recuperar ulteriores datos pertinentes.

El grupo de ficheros transferidos dentro de una llamada a esta función se relaciona con una sola TDD de respuesta (obtenida por una llamada a la función EGetTDD). Este grupo se identifica por el parámetro identificador de datos (proporcionado por la LA).

### 7.2.6.3 Parámetros

Véase el Cuadro 44.

CUADRO 44/T.611

#### Parámetros de la función EGetData

Parámetro	Tipo de datos	Observaciones	Sentido
Connection-ID (identificador de conexión)	Integer (entero)	Indicador (handle) del canal de datos LA-CA retornado por la función Elogin	Entrada
CA-ID	Integer (entero)	Identificador de la CA	Entrada
Data-ID (identificador de datos)	Integer (entero)	Identificador del grupo de datos	Entrada
Data-Descriptor (descriptor de datos)	Memory address (dirección memoria)	Almacena la dirección de un descriptor para los datos que van a ser recuperados, pasados a la CA en entrada, e insertados por la CA en salida	Entrada/salida
Next (siguiente)	Boolean (booleano)	Indica si se deberá o no recuperar más datos	Salida
Status (situación)	Integer (entero)	Código de retorno; el valor cero indica éxito en la operación	Salida

### 7.2.7 Función ESetAlarm

La función ESetAlarm puede ser soportada facultativamente por la CA. En este caso, la LA podrá invocarla si así lo desea. Si LA opta por utilizar la función ESetAlarm, deberá suministrar un manejador de alarmas, AlarmHandler. El AlarmHandler es una función que recibe las alarmas subsiguientes generadas por la CA.

Una CA deberá declarar el soporte de la función ESetAlarm en el ICE.

#### 7.2.7.1 Finalidad

Esta función permite a la LA registrar en la CA conectada el tipo de eventos de alarma que ella soporta. No se invocará esta función si la CA no entiende el mecanismo de alarma. La información que dice si la CA entiende o no el mecanismo de alarma se obtiene del elemento ALARM del ICE (véase la cláusula 9).

La función ESetAlarm informa a la CA que puede dar una alarma a la LA invocando el manejador de alarmas suministrado.

La LA puede inhabilitar temporalmente un evento de alarma inhabilitando la dirección correspondiente del manejador de alarmas, por ejemplo poniéndola a NULL. La restauración del evento de alarma para ese tipo de alarma se obtiene habilitando la dirección del manejador de alarmas, por ejemplo fijando como dirección el punto de entrada real del manejador de alarmas (AlarmHandler).

Los eventos de alarma definidos se indican en el Cuadro 45.

#### 7.2.7.2 Comportamiento

La CA que soporta la función ESetAlarm registrará la ubicación de la función AlarmHandler asignada por la LA. Como mínimo, la CA puede registrar tantas ubicaciones de AlarmHandler como haya LA diferentes enganchadas (logged-in). Si se produce un evento registrado con relación a una determinada LA, la CA enviará la alarma a esta LA invocando su manejador de alarmas.

CUADRO 45/T.611

**Eventos de alarma para la función ESetAlarm**

Nombre del evento	Observaciones
ASYNCR_RESPONSES	La LA no interrogará a la CA. La CA debe enviar una alarma cada vez que una TDD de respuesta esté disponible
QUEUE_FULL	La CA no aceptará ulteriores TDD, a menos que sea interrogada
DOCUMENT_RECEIVED	Puede recibirse un nuevo documento
CONNECTION_LOST	La conexión LA-CA se ha perdido
SEND_SUCCESS	Se envió con éxito un documento
SEND_FAILED	Fracasó el envío de un documento
CORRUPTED_TDD	La CA recibió una TDD corrompida/irreconocible
SEND_EVENT	Se produjo un evento de envío en la CA
RECEIVE_EVENT	Se produjo un evento de recepción en la CA
CA_WILL_STOP	La CA dejará de procesar peticiones. Efectuar Logout inmediatamente
ALARMS_UNAVAILABLE	Las alarmas ya no disponibles
TDD_RESP_AVAILABLE	Está disponible una respuesta de TDD. La LA debe interrogar a la CA cuanto antes
NOTA – Algunas alarmas necesitan datos adicionales para dar una información precisa sobre el evento.	

**7.2.7.3 Parámetros**

Véase el Cuadro 46.

CUADRO 46/T.611

**Parámetros de la función ESetAlarm**

Parámetro	Tipo de datos	Observaciones	Sentido
Connection-ID (identificador de conexión)	Integer (entero)	Indicador (handle) del canal de datos LA-CA retornado por la función Elogin	Entrada
CA-ID	Integer (entero)	Identificador de la CA	Entrada
Alarm-event (evento de alarma)	Integer (entero)	Combinación de los tipos de alarmas a que puede reaccionar la LA	Entrada
Alarm-Handler (manejador de alarmas)	Memory address (Dirección memoria)	Información sobre el punto de entrada del manejador de alarmas. Este punto de entrada será llamado por la CA cuando se produce uno de los eventos registrados  Este parámetro, al igual que el manejador de alarmas propiamente dicho, depende de la plataforma	Entrada
Status (situación)	Integer (entero)	Código de retorno; el valor cero indica éxito en la operación	Salida

#### **7.2.7.4 Función AlarmHandler**

De acuerdo con la plataforma, la alarma activada puede tener parámetros que detallan la razón de la alarma.

La función AlarmHandler define un mecanismo que permite a la CA avisar a la LA que están disponibles algunas TDD de respuesta. La utilización de este mecanismo facultativo puede mejorar el control de flujo entre las LA y las CA en sistemas muy cargados.

La función AlarmHandler es realizada por la LA. La CA llama a esta función cuando necesita activar una alarma. Las alarmas que la CA puede activar están especificadas por la LA mediante la función ESetAlarm.

Algunas alarmas pueden tener parámetros, que se pasan a la LA junto con la llamada de función.

Algunas alarmas entrañan que la LA interroga a la CA después de haber recibido la alarma (por ejemplo, cuando la CA ha recibido un nuevo documento). Si la LA no interroga a la CA con suficiente rapidez, la CA puede activar la alarma repetidamente hasta que la LA efectúe la acción esperada.

Como la función AlarmHandler depende de la plataforma, se describe con más detalle en las cláusulas pertinentes de la Parte IV de esta Recomendación.

#### **7.2.8 Función EAbortData**

La función EAbortData permite cancelar la transferencia de datos en curso. Tiene por efecto cancelar también la TDD correspondiente.

##### **7.2.8.1 Finalidad**

Esta función es invocada por la LA para cancelar una transferencia de datos iniciada por la función EPutData o EGetData.

##### **7.2.8.2 Comportamiento**

Cuando se aborta una transferencia de datos iniciada por la función EPutData, la CA destruirá la correspondiente TDD de petición que fue transportada por la anterior llamada a la función EPutTDD.

Cuando se aborta una transferencia de datos iniciada por la función EGetData, la CA retendrá la correspondiente TDD de respuesta que fue recuperada por la llamada a la función EGetTDD anterior.

El aborto de las transferencias de datos en cualquier sentido de transmisión puede producirse solamente entre dos llamadas a las funciones EGetData o EPutData, respectivamente.

Como la LA es responsable de la transferencia de datos entre la LA y la CA, la llamada a EAbortData después del último bloque de datos no tiene efecto (de todas maneras la transferencia de datos se completa). Por la misma razón la llamada a EAbortData antes del primer bloque de datos tiene por efecto cancelar toda la transferencia de datos y la TDD correspondiente.

##### **7.2.8.3 Parámetros**

Véase el Cuadro 47.

#### **7.2.9 Función ELogout**

La función ELogout deberá ser soportada por la CA. La invocará la LA cuando haya concluido el intercambio de TDD de petición y respuesta entre las LA y la CA.

##### **7.2.9.1 Finalidad**

La función ELogout retorna a la LA un status que indica si la interacción LA-CA ha terminado ordenadamente.

##### **7.2.9.2 Comportamiento**

Antes de concluir el diálogo LA-CA, la CA puede (pero no está obligada a) procesar todas las TDD de petición pendientes que fueron emitidas por esa LA.

CUADRO 47/T.611

**Parámetros de la función EAbortData**

Parámetro	Tipo de datos	Observaciones	Sentido
Connection-ID (identificador de conexión)	Integer (entero)	Indicador (handle) del canal de datos LA-CA retornado por la función Elogin	Entrada
CA-ID	Integer (entero)	Identificador de la CA	Entrada
Data-ID (identificador de datos)	Integer (entero)	Identificador de grupo de datos	Entrada
Status (situación)	Integer (entero)	Código de retorno; el valor cero indica éxito en la operación	Salida

**7.2.9.3 Parámetros**

Véase el Cuadro 48.

CUADRO 48/T.611

**Parámetros de la función ELogout**

Parámetro	Tipo de datos	Observaciones	Sentido
Connection-ID (identificador de conexión)	Integer (entero)	Indicador (handle) del canal de datos LA-CA retornado por la función Elogin	Entrada
CA-ID	Integer (entero)	Identificador de la CA	Entrada
Status (situación)	Integer (entero)	Código de retorno; el valor cero indica éxito en la operación	Salida

**7.3 Ejecución de las funciones del método de intercambio básico**

Las funciones del método de intercambio básico ofrecen una interfaz funcional que es necesario adaptar a la plataforma subyacente de una manera bien definida para proporcionar una interfaz compatible en binario.

Con este fin, la presente Recomendación define dos mecanismos de intercambio reales:

- 1) el método de intercambio de primitivas, y
- 2) el método de intercambio de ficheros.

El método de intercambio de primitivas es rápido, pero depende de la plataforma; el método de intercambio de ficheros es mucho menos dependiente de la plataforma, pero es más lento y no permite todas las características ofrecidas por las funciones del método de intercambio básico.

Las plataformas pueden imponer el método de intercambio que admiten. Véase la Parte IV de esta Recomendación.

**7.3.1 Método de intercambio de primitivas**

Para aplicar el método de intercambio de primitivas se utilizan llamadas a funciones. La CA debe proporcionar un punto de entrada para cada función del método de intercambio básico. Para asegurar la compatibilidad en binario, la provisión de este punto de entrada deberá ajustarse a las definiciones establecidas en esta Recomendación para cada plataforma (véase la Parte IV de la presente Recomendación).

### 7.3.2 Método de intercambio de ficheros

Para aplicar el método de intercambio de ficheros es necesario que las TDD se intercambien por medio de ficheros. Por esta razón hay que acordar y compartir directorios de ficheros comunes entre la LA y la CA. Por consiguiente la CA declarará los trayectos completos para esos directorios en la sección pertinente del ICE.

Hay que configurar tres directorios de ficheros, o, en un nivel más abstracto, tres zonas:

- Una zona de entrada para las TDD transferidas de la LA a la CA (jobs confiados por la LA a la CA);
- Una zona para las respuestas TDD;
- Una zona para las "tareas" que no pudieron procesarse debido a errores sintácticos o de otro tipo. La CA retira las TDD incorrectas de la zona entrada y las coloca en esta zona.

Estas tres zonas se designan simbólicamente por COM\_JOB, COM\_ACK y COM\_ERR, respectivamente.

Puesto que estas zonas son específicas de la realización y de la instalación, los trayectos completos a dichas zonas podrán ser configurados en la instalación de la CA.

Si una CA tiene que soportar múltiples conexiones de LA, se sugiere que dicha CA establezca tres zonas (o directorios) "madre", como se ha descrito anteriormente, y cree una subzona (subdirectorío) dentro de cada zona "madre" para cada LA a que pueda conectarse. El nombre de cada subzona puede obtenerse del LA-ID de las LA que están conectadas.

La adaptación de las funciones del método de intercambio básico al método de intercambio de ficheros es bastante compleja, ya que el adaptador tiene que emular las funciones del método de intercambio básico.

NOTA – Una realización de LA puede también (aunque ello no se aconseja en esta Recomendación) interactuar con el método de intercambio de ficheros directamente, sin utilizar funciones del método de intercambio básico. Este procedimiento tiene el inconveniente de hacer más difícil la portabilidad a otras plataformas basadas en primitivas.

#### 7.3.2.1 Transferencia de TDD

Para emitir una petición de función, la LA coloca una TDD como un fichero en la zona COM\_JOB.

Algunas implementaciones, en particular las sustentadas sobre plataformas que sólo pueden realizar una tarea en cada momento, requieren también que la LA ejecute un programa SYNC. SYNC es una forma abreviada de la palabra inglesa "synchronization" (sincronización). El programa SYNC asegura que la CA reciba el control y pueda procesar las TDD. En el descriptor de CA de la CA correspondiente se indica si es necesario o no ejecutar un programa SYNC.

Cuando una CA ha terminado de procesar la TDD, coloca la respuesta TDD en la zona de respuesta COM\_ACK y suprime la petición TDD en la zona de entrada COM\_JOB. La respuesta TDD colocada por la CA tiene el mismo nombre que la anterior petición TDD.

El ciclo completo de la transferencia de "tarea" se muestra en la Figura 11.

NOTA – No se garantiza que las respuestas TDD se proporcionen en la misma secuencia que las TDD de petición. El procesamiento no es necesariamente secuencial.

#### 7.3.2.2 Tratamiento de errores

Las peticiones TDD que la CA no puede procesar (debido a errores sintácticos, etc.) se copian en la zona COM\_ERR y se retiran de la zona COM\_JOB. Se inserta la palabra clave FATAL al final del sobre ("envelope") y se genera un mensaje de error con la forma [Num/Text], que se inserta en el campo de parámetro agregado.

#### 7.3.2.3 Mecanismo de sincronización (SYNC)

Si la CA tiene que ejecutar un programa de sincronización (SYNC) la LA actuará como sigue:

- Después de almacenada una petición TDD, la LA enviará un SYNC (señal de sincronización) a la CA (mediante la función "Sync\_Job").
- Antes de que la LA interroge para obtener una respuesta TDD, deberá enviarse otro SYNC a la CA (mediante función "SyncAck").

Cuando se está empleando como sistema operativo DOS, OS/2, UNIX u otros sistemas operativos similares a UNIX, el mecanismo SYNC deberá implementarse ejecutando el programa "APPLI/COM" desde la LA.

Las funciones "SyncJob" y "SyncAck" (antes mencionadas) se codifican como sigue:

SyncJob: EXEC APPLI/COM JOB

SyncAck: EXEC APPLI/COM ACK

EXEC representa la instrucción específica de la plataforma para lanzar un programa ejecutable.

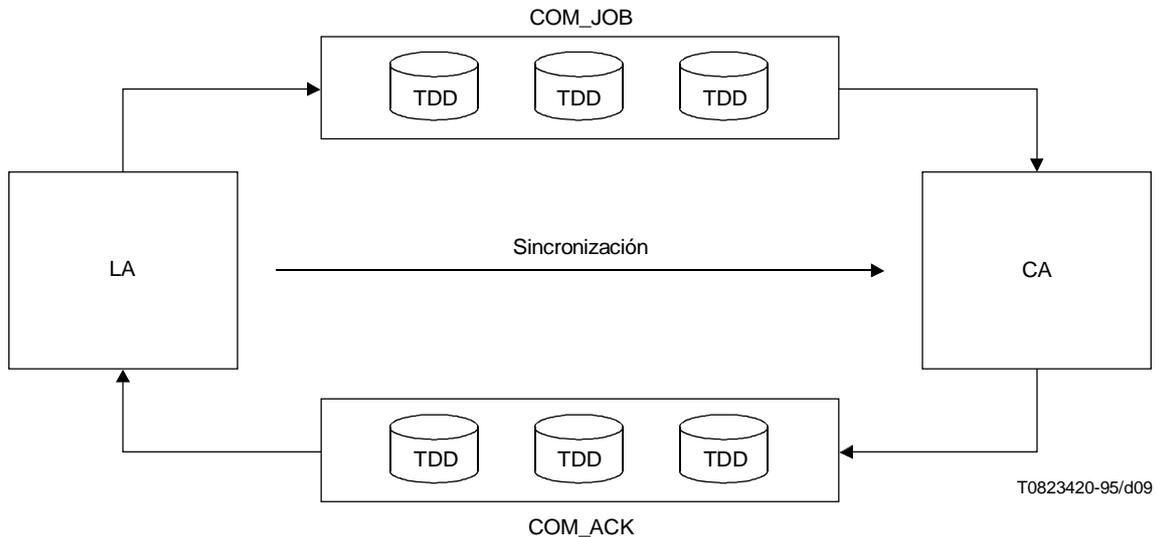


FIGURA 11/T.611

**Método de intercambio de ficheros: transferencia de TDD**

**8 Formatos de transferencia**

Para un funcionamiento correcto, el formato de los ficheros salientes y entrantes intercambiados localmente entre la LA y la CA debe cumplir ciertos requisitos. Los ficheros entrantes y salientes tienen un formato específico, que no es el mismo utilizado por los ficheros de transmisión, por ejemplo un fichero de tratamiento de texto que ha de intercambiarse en el modo básico del servicio teletax. En este caso se requieren conversiones de formato (que han de ser tratadas por la CA).

Para el intercambio de documentos entre la LA y la CA se definen en esta Recomendación varios "formatos de transferencia". Los formatos de transferencia se aplican a los "ficheros transferencia" definidos en 2.1. Estos formatos no deben confundirse con el formato transmitido a través de la red (formatos de transmisión) por los servicios de telecomunicaciones, ni con el formato utilizado por las TDD (codificación de TDD).

Son posibles varios formatos de transferencia:

- formatos de transferencia de textos;
- formatos de transferencia de gráficos;
- formatos de transferencia transparentes;
- formatos de transferencia privados, que pueden contener textos o gráficos.

Esta Recomendación define:

- tres formatos de textos (ASCII ampliado de APPLI/COM, ASCII estándar de APPLI/COM, y el formato teletex);
- un formato de transferencia de gráficos (TIFF de APPLI/COM);
- el formato de transferencia transparente.

El formato transparente puede utilizarse sólo si el documento se va a transmitir como un fichero binario por el servicio de telecomunicaciones. En este caso no se efectúa ninguna conversión del contenido del fichero, y éste se transmitirá inalterado.

Esta Recomendación admite el soporte de formatos de transferencia privados. Por consiguiente, los fabricantes de CA podrán utilizar otros formatos de transferencia (por ejemplo, postscript) para adaptar formatos originarios de programas de aplicación de uso difundido (por ejemplo, tratamientos de textos, bases de datos u hojas de cálculo). Para informar a la LA sobre este soporte, la CA colocará la palabra clave ADDCONV en el descriptor de CA del ICE (véase la cláusula 9).

Una CA que soporta un formato de transferencia adicional tiene que asegurarse de que los documentos se tratan correctamente para todos los servicios que ella soporta. Las restricciones impuestas por el propio servicio tienen que tratarse de una manera transparente. De todas formas, los fabricantes de las CA deberán describir en la documentación correspondiente los formatos de transferencia soportados por la CA.

El Cuadro 49 indica cómo los formatos de transferencia deberán ser soportados por la CA de acuerdo con el sistema operativo y con el servicio de telecomunicaciones ofrecido por la CA:

CUADRO 49/T.611

**Lista de los formatos de transferencia**

Formato de transferencia	Convert-id	Significa
ASCII ampliado de APPLI/COM	ASCII437	Formato de transferencia ASCII definido en 8.1.1. La utilización de este formato está limitada a algunos servicios. Véase 8.4
	ASCII	Formato de transferencia ASCII definido en la CA utilizada. Este formato de transferencia puede ser diferente del formato de transferencia ASCII437 empleado en los sistemas que no soportan el mismo juego de caracteres ASCII ampliado
ASCII estándar de APPLI/COM	T.50	Formato de transferencia T.50 (IRA) definido en 8.1.2. La utilización de este formato está limitada a algunos servicios. Véase 8.4
Formato teletex	T.61	Formato de transferencia T.61 definido en 8.2. La utilización de este formato está limitada a algunos servicios. Véase 8.4
TIFF de APPLI/COM	TIFF	Formato de transferencia TIFF definido en 8.3. La utilización de este formato está limitada a algunos servicios. Véase 8.4
Formato transparente	VOID	"No se efectúa conversión." La utilización está limitada a la transferencia transparente de documentos en diversos modos de "transferencia de ficheros" entre los servicios de telecomunicaciones

Los formatos de transferencia definidos en esta Recomendación podrán ser leídos y generados para los servicios apropiados, con los sistemas operativos apropiados, por las CA conformes.

NOTA 1 – Los formatos de transferencia de textos para el servicio telefax sólo son soportados en el sentido de salida.

Los documentos que se transfieran a la CA en un formato de textos deberán ser editados por la aplicación de manera que el formato y el juego de caracteres satisfagan los requisitos del servicio, para lo cual se fijará debidamente el número de "caracteres/línea", "líneas/página", "paso de caracteres", "espaciamiento de líneas", y atributos tales como "subrayado", "superíndice" y "subíndice".

La CA puede rechazar los documentos que empleen un formato o un juego de caracteres incorrectos.

NOTA 2 – Si una LA desea funcionar en forma independiente del servicio, puede utilizar para la transferencia de documentos uno de los formatos basados en ASCII, ya que éstos son los únicos formatos de transferencia que abarcan la mayoría de los servicios.

## 8.1 Formatos de transferencia ASCII ampliado y ASCII estándar de APPLI/COM

Los formatos de código mostrados en el Cuadro 50 se definen para los formatos de transferencia ASCII ampliado de APPLI/COM y ASCII estándar de APPLI/COM. No todos los servicios permiten todas las especificaciones de formato indicadas (véase 8.4). Se permite que la CA ignore las secuencias de escape (ESC). Además, según los tipos de escritura (en inglés "fonts") residentes utilizados para las conversiones de ASCII a T.4 (o T.6), se permite que la CA pase a la línea o a la página siguientes cuando sea necesarios, a menos que la LA haya inhabilitado explícitamente esta operación; en tal caso, la CA puede rechazar la conversión o efectuarla a un nivel inferior.

CUADRO 50/T.611

### Determinantes de formato para los formatos de transferencia ASCII de APPLI/COM

Formato	Valores posibles	Hex	ASCII	Por defecto
Orientación	Vertical (retrato)	1B 4F 30	ESC O 0	√
	Horizontal (paisaje)	1B 4F 31	ESC O 1	
Paso	Paso 10	1B 50 30	ESC P 0	√
	Paso 12	1B 50 31	ESC P 1	
	Paso 15	1B 50 32	ESC P 2	
Espaciamiento de líneas	6 líneas/pulgada (espaciamiento 1)	1B 4C 30	ESC L 0	√
	4 líneas/pulgada (espaciamiento 1,5)	1B 4C 31	ESC L 1	
	3 líneas/pulgada (espaciamiento 2)	1B 4C 32	ESC L 2	
	12 líneas/pulgada (espaciamiento 0,5)	1B 4C 33	ESC L 3	
Atributos	Subrayado desactivado	1B 55 30	ESC U 0	√
	Subrayado activado	1B 55 31	ESC U 1	
	Superíndice desactivado	1B 41 30	ESC A 0	√
	Superíndice activado	1B 41 31	ESC A 1	
	Subíndice desactivado	1B 56 30	ESC V 0	√
	Subíndice activado	1B 56 31	ESC V 1	
	Negrita desactivado	1B 42 30	ESC B 0	√
	Negrita activado	1B 42 31	ESC B 1	
	Tachado desactivado	1B 53 30	ESC S 0	√
	Tachado activado	1B 53 31	ESC S 1	
	Cursiva desactivado	1B 49 30	ESC I 0	√
Cursiva activado	1B 49 31	ESC I 1		
Composición del texto	Cambio de línea no admitido	1B 54 30	ESC T 0	
	Cambio de línea admitido	1B 54 31	ESC T 1	√
	Rotación de página no admitida	1B 52 30	ESC R 0	
	Rotación de página admitida	1B 52 31	ESC R 1	√
	Nueva línea	0D 0A	CR LF	
	Nueva página	0D 0C	CR FF	
Selección de tipo de escritura	Selección del tipo número n	1B 43 'n'	ESC C 'n'	'n' = 0

NOTA – Los tipos (de escritura) que se seleccionarán tienen que estar declarados en el ICE (véase la cláusula 9). Normalmente, los tipos tienen una anchura fija y un espaciamiento de carácter fijo. El proceso de conversión asegurará que ningún carácter se superponga a otro, ni se pierda. En el caso de tipos de gran tamaño, el espaciamiento de líneas puede aumentar y el formato puede cambiar. Sin embargo, incumbe a la CA asegurar que no haya pérdida del contenido y que las líneas del texto sean legibles.

### 8.1.1 Juego de caracteres del formato de transferencia ASCII ampliado de APPLI/COM

Los caracteres admitidos por el formato de transferencia ASCII ampliado de APPLI/COM para los servicios teletex, télex y telefax se muestran en los Cuadros 51, 52 y 53<sup>6)</sup>. Este formato de transferencia puede invocarse de dos maneras:

- especificando el valor de parámetro "ASCII437" en la palabra clave "Convert", en los sistemas que emplean la página de códigos 437. Estos sistemas lo declararán en el ICE.
- especificando el valor de parámetro "ASCII" en la palabra clave "Convert", en los sistemas que emplean la página de códigos 437 como juego de caracteres originario.

De lo contrario, sólo la parte baja del cuadro (o sea, los octetos de 20<sub>hex</sub> a 7E<sub>hex</sub>) tiene garantizada una conversión correcta.

NOTA – Los cuadros siguientes deben leerse seleccionando primero una columna y después una fila, por ejemplo el carácter "A" está en la columna 4, fila 1.

### 8.1.2 Juegos de caracteres del formato de transferencia ASCII estándar de APPLI/COM

Los caracteres soportados por el formato de transferencia ASCII estándar de APPLI/COM para los servicios teletex, télex y telefax se muestran en los Cuadros 54, 55 y 56.

## 8.2 Formato de transferencia T.61 de APPLI/COM

El formato de transferencia T.61 de APPLI/COM corresponde exactamente al de la Recomendación UIT-T T.61 para el servicio teletex. Los documentos que la aplicación transfiere en este formato se hacen corresponder con el servicio teletex sin modificación. La CA comprobará el formato y los códigos, pero no se efectuará conversión. Este formato es adecuado para aplicaciones que ya lo generan o que han de transmitir textos con una disposición compleja a través del servicio teletex.

## 8.3 Formato de transferencia TIFF de APPLI/COM

TIFF es un formato de transferencia de gráficos. La sigla TIFF significa "Tagged Image File Format" (formato de fichero de imagen rotulada). El formato de los ficheros TIFF incluye los atributos que describen la imagen, tales como resolución y dimensiones, que se incorporan mediante rótulos en el encabezamiento de los ficheros TIFF. Puesto que la información puede obtenerse a través de los rótulos, el programa que genera ficheros TIFF (escritor TIFF) no está ligado a una estructura de fichero constante, ya que un programa de lectura TIFF (lector TIFF) sólo necesita conocer el algoritmo necesario para localizar los rótulos.

El formato de transferencia TIFF presenta las siguientes características:

- contiene información de elemento de imagen (píxel);
- soporta formatos de compresión de datos;
- es independiente del equipo físico del sistema generador (trabaja con cualquier orden bytes);
- es flexible, debido a sus propias estructuras de rótulos.

Dado que el número de posibles combinaciones de rótulos es bastante elevado y que no todos los rótulos definidos son necesarios para definir una imagen, en el curso del desarrollo del TIFF se han creado varias clases de formatos TIFF. Por esta razón, esta Recomendación proporciona una definición del perfil adoptado como base para un fichero TIFF compatible con la interfaz.

Como un escritor TIFF, una CA puede generar cuatro clases de ficheros:

- Formato TIFF estándar, denominado aquí clase 1. Esta es la clase por defecto para una CA escritor TIFF.
- Formato TIFF con valor de compresión 2, denominado aquí clase 2. El soporte de este formato es una característica facultativa de la CA.
- Formato TIFF con compresión del grupo 3 de la UIT, denominado aquí clase 3. Este formato es soportado por todas las CA empleadas en el servicio telefax del grupo 3.
- Formato TIFF con compresión del grupo 4 de la UIT, denominado aquí clase 4. Este formato es soportado por todas las CA empleadas en el servicio telefax del grupo 4.

<sup>6)</sup> El juego de caracteres del formato de transferencia ASCII ampliado de APPLI/COM es un subconjunto del juego de caracteres IBM-PC, por lo que es muy apropiado para las implementaciones en que se emplean estos sistemas.

CUADRO 51/T.611

Juego de caracteres del formato de transferencia ASCII ampliado de APPLI/COM para el servicio telefax grupo 3 y grupo 4 (corresponde al juego de caracteres ASCII 437)

Hex	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			SP	0	@	P	'	p	Ç	É	á	⋮	⌒	⊥	α	≡
01			!	1	A	Q	a	q	ü	æ	í	⋈	⊥	⊥	β	±
02			"	2	B	R	b	r	é	Æ	ó	⋈	⊥	⊥	Γ	≥
03			#	3	C	S	c	s	â	ô	ú		⊥	⊥	π	≤
04		¶	\$	4	D	T	d	t	ä	ö	ñ	⊥	—	⊥	Σ	∫
05		§	%	5	E	U	e	u	à	ò	Ñ	⊥	⊥	⊥	σ	J
06			&	6	F	V	f	v	å	û	ª	⊥	⊥	⊥	μ	÷
07			'	7	G	W	g	w	ç	ù	º	⊥	⊥	⊥	τ	≈
08			(	8	H	X	h	x	ê	ÿ	¿	⊥	⊥	⊥	Φ	°
09			)	9	I	Y	i	y	ë	Ö	⌒	⊥	⊥	⊥	Θ	•
0A			*	:	J	Z	j	z	è	Ü	⌒	⊥	⊥	⊥	Ω	·
0B			+	;	K	[	k	{	ï	ø	½	⊥	⊥	■	δ	√
0C			,	<	L	\	l		î	£	¼	⊥	⊥	■	∞	ⁿ
0D			-	=	M	]	m	}	ì	¥	¡	⊥	⊥	■	∅	²
0E			.	>	N	^	n	~	Ä	₣	«	⊥	⊥	■	€	■
0F			/	?	O	_	o		Å	f	»	⊥	⊥	■	∩	

T0823430-95/d10

Sentido de la conversión	Acción ejecutada...
Saliente	Todos los caracteres indicados en el cuadro se aceptan en el juego de caracteres del servicio telefax
Entrante	El formato de transferencia no se genera en este sentido [Salvo en las configuraciones en las que la CA tiene la capacidad de reconocimiento opcional de caracteres (OCR, <i>optional characterrecognition</i> )]
<p>NOTA – Los caracteres «» (barra inversa) (5C<sub>HEX</sub>) y «{»,«}» («llaves») (7B<sub>HEX</sub> y 7D<sub>HEX</sub>), que suelen utilizarse con los sistemas operativos DOS y OS/2, no pertenecen al juego de caracteres teletex. Por consiguiente, no pueden utilizarse en documentos enviados a través de este servicio.</p>	

CUADRO 52/T.611

Juego de caracteres del formato de transferencia ASCII ampliado de APPLI/COM para el servicio teletex

Hex	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			SP	0	@	P	'	p	Ç	É	á					
01			!	1	A	Q	a	q	ü	æ	í				β	±
02			"	2	B	R	b	r	é	Æ	ó					
03			#	3	C	S	c	s	â	ô	ú					
04		¶	\$	4	D	T	d	t	ä	ö	ñ					
05		§	%	5	E	U	e	u	à	ò	Ñ					
06			&	6	F	V	f	v	å	û	ª				μ	÷
07			'	7	G	W	g	w	ç	ù	º					
08			(	8	H	X	h	x	ê	ÿ	¿					°
09			)	9	I	Y	i	y	ë	Ö						•
0A			*	:	J	Z	j	z	è	Ü					Ω	·
0B			+	;	K	[	k		ï	ø	½					
0C			,	<	L		l		î	£	¼					
0D			-	=	M	]	m		ì	¥	¡					²
0E			.	>	N	^	n	~	Ä		«					
0F			/	?	O	_	o		Å		»					

T0823440-95/d11

Sentido de la conversión	Acción ejecutada...
Saliente	Todos los caracteres indicados en el cuadro se aceptan en el juego de caracteres del servicio teletex
Entrante	En el sentido de recepción, algunos caracteres que provienen del servicio pueden no estar contenidos en el cuadro. Estos caracteres deberán convertirse en caracteres de aspecto similar al original, o en el carácter «?» (3F <sub>HEX</sub> ). El carácter FA <sub>HEX</sub> no se genera en el sentido de recepción en ningún caso. En su lugar se genera el carácter F <sub>HEX</sub>

CUADRO 53/T.611

Juego de caracteres del formato de transferencia ASCII ampliado de APPLI/COM para el servicio télex

Hex	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			SP	0		P		p								
01				1	A	Q	a	q								
02				2	B	R	b	r								
03				3	C	S	c	s								
04				4	D	T	d	t								
05				5	E	U	e	u								
06				6	F	V	f	v								
07			'	7	G	W	g	w								
08			(	8	H	X	h	x								
09			)	9	I	Y	i	y								
0A			:		J	Z	j	z								
0B			+		K		k									
0C			,		L		l									
0D			-	=	M		m									
0E			.		N		n									
0F			/	?	O		o									

T0823450-95/d12

Sentido de la conversión	Acción ejecutada...
Saliente	Todos los caracteres indicados en el cuadro se aceptan en el juego de caracteres del servicio télex. Las letras mayúsculas se interpretan como minúsculas
Entrante	En el sentido de recepción, las letras siempre se codifican como minúsculas (códigos 61 <sub>HEX</sub> a 7A <sub>HEX</sub> ). No se generan letras mayúsculas

CUADRO 54/T.611

Juego de caracteres del formato de transferencia ASCII estándar de APPLI/COM para el servicio telefax grupos 3 y 4

Hex	00	10	20	30	40	50	60	70
00			SP	0	@	P	'	p
01			!	1	A	Q	a	q
02			"	2	B	R	b	r
03			#	3	C	S	c	s
04			\$	4	D	T	d	t
05			%	5	E	U	e	u
06			&	6	F	V	f	v
07			'	7	G	W	g	w
08			(	8	H	X	h	x
09			)	9	I	Y	i	y
0A			*	:	J	Z	j	z
0B			+	;	K	[	k	{
0C			,	<	L	\	l	
0D			-	=	M	]	m	}
0E			.	>	N	^	n	~
0F			/	?	O	_	o	

T0823460-95/d13

Sentido de la conversión	Acción ejecutada...
Saliente	Todos los caracteres indicados en el cuadro se aceptan en el juego de caracteres del servicio telefax grupos 3 y 4
Entrante	El formato de transferencia no se genera en este sentido
<p>NOTA – Los caracteres «\» (barra inversa) (5C<sub>HEX</sub>) y «{»,«}» («llaves») (7B<sub>HEX</sub> y 7D<sub>HEX</sub>), que suelen utilizarse con los sistemas operativos DOS y OS/2, no pertenecen al juego de caracteres teletex. Por consiguiente, no pueden utilizarse en documentos enviados a través de este servicio.</p>	

CUADRO 55/T.611

Juego de caracteres del formato de transferencia ASCII estándar de APPLI/COM para el servicio teletex

Hex	00	10	20	30	40	50	60	70
00			SP	0	@	P	'	p
01			!	1	A	Q	a	q
02			"	2	B	R	b	r
03			#	3	C	S	c	s
04			\$	4	D	T	d	t
05			%	5	E	U	e	u
06			&	6	F	V	f	v
07			'	7	G	W	g	w
08			(	8	H	X	h	x
09			)	9	I	Y	i	y
0A			*	:	J	Z	j	z
0B			+	;	K	[	k	
0C			,	<	L		l	
0D			-	=	M	]	m	
0E			.	>	N	^	n	~
0F			/	?	O	_	o	

T0823470-95/d14

Sentido de la conversión	Acción ejecutada...
Saliente	Todos los caracteres indicados en el cuadro se aceptan en el juego de caracteres del servicio teletex
Entrante	En el sentido de recepción pueden llegar caracteres que no estén contenidos en el anterior cuadro. Estos caracteres pueden sustituirse por el carácter «?» (3F <sub>HEX</sub> )

CUADRO 56/T.611

Juego de caracteres del formato de transferencia ASCII estándar de APPLI/COM para el servicio télex

Hex	00	10	20	30	40	50	60	70
00			SP	0		P		p
01				1	A	Q	a	q
02				2	B	R	b	r
03				3	C	S	c	s
04				4	D	T	d	t
05				5	E	U	e	u
06				6	F	V	f	v
07			'	7	G	W	g	w
08			(	8	H	X	h	x
09			)	9	I	Y	i	y
0A			:	J	Z	j	z	
0B			+		K		k	
0C			,		L		l	
0D			-	=	M		m	
0E			.		N		n	
0F			/	?	O		o	

T0823480-95/d15

Sentido de la conversión	Acción ejecutada...
Saliente	Todos los caracteres indicados en el cuadro se aceptan en el juego de caracteres del servicio télex. Las letras mayúsculas se interpretan como minúsculas
Entrante	En el sentido de recepción, las letras siempre se codifican como minúsculas (códigos 61 <sub>HEX</sub> a 7A <sub>HEX</sub> ). No se generan letras mayúsculas

### 8.3.1 Perfil TIFF de APPLI/COM

En el Cuadro 57 se resumen los rótulos reconocidos por las CA y su tratamiento.

**Cuadro del perfil TIFF de APPLI/COM**

Rótulo		Lector TIFF		Escritor TIFF			
HEX	Nombre	Valor aceptado (dec)	Valor por defecto (dec)	Valor clase 1 (dec)	Valor clase 2 (dec)	Valor clase 3 (dec)	Valor clase 4 (dec)
FE	NewSubfileType (Tipo nuevo subfichero) (Nota 1)	"0", "2"	"0"	"0"	"0"	"0"	"0"
FF	SubfileType (Tipo subfichero) (Nota 1)	"0", "2"	"0"	"0"	"0"	"0"	"0"
100	ImageWidth (Anchura de imagen)	'Width' (anchura) (Nota 2)	Rechazar	'Width' (anchura)	'Width' (anchura)	'Width' (anchura)	'Width' (anchura)
101	ImageLength (Longitud de imagen)	'Length' (longitud) (Nota 3)	Rechazar	'Length' (longitud)	'Length' (longitud)	'Length' (longitud)	'Length' (longitud)
102	BitsperSample (Bits por muestra)	"1"	"1"	"1"	"1"	"1"	"1"
103	Compression (Compresión)	"1", "2", "3", "4", "32773" (Nota 4)	"1"	"1"	"2"	"3"	"4"
106	Photometric-Interpretation (Interpretación fotométrica)	"0", "1"	"0"	"0"	"0"	"0"	"0"
107	Thresholding (Fijación de umbral)	Ignorar		Ninguno			
108	CellWidth (Anchura de célula)	Ignorar		Ninguno			
109	CellLength (Longitud de célula)	Ignorar		Ninguno			
10A	FillOrder (Orden de relleno)	"1", "2" (Nota 5)	"1"	"1"	"1"	"2"	"2"
10D	DocumentName (Nombre de documento)	Ignorar		Ninguno			
10E	ImageDescription (Descripción de imagen)	Ignorar		Ninguno			
10F	Make (Fabricante)	Ignorar		Ninguno			
110	Model (Modelo)	Ignorar		Ninguno			
111	StripOffsets (Desplazamientos de bandas)	'Offset' (desplazamiento)	Rechazar	'Offset' (desplazamiento)	'Offset' (desplazamiento)	'Offset' (desplazamiento)	'Offset' (desplazamiento)
112	Orientation (Orientación)	"1"	"1"	"1"	"1"	"1"	"1"
115	SamplesPerPixel (Muestras por píxel)	"1"	"1"	"1"	"1"	"1"	"1"
116	RowsPerStrip (Filas por banda)	'Rows' (filas)	Rechazar	'Rows' (filas)	'Rows' (filas)	'Rows' (filas)	'Rows' (filas)
117	StripByteCounts (Cuentas de bytes banda)	'Count' (cuenta)	Rechazar	'Count' (cuenta)	'Count' (cuenta)	'Count' (cuenta)	'Count' (cuenta)
118	MinSampleValue (Valor de muestra mínimo)	"0"	"0"	"0"	"0"	"0"	"0"
119	MaxSampleValue (Valor de muestra máximo)	"1"	"1"	"1"	"1"	"1"	"1"
11A	XResolution (Resolución X)	"300" (Nota 6)	"300"	"300"	"300"	'X-res'	'X-res'
11B	YResolution (Resolución Y)	"300" (Nota 6)	"300"	"300"	"300"	'Y-res'	'Y-res'

**Cuadro del perfil TIFF de APPLI/COM**

Rótulo		Lector TIFF		Escritor TIFF			
HEX	Nombre	Valor aceptado (dec)	Valor por defecto (dec)	Valor clase 1 (dec)	Valor clase 2 (dec)	Valor clase 3 (dec)	Valor clase 4 (dec)
11C	PlanarConfiguration (Configuración planar)	"1"	"1"	"1"	"1"	"1"	"1"
11D	PageName (Nombre de página)	Ignorar		Ninguno			
11E	XPosition (Posición X)	Ignorar		Ninguno			
11F	YPosition (Posición Y)	Ignorar		Ninguno			
120	FreeOffsets (Desplazamientos libres)	Ignorar		Ninguno			
121	FreeByteCount (Cuenta de bytes libres)	Ignorar		Ninguno			
122	GrayResponseUnit (Unidad de respuesta gris)	Ignorar		Ninguno			
123	GrayResponseCurve (Curva de respuesta gris)	Ignorar		Ninguno			
124	Group3Options (Opciones del grupo 3)	"0", "4" (Nota 7)	"0"	"0"	"0"	"4"	"4"
125	Group4Options (opciones del grupo 4)	"0", "4" (Nota 7)	"0"	"0"	"0"	"4"	"0"
128	ResolutionUnit (Unidad de resolución)	"2"	"2"	"2"	"2"	"2"	"2"
129	PageNumber (Número de página)	'Page' (página)	Rechazar	'Page' (página)	'Page' (página)	'Page' (página)	'Page' (página)
12C	ColorResponseUnit (Unidad de respuesta color)	Ignorar		Ninguno			
12D	ColorResponseCurve (Curva de respuesta color)	Ignorar		Ninguno			

**NOTAS**

- 1 Los rótulos NewSubfileType y SubfileType deberán ser aceptados por los lectores. El escritor deberá generar el rótulo NewSubfileType sólo cuando éste haya sustituido al rótulo SubfileType.
- 2 Se aplica lo siguiente: Anchura de imagen/resolución X <= 215 mm. Si se excede este cociente, el documento puede rechazarse. Por consiguiente, si la resolución X es 204 dpi (norma telefax G3), no ha de rebasarse el valor de 1728 para la anchura de imagen.
- 3 La longitud de imagen se procesa si se cumple lo siguiente: cociente longitud de imagen/resolución Y < 297 mm. Esto corresponde a la longitud de una hoja de papel DIN A4. La garantía se aplica hasta una resolución de 300 dpi. El tratamiento de valores de longitud de imagen más altos es una característica facultativa de la CA.
- 4 Admisión de valores de compresión:
  - El valor 1 (sin compresión) se garantiza únicamente si el valor de la resolución (resolución X, resolución Y) es de 300 dpi o corresponde exactamente al del servicio telefax seleccionado.
  - La admisión del valor 2 es una característica facultativa de la CA.
  - La compresión 3 ó 4 del grupo 3/4 de la UIT sólo se garantiza si el valor de la resolución (resolución X, resolución Y) corresponde exactamente a las resoluciones de los servicios telefax seleccionados.
  - 32773 (compresión "pack-bits") es una característica facultativa de la CA.
- 5 El valor de orden de relleno 2 [= 'reverse bit-order' (orden de bits inverso)] sólo se admite si el valor de compresión es 3 ó 4.
- 6 Para los valores de compresión 3 y 4 se especificará el valor de resolución real (resolución X, resolución Y), en dpi, que corresponda exactamente al del servicio telefax seleccionado. No se aplica el valor por defecto del lector, ya que se deberá aplicar la resolución X e Y. La LA es responsable de la conversión local de las resoluciones recibidas.
- 7 Si el valor de compresión es 3, el valor de las opciones del grupo 3 es 4; de lo contrario, es cero. Se rechazará cualquier otra combinación.

En el Cuadro 58 se resumen las reglas aplicables al cuadro del perfil TIFF.

CUADRO 58/T.611

**Reglas aplicables al cuadro del perfil TIFF**

Columna	Regla
Rótulo	Identifica el nombre del rótulo y su valor en notación hexadecimal. Si un lector TIFF encuentra un rótulo no indicado aquí, puede rechazar el fichero TIFF
Lector – Valor aceptado	Recapitula los valores aceptados. Sólo está garantizada la aceptación por la CA de los valores que estén indicados. La CA puede rechazar otros valores, por lo que la LA generadora ha de evitarlos. Si un valor indica "ignorar", el lector hace caso omiso del rótulo
Lector – Valor por defecto	Indica el valor por defecto aplicable si el rótulo no está especificado. Si un valor indica "rechazar". El fichero TIFF deberá rechazarse si el rótulo está ausente. Si un valor indica "ignorar", no se aplica ningún valor por defecto, ya que el lector no tiene en cuenta el rótulo
Escritor	La columna correspondiente al escritor contiene los valores generados por la clase apropiada de escritor TIFF de la CA. Si un valor indica "ninguno", no se genera ningún rótulo ni valor en esa clase

**8.4 Limitaciones de servicio aplicables a los formatos de transferencia**

Por la propia naturaleza de los servicios de telecomunicaciones, los formatos de transferencia están sujetos a varias limitaciones. Estas se resumen en el Cuadro 59.

CUADRO 59/T.611

**Limitaciones de los formatos de transferencia**

Servicio de telecomunicaciones	Formato de transferencia permitido	Limitaciones
Télex	ASCII, T.50	No se permiten especificaciones de formato, salvo las relativas a la composición del texto. Las demás especificaciones deberán ignorarse. Hay también limitaciones en cuanto al juego de caracteres
Teletex	ASCII, T.50, T.61	Sólo se permiten las especificaciones de formato propias del teletex. Todas las demás especificaciones deberán ignorarse, o deberá rechazarse el fichero. Hay también limitaciones en cuanto al juego de caracteres
	VOID	Sólo se permite en combinación con una selección de tipo TFT (transferencia de ficheros telemáticos)
Telefax (G3/G4)	ASCII, T.50 <sup>a)</sup>	Se permiten todos los formatos y atributos. Si una CA no admite un atributo o formato determinado, le está permitido ignorarlo
	TIFF	Sólo se permite si no se ha seleccionado ninguno de los tipos TFT (transferencia de ficheros telemáticos)
	VOID	Sólo se permite en combinación con una selección de tipo TFT (transferencia de ficheros telemáticos)
<sup>a)</sup> Sólo es aplicable en el sentido saliente.		

## 9 Entorno de configuración de interfaz (ICE)

Antes que una LA pueda iniciar una interacción con una CA, la LA necesita conocer que CA requiere (sobre la base de criterios como el tipo de servicio de telecomunicación admitido) y cómo "hablar" con esa CA. Dado que la interfaz descrita en esta Recomendación está orientada a múltiples plataformas, el mecanismo prescrito para obtener información de las CA antes de la interacción con ellas se basa en la lectura de la información de configuración. Esta información de configuración se denomina entorno de configuración de interfaz (ICE).

El proveedor de la CA suministrará esta información de configuración de modo tal que las LA puedan obtener información de una determinada CA y acceder a la misma.

La parte del ICE que suministra información de una CA determinada se denomina descriptor de CA. Según el sistema sea independiente o utilice una red de zona local (LAN), la información sobre las CA asequibles se puede hallar en diferentes lugares. Como múltiples CA pueden ser alcanzadas por una LA, es necesario centralizar las ubicaciones de cada descriptor de CA en un fichero "ICE maestro". La ubicación del fichero ICE maestro propiamente dicho se define en una base por plataforma (véase el Anexo B).

Por tanto, en un sistema dado, una LA leerá primero el fichero ICE maestro para descubrir qué CA puede alcanzar así como la ubicación de esos descriptores de CA de las CA. Posteriormente, la LA lee el descriptor de CA de la CA que desea utilizar. Por último, la LA se engancha a esa CA mediante la información hallada en el mencionado descriptor de CA de la CA.

Para ilustrar estos conceptos, se pueden considerar tres configuraciones típicas:

- un sistema independiente que admite múltiples LA y múltiples CA;
- un sistema basado en LAN en el que las LA y las CA están ubicadas en diferentes máquinas, y hay un servidor de ficheros;
- un sistema basado en LAN en el que las LA y las CA están ubicadas en diferentes máquinas, y no hay ningún servidor de ficheros.

En el caso a), el ICE maestro está ubicado en un determinado trayecto de fichero (la ubicación precisa depende de la plataforma, véase el Anexo B), y el ICE maestro da una lista de todas las CA de ese sistema con la ubicación de sus respectivos descriptores de CA (véase la Figura 12).

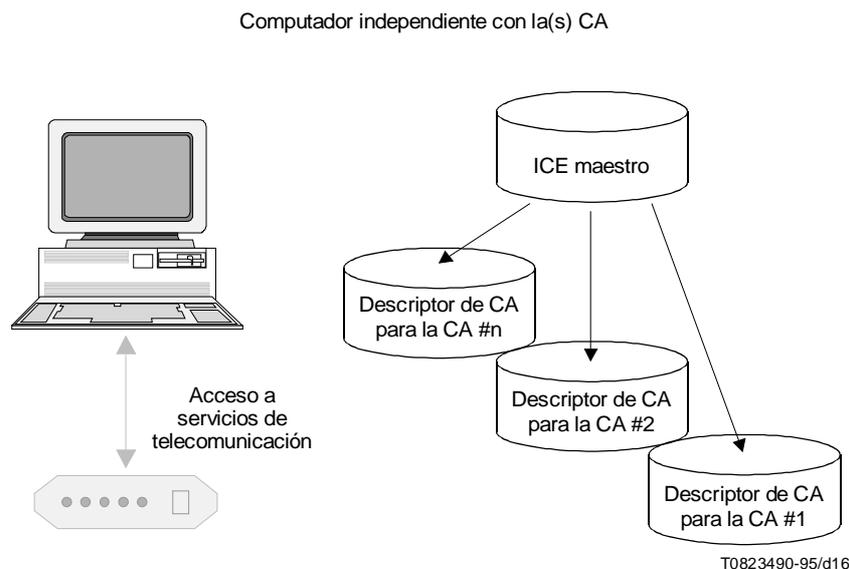


FIGURA 12/T.611

**Sistema independiente que admite múltiples CA**

En el caso b), el ICE maestro está ubicado en el servidor de ficheros, en un trayecto de fichero que todas las estaciones de trabajo pueden alcanzar. El ICE maestro da una lista de todas las CA en la LAN y presenta los trayectos absolutos de su descriptor de CA respectivo (que puede estar ubicado en otras máquinas distintas del servidor de ficheros) (véase la Figura 13).

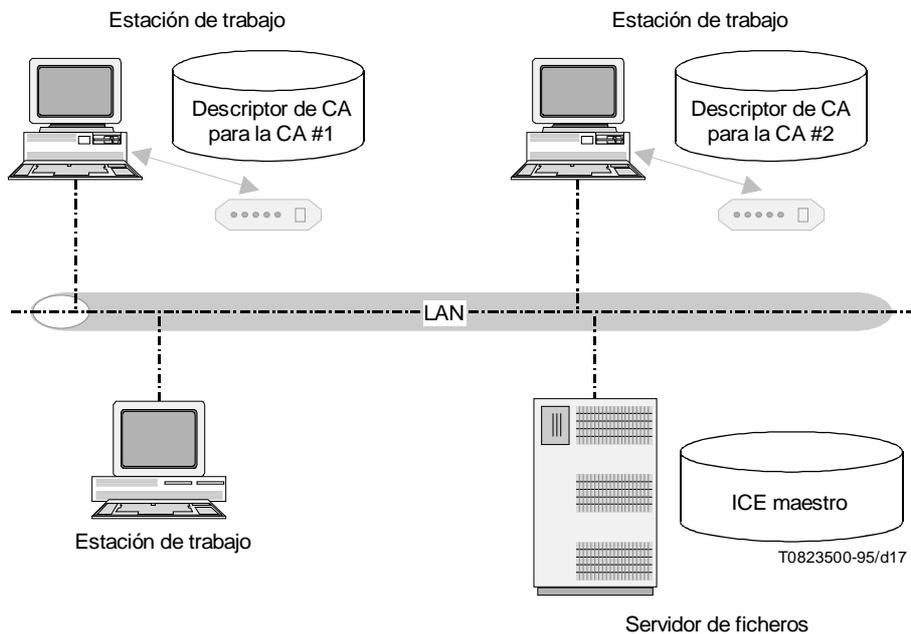


FIGURA 13/T.611

**Sistema basado en LAN con servidor de ficheros**

En el caso c), como ningún servidor de fichero puede centralizar el fichero ICE maestro para la totalidad del sistema, se ubica un fichero ICE maestro en cada máquina que admite una o más CA, en una ubicación fija como en el caso a). La ubicación de los descriptores de CA se registra en el ICE maestro con trayectos relativos a esa máquina (véase la Figura 14).

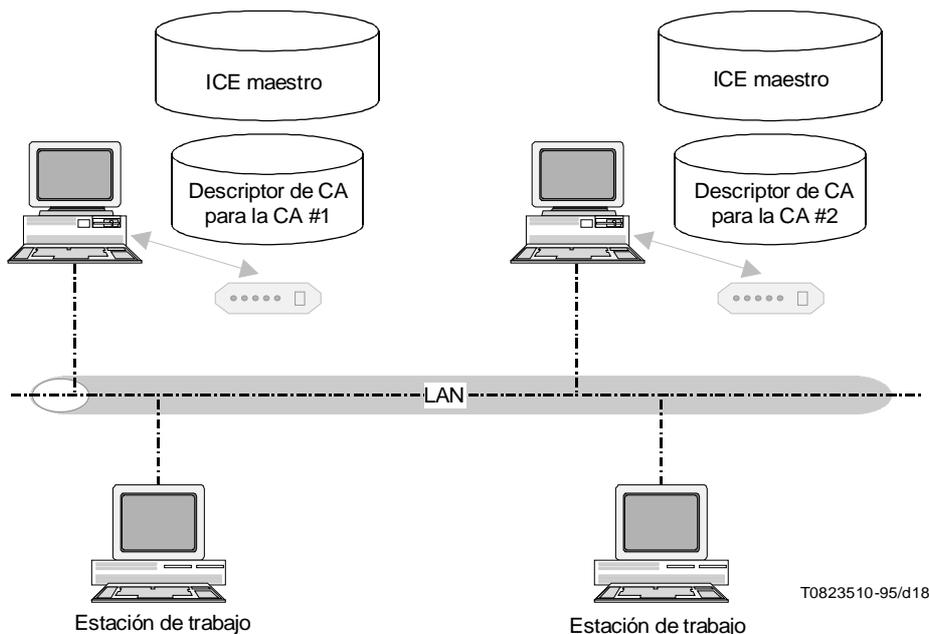


FIGURA 14/T.611

**Sistema LAN entre entidades pares**

## 9.1 Presentación del ICE

El ICE se presenta utilizando una codificación de texto legible, como la descrita para las TDD en 6.4.1. Salvo en aquellos sistemas en que la codificación natural de caracteres sea EBCDIC (Code-ID="E"), se utiliza la codificación IRA (Code-ID="I"). El ICE (ICE maestro y descriptores de CA) se presentarán en forma de líneas utilizando el repertorio de caracteres IRA, con el formato descrito para las TDD en 6.4.1.

## 9.2 Acceso a la información de ICE

El acceso al ICE maestro se efectuará por un "método de acceso a fichero" que es común a la mayor parte de los sistemas operativos. A su vez, todos los métodos de acceso disponibles para obtener información de descriptor de CA deberán estar definidos en el ICE maestro. Los métodos de acceso pueden estar "basados en ficheros" o pueden servirse de métodos más dinámicos que pueden estar disponibles en un determinado entorno operativo [por ejemplo, DLL para Windows y ficheros ejecutables y controles de dispositivos (device drivers) para DOS].

## 9.3 ICE maestro

El fichero ICE maestro da una lista de todas las CA que puede alcanzar una LA. Contiene información sobre el identificador de CA, los servicios de comunicaciones soportados, el tipo de mecanismo de intercambio admitido y la ubicación del descriptor de CA de la CA donde se puede obtener más información.

### 9.3.1 Descripción formal

Esta subcláusula describe la sintaxis del ICE maestro. El primer elemento de sintaxis en el ICE maestro es el <Master ICE Header>. Antes de este encabezamiento no se permite ningún otro elemento, ni siquiera los determinantes de formato ESPACIO y TABULACIÓN. A continuación se muestra la sintaxis detallada, descrita en gramática estilo BNF (para obtener una descripción de la gramática estilo BNF, véase A.1):

```
<Master ICE> :=          <Master ICE Header> <CA Entries>

<Master ICE Header> :=   "I*APPLI/COM*" <Version> "*ITU-T*MASTER_ICE"
-- for definition of <ICE Header> see also 6.3.2.3

<Version> :=             "1994"
-- <Version> denotes year of approval of this Recommendation

<CA Entries> :=          "#" <CA Entry> { "#" <CA Entry> }

<CA Entry> :=            <CA Identifier> <CA Product Info> <CA EM>
                        <CA Service> { <CA Service> }
                        <CA ICE Location> { <CA ICE Location> }

<CA Identifier> :=       "CA-ID" ":" NUMERIC-STRING
-- identifies the CA

<CA Product Info> :=     "APPLI/COM" ":" STRING(SIZE(1..255))
-- Information about the product, CA manufacturer-provided

<CA EM> :=               "EM" ":" "FILE" | "PRIMITIVE"
-- exchange mechanism supported by the CA
-- refer to 7.3

<CA Service> :=          "SERVICE" ":" <Service-id-parameter>
-- telecommunication services supported by the CA
-- refer to section 6.4.5.1 and 10.4

<CA ICE Location> :=     "ACCESS" ":" PATH
-- points to the location of the CA-Descriptor file or driver
```

### 9.3.2 Configuración

El ICE maestro se puede configurar manualmente por medio de un programa "editor" en el curso de la instalación de una determinada CA, o dinámicamente por medio de un programa de control (driver) adecuado proporcionado por el fabricante de la CA<sup>7)</sup>.

El ICE se configurará de modo que tenga en cuenta todas las posibles aplicaciones de comunicaciones accesibles desde el interior de un sistema<sup>8)</sup>.

## 9.4 Descriptor de CA

El descriptor de CA contiene la información pertinente a una determinada CA. Se compone de secciones que agrupan entradas relacionadas lógicamente.

### 9.4.1 Descripción formal

Esta subcláusula describe la sintaxis del descriptor de CA. El primer elemento de sintaxis en el descriptor de CA es el <CA-Descriptor Header>. Antes del encabezamiento no se permite ningún otro elemento, ni siquiera los determinantes de formato ESPACIO y TABULACIÓN. A continuación se muestra la sintaxis detallada, descrita en gramática estilo BNF:

```
<CA-Descriptor> :=          <CA-Descriptor Header> <CA-Descriptor-component>
<CA-Descriptor Header> :=   "I*APPLI/COM*" <Version> "ITU-T*ICE"
                             -- for definition of <CA-Descriptor Header> see also 6.3.2.3
<Version> :=                 "1994"
                             -- <Version> denotes year of approval of this Recommendation
<CA-Descriptor-component> := "#" <Section> {<Section>}
<Section> :=                 <Section-Parameter> {<Section-Parameter>}
<Section-Parameter> :=       <Parameter-Name> ":" <Parameter-Value>
<Parameter-Name> :=          STRING(SIZE(1..16))
<Parameter-Value> :=         STRING(SIZE(1..255))
```

### 9.4.2 Configuración

La información del descriptor de CA será proporcionada por el fabricante de la CA en forma de fichero o por un medio dinámico como un programa de control de dispositivo, un fichero ejecutable, o DLL. El método o los métodos de acceso al descriptor de CA se definirán en el ICE maestro como parte del procedimiento de instalación para cada CA. El descriptor de CA se puede configurar, sea manualmente por medio de un programa "editor" en el curso de la instalación de una CA dada, o dinámicamente por medio de un programa de control de dispositivo proporcionado por el fabricante de la CA. Sin embargo, una LA que se basa en la información obtenida del ICE maestro o en el descriptor de CA con relación a una CA dada, no supondrá que la CA está activa en ese mismo momento (es decir, que ha sido cargada y está funcionando). Para asegurarse que una determinada CA está realmente activa la LA tiene que enganchar con esa CA. Sólo cuando el enganche se ha efectuado correctamente puede considerarse activa la CA. Para mayor información sobre el procedimiento enganche, véanse 7.1 y 7.2.1.

## 9.5 Componentes del descriptor de CA

En los Cuadros 60, 61 y 62 se describen las palabras clave y los correspondientes valores de parámetros del descriptor de CA. En la subcláusula 6.4.4 se da información complementaria sobre la codificación de los valores de parámetros.

<sup>7)</sup> Puesto que el acceso a ficheros y a programas de control de dispositivos del sistema operativo son similares en los sistemas operativos más frecuentemente utilizados, el propio ICE maestro puede realizarse estáticamente como un fichero, o dinámicamente en la forma de un programa de control de dispositivo del sistema operativo.

<sup>8)</sup> No es necesario que la CA esté situada en el mismo equipo físico que la LA. La LA sólo tiene que proporcionar acceso por medio de un programa de control de dispositivo o de otra forma similar.

## 9.5.1 Parámetros de la sección GENERAL

Véase el Cuadro 60.

CUADRO 60/T.611

### Ítems de información del descriptor de CA que son independientes del sistema operativo y del mecanismo de intercambio

Palabra clave <sup>a)</sup>	Parámetro	Interpretación
ALIAS	"yes"   "no"	Especifica si la CA soporta o no nombres de alias (nombres fáciles de manejar)
CODEPAGE*	STRING	Especifica las páginas de códigos adicionales para los juegos de caracteres ASCII ampliado que son soportadas por la CA. La cadena de caracteres indica el número de la página de códigos (por ejemplo, "850")
CODING*	Code-ID	Especifica el esquema de codificación soportado por la CA. Para los valores soportados que pueden especificarse, véase 6.3. Si el identificador de código "E" tiene que aparecer en el ICE, su valor binario debe fijarse de acuerdo con la presentación de código elegida para el propio ICE, es decir, si el identificador de código del encabezamiento ICE es "T" (Standard ASCII de APPLI/COM), el valor binario del identificador de código "E" deberá codificarse como 45 <sub>HEX</sub>
CONVCHK*	<Convert-id-parameter>	Declara los formatos de transferencia soportados en la TDD de depósito (Submit) función CONVERT y/o CHECK. Para la sintaxis del <Convert-id-parameter> véase también 6.4.5.3
COUNTRY	STRING	Especifica el país para el cual está configurada la CA. Se utiliza para registrar características específicas de un país como las que dan acceso a la facilidad de conversión, o a la lista negra de números que podrían marcarse. El valor a colocar en este parámetro deberá tomarse de la Recomendación UIT-T T.35. Se presentará como una cadena numérica de base decimal, por ejemplo "154" para Seychelles
DRF	"yes"   "no"	Parámetro booleano; indica si la CA soporta o no la facilidad "Dispatch received files" (despacho de ficheros recibidos)
EXTEND*	<keyword>	Da la posibilidad de ampliaciones a la Recomendación. Sólo puede implementarse como cambios formales a la Recomendación. Deberán indicarse todas las palabras clave soportadas por la CA
FC	"A"   "B"	Indica la clase funcional soportada por la CA
FONT0	NUMERIC-STRING	Un dígito en la gama "1..9" para especificar el número de tipos de escritura (fonts) declarados. Los tipos de escritura se designan por una serie de identificadores desde "Font1" hasta "Font9" (véase más adelante)
FONTx*	STRING "," NUMERIC-STRING	El nombre del tipo de escritura seguido por el "paso" ("pitch") expresado en caracteres por pulgada. El valor de "x" es de 1 a 9
NATIONAL*	<keyword>	Da la posibilidad de ampliaciones nacionales a la Recomendación. Sólo puede implementarse con la aprobación de las Administraciones nacionales. (Deberán indicarse todas las palabras clave soportadas)
PRINT*	<Printer-id-parameter>	Declara las impresoras que pueden ser controladas por la CA en la TDD de depósito (Submit) función PRINT. Para la sintaxis del <Printer-id-parameter> véase también 6.4.4.10
PRIVATE*	<keyword>	Da la posibilidad de ampliaciones privadas a la Recomendación. (Deberán indicarse todas las palabras clave soportadas)
RECORD*	<keyword> "," NUMERIC-STRING	Da la lista completa de los nombres de los campos del Registro de CA soportados por la CA, en el orden en que se encuentran en el fichero producido por la función Trace:COPY. La CA indicará la palabra clave, seguida de un carácter "," (coma) como separador, seguido de la longitud que tendrá el campo en el fichero resultante. Véase también el Cuadro 23
SUBMIT*	"PRINT"   "CONVERT"   "CHECK"	Declara las funciones soportadas en la función de la TDD de depósito (Submit). Esta palabra clave se repetirá las veces que sea necesario

<sup>a)</sup> Un "\*" (asterisco) al final de una palabra clave indica que puede repetirse.

### 9.5.2 Parámetros de la sección EM

Véase el Cuadro 61.

CUADRO 61/T.611

#### Ítems de información del descriptor de CA aplicables al mecanismo de intercambio de ficheros

Palabra clave	Parámetro	Interpretación
EM	"file"   "primitive"	Método de intercambio utilizado para intercambiar las TDD entre las LA y las CA "file" y "primitive" son los valores soportados (para más detalles véase 7.3)
SYNC	"yes"   "no"	Indica si la CA es o no accionada por sincronización ("sync-driven"). Para más detalles, véase 7.3
F_JOB_Q	PATH	Especifica el trayecto de los ficheros de petición TDD. Para más detalles, véase 7.3
F_ACK_Q	PATH	Especifica el trayecto de los ficheros de respuesta TDD. Para más detalles, véase 7.3
ERROR_Q	PATH	Especifica el trayecto de los ficheros de respuesta TDD relativos a errores. Para más detalles, véase 7.3
ALARM	"yes"   "no"	Este parámetro booleano indica si la CA soporta o no la función SetAlarm

### 9.5.3 Parámetros de la sección SISTEMA OPERATIVO

Véase el Cuadro 62.

### 9.5.4 Parámetros de la sección SERVICIO

Véase el Cuadro 63.

En las cláusulas pertinentes de la Parte II de esta Recomendación se describen más ítems de información específica del servicio.

## 10 Clases funcionales y perfiles de servicio

### 10.1 Clase funcional A

La clase funcional A (FCA, *functional class A*) especifica un conjunto de formatos de transferencia, así como las interacciones entre las LA y las CA para enviar y recibir documentos.

La clase funcional A requiere el soporte de:

- <SendTDD> y <SendAckTDD>, que permiten a la LA enviar un documento a través de una CA;
- <ReceiveTDD> que permite a la LA recuperar documentos recibidos por una CA.

Además, la LA puede seleccionar diversas opciones ofrecidas por la CA, como la utilización de palabras clave adicionales ("ADDKEYS") o la provisión de la facilidad de depósito ("SUBMIT").

### 10.2 Clase funcional B

Además de las características de la FCA, la clase funcional B (FCB) proporciona la funcionalidad de rastreo completa.

La FBC es un superconjunto de la FCA y proporciona una mayor integración de las funciones de comunicaciones a las aplicaciones de los usuarios.

CUADRO 62/T.611

## Ítems de información del descriptor de CA aplicables al sistema operativo

Palabra clave <sup>a)</sup>	Parámetro	Interpretación
ENVIRON*	"MSDOS"   "WINDOWS"   "UNIX"   "OS2"   "MacOS"	Esta palabra clave especifica el entorno operativo de la CA. Si una CA admite varios entornos, el ICE contendrá tantas veces la palabra clave ENVIRON como sistemas operativos diferentes admitidos
DRIVER	PATH	Nombre del control que hay que abrir para iniciar diálogos con la CA. Véase el Anexo E para más detalles
INT	hex,hex	Indica el número del interrupt. Está formado por dos números hexadecimales, el primero especifica el número múltiplex, y el segundo el código de programa. Si el interrupt no está multiplexado, no se especificará el segundo número hexadecimal
LIB	"yes"   "no"	Este parámetro booleano indica si la CA es o no una biblioteca estática. La LA tiene que estar vinculada a ella
LIB-NAME*	PATH	Trayecto(s) de la(s) biblioteca(s) (se utiliza en combinación con la palabra clave LIB)
DLL	"yes"   "no"	Biblioteca de vinculación dinámica ( <i>dynamic link library</i> ). Véase el Anexo E. La palabra clave DLL-NAME deberá ser soportada solamente si se soporta el mecanismo de intercambio DLL
DLL-NAME*	PATH	Trayecto(s) del (de los) fichero(s) DLL (se utiliza en combinación con la palabra clave DLL)
DDE	"yes"   "no"	Mecanismo de intercambio dinámico de datos. En el entorno WINDOWS, si la aplicación soporta el mecanismo de intercambio DDE, deberá especificar "sí". Véase el Anexo E. Las tres palabras clave siguientes deberán incluirse en el ICE si se utiliza el mecanismo DDE
WIN-APP	STRING	Nombre de la aplicación (formato MsDos) XXXXXXXXX.XXX
SUBJECT*	STRING	Deberán mencionarse todos los "Subjects" de la CA (si existen); de lo contrario, deberán dejarse vacíos (para utilizarlos con la palabra clave DDE)
ITEM*	STRING	Deberán mencionarse todos los "ítems" de la CA (si existen); de lo contrario deberán dejarse vacíos (para utilizarlos con la palabra clave DDE)
<sup>a)</sup> Un "*" (asterisco) al final de una palabra clave indica que puede repetirse.		

CUADRO 63/T.611

## Ítems de información del descriptor de CA aplicables a la sección servicio

Palabra clave <sup>a)</sup>	Parámetro	Interpretación
ADDCONV*	STRING (SIZE (1..8))	Enumera formatos de transferencia adicionales suministrados por la CA. Los nombres aquí indicados se utilizarán para seleccionar los formatos de transferencia en los lugares apropiados
ADDKEYS*	<keyword>	Enumera las palabras clave adicionales admitidas por la CA. Sólo las palabras clave clasificadas como "+" en los cuadros de las TDD de la cláusula 6 se pueden especificar aquí
<sup>a)</sup> Un "*" (asterisco) al final de una palabra clave indica que puede repetirse.		

### **10.3 Funciones adicionales**

Las funcionalidades de depósito (Submit) y ampliación (Extend) son facilidades adicionales que deben ser declaradas en el ICE por la CA (véase también 9.5).

### **10.4 Perfiles de servicio**

El soporte de todas las características de un servicio dado de telecomunicaciones puede ser imposible, debido al gran número de funciones que deben implementarse. Por otra parte, el ofrecimiento de un número demasiado grande de opciones para el soporte de un servicio determinado tiene por consecuencia que haya implementaciones que no sean compatibles entre sí, aunque cada una de ellas cumple las especificaciones.

A fin de reducir las incompatibilidades entre aplicaciones obtenidas de vendedores diferentes, en esta Recomendación se emplea el concepto de perfiles de servicio. Un perfil de servicio agrupa un conjunto definido de características de servicio que tienen que ser soportadas por las implementaciones que pretenden ser compatibles con esta Recomendación.

El conjunto de características que deben ser soportadas para un perfil de servicio dado se define individualmente para cada servicio.

Una implementación puede así alegar ser compatible con un perfil de servicio y soportar características adicionales. Con esto queda garantizado que dos implementaciones (por ejemplo, una CA y una LA) que soporten el mismo perfil de servicio podrán ofrecer todas las características mencionadas en el perfil de servicio y, facultativamente, características adicionales, no obligatorias para ese perfil de servicio.

Las CA declararán en el ICE maestro los perfiles de servicio soportados (véase 9.5).

## PARTE II – DEPENDENCIAS DE SERVICIOS

### 11 Servicio: Telefax del grupo 3

Además de las palabras clave específicas del servicio telefax para <SendTDD>, <SendAckTDD> y <ReceiveTDD>, se define una función adicional: Poll (Interrogar) – para "interrogar" a una estación facsímil distante con miras a recibir de ella un documento. Esta función se realizará utilizando <ExtendTDD>. La extensión se denomina: <PollExtension>.

Junto con <PollExtension> se puede extender (o ampliar) también <SendTDD> y <SendAckTDD> para realizar una operación de interrogación (Poll) mientras se envía un documento.

Las extensiones se describen en los lugares apropiados de esta cláusula.

#### 11.1 Elementos de sintaxis específicos del servicio

```
<ServiceDependentKeywordsSend> :=  
    ((<Recipient> [<PollSendExtension>]) | <RecipientSpec>)  
    ((<Document> <Convert> [<Type>] [From] [To]) | <DocumentSpec>)  
    [<SubAddress>] [<G3Speed>] [<GenCil>] [<HighRes>] [<UseEcm>]
```

```
<ServiceDependentKeywordsSendack> :=  
    <Recipient> [<PollSendExtension>]  
    ((<Document> <Convert> [<Type>] [From] [To]) | <DocumentSpec>)  
    [<SubAddress>] [<G3Speed>] [<GenCil>] [<HighRes>] [<UseEcm>]
```

```
<ServiceDependentKeywordsReceive> :=  
    [<Originator>]  
    ((<Document> <Convert> <Type>) | <DocumentSpec>)  
    [SubAddress] [<CvFax3>] [<G3Speed>]
```

```
<ExtendSubFunctionKeywords> :=  
    <PollExtension>
```

```
<PollSendExtension> :=    <DoPoll> <PollPassword> [<PollSelector>]
```

```
<PollExtension> :=    <PollSubFunction> <Recipient> <PollPassword> [<PollSelector>] [<SendTime>]  
    [<ComId>] [<Minor>] [<Warning>]
```

Véase el Cuadro 64.

#### 11.2 Codificación basada en texto

##### 11.2.1 Correspondencia de palabras clave

Véase el Cuadro 65.

##### 11.2.2 Codificación de parámetros

Véase también 6.4.4 para la codificación de los parámetros que no dependen del servicio.

###### 11.2.2.1 Parámetro identificador de servicio

El parámetro Service-id-parameter se codifica como STRING fijada al valor constante "FX3".

*Sintaxis:*

```
<Service-id-parameter> :=    "FX3"
```

**Elementos de sintaxis adicionales para el servicio telefax del grupo 3**

Elemento de sintaxis	Finalidad
<Convert>	Especifica el formato de transferencia que va a utilizarse
<CvFax3>	Especifica el formato de transferencia deseado para los ficheros recibidos
<Document>, <DocumentSpec>	Especifica el documento o documentos que se va a enviar o que se están recibiendo
<DoPoll>	Especifica si el recipiente debe ser interrogado en una <SendTDD> o en una <SendAckTDD>
<G3Speed>	Especifica la velocidad de modulación deseada (entrada) o utilizada (salida)
<GenCil>	Especifica la inserción de un CIL cuando se genera el fax saliente
<HighRes>	Indica la resolución
<Minor>	Retorna un código de error adicional en la respuesta TDD
<Originator>	Especifica la dirección de comunicaciones del originador
<PollPassword>	Especifica la contraseña para interrogar un dispositivo distante
<PollSelector>	Da un (handle) al recipiente para que interroge con miras a un documento específico
<PollSubFunction>	Especifica la función de la Poll <ExtendTDD>
<Recipient>, <RecipientSpec>	Especifica la(s) dirección(es) de comunicaciones del (de los) recipiente(s)
<SubAddress>	Especifica la subdirección del originador
<To>	La transmisión debe terminar en el número de página especificado
<Type>	Especifica el formato de transmisión utilizado
<UseEcm>	ECM se desea (entrada) o fue activado (salida)
<Warning>	Este elemento de sintaxis retorna un código de advertencia adicional en la respuesta TDD

**11.2.2.2 Parámetro identificador de tipo**

El parámetro Type-id-parameter se codifica como una STRING fijada a uno de los siguientes valores:

"STD"	Servicio telefax grupo 3 básico (MH)
"BTM"	Servicio de transferencia de ficheros telemáticos (TFT) del servicio telefax G3: modo transparente básico
"DTM"	Transferencia de ficheros telemáticos (TFT) del servicio telefax G3: modo transparente al documento
"EDI"	Transferencia de ficheros telemáticos (TFT) del servicio telefax G3: Edifact
"BFT"	Servicio telefax G3: transferencia de ficheros binarios

NOTA – El formato de transmisión STD (= código Huffman modificado, MH) se define en la Recomendación UIT-T T.4; los formatos TFT BTM, DTM y EDI se definen en las Recomendaciones UIT-T T.571 y T.30; y el formato de transmisión BFT se define en la Recomendación UIT-T T.434.

*Sintaxis:*

<Type-id-parameter> := "STD" | "BTM" | "DTM" | "BFT" | "EDI"

**Codificación basada en texto de los elementos de sintaxis adicionales para el servicio telefax del grupo 3**  
 (El símbolo ↵ representa el determinante de formato de nueva línea)

Elemento de sintaxis	Par palabra clave/Parámetro
<Convert>	"CONVERT" ":" <Convert-id-parameter> ↵
<CvFax3>	"CVFAX3" ":" <Convert-id-parameter> ↵
<Document>	"FILENAME" ":" <File-parameter> ↵
<DocumentSpec>	"FILENAME" ":" "@" <File-of-filespec> ↵
<DoPoll>	"DOPOLL" ":" <Boolean-parameter> ↵
<From>	"FROM" ":" NUMERIC-STRING ↵
<G3Speed>	"G3SPEED" ":" <G3-speed-parameter> ↵
<GenCil>	"GENCIL" ":" <Boolean-parameter> ↵
<HighRes>	"HIGHRES" ":" <Resolution-parameter> ↵
<Minor>	"MINOR" ":" <Error-parameter> ↵
<Originator>	"ADDRESS" ":" <Address-parameter> ↵
<PollPassword>	"PASSWORD" ":" <Password-parameter> ↵
<PollSelector>	"POLLSELECT" ":" <Poll-select-parameter> ↵
<PollSubFunction>	"SUBFUNC" ":" "Poll" ↵
<Recipient>	"ADDRESS" ":" <Address-parameter> ↵
<RecipientSpec>	"ADDRESS" ":" "@" <File-of-addrspec> ↵
<SubAddress>	"SUBADDR" ":" <Sub-address-parameter> ↵
<To>	"TO" ":" NUMERIC-STRING ↵
<Type>	"TYPE" ":" <Type-id-parameter> ↵
<UseEcm>	"USEECM" ":" <Boolean-parameter> ↵
<Warning>	"WARNING" ":" <Error-parameter> ↵

**11.2.2.3 Parámetro identificador de conversión**

El parámetro Convert-id-parameter se codifica como una STRING fijada a uno de los siguientes valores:

- "ASCII" ASCII ampliado de APPLI/COM (para ficheros salientes solamente)
- "ASCII437" ASCII ampliado de APPLI/COM (para ficheros salientes solamente)
- "T.50" ASCII estándar de APPLI/COM (para ficheros salientes solamente)
- "TIFF" Formato de transferencia TIFF definido en 8.3
- "TIFF2" (Para ficheros entrantes solamente.) Clase 2 de TIFF de APPLI/COM, especificada en 8.3
- "TIFF3" (Para ficheros entrantes solamente.) Clase 3 de TIFF de APPLI/COM, especificada en 8.3
- "VOID" No hay conversión

NOTA – Si una LA solicita un documento telefax que ha sido recibido, es posible:

- obtener el documento en la clase 2 de TIFF de APPLI/COM, especificando TIFF2;
- obtener el documento en la clase 3 de TIFF de APPLI/COM, especificando TIFF3;

a condición de que la CA pueda generar ese formato TIFF especial. Si sólo se especifica TIFF, el documento se entregará en la clase por defecto de TIFF de APPLI/COM (la clase 1). Sin embargo, en el sentido "envío" basta con especificar TIFF solamente (sin número agregado), ya que la información de compresión está contenida en el propio formato de transferencia.

La utilización del parámetro identificador de conversión depende también del identificador de tipo seleccionado. Esto se muestra en el Cuadro 66.

CUADRO 66/T.611

**Asignaciones de identificador de conversión permitidas que dependen del identificador de tipo y del sentido del tráfico**

Identificador de tipo	Identificador de conversión para tráfico saliente	Identificador de conversión para tráfico entrante
STD	ASCII, ASCIIxxx <sup>a)</sup> , T.50, TIFF	TIFF, TIFF <sub>x</sub> <sup>b)</sup>
BTM, DTM, BFT, EDI	VOID	
<sup>a)</sup> xxx representa una página de códigos, declarada en el ICE, por ejemplo ASCII437 si se había declarado la página de códigos 437. <sup>b)</sup> x representa la clase TIFF que habrá de leerse, que es un valor entre 2 y 3.		

*Sintaxis:*

```

<Convert-id-parameter> := <Convert-std> | <Convert-bin>
<Convert-std> := <Convert-std-in> | <Convert-std-out>
<Convert-std-in> := "TIFF" | "TIFF2" | "TIFF3"
-- documentos entrantes
<Convert-std-out> := <Ascii> | "T.50" | "TIFF"
-- documentos salientes
<Ascii> := "ASCII" | STRING ("ASCII" + <Code-page>)
<Code-page> := <digit> <digit> <digit>
<digit> := "0" | ... | "9"
<Convert-bin> := "VOID"
  
```

**11.2.2.4 Parámetro de fichero de especificaciones de direcciones (File-of-addrspec) y parámetro de dirección (Address-parameter)**

El parámetro File-of-addrspec se codifica como un PATH que apunta a un fichero que contiene parámetros Addrspec-parameters, que a su vez contiene el Address-parameter.

El parámetro Address-parameter se codifica como una STRING, que contiene el número de teléfono. Si el número de teléfono comienza con un carácter "!", puede tener caracteres especiales que se tratan como operadores (o modificadores) y no como "dígito marcado", como se muestra en el Cuadro 67.

El número de teléfono puede terminarse por un carácter "#" (signo de número) y puede ir seguido de una subdirección (subdirección del equipo distante). La codificación de la subdirección se define en 11.2.2.7.

Como otra posibilidad, puede darse un nombre de alias en lugar del número de teléfono, a condición de que el alias se introduzca por un carácter "&". Se supone que la CA sabe cómo decodificar el alias especificado. Esta característica se usa típicamente cuando la CA realiza una guía telefónica.

**Modificadores de la cadena de marcación en la RTPC**

Entrada	Tipo	Explicación
"0"   ...   "9"   "*"   "#"	Dígito marcado	Los números marcados no son interpretados
"!"	Modificador	Primer carácter en la cadena de la secuencia marcada: se pasa al "modo de marcación bruta"
";"	Modificador	Pausa en el proceso de marcación. Su duración depende de la CA. Obsérvese que ";" tiene que ir precedido por el carácter de escape "\"; de lo contrario la CA puede interpretarlo como la introducción de un comentario
":"	Modificador	Similar a ";", pero no necesita ir precedido por el carácter de escape
,	Modificador	Pausa de 2 segundos en la marcación
"T"   "t"	Modificador	Establece la marcación "por tonos" para los dígitos que sigan
"P"   "p"	Modificador	Establece la marcación "por impulsos" para los dígitos que sigan
"W"   "w"	Modificador	La CA espera, y escucha durante 3 segundos, un tono de marcar continuo
"@"	Modificador	La CA escucha la señal de llamada (timbre) del equipo distante seguida de un período de silencio de 5 segundos. Si no se detecta la contestación del equipo distante, la CA da la respuesta por defecto
NOTA – Los modificadores de la cadena de marcación pueden ser objeto de acuerdos entre las Administraciones nacionales.		

*Sintaxis:*

- <File-of-addrspec> := PATH  
 -- El trayecto apunta a un fichero que contiene uno o más  
 -- <Addrsec-parameter>s
- <Addrsec-parameter> := <Address-parameter>  
 ["," <Cover-path> "," <Cover-conv> ["," [<Cover-type>] ["," <Nopgbrk>]]]
- <Address-parameter> := (<Phone-number> ["#" <Subaddress>]) |  
 ("!" <Dial-command> ["#" <Subaddress>]) |  
 ("&" <Alias>)  
 -- Contiene la secuencia de marcación del número de teléfono o un alias
- <Phone-number> := NUMERIC-STRING
- <Dial-command> := <Dial-operator> {<Dial-operator>}
- <Dial-operator> := "0" | ... | "9" | "\*" | "#" | ";" | ":" | "," | "T" | "t" | "P" | "p" | "W" | "w" | "@"
- <Subaddress> := <Subaddress-parameter>  
 -- Véase 11.2.2.7
- <Alias> := STRING
- <Cover-path> := PATH  
 -- Trayecto a un fichero que contiene una portada para un destinatario dado

<Cover-conv> := <Convert-id-parameter>  
 -- Especifica el formato de transferencia del fichero de portada

<Cover-type> := <Type-id-parameter>  
 -- Especifica el formato de transmisión de la portada

<Nopgrbrk> := <Boolean-parameter>  
 -- Si está fijado a verdadero ("Yes"), no habrá corte de página entre  
 -- la portada y el documento principal

### 11.2.2.5 Parámetro de fichero de especificaciones de ficheros (File-of-filespec) y parámetro de fichero (File-parameter)

El parámetro File-of-filespec se codifica como un PATH que apunta a un fichero que contiene parámetros Filespec-parameter, que a su vez contiene el File-parameter. El File parameter en sí se codifica también como un PATH, que apunta al fichero transferido.

*Sintaxis:*

<File-of-filespec> := PATH  
 -- El trayecto apunta a un fichero que contiene uno o más  
 -- <Filespec-parameter>s

<Filespec-parameter> := <File-parameter> "," <File-conv> ["," <File-type>]

<File-parameter> := PATH  
 -- Trayecto del fichero transferido

<File-conv> := <Convert-id-parameter>  
 -- Especifica el formato de transferencia del fichero

<File-type> := <Type-id-parameter>  
 -- Especifica el formato de transmisión del fichero

### 11.2.2.6 Parámetro de velocidad del grupo 3

El parámetro G3-speed-parameter se codifica como una NUMERIC-STRING fijada a uno de los siguientes valores:

"2400"	2400 bits por segundo
"4800"	4800 bits por segundo
"7200"	7200 bits por segundo
"9600"	9600 bits por segundo
"12200"	12200 bits por segundo
"14400"	14400 bits por segundo

*Sintaxis:*

<G3-speed-parameter> := "2400" | "4800" | "7200" | "9600" | "12200" | "14400"

### 11.2.2.7 Parámetro de subdirección

El parámetro Sub-address-parameter se codifica como una STRING que representa una subdirección. De acuerdo con la definición de la Recomendación UIT-T T.30 (1994), la cadena está limitada a los dígitos ("0".."9") y a los caracteres "+" y "-". La CA traducirá el carácter "\_" (subrayado) en un carácter de espacio IRA (20<sub>HEX</sub>). La longitud del parámetro de subdirección está limitada a 20 octetos.

*Sintaxis:*

<Sub-address-parameter> := <Subaddress-digit> {<Subaddress-digit>}

<Subaddress-digit> := "0" | ... | "9" | "\*" | "#"  
 -- La longitud del <Sub-address-parameter> está limitada a 20 octetos.

### 11.2.2.8 Parámetro de resolución

El Resolution-parameter se codifica como una NUMERIC-STRING fijada a uno de los siguientes valores:

"0"	8 pels por mm horizontal, 3,85 líneas por mm vertical (98 dpi vertical)
"1"	8 pels por mm horizontal, 7,7 líneas por mm vertical (196 dpi vertical)
"2"	200 dpi horizontal, 200 vertical (200 dpi)
"3"	300 dpi horizontal, 300 vertical (300 dpi)
"4"	400 dpi horizontal, 400 vertical (400 dpi)
"5"	8 pels por mm horizontal, 15,4 líneas por mm vertical (392 dpi vertical)
"6"	16 pels por mm horizontal, 15,4 líneas por mm vertical (392 dpi vertical)

*Sintaxis:*

<Resolution-parameter> := "0" | ... | "6"

### 11.2.2.9 Parámetro de contraseña (Password-parameter)

El Password-parameter se codifica como una STRING. De acuerdo con la definición de la Recomendación UIT-T T.30 (1994), el conjunto de caracteres de la cadena está limitado a los dígitos ("0".."9") y a los caracteres "+" y "-". La CA traducirá el carácter "\_" (subrayado) en un carácter de espacio IRA (20<sub>HEX</sub>).

*Sintaxis:*

<Password-parameter> := <Password-digit> {<Password-digit>}

<Password-digit> := "0" | ... | "9" | "\*" | "#"  
-- La longitud del <Password-parameter> está limitada a 20 octetos.

### 11.2.2.10 Parámetro de selección de interrogación (Poll-select-parameter)

El Poll-select-parameter se codifica como una STRING. De acuerdo con la definición de la Recomendación UIT-T T.30 (1994), el conjunto de caracteres de la cadena está limitado a los dígitos ("0".."9") y a los caracteres "+" y "-". La CA traducirá el carácter "\_" (subrayado) en un carácter de espacio IRA (20<sub>HEX</sub>).

*Sintaxis:*

<Poll-select-parameter> := <Poll-select-digit> {<Poll-select-digit>}

<Poll-select-digit> := "0" | ... | "9" | "\*" | "#"  
-- La longitud del <Poll-select-parameter> está limitada a 20 octetos.

## 11.3 Funcionalidad adicional

### 11.3.1 Función: Send y SendAck

Véanse los Cuadros 68 y 69.

### 11.3.2 Función: Receive

Véase el Cuadro 70.

### 11.3.3 Función: Poll (Interrogación)

Esta extensión tiene por objeto permitir a la LA recuperar una serie de documentos remitidos (posted) por un equipo facsímil distante.

Por ejemplo, si el usuario "Juan" envía documento a un equipo facsímil distante "Tomás", y "Tomás" ha remitido documentos para que sean recuperados por "Juan", la extensión "Polling" (interrogación) permite a "Juan" recibir los documentos de "Tomás" en la misma sesión de comunicación.

La <ExtendTDD>, ampliada con la <PollExtension>, se utiliza para recuperar específicamente los documentos remitidos por el equipo distante (véase el Cuadro 71).

**Funcionalidad adicional de la <SendTDD> para el servicio telefax del grupo 3**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<G3Speed>	B	o	I	G3SPEED	<G3-speed-parameter>	Velocidad más alta disponible	Velocidad de modulación. Si la CA no puede retornar la velocidad de modulación utilizada, dejará el campo vacío
<GenCil>	+	o	I	GENCIL	<Boolean-parameter>	"Yes" (Sí)	Pide a la CA que genere un CIL en el fichero transmitido
<HighRes>	+	o	I	HIGHRES	<Resolution-parameter>	"0"	Establece una resolución más alta cuando está fijado a valores mayores que "0". Véase también 11.2.2.8
<SubAddress>	+	o	I	SUBADDR	<Sub-address-parameter>	-	Subdirección del originador
<UseEcm>	+	o	I	USEECM	<Boolean-parameter>	"Yes" (Sí)	Indica si deberá o no utilizarse el modo corrección de error
Envío a un solo destinatario							
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	-	Especifica el número de llamada de un recipiente
<DoPoll>	+	o	I	DOPOLL	<Boolean-parameter>	"No"	Si está fijado a "Yes", se interroga al receptor con miras al envío de un documento
<PollPassword>	+	o	I	PASSWORD	<Password-parameter>	-	
<PollSelector>	+	o	I	SELECT	<Poll-select-parameter>	-	
Envío a uno o varios destinatarios							
<RecipientSpec>	+	m	I	ADDRESS	"@" <File-of-addrspec>	-	Especifica una lista de recipientes
Envío de un solo fichero							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	-	Indica el formato de transferencia del fichero saliente
<Document>	B	m	I	FILENAME	<File-parameter>	-	Fichero único a transmitir y entregar con el formato de transferencia especificado por la palabra clave Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Especifica el tipo de documento a enviar
<From>	+	o	I	FROM	NUMERIC-STRING	Primera página	Primera página que se enviará efectivamente; sólo es aplicable a los ficheros de texto
<To>	+	o	I	TO	NUMERIC-STRING	Última página	Última página que se enviará efectivamente; sólo es aplicable a los ficheros de texto
Envío de uno o varios ficheros							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	-	Especifica una lista de ficheros; utiliza una sintaxis especial

**Funcionalidad adicional de la <SendAckTDD> para el servicio telefax del grupo 3**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<Recipient>	B	m	I/O	ADDRESS	<Address-parameter>	–	Especifica el número de llamada de un recipiente
<G3Speed>	B	o	I/O	G3SPEED	<G3-speed-parameter>	Velocidad más alta disponible	Velocidad de modulación. Si la CA no puede retornar la velocidad de modulación utilizada, dejará el campo vacío
<GenCil>	+	o	I	GENCIL	<Boolean-parameter>	"Yes" (Sí)	Pide a la CA que genere un CIL en el fichero transmitido
<HighRes>	+	o	I	HIGHRES	<Resolution-parameter>	"0"	Establece una resolución más alta cuando está fijado a valores mayores que "0". Véase también 11.2.2.8
<SubAddress>	+	o	I	SUBADDR	<Sub-address-parameter>	–	Subdirección del originador
<UseEcm>	+	o	I/O	USEECM	<Boolean-parameter>	"Yes" (Sí)	Indica si deberá o no utilizarse el modo corrección de error
<DoPoll>	+	o	I	DO POLL	<Boolean-parameter>	"No"	Si está fijado a "Yes", se interroga al receptor con miras al envío de un documento
<PollPassword>	+	o	I	PASSWORD	<Password-parameter>	–	
<PollSelector>	+	o	I	SELECT	<Poll-select-parameter>	–	
Envío de un solo fichero							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Indica el formato de transferencia del fichero saliente
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichero único a transmitir y entregar con el formato de transferencia especificado por la palabra clave Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Especifica el tipo de documento a enviar
<From>	+	o	I	FROM	NUMERIC-STRING	Primera página	Primera página que se enviará efectivamente; sólo es aplicable a los ficheros de texto
<To>	+	o	I	TO	NUMERIC-STRING	Última página	Última página que se enviará efectivamente; sólo es aplicable a los ficheros de texto
Envío de uno o varios ficheros							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Especifica una lista de ficheros; utiliza una sintaxis especial

**Funcionalidad adicional de la <ReceiveTDD> para el servicio telefax del grupo 3**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<CvFax3>	B	o	I	CVFAX3	<Convert-id-parameter>	"TIFF"	Formato de transferencia deseado por la LA
<Originator>	B	o	O	ADDRESS	<Address-parameter>	–	Especifica el número de teléfono del originador
<G3Speed>	+	o	O	G3SPEED	<G3-speed-parameter>	–	Velocidad de modulación. Si la CA no puede retornar la velocidad de modulación utilizada, dejará el campo vacío
<SubAddress>	+	o	I/O	SUBADDR	<Sub-address-parameter>	–	Subdirección del destinatario; si está especificada en la petición, se utiliza como un selector para la recuperación
Recepción de un solo fichero							
<Convert>	B	m	O	CONVERT	<Convert-id-parameter>	–	Indica el formato de transferencia del fichero recibido
<Document>	B	m	I/O	FILENAME	<File-parameter>	–	El nombre del fichero puede haber sido fijado previamente por la LA. Si se recibe un solo fichero, la CA retendrá el nombre. Si se reciben varios ficheros, la CA puede escribir "aplastando" el nombre anteriormente escrito
<Type>	B	o	O	TYPE	<Type-id-parameter>	–	Especifica el tipo de documento recibido
Recepción de varios ficheros							
<DocumentSpec>	+	m	O	FILENAME	"@" <File-of-filespec>	–	Especifica una lista de ficheros; utiliza una sintaxis especial

**11.4 Fijaciones del descriptor de CA**

Una CA que admite el servicio FX3 especificará el identificador de tipo (formato de transmisión) en el descriptor de CA (véase 9.5).

Para el soporte de la TDD de interrogación, la palabra clave EXTEND del descriptor de CA deberá fijarse a los valores adecuados. El Cuadro 72 muestra los posibles valores.

CUADRO 71/T.611

**Elementos de sintaxis de <PollExtension> para el servicio telefax del grupo 3**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<PollSubFunction>	B	m	I	SUBFUNC	"Poll"	–	La CA generará una respuesta
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	–	Especifica el número de llamada de un recipiente
<PollPassword>	B	m	I	PASSWORD	<Password-parameter>	–	
<PollSelector>	B	o	I	SELECT	<Poll-select-parameter>		
<SendTime>	B	o	I	SENDTIME	<Send-time-parameter>	"IMMEDIATE"	La CA procesará efectivamente la petición en el tiempo especificado
<ComId>	B	o	O	COMID	<Com-id-parameter>	–	Identificación de la comunicación (calculada por la CA)

CUADRO 72/T.611

**Valores adicionales del descriptor de CA para el servicio telefax del grupo 3**

Palabra clave	Parámetro	CA declara
FX3	"STD"	El soporte del servicio telefax básico (MH)
FX3	"BTM"	El soporte del modo transparente básico
FX3	"DTM"	El soporte del modo transparente al documento
FX3	"BFT"	El soporte de la transferencia de ficheros binarios
FX3	"EDI"	El soporte del intercambio electrónico de datos
EXTEND	"Poll"	El soporte de la funcionalidad Poll (interrogación)

## 12 Servicio: Telefax del grupo 4

### 12.1 Elementos de sintaxis específicos del servicio

```

<ServiceDependentKeywordsSend> :=
    (<Recipient> | <RecipientSpec>)
    ((<Document> <Convert> [<Type>] [<Name>] [<UserInfo>] [<Prolog>] [From]
    [To]) | <DocumentSpec>) [<SubAddress>] [<HighRes>]

<ServiceDependentKeywordsSendAck> :=
    <Recipient>
    ((<Document> <Convert> [<Type>] [<Name>] [<UserInfo>] [<Prolog>] [<Cil>]
    [From] [To]) | <DocumentSpec>) [<SubAddress>] [<HighRes>]

<ServiceDependentKeywordsReceive> :=
    [<Originator>]
    ((<Document> <Convert> <Type> [<Name>] [<UserInfo>] [<Prolog>] [<Cil>]
    [<FirstPg>]) | <DocumentSpec>) [<CvFax4>] [<SubAddress>]
  
```

Véase el Cuadro 73.

CUADRO 73/T.611

#### Elementos de sintaxis adicionales para el servicio telefax del grupo 4

Elemento de sintaxis	Finalidad
<Cil>	Especifica el contenido del renglón de identificación de la llamada, definido en la Recomendación UIT-T F.200. Véase también G.1
<Convert>	Especifica el formato de transferencia que va a utilizarse
<CvFax4>	Especifica el formato de transferencia deseado para los ficheros recibidos
<Document>, <DocumentSpec>	Especifica el documento o documentos que se van a enviar o que se están recibiendo
<FirstPg>	Especifica el número de la primera página recibida
<From>	La transmisión debe comenzar en el número de página especificado
<HighRes>	Indica la resolución
<Name>	Asigna un nombre al documento que se va a enviar, o indica el nombre del documento recibido, como se define en la Recomendación UIT-T T.571
<Originator>	Indica la dirección de comunicaciones del originador
<Prolog>	Trayecto hasta el documento "prólogo". Este es un documento, codificado de acuerdo con la Recomendación UIT-T T.61, que se añade al principio del documento principal que se va a enviar
<Recipient>, <RecipientSpec>	Especifica la dirección o direcciones del recipiente o recipientes
<SubAddress>	Especifica la subdirección del originador
<To>	La transmisión debe terminar en el número de página especificado
<Type>	Especifica el formato de transmisión utilizado
<UserInfo>	Especifica un comentario asociado a un documento que se va a enviar o recibir. El comentario se transmite a través del servicio de telecomunicaciones

## 12.2 Codificación basada en texto

### 12.2.1 Correspondencia de las palabras clave

Véase el Cuadro 74.

CUADRO 74/T.611

**Codificación basada en texto de los elementos de sintaxis adicionales para el servicio telefax del grupo 4**  
(El símbolo ↵ representa el determinante de formato de nueva línea)

Elemento de sintaxis	Par palabra clave/Parámetro
<Cil>	"CIL" ":" <Cil-parameter> ↵
<Convert>	"CONVERT" ":" <Convert-id-parameter> ↵
<CvFax4>	"CVFAX4" ":" <Convert-id-parameter> ↵
<Document>	"FILENAME" ":" <File-parameter> ↵
<DocumentSpec>	"FILENAME" ":" "@" <File-of-filespec> ↵
<FirstPg>	"FIRSTPG" ":" NUMERIC-STRING ↵
<From>	"FROM" ":" NUMERIC-STRING ↵
<HighRes>	"HIGHRES" ":" <Resolution-parameter> ↵
<Name>	"NAME" ":" STRING (SIZE(1..12)) ↵
<Originator>	"ADDRESS" ":" <Address-parameter> ↵
<Prolog>	"PROLOG" ":" <Path-parameter> ↵
<Recipient>	"ADDRESS" ":" <Address-parameter> ↵
<RecipientSpec>	"ADDRESS" ":" "@" <File-of-addrspec> ↵
<SubAddress>	"SUBADDR" ":" <Sub-address-parameter> ↵
<To>	"TO" ":" NUMERIC-STRING ↵
<Type>	"TYPE" ":" <Type-id-parameter> ↵
<UserInfo>	"USERINFO" ":" STRING (SIZE(1..12)) ↵

### 12.2.2 Codificación de parámetros

Véase también 6.4.4 para la codificación de los parámetros que no dependen del servicio.

#### 12.2.2.1 Parámetro identificador de servicio

El parámetro Service-id-parameter se codifica como una STRING fijada al valor constante "FX4".

*Sintaxis:*

<Service-id-parameter> := "FX4"

### 12.2.2.2 Parámetro identificador de tipo

El parámetro Type-id-parameter se codifica como una STRING fijada a uno de los siguientes valores:

"STD"	Servicio telefax G4 básico (MR)
"DTM"	Transferencia de ficheros telemáticos (TFT) del servicio telefax G4: modo transparente al documento
"EDI"	Transferencia de ficheros telemáticos (TFT) del servicio telefax G4: Edifact
"BFT"	Servicio telefax G4: transferencia de ficheros binarios

NOTA – El formato de transmisión STD (= código Read modificado, MC) se define en la Recomendación UIT-T T.6; los formatos TFT DTM y EDI se definen en la Recomendación UIT-T T.571; y el formato de transmisión BFT se define en la Recomendación UIT-T T.434.

*Sintaxis:*

<Type-id-parameter> := "STD" | "DTM" | "BFT" | "EDI"

### 12.2.2.3 Parámetro identificador de conversión

El parámetro Convert-id-parameter se codifica como una STRING fijada a uno de los siguientes valores:

"ASCII"	ASCII ampliado de APPLI/COM (para ficheros salientes solamente)
"ASCII437"	ASCII ampliado de APPLI/COM (para ficheros salientes solamente)
"T.50"	ASCII estándar de APPLI/COM (para ficheros salientes solamente)
"T.61"	Formato de transferencia T.61 de APPLI/COM
"TIFF"	Formato de transferencia TIFF definido en 8.3
"TIFF2"	(Para ficheros entrantes solamente.) Clase 2 de TIFF de APPLI/COM especificada en 8.3
"TIFF4"	(Para ficheros entrantes solamente.) Clase 4 de TIFF de APPLI/COM especificada en 8.3
"VOID"	No hay conversión

NOTA – Si una LA solicita un documento telefax que ha sido recibido, es posible:

- obtener el documento en la clase 2 de TIFF de APPLI/COM, especificando TIFF2;
- obtener el documento en la clase 4 de TIFF de APPLI/COM, especificando TIFF4;

a condición de que la CA pueda generar ese formato TIFF especial. Si sólo se especifica TIFF, el documento se entregará en la clase por defecto de TIFF de APPLI/COM (la clase 1). Sin embargo, en el sentido "envío" basta con especificar TIFF solamente (sin número agregado), ya que la información de compresión está contenida en el propio formato de transferencia.

La utilización del parámetro identificador de conversión depende también del identificador de tipo seleccionado. Esto se muestra en el Cuadro 75.

CUADRO 75/T.611

#### Asignaciones de identificador de conversión permitidas que dependen del identificador de tipo y del sentido del tráfico

Identificador de tipo	Identificador de conversión para tráfico saliente	Identificador de conversión para tráfico entrante
STD	ASCII, ASCIIxxx <sup>a)</sup> , T.50, TIFF	TIFF, TIFFx <sup>b)</sup>
DTM, BFT, EDI	VOID	
<p>a) xxx representa una página de códigos, declarada en el ICE, por ejemplo ASCII437 si se había declarado la página de códigos 437.</p> <p>b) x representa la clase TIFF que habrá de leerse, los valores posibles son 2 ó 4.</p>		

*Sintaxis:*

```
<Convert-id-parameter> := <Convert-std> | <Convert-bin> | <Convert-txt>
<Convert-std> := <Convert-std-in> | <Convert-std-out>
<Convert-std-in> := "TIFF" | "TIFF2" | "TIFF4"
-- documentos entrantes
<Convert-std-out> := <Ascii> | "T.50" | "TIFF"
-- documentos salientes
<Convert-txt> := <Ascii> | "T.50" | "T.61"
<Ascii> := "ASCII" | STRING ("ASCII" + <Code-page>)
<Code-page> := <digit> <digit> <digit>
<digit> := "0" | ... | "9"
<Convert-bin> := "VOID"
```

#### 12.2.2.4 Parámetro de fichero de especificaciones de direcciones (File-of-addrspec) y parámetro de dirección (Address-parameter)

El parámetro File-of-addrspec se codifica como un PATH que apunta a un fichero que contiene parámetros Addr-spec-parameters, que a su vez contiene el Address-parameter.

El parámetro Address-parameter se codifica como una STRING. Esta consiste en el identificador del terminal del recipiente, definido en la Recomendación UIT-T F.184. La parte nemónica del identificador de terminal puede omitirse o especificarse como un signo de cierre de interrogación ("?"), en cuyo caso no se verifica la dirección del abonado. Véase también G.2.

Como otra posibilidad, puede darse un nombre de alias en lugar del identificador de terminal, a condición de que el alias se introduzca por un carácter "&". Se supone que la CA sabe cómo decodificar el alias especificado. El uso típico de esta característica es cuando la CA realiza una guía telefónica.

*Sintaxis:*

```
<File-of-addrspec> := PATH
-- El trayecto apunta a un fichero que contiene uno o más
-- <Addr-spec-parameter>s
<Addr-spec-parameter> := <Address-parameter>
["," <Cover-path> "," <Cover-conv> ["," <Cover-type>] ["," <Nopgbrk>]]
<Address-parameter> := <Terminal-id> | ("&" <Alias>)
<Terminal-id> := STRING
-- La cadena presentará el identificador de terminal definido en la
-- Recomendación UIT-T F.184
<Alias> := STRING
<Cover-path> := PATH
-- Trayecto a un fichero que contiene una portada para un destinatario específico
<Cover-conv> := <Convert-id-parameter>
-- Especifica el formato de transferencia del fichero de portada
<Cover-type> := <Type-id-parameter>
-- Especifica el formato de transmisión de la portada
<Nopgbrk> := <Boolean-parameter>
-- Si está fijado a verdadero ("Yes"), no habrá corte de página
-- entre la portada y el documento principal
```

### 12.2.2.5 Parámetro de fichero de especificaciones de ficheros (File-of-filespec) y parámetro de fichero (File-parameter)

Parámetro de fichero de especificaciones de ficheros (File-of-filespec) y parámetro de fichero (File-parameter)

*Sintaxis:*

```
<File-of-filespec> :=          PATH
-- El trayecto apunta a un fichero que contiene uno o más
-- <Filespec-parameter>s

<Filespec-parameter> :=      <File-parameter> "," <File-conv> ["," [<File-type>]
["," <Send-parameter> | <SendAck-parameter> | <Receive-parameter>] ]

<Send-parameter> :=          [<File-name>] ["," <File-userinfo>]

<SendAck-parameter> :=       [<File-name>] ["," [<File-userinfo>] ["," <File-cil>] ]

<Receive-parameter> :=       [<File-name>] ["," [<File-userinfo>] ["," <File-cil>] ]

<File-parameter> :=          PATH
-- Trayecto del fichero transferido

<File-conv> :=                <Convert-id-parameter>
-- Especifica el formato de transferencia del fichero

<File-type> :=                <Type-id-parameter>
-- Especifica el formato de transmisión del fichero

<File-name> :=                STRING (SIZE(1..12))
-- Especifica el nombre del fichero que se va a transmitir

<File-userinfo> :=            STRING (SIZE(1..12))
-- Especifica información adicional de usuario a usuario que se va a transmitir

<File-cil> :=                 Cil-parameter
-- Véase 12.2.2.6
```

### 12.2.2.6 Parámetro de renglón de identificación de la llamada

El parámetro de renglón de identificación de la llamada (Cil-parameter) es una cadena de 72 caracteres definida en la Recomendación UIT-T F.200 y tiene la siguiente composición (véase también G.1):

Identificador de terminal de recibiente	Longitud: 24 caracteres
Separador	El carácter "/" (2F <sub>HEX</sub> )
Identificador de terminal de originador	Longitud: 24 caracteres
Separador	El carácter "/" (2F <sub>HEX</sub> )
Fecha y hora locales de originador	Longitud: 14 caracteres, formato: "YY-MM-DD-HH:MM"
Separador	El carácter "/" (2F <sub>HEX</sub> )
Información de referencia	Longitud: 7 caracteres

*Sintaxis:*

```
<Cil-parameter> :=          <Recipient-tid> "/" <Originator-tid> "/" <Local-time> "/" <Ref-info>

<Recipient-tid> :=          STRING (SIZE(24))
-- La cadena representa el terminal-id definido en la
-- Recomendación UIT-T F.184

<Originator-tid> :=         STRING (SIZE(24))
-- La cadena representa el terminal-id definido en la
-- Recomendación UIT-T F.184

<Local-time> :=             <Date-time-parameter>

<Ref-info> :=               STRING (SIZE(7))
```

### 12.2.2.7 Parámetro identificador de subdirección

El parámetro de subdirección Sub-address-parameter se codifica como una NUMERIC-STRING que representa una subdirección.

*Sintaxis:*

<Sub-address-parameter> := NUMERIC-STRING (SIZE(1..4))

### 12.2.2.8 Parámetro de resolución

El Resolution-parameter se codifica como una NUMERIC-STRING fijada a uno de los siguientes valores:

"0"	200 dpi
"1"	240 dpi
"2"	300 dpi
"3"	400 dpi
"4"	400 dpi

*Sintaxis:*

<Resolution-parameter> := "0" | ... | "4"

## 12.3 Funcionalidad adicional

### 12.3.1 Función: Send y SendAck

Véanse los Cuadros 76 y 77.

### 12.3.2 Función: Receive

Véase el Cuadro 78.

## 12.4 Fijaciones del descriptor de CA

Una CA que admite el servicio FX4 especificará el identificador de tipo (formato de transmisión de documento) soportado en el descriptor de CA (véase 9.5). El Cuadro 79 muestra las posibles asignaciones.

## 13 Servicio: Teletex

### 13.1 Elementos de sintaxis específicos del servicio

<ServiceDependentKeywordsSend> :=  
(<Recipient> | <RecipientSpec>)  
(((<Document> <Convert> [<Type>] [<Name>] [<UserInfo>] [<Prolog>] [From]  
[To] [T61Options]) | <DocumentSpec>) [<SubAddress>]

<ServiceDependentKeywordsSendAck> :=  
<Recipient>  
(((<Document> <Convert> [<Type>] [<Name>] [<UserInfo>] [<Prolog>] [<Cil>]  
[From] [To] [T61Options]) | <DocumentSpec>) [<SubAddress>]

<ServiceDependentKeywordsReceive> :=  
[<Originator>]  
(((<Document> <Convert> <Type> [<Name>] [<UserInfo>] [<Prolog>] [<Cil>]  
[<FirstPg>]) | <DocumentSpec>) [<CvTtx>] [<SubAddress>]

Véase el Cuadro 80.

CUADRO 76/T.611

**Funcionalidad adicional de la <SendTDD> para el servicio telefax del grupo 4**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<HighRes>	+	o	I	HIGHRES	<Resolution-parameter>	"0"	Establece una resolución más alta cuando está fijado a valores mayores que "0". Véase también 12.2.2.8
<SubAddress>	+	o	I	SUBADDR	<Sub-address-parameter>	-	Subdirección del originador
Envío a un solo destinatario							
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	-	Especifica el número de llamada de un recipiente
Envío a uno o varios destinatarios							
<RecipientSpec>	+	m	I	ADDRESS	"@" <File-of-addrspec>	-	Especifica una lista de recipientes
Envío de un solo fichero							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	-	Indica el formato de transferencia del fichero saliente
<Document>	B	m	I	FILENAME	<File-parameter>	-	Fichero único a transmitir y entregar con el formato de transferencia especificado por la palabra clave Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Especifica el tipo de documento a enviar
<Name>	+	o	I	NAME	STRING (SIZE(1..12))	-	Asigna un "nombre" a los documentos que se van a enviar. Longitud máxima: 12 caracteres. Corresponde al parámetro de nombre de fichero en TFT (véase la Rec. UIT-T T.571)
<UserInfo>	+	o	I	USERINFO	STRING (SIZE(1..12))	-	Asigna un comentario de usuario al documento. Este comentario se transmite con el documento. Longitud máxima: 12 caracteres
<Prolog>	+	o	I	PROLOG	PATH	-	Nombre de fichero del "documento de control" (trayecto completo)
<From>	+	o	I	FROM	NUMERIC-STRING	Primera página	Primera página que se enviará efectivamente; sólo es aplicable a los ficheros de texto
<To>	+	o	I	TO	NUMERIC-STRING	Última página	Última página que se enviará efectivamente; sólo es aplicable a los ficheros de texto
Envío de uno o varios ficheros							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	-	Especifica una lista de ficheros; utiliza una sintaxis especial

**Funcionalidad adicional de la <SendAckTDD> para el servicio telefax del grupo 4**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	–	Especifica el número de llamada de un recipiente
<HighRes>	+	o	I	HIGHRES	<Resolution-parameter>	"0"	Establece una resolución más alta cuando está fijado a valores mayores que "0". Véase también 12.2.2.8
<SubAddress>	+	o	I	SUBADDR	<Sub-address-parameter>	–	Subdirección del originador
Envío de un solo fichero							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Indica el formato de transferencia del fichero saliente
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichero único a transmitir y entregar con el formato de transferencia especificado por la palabra clave Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Especifica el tipo de documento a enviar
<Cil>	B	o	O	CIL	<Cil-parameter>	–	Renglón de identificación de la llamada construido por la CA (véase la Rec. UIT-T F.200)
<Name>	+	o	I	NAME	STRING (SIZE(1..12))	–	Asigna un "nombre" a los documentos que se van a enviar. Longitud máxima: 12 caracteres. Corresponde al parámetro de nombre de fichero en TFT (véase la Rec. UIT-T T.571)
<UserInfo>	+	o	I	USERINFO	STRING (SIZE(1..12))	–	Asigna un comentario de usuario al documento. Este comentario se transmite con el documento
<Prolog>	+	o	I	PROLOG	PATH	–	Nombre de fichero del "documento de control" (trayecto completo)
<From>	+	o	I	FROM	NUMERIC-STRING	Primera página	Primera página que se enviará efectivamente; sólo es aplicable a los ficheros de texto
<To>	+	o	I	TO	NUMERIC-STRING	Última página	Última página que se enviará efectivamente; sólo es aplicable a los ficheros de texto
Envío de uno o varios ficheros							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Especifica una lista de ficheros; utiliza una sintaxis especial

**Funcionalidad adicional de la <ReceiveTDD> para el servicio telefax del grupo 4**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<Originator>	B	o	O	ADDRESS	<Address-parameter>	--	Especifica el número de teléfono del originador
<CvFax4>	B	o	I	CVFAX4	<Convert-id-parameter>	"TIFF"	Formato de transferencia deseado por la LA
<SubAddress>	+	o	I/O	SUBADDR	<Sub-address-parameter>	--	Subdirección del recipiente; si está especificada en la petición, se utiliza como un selector para la recuperación
Recepción de un solo fichero							
<Convert>	B	m	O	CONVERT	<Convert-id-parameter>	--	Indica el formato de transferencia del fichero recibido
<Document>	B	m	I/O	FILENAME	<File-parameter>	--	El nombre del fichero puede haber sido fijado previamente por la LA. Si se recibe un solo fichero, la CA retendrá el nombre. Si se reciben varios ficheros, la CA puede escribir "aplastando" el nombre anteriormente escrito
<Type>	B	o	O	TYPE	<Type-id-parameter>	--	Especifica el tipo de documento recibido
<Cil>	B	o	O	CIL	<Cil-parameter>	--	Renglón de identificación de la llamada construido por la CA (véase la Rec. UIT-T F.200)
<Name>	+	o	O	NAME	STRING (SIZE(1..12))	--	Retorna el nombre del fichero entrante (longitud máxima: 12 caracteres)
<UserInfo>	+	o	O	USERINFO	STRING (SIZE(1..12))	--	Asigna un comentario de usuario al documento. Este comentario se transmite con el documento. Longitud máxima: 12 caracteres
<FirstPg>	+	o	O	FIRSTPG	NUMERIC-STRING	--	Número de la primera página recibida
<Prolog>	+	o	I	PROLOG	PATH	--	Nombre de fichero del "documento de control"
Recepción de varios ficheros							
<DocumentSpec>	+	m	O	FILENAME	"@" <File-of-filespec>	--	Especifica una lista de ficheros; utiliza una sintaxis especial

CUADRO 79/T.611

**Valores adicionales del descriptor de CA para el servicio telefax del grupo 4**

Palabra clave	Parámetro	CA declara
FX4	"STD"	El soporte del servicio telefax básico
FX4	"OPD"	El soporte de la transmisión de documento operador
FX4	"MD"	El soporte de la transmisión de documento de supervisión
FX4	"CTL"	El soporte de la transmisión de documento de control
FX4	"DTM"	El soporte del modo transparente al documento
FX4	"BFT"	El soporte de la transferencia de ficheros binarios
FX4	"EDI"	El soporte del intercambio electrónico de datos

CUADRO 80/T.611

**Elementos de sintaxis adicionales para el servicio teletex**

Elemento de sintaxis	Finalidad
<Cil>	Especifica el contenido del renglón de identificación de la llamada, definido en la Recomendación UIT-T F.200. Véase también G.1
<Convert>	Especifica el formato de transferencia que va a utilizarse
<CvTx>	Especifica el formato de transferencia deseado para los ficheros recibidos
<Document>, <DocumentSpec>	Especifica el documento o los documentos que se van a enviar o que se están recibiendo
<FirstPg>	Especifica el número de la primera página recibida
<From>	La transmisión debe comenzar en el número de página especificado
<Name>	Asigna un nombre al documento que se va a enviar o indica el nombre del documento recibido, como se define en la Recomendación UIT-T T.571
<Originator>	Indica la dirección de comunicaciones del originador
<Prolog>	Trayecto al documento "prólogo"
<Recipient>, <RecipientSpec>	Especifica la dirección o direcciones del recipiente o recipientes
<SubAddress>	Especifica la subdirección del originador
<T61Options>	Indica las opciones específicas de la codificación T.61
<To>	La transmisión debe terminar en el número de página especificado
<Type>	Especifica el formato de transmisión utilizado
<UserInfo>	Especifica un comentario asociado a un documento que se va a enviar o recibir. El comentario se transmite a través del servicio de telecomunicaciones

## 13.2 Codificación basada en texto

### 13.2.1 Correspondencia de las palabras clave

Véase el Cuadro 81.

CUADRO 81/T.611

**Codificación basada en texto de los elementos de sintaxis adicionales para el servicio teletex**  
(El símbolo ↵ representa el determinante de formato de nueva línea)

Elemento de sintaxis	Par palabra clave/Parámetro
<Cil>	"CIL" ":" <Cil-parameter> ↵
<Convert>	"CONVERT" ":" <Convert-id-parameter> ↵
<CvTtx>	"CVTTX" ":" <Convert-id-parameter> ↵
<Document>	"FILENAME" ":" <File-parameter> ↵
<DocumentSpec>	"FILENAME" ":" "@" <File-of-filespec> ↵
<FirstPg>	"FIRSTPG" ":" NUMERIC-STRING ↵
<From>	"FROM" ":" NUMERIC-STRING ↵
<Name>	"NAME" ":" (SIZE (1..12)) ↵
<Originator>	"ADDRESS" ":" <Address-parameter> ↵
<Prolog>	"PROLOG" ":" <Path-parameter> ↵
<Recipient>	"ADDRESS" ":" <Address-parameter> ↵
<RecipientSpec>	"ADDRESS" ":" "@" <File-of-addrspec> ↵
<SubAddress>	"SUBADDR" ":" <Sub-address-parameter> ↵
<T61Options>	"T61OPTS" ":" <T61-options-parameter> ↵
<To>	"TO" ":" NUMERIC-STRING ↵
<Type>	"TYPE" ":" <Type-id-parameter> ↵
<UserInfo>	"USERINFO" ":" STRING (SIZE (1..12)) ↵

### 13.2.2 Codificación de parámetros

Véase también 6.4.4 para la codificación utilizada para los parámetros que no dependen del servicio.

#### 13.2.2.1 Parámetro identificador de servicio

El parámetro Service-id-parameter se codifica como una STRING fijada al valor constante "TTX".

*Sintaxis:*

<Service-id-parameter> := "TTX"

### 13.2.2.2 Parámetro identificador de tipo

El parámetro Type-id-parameter se codifica como una STRING fijada a uno de los siguientes valores:

"STD"	Servicio teletex básico (T.61)
"OPD"	Servicio teletex básico: documento operador
"MD"	Servicio teletex básico: documento de supervisión
"CTL"	Servicio teletex básico: documento de control
"DTM"	Transferencia de ficheros telemáticos (TFT) del servicio teletex: modo transparente al documento
"EDI"	Transferencia de ficheros telemáticos (TFT) del servicio teletex: Edifact
"BFT"	Servicio teletex: transferencia de ficheros binarios

NOTA – El formato de transmisión STD se define en la Recomendación UIT-T T.61; los tipos de documento OPD, MD y CTL se definen en la Recomendación UIT-T T.62 (Anexo E); los formatos TFT DTM y EDI se definen en la Recomendación UIT-T T.571; y el formato de transmisión BFT se define en la Recomendación UIT-T T.434.

*Sintaxis:*

<Type-id-parameter> := "STD" | "OPD" | "MD" | "CTL" | "DTM" | "BFT" | "EDI"

### 13.2.2.3 Parámetro identificador de conversión

El parámetro Convert-id se codifica como una STRING fijada a uno de los siguientes valores:

"ASCII"	ASCII ampliado de APPLI/COM
"ASCII437"	ASCII ampliado de APPLI/COM
"T.50"	ASCII estándar de APPLI/COM
"T.61"	Codificación del servicio teletex básico definido en la Recomendación UIT-T T.61
"VOID"	No hay conversión

La utilización del parámetro identificador de conversión depende también del identificador de tipo seleccionado. Esto se muestra en el Cuadro 82.

CUADRO 82/T.611

#### Asignaciones de identificador de conversión permitidas que dependen del identificador de tipo

Identificador de tipo (Type-id)	Identificador de conversión (Convert-id) permitido para tráfico entrante y saliente
STD, OPD, MD, CTL	ASCII, ASCII <sup>a)</sup> , T.50, T.61
DTM, BFT, EDI	VOID
<p>a) xxx significa una página de códigos, declarada en el ICE, por ejemplo ASCII437 si se ha declarado la página de códigos 437.</p>	

*Sintaxis:*

<Convert-id-parameter> := <Convert-txt> | <Convert-bin>  
 <Convert-txt> := <Ascii> | "T.50" | "T.61"  
 <Ascii> := "ASCII" | STRING ("ASCII" + <Code-page>)  
 <Code-page> := <digit> <digit> <digit>  
 <digit> := "0" | ... | "9"  
 <Convert-bin> := "VOID"

#### 13.2.2.4 Parámetro de fichero de especificaciones de direcciones (File-of-addrspec) y parámetro de dirección (Address-parameter)

El parámetro File-of-addrspec se codifica como un PATH que apunta a un fichero que contiene parámetros Addrspec-parameters, que a su vez contiene el Address-parameter.

El parámetro Address se codifica como una STRING. La STRING consiste en el identificador del terminal del recipiente definido en la Recomendación UIT-T F.200. La parte nemónica del identificador de terminal puede omitirse o especificarse como un signo de cierre de interrogación ("?"), en cuyo caso no se verifica la dirección del abonado. Véase también el anexo G.2.

Como otra posibilidad, la presencia del número de llamada como un número "bruto" de marcación puede indicarse insertando el signo de cierre de admiración ("!") inmediatamente antes del primer dígito. Esto significa que el equipo llamante tiene que marcar los dígitos que siguen al signo de admiración tal como aparecen, sin interpretación.

Además, puede darse un nombre de alias en lugar del identificador de terminal, a condición de que se introduzca por un carácter "&". Se supone que la CA sabe cómo decodificar el alias especificado.

*Sintaxis:*

```
<File-of-addrspec> :=          PATH
-- El trayecto apunta a un fichero que contiene uno o más
-- <Addrsec-parameter>s

<Addrsec-parameter> :=       <Address-parameter>

<Address-parameter> :=       <Terminal-id> | ("!" <Dial-sequence>) | ("&" <Alias>)

<Terminal-id> :=              STRING
-- La cadena presentará el identificador de terminal definido en la
-- Recomendación UIT-T F.200

<Dial-sequence> :=           NUMERIC-STRING

<Alias> :=                     STRING
```

#### 13.2.2.5 Parámetro de fichero de especificaciones de ficheros (File-of-filespec) y parámetro de fichero (File-parameter)

El parámetro File-of-filespec se codifica como un PATH que apunta a un fichero que contiene parámetros Filespec-parameter, que contiene el File-parameter. El File-parameter en sí se codifica también como un PATH, que apunta al fichero transferido.

*Sintaxis:*

```
<File-of-filespec> :=          PATH
-- El trayecto apunta a un fichero que contiene uno o más
-- <Filespec-parameter>s

<Filespec-parameter> :=       <File-parameter> ", " <File-conv> [ ", " [<File-type>]
[" , " <Send-parameter> | <SendAck-parameter> | <Receive-parameter> ] ]

<Send-parameter> :=           [<File-name>] [ ", " <File-userinfo> ]

<SendAck-parameter> :=        [<File-name>] [ ", " [<File-userinfo>] [ ", " <File-cil> ] ]

<Receive-parameter> :=        [<File-name>] [ ", " [<File-userinfo>] [ ", " <File-cil> ] ]

<File-parameter> :=           PATH
-- Trayecto del fichero transferido

<File-conv> :=                 <Convert-id-parameter>
-- Especifica el formato de transferencia del fichero

<File-type> :=                 <Type-id-parameter>
-- Especifica el formato de transmisión del fichero

<File-name> :=                 STRING (SIZE(1..12))
-- Especifica el nombre del fichero a transmitir
```

<File-userinfo> := STRING (SIZE(1..12))  
 -- Especifica más información de usuario a usuario a transmitir

<File-cil> := Cil-parameter  
 -- Véase 13.2.2.6

### 13.2.2.6 Parámetro de renglón de identificación de la llamada

El parámetro Cil (*call identification line*) es una cadena de 72 caracteres definida en la Recomendación UIT-T F.200, y tiene la siguiente composición (véase también G.1):

Identificador de terminal de recibiente	Longitud: 24 caracteres
Separador	El carácter "/" (2F <sub>HEX</sub> )
Identificador de terminal de originador	Longitud: 24 caracteres
Separador	El carácter "/" (2F <sub>HEX</sub> )
Fecha y hora locales de originador	Longitud: 14 caracteres, formato: "YY-MM-DD-HH:MM"
Separador	El carácter "/" (2F <sub>HEX</sub> )
Información de referencia	Longitud: 7 caracteres

*Sintaxis:*

<Cil-parameter> := <Recipient-tid> "/" <Originator-tid> "/" <Local-time> "/" <Ref-info>

<Recipient-tid> := STRING (SIZE(24))  
 -- La cadena representa el terminal-id definido en la  
 -- Recomendación UIT-T F.184

<Originator-tid> := STRING (SIZE(24))  
 -- La cadena representa el terminal-id definido en la  
 -- Recomendación UIT-T F.184

<Local-time> := <Date-time-parameter>

<Ref-info> := STRING (SIZE(7))

### 13.2.2.7 Parámetro identificador de subdirección

El parámetro de subdirección Sub-address-parameter se codifica como una NUMERIC-STRING que representa una subdirección.

*Sintaxis:*

<Sub-address-parameter> := NUMERIC-STRING (SIZE(1..4))

### 13.2.2.8 Parámetro de opciones T.61

Queda en estudio.

*Sintaxis:*

<T61-options-parameter> := STRING

## 13.3 Funcionalidad adicional

### 13.3.1 Función: Send y SendAck

Véanse los Cuadros 83 y 84.

### 13.3.2 Función: Receive

Véase el Cuadro 85.

**Funcionalidad adicional de la <SendTDD> para el servicio teletex**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<SubAddress>	+	o	I	SUBADDR	<Sub-address-parameter>	–	Subdirección del originador
Envío a un solo destinatario							
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	–	Especifica el número de llamada de un recipiente
Envío a uno o varios destinatarios							
<RecipientSpec>	+	m	I	ADDRESS	"@" <File-of-addrspec>	–	Especifica una lista de recipientes
Envío de un solo fichero							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Formato de transferencia del fichero saliente
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichero único a transmitir
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Especifica el tipo de documento a enviar
<Name>	+	o	I	NAME	STRING (SIZE(1..12))	–	Parámetro de nombre de fichero en TFT (Rec. UIT-T T.571)
<UserInfo>	+	o	I	USERINFO	STRING (SIZE(1..12))	–	Userinfo se transmite con el documento
<Prolog>	+	o	I	PROLOG	PATH	–	Trayecto al documento de control (Rec. UIT-T T.62)
<From>	+	o	I	FROM	NUMERIC-STRING	Primera página	Primera página efectivamente enviada; sólo es aplicable a los ficheros de texto
<To>	+	o	I	TO	NUMERIC-STRING	Última página	Última página efectivamente enviada; sólo es aplicable a los ficheros de texto
<T61Options>	+	o	I	T61OPTS	<T61-options-parameter>	–	
Envío de uno o varios ficheros							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Especifica una lista de ficheros; utiliza una sintaxis especial

CUADRO 84/T.611

**Funcionalidad adicional de la <SendTDD> para el servicio teletex**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	–	Especifica el número de llamada de un recipiente
<SubAddress>	+	o	I	SUBADDR	<Sub-address-parameter>	–	Subdirección del originador
Envío de un solo fichero							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Formato de transferencia del fichero saliente
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichero único a transmitir y entregar con el formato de transferencia especificado por la palabra clave Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Especifica el tipo de documento a enviar
<Cil>	B	o	O	CIL	<Cil-parameter>	–	Renglón de identificación de la llamada construido por la CA (véase la Rec. UIT-T F.200)
<Name>	+	o	I	NAME	STRING (SIZE(1..12))	–	Asigna un nombre a los ficheros que se van a enviar. Corresponde al parámetro de nombre de fichero en TFT (véase la Rec. UIT-T T.571)
<UserInfo>	+	o	I	USERINFO	STRING (SIZE(1..12))	–	Asigna un comentario de usuario al documento. El comentario se transmite con el documento
<Prolog>	+	o	I	PROLOG	PATH	–	Nombre de fichero del "documento de control" (trayecto completo)
<From>	+	o	I	FROM	NUMERIC-STRING	Primera página	Primera página efectivamente enviada; sólo es aplicable a los ficheros de texto
<To>	+	o	I	TO	NUMERIC-STRING	Última página	Última página efectivamente enviada; sólo es aplicable a los ficheros de texto
<T61Options>	+	o	I	T61OPTS	<T61-options-parameter>	–	
Envío de uno o varios ficheros							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Especifica una lista de ficheros; utiliza una sintaxis especial

**Funcionalidad adicional de la <ReceiveTDD> para el servicio teletex**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<Originator>	B	o	O	ADDRESS	<Address-parameter>	-	Especifica el número de teléfono del originador
<CvTx>	B	o	I	CVTTX	<Convert-id-parameter>	"T.61"	Formato de transferencia deseado por la LA
<SubAddress>	+	o	I/O	SUBADDR	<Sub-address-parameter>	-	Subdirección del recipiente; si está especificada en la petición, se utiliza como un selector para la recuperación
Recepción de un solo fichero							
<Convert>	B	m	O	CONVERT	<Convert-id-parameter>	-	Indica el formato de transferencia del fichero recibido
<Document>	B	m	I/O	FILENAME	<File-parameter>	-	El nombre del fichero puede haber sido fijado previamente por la LA. Si se recibe un solo fichero, la CA retendrá el nombre. Si se reciben varios ficheros, la CA puede escribir "aplastando" el nombre anteriormente escrito
<Type>	B	o	O	TYPE	<Type-id-parameter>	-	Especifica el tipo de documento recibido
<Cil>	B	o	O	CIL	<Cil-parameter>	-	Renglón de identificación de la llamada construido por la CA (véase la Rec. UIT-T F.200)
<Name>	+	o	O	NAME	STRING (SIZE(1..12))	-	Retorna el nombre del fichero entrante (longitud máxima: 12 caracteres)
<UserInfo>	+	o	O	USERINFO	STRING (SIZE(1..12))	-	Asigna un comentario de usuario al documento. Este comentario se transmite con el documento
<FirstPg>	+	o	O	FIRSTPG	NUMERIC-STRING	-	Número de la primera página recibida
<Prolog>	+	o	I	PROLOG	PATH	-	Nombre de fichero del "documento de control"
Recepción de varios ficheros							
<DocumentSpec>	+	m	O	FILENAME	"@" <File-of-filespec>	-	Especifica una lista de ficheros; utiliza una sintaxis especial

**13.4 Fijaciones del descriptor de CA**

Una CA que admite el servicio TTX (teletex) especificará el identificador de tipo (formato de transmisión de documento) soportado en el descriptor de CA (véase 9.5). El Cuadro 86 muestra las posibles asignaciones.

CUADRO 86/T.611

Valores adicionales del descriptor de CA para transferencia de ficheros

Palabra clave	Parámetro	CA declara
TTX	"STD"	El soporte del servicio teletex básico (Rec. T.61)
TTX	"OPD"	El soporte de la transmisión de documento operador
TTX	"MD"	El soporte de la transmisión de documento de supervisión
TTX	"CTL"	El soporte de la transmisión de documento de control
TTX	"DTM"	El soporte del modo transparente al documento
TTX	"BFT"	El soporte de la transferencia de ficheros binarios
TTX	"EDI"	El soporte del intercambio electrónico de datos

14 Servicio: Télex vía teletex

14.1 Elementos de sintaxis específicos del servicio

<ServiceDependentKeywordsSend> :=  
 (<Recipient> | <RecipientSpec>)  
 ((<Document> <Convert> [<Type>] [From] [To]) | <DocumentSpec>)  
 [<SubAddress>] [<Notify>]

<ServiceDependentKeywordsSendAck> :=  
 <Recipient>  
 ((<Document> <Convert> [<Type>] [From] [To]) | <DocumentSpec>)  
 [<SubAddress>] [<Notify>]

<ServiceDependentKeywordsReceive> :=  
 [<Originator>] ((<Document> <Convert> <Type>) | <DocumentSpec>)  
 [<CvTx>] [<SubAddress>]

Véase el Cuadro 87.

CUADRO 87/T.611

Elementos de sintaxis adicionales para el servicio télex vía teletex

Elemento de sintaxis	Finalidad
<Convert>	Especifica el formato de transferencia que va a utilizarse
<CvTx>	Especifica el formato de transferencia deseado para los ficheros recibidos
<Document>, <DocumentSpec>	Especifica el documento o documentos que se van a enviar o que se están recibiendo
<From>	La transmisión debe comenzar en el número de página especificado
<Notify>	Especifica si se ha solicitado o no la notificación de entrega
<Originator>	Especifica la dirección de comunicaciones del originador
<Recipient>, <RecipientSpec>	Especifica la dirección o direcciones de comunicaciones del recipiente o recipientes
<SubAddress>	Especifica la subdirección del originador
<To>	Especifica el número de la página en que deberá terminar la transmisión
<Type>	Especifica el formato de transmisión utilizado

## 14.2 Codificación basada en texto

### 14.2.1 Correspondencia de las palabras clave

Véase el Cuadro 88.

CUADRO 88/T.611

**Codificación basada en texto de los elementos de sintaxis adicionales para el servicio télex vía teletex**  
(El símbolo ↵ representa el determinante de formato de nueva línea)

Elemento de sintaxis	Par palabra clave/Parámetro
<Convert>	"CONVERT" ":" <Convert-id-parameter> ↵
<CvTx>	"CVTX" ":" <Convert-id-parameter> ↵
<Document>	"FILENAME" ":" <File-parameter> ↵
<DocumentSpec>	"FILENAME" ":" "@" <File-of-filespec> ↵
<From>	"FROM" ":" NUMERIC-STRING ↵
<Notify>	"NOTIFY" ":" <Boolean-parameter> ↵
<Originator>	"ADDRESS" ":" <Address-parameter> ↵
<Recipient>	"ADDRESS" ":" <Address-parameter> ↵
<RecipientSpec>	"ADDRESS" ":" "@" <File-of-addrspec> ↵
<SubAddress>	"SUBADDR" ":" <Sub-address-parameter> ↵
<To>	"TO" ":" NUMERIC-STRING ↵
<Type>	"TYPE" ":" <Type-id-parameter> ↵

### 14.2.2 Codificación de parámetros

Véase también el § 6.4.4 para la codificación utilizada para los parámetros que no dependen del servicio.

#### 14.2.2.1 Parámetro identificador de servicio

El parámetro Service-id-parameter se codifica como una STRING fijada al valor constante "TX".

*Sintaxis:*

<Service-id-parameter> := "TX"

#### 14.2.2.2 Parámetro identificador de tipo

El parámetro Type-id-parameter se codifica como una STRING fijada al valor constante "STD".

*Sintaxis:*

<Type-id-parameter> := "STD"

NOTA – El formato de transmisión STD se define en la Recomendación UIT-T S.1.

### 14.2.2.3 Parámetro identificador de conversión

El parámetro Convert-id-parameter se codifica como una STRING fijada a uno de los siguientes valores:

"ASCII"	ASCII ampliado de APPLI/COM
"ASCII437"	ASCII ampliado de APPLI/COM
"T.50"	ASCII estándar de APPLI/COM

*Sintaxis:*

```
<Convert-id-parameter> := <Ascii> | "T.50"  
<Ascii> := "ASCII" | STRING ("ASCII" + <Code-page>)  
<Code-page> := <digit> <digit> <digit>  
<digit> := "0" | ... | "9"
```

### 14.2.2.4 Parámetro de fichero de especificaciones de direcciones (File-of-addrspec) y parámetro de dirección (Address-parameter)

El parámetro File-of-addrspec se codifica como un PATH que apunta a un fichero que contiene parámetros Addrspec-parameters, que a su vez contiene el Address-parameter.

El parámetro Address-parameter se codifica como una STRING que forma el número de llamada. Si se debe verificar la identificación de un abonado, hay que entrar la unidad de distintivo del terminal receptor después del número de llamada, separado por el signo "igual" ("=").

Como otra posibilidad puede darse un nombre de alias en lugar del identificador del terminal, a condición que el alias se introduzca por un carácter "&". Se supone que la CA sabe cómo decodificar el alias especificado.

NOTA – Si se tiene el propósito de verificar la identificación del abonado, hay que entrar la unidad de distintivo completa después del signo igual, y no limitarse a entrar solamente el componente alfabético del receptor.

*Sintaxis:*

```
<File-of-addrspec> := PATH  
-- El trayecto apunta a un fichero que contiene uno o más  
-- <Addrspec-parameter>s  
<Addrspec-parameter> := <Address-parameter>  
<Address-parameter> := <Telex-address> | ("&" <Alias>)  
<Telex-address> := STRING  
-- La cadena presentará una dirección télex válida  
<Alias> := STRING
```

### 14.2.2.5 Parámetro de fichero de especificaciones de ficheros (File-of-filespec) y parámetro de fichero (File-parameter)

El parámetro File-of-filespec se codifica como un PATH que apunta a un fichero que contiene parámetros Filespec-parameter, que a su vez contiene el File-parameter. El File parameter en sí se codifica también como un PATH, que apunta al fichero transferido.

*Sintaxis:*

```
<File-of-filespec> := PATH  
-- El trayecto apunta a un fichero que contiene uno o más  
-- <Filespec-parameter>s  
<Filespec-parameter> := <File-parameter> ", " <File-conv> [", " <File-type>]  
<File-parameter> := PATH  
-- Trayecto del fichero transferido  
<File-conv> := <Convert-id-parameter>  
-- Especifica el formato de transferencia del fichero  
<File-type> := <Type-id-parameter>  
-- Especifica el formato de transmisión del fichero
```

### 14.2.2.6 Parámetro identificador de subdirección

El parámetro Sub-address-parameter se codifica como una NUMERIC-STRING que representa una subdirección.

*Sintaxis:*

<Sub-address-parameter> := NUMERIC-STRING

## 14.3 Funcionalidad adicional

### 14.3.1 Función: Send y SendAck

Véanse los Cuadros 89 y 90.

CUADRO 89/T.611

#### Funcionalidad adicional de la <SendTDD> para el servicio télex vía teletex

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<Notify>	+	o	I	NOTIFY	<Boolean-parameter>	"Yes" (Sí)	
<SubAddress>	+	o	I	SUBADDR	<Sub-address-parameter>	–	Subdirección del originador
Envío a un solo destinatario							
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	–	Especifica el número de llamada de un recipiente
Envío a uno o varios destinatarios							
<RecipientSpec>	+	m	I	ADDRESS	"@" <File-of-addrspec>	–	Especifica una lista de recipientes
Envío de un solo fichero							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Formato de transferencia del fichero saliente
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichero único a transmitir y entregar con el formato de transferencia especificado por la palabra clave Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Especifica el tipo de documento a enviar
<From>	+	o	I	FROM	NUMERIC-STRING	Primera página	Primera página efectivamente enviada; sólo es aplicable a los ficheros de texto
<To>	+	o	I	TO	NUMERIC-STRING	Última página	Última página efectivamente enviada; sólo es aplicable a los ficheros de texto
Envío de uno o varios ficheros							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Especifica una lista de ficheros; utiliza una sintaxis especial

**Funcionalidad adicional de la <SendAckTDD> para el servicio télex vía teletex**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	–	Especifica el número de llamada de un recipiente
<Notify>	+	o	I	NOTIFY	<Boolean-parameter>	"Yes" (Sí)	
<SubAddress>	+	o	I	SUBADDR	<Sub-address-parameter>	–	Subdirección del originador
Envío de un solo fichero							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Formato de transferencia del fichero saliente
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichero único a ser transmitido y entregado con el formato de transferencia especificado por la palabra clave Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Especifica el tipo de documento a enviar
<From>	+	o	I	FROM	NUMERIC-STRING	Primera página	Primera página efectivamente enviada; sólo es aplicable a los ficheros de texto
<To>	+	o	I	TO	NUMERIC-STRING	Última página	Última página efectivamente enviada; sólo es aplicable a los ficheros de texto
Envío de uno o varios ficheros							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Especifica una lista de ficheros; utiliza una sintaxis especial

**14.3.2 Función: recibir (receive)**

Véase el Cuadro 91.

**14.4 Fijaciones del descriptor de CA**

Una CA que admite el servicio TX especificará el identificador de tipo (formato de transmisión de documento) soportado en el descriptor de CA (véase 9.5). El Cuadro 92 muestra las posibles asignaciones.

CUADRO 91/T.611

**Funcionalidad adicional de la <ReceiveTDD> para el servicio télex vía teletex**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<Originator>	B	o	O	ADDRESS	<Address-parameter>	-	Especifica el número de teléfono del originador
<CvTx>	B	o	I	CVTX	<Convert-id-parameter>	"T.50"	Formato de transferencia deseado por la LA
<SubAddress>	+	o	I/O	SUBADDR	<Sub-address-parameter>	-	Subdirección del destinatario; si está especificada en la petición, se utiliza como un selector para la recuperación
Recepción de un solo fichero							
<Convert>	B	m	O	CONVERT	<Convert-id-parameter>	-	Indica el formato de transferencia del fichero recibido
<Document>	B	m	I/O	FILENAME	<File-parameter>	-	El nombre del fichero puede haber sido fijado previamente por la LA. Si se recibe un solo fichero, la CA retendrá el nombre. Si se reciben varios ficheros, la CA puede escribir "aplastando" el nombre anteriormente escrito
<Type>	B	o	O	TYPE	<Type-id-parameter>	-	Especifica el tipo de documento recibido
Recepción de varios ficheros							
<DocumentSpec>	+	m	O	FILENAME	"@" <File-of-filespec>	-	Especifica una lista de ficheros; utiliza una sintaxis especial

CUADRO 92/T.611

**Valores adicionales del descriptor de CA para el servicio télex vía teletex**

Palabra clave	Parámetro	CA declara
TX	"STD"	El soporte del servicio básico

## 15 Servicio: Télex

### 15.1 Elementos de sintaxis específicos del servicio

<ServiceDependentKeywordsSend> :=  
(<Recipient> | <RecipientSpec>)  
(((<Document> <Convert> [<Type>] [From] [To]) | <DocumentSpec>)  
[<SubAddress>])

<ServiceDependentKeywordsSendAck> :=  
<Recipient>  
(((<Document> <Convert> [<Type>] [From] [To]) | <DocumentSpec>)  
[<SubAddress>])

<ServiceDependentKeywordsReceive> :=  
[<Originator>] ((<Document> <Convert> <Type>) | <DocumentSpec>)  
[<CvTlx>] [<SubAddress>]

Véase el Cuadro 93.

CUADRO 93/T.611

#### Elementos de sintaxis adicionales para el servicio télex

Elemento de sintaxis	Finalidad
<Convert>	Especifica el formato de transferencia que va a utilizarse
<CvTlx>	Especifica el formato de transferencia deseado para los ficheros recibidos
<Document>, <DocumentSpec>	Especifica el documento o documentos que se van a enviar o que se están recibiendo
<From>	La transmisión debe comenzar en el número de página especificado
<Originator>	Especifica la dirección de comunicaciones del originador
<Recipient>, <RecipientSpec>	Especifica la dirección o dirección de comunicaciones del recipiente o recipientes
<SubAddress>	Especifica la subdirección del originador
<To>	La transmisión debe terminar en el número de página especificado
<Type>	Especifica formato de transmisión utilizado

### 15.2 Codificación basada en texto

#### 15.2.1 Correspondencia de las palabras clave

Véase el Cuadro 94.

#### 15.2.2 Codificación de parámetros

Véase también 6.4.4 para la codificación utilizada para los parámetros que no dependen del servicio.

##### 15.2.2.1 Parámetro identificador de servicio

El parámetro Service-id-parameter se codifica como una STRING fijada al valor constante "TLX".

*Sintaxis:*

<Service-id-parameter> := "TLX"

CUADRO 94/T.611

**Codificación basada en texto de los elementos de sintaxis adicionales para el servicio télex**

(El símbolo ↵ representa el determinante de formato de nueva línea)

Elemento de sintaxis	Par palabra clave/parámetro
<Convert>	"CONVERT" ":" <Convert-id-parameter> ↵
<CvTlx>	"CVTLX" ":" <Convert-id-parameter> ↵
<Document>	"FILENAME" ":" <File-parameter> ↵
<DocumentSpec>	"FILENAME" ":" "@" <File-of-filespec> ↵
<From>	"FROM" ":" NUMERIC-STRING ↵
<Originator>	"ADDRESS" ":" <Address-parameter> ↵
<Recipient>	"ADDRESS" ":" <Address-parameter> ↵
<RecipientSpec>	"ADDRESS" ":" "@" <File-of-addrspec> ↵
<SubAddress>	"SUBADDR" ":" <Sub-address-parameter> ↵
<To>	"TO" ":" NUMERIC-STRING ↵
<Type>	"TYPE" ":" <Type-id-parameter> ↵

**15.2.2.2 Parámetro identificador de tipo**

El parámetro Type-id-parameter se codifica como una STRING fijada al valor constante "STD".

*Sintaxis:*

<Type-id-parameter> := "STD"

NOTA – El formato de transmisión STD se define en la Recomendación UIT-T S.1.

**15.2.2.3 Parámetro identificador de conversión**

El parámetro Convert-id-parameter se codifica como una STRING fijada a uno de los siguientes valores:

- "ASCII" ASCII ampliado de APPLI/COM
- "ASCII437" ASCII ampliado de APPLI/COM
- "T.50" ASCII estándar de APPLI/COM

*Sintaxis:*

<Convert-id-parameter> := <Ascii> | "T.50"

<Ascii> := "ASCII" | STRING ("ASCII" + <Code-page>)

<Code-page> := <digit> <digit> <digit>

<digit> := "0" | ... | "9"

**15.2.2.4 Parámetro de fichero de especificaciones de direcciones (File-of-addrspec) y parámetro de dirección (Address-parameter)**

El parámetro File-of-addrspec se codifica como un PATH que apunta a un fichero que contiene parámetros Addr-spec-parameters, que contiene el Address-parameter.

El parámetro Address-parameter se codifica como una STRING que forma el número de llamada. Si se debe verificar la identificación de un abonado, hay que entrar la unidad de distintivo del terminal receptor después del número de llamada, separado por el signo "igual" ("=").

Como otra posibilidad, puede darse un nombre de alias en lugar del identificador del terminal, a condición que el alias se introduzca por un carácter "&". Se supone que la CA sabe cómo decodificar el alias especificado.

NOTA – Si se tiene el propósito de verificar la identificación del abonado, hay que entrar la unidad de distintivo completa después del signo igual, y no limitarse a entrar solamente el componente alfabético del receptor.

*Sintaxis:*

<File-of-addrspec> :=           PATH  
-- *El trayecto apunta a un fichero que contiene uno o más*  
-- <Addrsec-parameter>

<Addrsec-parameter> :=       <Address-parameter>

<Address-parameter> :=       <Telex-address> | ("&" <Alias>)

<Telex-address> :=           STRING  
-- *La cadena presentará una dirección télex válida*

<Alias> :=                    STRING

### 15.2.2.5 Parámetro de fichero de especificaciones de ficheros (File-of-filespec) y parámetro de fichero (File-parameter)

El parámetro File-of-filespec se codifica como un PATH que apunta a un fichero que contiene parámetros Filespec-parameter, que contiene el File-parameter. El File parameter en sí se codifica también como un PATH, que apunta al fichero transferido.

*Sintaxis:*

<File-of-filespec> :=       PATH  
-- *El trayecto apunta a un fichero que contiene*  
-- <Filespec-parameter>

<Filespec-parameter> :=   <File-parameter> ", " <File-conv> [", " <File-type>]

<File-parameter> :=       PATH  
-- *Trayecto del fichero transferido*

<File-conv> :=            <Convert-id-parameter>  
-- *Especifica el formato de transferencia del fichero*

<File-type> :=            <Type-id-parameter>  
-- *Especifica el formato de transmisión del fichero*

### 15.2.2.6 Parámetro identificador de subdirección

El parámetro Sub-address-parameter se codifica como una NUMERIC-STRING que representa una subdirección.

*Sintaxis:*

<Sub-address-parameter> :=    NUMERIC-STRING

## 15.3 Funcionalidad adicional

### 15.3.1 Función: Send y SendAck

Véanse los Cuadros 95 y 96.

**Funcionalidad adicional de la <SendTDD> para el servicio télex**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<SubAddress>	+	o	I	SUBADDR	<Sub-address-parameter>	-	Subdirección del originador
Envío a un solo destinatario							
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	-	Especifica el número de llamada de un recipiente
Envío a uno o varios destinatarios							
<RecipientSpec>	+	m	I	ADDRESS	"@" <File-of-addrspec>	-	Especifica una lista de recipientes
Envío de un solo fichero							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	-	Formato de transferencia del fichero saliente
<Document>	B	m	I	FILENAME	<File-parameter>	-	Fichero único a transmitir y entregar con el formato especificado por la palabra clave Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Especifica el tipo de documento a enviar
<From>	+	o	I	FROM	NUMERIC-STRING	Primera página	Primera página efectivamente enviada; sólo es aplicable a los ficheros de texto
<To>	+	o	I	TO	NUMERIC-STRING	Última página	Última página efectivamente enviada; sólo es aplicable a los ficheros de texto
Envío de uno o varios ficheros							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	-	Especifica una lista de ficheros; utiliza una sintaxis especial

CUADRO 96/T.611

Funcionalidad adicional de la <SendAckTDD> para el servicio télex vía teletex

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	–	Especifica el número de llamada de un recipiente
<SubAddress>	+	o	I	SUBADDR	<Sub-address-parameter>	–	Subdirección del originador
Envío de un solo fichero							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Formato de transferencia del fichero saliente
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichero único a ser transmitido y entregado con el formato de transferencia especificado por la palabra clave Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Especifica el tipo de documento a enviar
<From>	+	o	I	FROM	NUMERIC-STRING	Primera página	Primera página efectivamente enviada; sólo es aplicable a los ficheros de texto
<To>	+	o	I	TO	NUMERIC-STRING	Última página	Última página efectivamente enviada; sólo es aplicable a los ficheros de texto
Envío de uno o varios ficheros							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Especifica una lista de ficheros; utiliza una sintaxis especial

### 15.3.2 Función: recepción (receive)

Véase el Cuadro 97.

CUADRO 97/T.611

#### Funcionalidad adicional de la <RecieveTDD> para el servicio télex vía teletex

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<Originator>	B	o	O	ADDRESS	<Address-parameter>	–	Especifica el número de teléfono del originador
<CvTx>	B	o	I	CVTLX	<Convert-id-parameter>	"T.50"	Formato de transferencia deseado por la LA
<SubAddress>	+	o	I/O	SUBADDR	<Sub-address-parameter>	–	Subdirección del recipiente; si está especificada en la petición, se utiliza como un selector para la recuperación
Recepción de un solo fichero							
<Convert>	B	m	O	CONVERT	<Convert-id-parameter>	–	Indica el formato de transferencia del fichero recibido
<Document>	B	m	I/O	FILENAME	<File-parameter>	–	El nombre del fichero puede haber sido fijado previamente por la LA. Si se recibe un solo fichero, la CA retendrá el nombre. Si se reciben varios ficheros "aplastando" el nombre anteriormente escrito
<Type>	B	o	O	TYPE	<Type-id-parameter>	–	Especifica el tipo de documento recibido
Recepción de varios ficheros							
<DocumentSpec>	+	m	O	FILENAME	"@" <File-of-filespec>	–	Especifica una lista de ficheros; utiliza una sintaxis especial

### 15.4 Fijaciones del descriptor de CA

Una CA que soporta el servicio TX especificará el identificador de tipo (formatos de transmisión de documento) soportado en el descriptor de CA (véase 9.5). El Cuadro 98 muestra las posibles asignaciones.

CUADRO 98/T.611

#### Valores adicionales del descriptor de CA para el servicio télex

Palabra clave	Parámetro	CA declara
TX	"STD"	El soporte del servicio básico

## 16 Servicio: Correo electrónico

Esta Recomendación permite el envío y recepción de información mediante servicios de correo electrónico ("E-Mail"). No se pretende un control exhaustivo de los servicios de correo electrónico a través de la interfaz.

Las definiciones que aquí siguen permiten asimismo el acceso al sistema de tratamiento de mensajes (MHS, *message handling system*), descrito en las Recomendaciones UIT-T de la serie X.400. Sobre la base del MHS, la interfaz proporciona acceso a la mensajería interpersonal.

La implementación de la CA puede basarse en otras interfaces que soporten el correo electrónico, como muestra la Figura 15. La CA mostrada utiliza una interfaz de correo electrónico subyacente para comunicarse con el proveedor de servicio de correo electrónico. En general, la CA puede utilizar, libremente, *cualquier* interfaz de correo electrónico para ganar acceso y controlar servicios de correo electrónico.

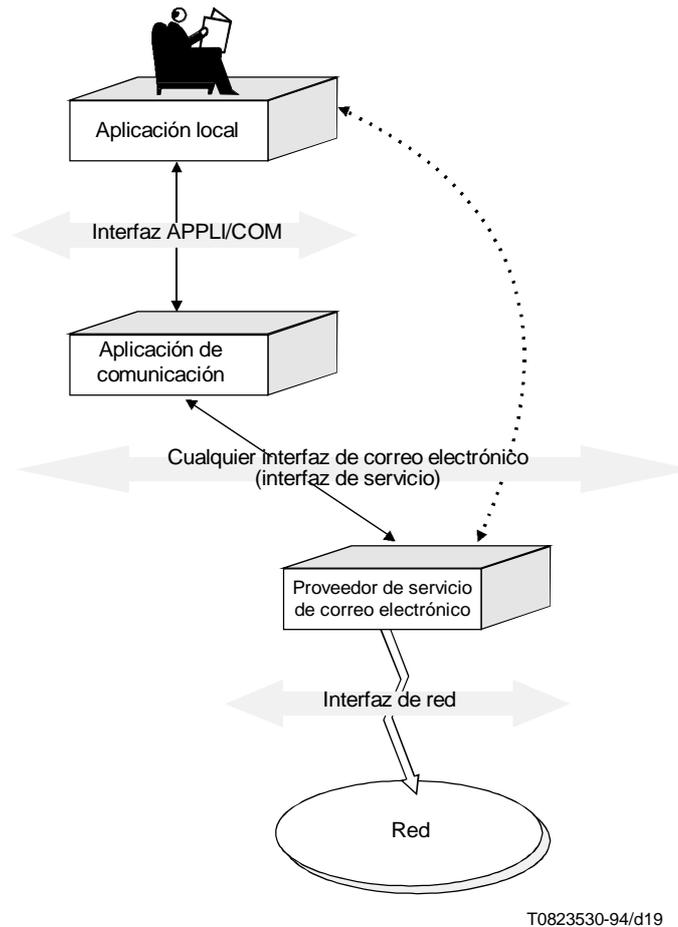


FIGURA 15/T.611

### Modelo de acceso al correo electrónico

Los sistemas de tratamiento de mensajes (MHS) y de correo electrónico normalizan la sintaxis y la semántica de los mensajes interpersonales (IPM, *interpersonal messages*)<sup>9)</sup> así como la sintaxis y la semántica de notificaciones interpersonales (IPN, *interpersonal notification*) e informes de entrega (DR, *delivery report*).

<sup>9)</sup> Para el MHS, la Recomendación UIT-T X.420 normaliza la sintaxis de los IPM y las IPN. En la versión de 1984 de la Recomendación UIT-T X.420, las notificaciones se denominaron SR-UAPDU.

Un recipiente puede crear y enviar una IPN automáticamente o a petición de la LA (notificación explícita). Las IPN pueden ser:

- una notificación de recepción (RN, *receipt notification*): se recibió el IPM; o bien
- una notificación de no recepción (NRN, *non receipt notification*): el IPM no se entregó; en este caso, el IPM no entregado puede transmitirse con la NRN.

A diferencia de la IPN, el DR es creado por el sistema de transferencia de mensajes. Los DR pueden ser:

- informe de entrega (DR);
- informe de no entrega (NDR, *non-delivery report*).

En las TDD de envío ("Send") y de acuse de recibo de envío ("SendAck"), la LA puede especificar la notificación interpersonal (IPN) o el informe de entrega (DR) que será generado para cada recipiente. Para mayor información, véanse 16.2.2.6, 16.4 y 16.5.

El MHS y otros sistemas de correo electrónico permiten transferir IPM completos (constituídos por un encabezamiento y partes de cuerpo) como una parte de cuerpo en un mensaje.

Esta Recomendación soporta la transmisión de esas partes de cuerpo en las TDD de envío (acuse de recibo de envío) y recepción por medio de un parámetro identificador de tipo (Type-id) específico (el identificador de tipo "MESSAGE", véase 16.2.2.2). Dado que el contenido de esa parte de cuerpo está codificado<sup>10)</sup>, esta Recomendación permite decodificar y codificar esta parte de cuerpo mediante el empleo de las TDD EncodeIPM y DecodeIPM.

Las TDD EncodeIPM y DecodeIPM se implementarán utilizando la <extendTDD>. Las extensiones (o ampliaciones) se denominan:

- <EncodeIPMExtension>; y
- <DecodeIPMExtension>, respectivamente.

Las TDD EncodeIPM y DecodeIPM se describen en 16.3.3.

## 16.1 Elementos de sintaxis específicos del servicio

A continuación se indican las extensiones dependientes del servicio de la sintaxis de TDD genérica a que se hace alusión en 6.1:

```
<ServiceDependentKeywordsSend> :=  
    <S-RecipientSpec> [<S-OriginatorSpec>]  
    ((<Document> <Convert> [<Type>]) | <DocumentSpec>)  
    <IpmlId> [<Alternate>] [<ContType>] [<DiscloRec>] [<ExpiryTime>]  
    [<ImplicitConv>] [<Importance>] [<Language>] [<Priority>]  
    [<RelatedSpec>] [<ReplyId>] [<ReplyTime>] [<Sensitivity>] [<Subject>]  
    [<UserInfo>]
```

```
<ServiceDependentKeywordsSendAck> :=  
    <S-RecipientSpec> [<S-OriginatorSpec>]  
    ((<Document> <Convert> [<Type>]) | <DocumentSpec>)  
    <IpmlId> [<Alternate>] [<ContType>] [<DiscloRec>] [<ExpiryTime>]  
    [<ImplicitConv>] [<Importance>] [<Language>] [<Priority>]  
    [<RelatedSpec>] [<ReplyId>] [<ReplyTime>] [<Sensitivity>] [<Subject>]  
    [<UserInfo>] [<MsgSubId>] [<SubmitTime>]
```

```
<ServiceDependentKeywordsReceive> :=  
    <R-RecipientSpec> [<R-OriginatorSpec>]  
    ((<Document> <Convert> <Type>) | <DocumentSpec>)  
    <IpmlId> [<RelatedSpec>] [<ContType>] [<ExpiryTime>] [<Forwarded>]  
    [<Importance>] [<IncompleteCopy>] [<Language>] [<Priority>]  
    [<ReplyId>] [<ReplyTime>] [<Sensitivity>] [<Subject>] [<SubmitTime>]  
    [<UserInfo>]
```

<sup>10)</sup> En el caso del MHS, el IPM se codifica conforme a las reglas de codificación básica de ASN.1.

```

<ExtendSubFunctionKeywords> :=
    <EncodeIPMExtension> | <DecodeIPMExtension>

<EncodeIPMExtension> :=
    <EncodeIPMSubFunction>
    ((<Document> <Convert> [<Type>]) | <DocumentSpec>) <Message>
    <S-RecipientSpec> [<S-OriginatorSpec>] <IpmId> [<ContType>] [<ExpiryTime>]
    [<Importance>] [<Priority>] [<RelatedSpec>] [<ReplyId>] [<ReplyTime>]
    <S-Originator> [<Sensitivity>] [<Subject>] [<UserInfo>]

<DecodeIPMExtension> :=
    <DecodeIPMSubFunction>
    <Message> ((<Document> <Convert> <Type>) | <DocumentSpec>)
    <S-RecipientSpec> [<S-OriginatorSpec>] <IpmId> [<ContType>] [<ExpiryTime>]
    [<Forwarded>] [<Importance>] [<Language>] [<Priority>] [<RelatedSpec>]
    [<ReplyId>] [<ReplyTime>] [<Sensitivity>] [<Subject>] [<UserInfo>]

```

El Cuadro 99 describe los elementos de sintaxis adicionales para los servicios de correo electrónico. Para la correspondencia de los elementos de servicio del MHS con los elementos de sintaxis utilizados en esta Recomendación, véase el Cuadro 112 en 16.6.

## 16.2 Codificación basada en texto

### 16.2.1 Correspondencia de palabras clave

Véase el Cuadro 100.

### 16.2.2 Codificación de parámetros

Véase también 6.4.4 para la codificación de los parámetros que no dependen del servicio.

#### 16.2.2.1 Parámetro identificador de servicio

El parámetro identificador de servicio se codifica como una STRING fijada al valor constante "EMAIL".

*Sintaxis:*

```
<Service-id-parameter> := "EMAIL"
```

#### 16.2.2.2 Parámetro identificador de tipo

El parámetro identificador de tipo se codifica como una STRING fijada a uno de los siguientes valores:

"STD"	Parte de cuerpo AI5
"TELETEX"	Parte de cuerpo teletex
"G3FAX"	Parte de cuerpo telefax grupo 3
"G4CLASS1"	Parte de cuerpo telefax grupo 4
"VIDEOTEX"	Parte de cuerpo videotex
"MESSAGE"	Parte de cuerpo que contiene un IPM (constituido por encabezamiento y partes de cuerpo), transferido por el sistema de correo electrónico
"BILATERAL"	Contenido de parte de cuerpo definido por acuerdo bilateral
"NATIONAL"	Contenido de parte de cuerpo definido en el plano nacional
"ODA"	Parte de cuerpo ODA

*Sintaxis:*

```
<Type-id-parameter >:=
    "STD" | "TELETEX" | "G3FAX" | "G4CLASS1" | "VIDEOTEX" |
    "MESSAGE" | "BILATERAL" | "NATIONAL" | "ODA"11)
```

<sup>11)</sup> Para el MHS, la Recomendación X.420 define un conjunto de parámetros relacionados con cada parte de cuerpo. Estos parámetros se transfieren utilizando el parámetro Filespec-parameter (véase 16.2.2.8).

CUADRO 99/T.611

**Elementos de sintaxis adicionales para servicios de correo electrónico**

Elemento de sintaxis	Finalidad
<Alternate>	Especifica si se permiten recibientes alternativos
<ContType>	Especifica el tipo de contenido del mensaje intercambiado
<Convert>	Especifica el formato de transferencia que se va a utilizar
<DecodeIPMSubFunction>	Identifica la subfunción DecodeIPM
<DiscloRec>	Revelación de recibientes
<Document>, <DocumentSpec>	Especifica el documento o los documentos que se enviarán o que se están recibiendo (o que se intercambiarán con las subfunciones EncodeIPM y DecodeIPM)
<EncodeIPMSubFunction>	Identifica la subfunción DecodeIPM
<ExpiryTime>	Fecha/hora de expiración del mensaje (tiempo de expiración)
<Forwarded>	Indicación de reenvío automático
<ImplicitConv>	Conversión implícita
<Importance>	Importancia del contenido del mensaje
<IncompleteCopy>	Refleja la indicación de copia incompleta del MHS
<IpMId>	Identificador de mensaje
<Language>	Identificador del idioma utilizado
<Message>	Trayecto del mensaje codificado según las reglas de codificación básica de ASN.1 de las subfunciones EncodeIPM y DecodeIPM
<MsgSubId>	Identificador de depósito de mensaje retornado por el proveedor del servicio de correo electrónico después de haber aceptado la petición
<Priority>	Prioridad del contenido del mensaje
<R-OriginatorSpec>, <S-OriginatorSpec>	Trayecto del fichero que contiene las especificaciones de direcciones del originador, del usuario o usuarios autorizantes, y la respuesta al usuario o usuarios, del mensaje
<R-RecipientSpec>, <S-RecipientSpec>	Trayecto del fichero que contiene las especificaciones de dirección de todos los recibientes deseados del mensaje, es decir, el recibiente o recibientes primarios de copia, y de copia ciega
<RelatedSpec>	Nombre de trayecto del fichero que contiene las "Relatedspecs" de todos los mensajes relacionados con el presente mensaje
<ReplyId>	Identificador del mensaje a que se responde con éste
<ReplyTime>	Fecha y hora en que el recibiente o recibientes del mensaje deben responder a los usuarios autorizantes (tiempo de respuesta)
<Sensitivity>	Sensibilidad del contenido del mensaje
<Subject>	Asunto a que se refiere el mensaje
<SubmitTime>	Fecha y hora en que el proveedor del servicio de correo electrónico procesó la petición (tiempo de depósito del mensaje)
<Type>	Especifica el formato de transmisión utilizado
<UserInfo>	Identificador de sobre facultativo proporcionado por el usuario, que se envía junto con el mensaje. Se puede utilizar para fines de identificación específicos del usuario

CUADRO 100/T.611

**Codificación basada en texto de los elementos de sintaxis adicionales para servicios de correo electrónico**

(El símbolo ↵ significa nueva línea)

Elemento de sintaxis	Par palabra clave/parámetro
<Alternate>	"ALTERNATE" ":" <Boolean-parameter> ↵
<ContType>	"CONT-TYPE" ":" <Content-type-parameter> ↵
<Convert>	"CONVERT" ":" <Convert-id-parameter> ↵
<DecodeIPMSubFunction>	"SUBFUNC" ":" "DecodeIPM" ↵
<DiscloRec>	"DISCLO-REC" ":" <Boolean-parameter> ↵
<Document>	"FILENAME" ":" <File-parameter> ↵
<DocumentSpec>	"FILENAME" ":" "@" <File-of-filespec> ↵
<EncodeIPMSubFunction>	"SUBFUNC" ":" "EncodeIPM" ↵
<ExpiryTime>	"EXPIRYTIME" ":" <Date-time-parameter> ↵
<Forwarded>	"FORWARDED" ":" <Boolean-parameter> ↵
<ImplicitConv>	"IMPLICIT-CONV" ":" <Boolean-parameter> ↵
<Importance>	"IMPORTANCE" ":" <Importance-parameter> ↵
<IncompleteCopy>	"INC-COPY" ":" <Boolean-parameter> ↵
<IpmId>	"IPM-ID" ":" <Ipm-id-parameter> ↵
<Language>	"LANGUAGE" ":" <Language-id-parameter> ↵
<Message>	"MESSAGE" ":" <Path-parameter> ↵
<MsgSubId>	"MSG-SUB-ID" ":" <Msg-sub-id-parameter> ↵
<Priority>	"PRIORITY" ":" <Priority-parameter> ↵
<R-OriginatorSpec>	"ADDRESS" ":" "@" <File-of-originatorspec> ↵
<R-RecipientSpec>	"RECIPIENT" ":" "@" <File-of-r-recipientspec> ↵
<RelatedSpec>	"RELATED" ":" "@" <File-of-relatedspec> ↵
<ReplyId>	"REPLYID" ":" <Ipm-id-parameter> ↵
<ReplyTime>	"REPLYTIME" ":" <Date-time-parameter> ↵
<S-OriginatorSpec>	"ORIGINATOR" ":" "@" <File-of-originatorspec> ↵
<S-RecipientSpec>	"ADDRESS" ":" "@" <File-of-s-recipientspec> ↵
<Sensitivity>	"SENSITIVITY" ":" <Sensitivity-parameter> ↵
<Subject>	"SUBJECT" ":" <Subject-parameter> ↵
<SubmitTime>	"SUBMITTIME" ":" <Date-time-parameter> ↵
<Type>	"TYPE" ":" <Type-id-parameter> ↵
<UserInfo>	"USERINFO" ":" <Userinfo-parameter> ↵

### 16.2.2.3 Parámetro identificador de conversión

El parámetro identificador de conversión (Convert-id) se codifica como una STRING fijada a uno de los siguientes valores:

"ASCII"	ASCII ampliado de APPLI/COM
"ASCII437"	ASCII ampliado de APPLI/COM
"T.50"	ASCII estándar de APPLI/COM
"T.61"	Formato teletex de APPLI/COM
"TIFF"	TIFF de APPLI/COM
"VOID"	No hay conversión
"PROBE"	Especifica un documento vacío (es decir, que no hay ningún documento) para procedimientos de comprobación de la entrega

La utilización del parámetro identificador de conversión depende también del identificador de tipo seleccionado. Esto se muestra en el Cuadro 101.

CUADRO 101/T.611

#### Asignaciones de identificador de conversión permitidas que dependen del identificador de tipo

Identificador de tipo (Type-id)	Identificador de conversión (Convert-id) para tráfico saliente	Identificador de conversión (Convert-id) para tráfico entrante
STD	T.50	
TELETEX	ASCII, ASCII <sub>xxx</sub> <sup>a)</sup> , T.50, T.61	
G3FAX, G4CLASS1	ASCII, ASCII <sub>xxx</sub> , T.50, TIFF	TIFF
VIDEOTEX, MESSAGE, BILATERAL, NATIONAL, ODA	VOID	
<sup>a)</sup> xxx significa una página de códigos, declarada en el ICE, por ejemplo, ASCII437 si se ha declarado la página de códigos 437.		

*Sintaxis:*

```

<Convert-id-parameter> := <Convert-probe> | <Convert-std> | <Convert-ttx> | <Convert-img> | <Convert-bin>
<Convert-probe> := "PROBE"
-- Modo especial en el que no se transmiten documentos;
-- este modo sólo puede ser enviado

<Convert-std> := "T.50"
<Convert-ttx> := <Ascii> | "T.50" | "T.61"
<Convert-img> := <Ascii> | "T.50" | "TIFF"
<Convert-bin> := "VOID"
<Ascii> := "ASCII" | STRING ("ASCII" + <Code-page>)
<Code-page> := <digit> <digit> <digit>
<digit> := "0" | ... | "9"
  
```

### 16.2.2.4 Direcciones MHS

Las direcciones MHS son referenciadas por los parámetros Originatorspec-parameter, R-Recipientspec-parameter y S-Recipientspec-parameter, o Ipm-id-parameter (véanse 16.2.2.5, 16.2.2.6 y 16.2.2.11).

La sintaxis de las direcciones MHS, a saber, Mhs-or-descriptor, Mhs-or-name, Mhs-dl-name y Mhs-or-address se especifica más adelante.

El parámetro MHS-or-address se codifica como una STRING que representa una compilación de elementos de dirección, cada uno de los cuales especifica una parte de la dirección.

Un elemento de dirección se compone de un identificador de campo de dirección, seguido de un separador (signo "=") y el atributo de campo de dirección.

El siguiente Cuadro 102 contiene los identificadores de campo de dirección predefinidos para los servicios MHS en esta Recomendación.<sup>12)</sup>

NOTA – El esquema de direccionamiento del Cuadro 102 también se puede utilizar para otros entornos de correo electrónico, si procede. Es posible la creación de elementos de dirección adicionales, no predefinidos, utilizando la descripción de sintaxis formal de estilo BNF del <Private-addr-element> más abajo.

CUADRO 102/T.611

**Identificadores y atributos de campo de dirección**

Identificador de campo de dirección	Atributo de campo de dirección	Longitud máxima del campo de atributo
C	CountryName	3
A	AdministrationDomainName	16
P	PrivateDomainName	16
O	OrganizationName	64
OUn (0 ≤ n ≤ 4)	OrganizationUnitName	32
Q	GenerationQualifier	3
S	Surname	40
G	GivenName	16
I	Initials	5
X121	X.121Address	16
T-ID	TerminalIdentifier	24
PD-BOX	PostalOfficeBoxAddress	60

*Sintaxis:*

-- Sintaxis de <Mhs-or-address> (referenciado por otras sintaxis de parámetros)

<Mhs-or-address> :=

```
"/" <Mandatory-address-element> { "/" <Mandatory-address-element> }
{ "/" <Optional-address-element> } { "/" <Private-addr-element> }
-- para uso con sistemas MHS
-- debe estar presente al menos un <Mandatory-address-element>
```

<Mandatory-address-element> :=

```
("C" "=" STRING (SIZE(1..3))) | -- CountryName
("A" "=" STRING (SIZE(1..16))) | -- AdministrationDomainName
("P" "=" STRING (SIZE(1..16))) | -- PrivateDomainName
("O" "=" STRING (SIZE(1..64))) | -- OrganizationName
-- Véase también el Cuadro 102 para las asignaciones
```

<Optional-address-element> :=

```
("OU"<n> "=" STRING (SIZE(1..32))) | -- OrganizationUnitName
("Q" "=" STRING (SIZE(1..3))) | -- GenerationQualifier
("S" "=" STRING (SIZE(1..40))) | -- Surname
```

<sup>12)</sup> Cada identificador de campo de dirección señala un atributo O/R-Name especificado en la Recomendación UIT-T X.402.

```

("G" "=" STRING (SIZE(1..16))) | -- GivenName
("I" "=" STRING (SIZE(1..5))) | -- Initials
("X121" "=" STRING (SIZE(1..16))) | -- X.121 Address
("T-ID" "=" STRING (SIZE(1..24))) | -- Terminal identifier
("PD-BOX" "=" STRING (SIZE(1..60))) | -- PostalOfficeBoxAddress
-- Véase también el Cuadro 102 para las asignaciones

<n> := "0" | ... | "4"

<Private-addr-element> := <Field-identifier> "=" <Field-attribute>
-- elemento de dirección privado, o definido adicionalmente

-- Sintaxis de <Mhs-or-descriptor> (referenciado por otras sintaxis de parámetros)

<Mhs-or-descriptor> := (<Mhs-or-name> ["+" [<Mhs-free-name>] ["+" <Mhs-phone-number>]]) |
("+" <Mhs-free-name> ["+" <Mhs-phone-number>])

<Mhs-free-name> := STRING (SIZE(1..64))

<Mhs-phone-number> := STRING (SIZE(1..32))

-- Sintaxis de <Mhs-or-name> (referenciado por otras sintaxis de parámetro)

<Mhs-or-name> := (<Mhs-or-address> ["+" <Mhs-dl-name>]) | ("+" <Mhs-dl-name>)

<Mhs-dl-name> := STRING (SIZE(1..64))
-- significa nombre de lista de distribución, de acuerdo con
-- la Recomendación UIT-T X.402

```

NOTA – El <Mhs-dl-nombre> es diferente del <Alias> utilizado en algunas sintaxis, ya que la expansión del alias se efectúa localmente (en la CA), mientras que la expansión del <Mhs-dl-name> la efectúa el MTS.

### 16.2.2.5 Fichero de originadores (File-of-originatorspec)

La construcción File-of-originatorspec especifica los originadores del mensaje. Se codifica como PATH. El trayecto dado conducirá a un fichero que contiene uno o más parámetros Originatorspec-parameter.

*Sintaxis:*

```

<File-of-originatorspec> := PATH
-- El trayecto conduce a un fichero que contiene uno o más
-- parámetros <Originatorspec-parameter>

<Originatorspec-parameter> := <Addr-descriptor> | <Mhs-or-descriptor> "," <O-type>
-- para la definición de <Mhs-or-descriptor> véase 16.2.2.4

<Addr-descriptor> := STRING
-- El contenido será el requerido por el sistema E-Mail (no-MHS)
-- subyacente; no incluirá caracteres ","

<O-type> := "Authorizing" | "Originator" | "ReplyTo"

```

### 16.2.2.6 File-of-r-recipientspec y File-of-s-recipientspec

Las construcciones File-of-r-recipientspec y File-of-s-recipientspec especifican los recibientes del mensaje. Se codifican como PATH. El trayecto conducirá a un fichero que contiene uno o más parámetros R-recipientspec-parameter o S-recipientspec-parameter, respectivamente.

*Sintaxis:*

```

<File-of-r-recipientspec> := PATH
-- El trayecto conduce a un fichero que contiene uno o más
-- parámetros <R-recipientspec-parameter>

```

```

<File-of-s-recipientspec> :=      PATH
-- El trayecto conduce a un fichero que contiene uno o más
-- parámetros <S-recipientspec-parameter>

<R-recipientspec-parameter> :=
<Addr-descriptor> | <Mhs-or-descriptor>
", " <R-type> [", " [<Reply>] [", " [<Notify>] [", " <Report>] ] ]
-- Lado receptor
-- para la definición de <Mhs-or-descriptor> véase 16.2.2.4

<S-Recipientspec-parameter> :=
("&" <Alias>) | <Addr-descriptor> | <Mhs-or-descriptor>
", " <S-type> [", " [<Reply>] [", " [<Notify>] [", " <Report>] ] ]
-- El lado emisor puede especificar un alias (local) en lugar de un
-- descriptor, para la definición de <Mhs-or-descriptor> véase 16.2.2.4

<Alias> :=                        STRING
-- El alias no incluirá caracteres ", " o ";"

<Addr-descriptor> :=              STRING
-- El contenido será el requerido por el sistema E-Mail (no-MHS)
-- subyacente; no comprenderá caracteres ", "

<R-type> :=                       "Primary" | "Copy" | "Blind" | "Intended"

<S-type> :=                       "Primary" | "Copy" | "Blind"

<Reply> :=                        "NoReply" | "Reply"
-- Si no se especifica, se toma NoReply como valor por defecto

<Notify> :=                       "NoNotify" | "NotifyNotReceived" | "NotifyReceived" | "NotifyExplicit"
-- Si no se especifica, se toma NoNotify como valor por defecto

<Report> :=                       "NoReport" | "BasicReport" | "ConfirmedReport"13)
-- Si no se especifica, se toma noReport como valor por defecto

```

### 16.2.2.7 File-of-relatedspec

La construcción File-of-relatedspec especifica identificadores de IPM ("IPM-ID") con los cuales está relacionado un mensaje. Se codifica como PATH. El trayecto dado conducirá a un fichero que contiene uno o más parámetros Relatedspec-parameter.

*Sintaxis:*

```

<File-of-relatedspec> :=          PATH
-- El trayecto conduce a un fichero que contiene uno o más
-- parámetros <Relatedspec-parameter>

<Relatedspec-parameter> :=       <Ipm-Id-parameter> ", " <Relation>
-- Para la definición del Ipm-id-parameter véase 16.2.2.11

<Relation> :=                   "Reference" | "Obsolete"

```

### 16.2.2.8 Parámetro de fichero de originadores (File-of-filespec) y parámetro de fichero (File-parameter)

El parámetro File-of-filespec se codifica como un PATH que apunta a un fichero que contiene parámetros Filespec-parameters, que a su vez contiene el File-parameter. El propio File-parameter se codifica también como un PATH que apunta al fichero transferido.

Si se especifica el parámetro especial Convert-id-parameter "PROBE", se puede omitir File-parameter. Véase también 16.2.2.3.

<sup>13)</sup> En el caso de MHS, "BasicReport" corresponde al informe de no entrega y "ConfirmedReport" corresponde al informe de la Recomendación UIT-T X.411.

*Sintaxis:*

```
<File-of-filespec> :=          PATH
-- El trayecto conduce a un fichero que contiene uno o más
-- <Filespec-parameter>

<Filespec-parameter> :=      <File-parameter> "," <File-conv>
[ [ "," <File-type> ] [ "," <Body-part-parameter> ] ]

<File-parameter> :=          PATH
-- Trayecto del fichero transferido

<File-conv> :=                <Convert-id-parameter>
-- Especifica el formato de transferencia del fichero

<File-type> :=                <Type-id-parameter>
-- Especifica el formato de transmisión del fichero

<Body-part-parameter> :=     STRING
-- la sintaxis queda en estudio
```

### 16.2.2.9 Parámetro tipo de contenido (Content-type)

El Content-type-parameter se codifica como una STRING fijada a uno de los siguientes valores:

"mhsIPM84"	Mensaje interpersonal (IPM) de acuerdo con el protocolo 2 definido en la Recomendación UIT-T X.420 (1984)
"mhsIPM88"	Mensaje interpersonal (IPM) de acuerdo con el protocolo 2 definido en la Recomendación UIT-T X.420 (1988)
"mhsIPN84"	Notificación interpersonal (IPN) de acuerdo con el protocolo 2 definido en la Recomendación UIT-T X.420 (1984)
"mhsIPN88"	Notificación interpersonal (IPN) de acuerdo con el protocolo 2 definido en la Recomendación UIT-T X.420 (1988)
"mhsDR"	Informe de entrega (DR) definido en la Recomendación UIT-T X.420
"cmcIPM"	Reservado para mensajes interpersonales (IPM) conformes a la interfaz E-Mail de llamada de mensajería común
"cmcIPN"	Reservado para notificaciones interpersonales (IPN) conformes a la interfaz E-Mail de llamada de mensajería común
"cmcDR"	Reservado para informes de entrega (DR) conformes a la interfaz E-Mail de llamada de mensajería común
cualquier otro	Reservado para ulterior estudio

*Sintaxis:*

```
<Content-type-parameter>:=    "mhsIPM84" | "mhsIPM88" | "mhsIPN84" | "mhsIPN88" | "mhsDR"
| "cmcIPM" | "cmcIPN" | "cmcDR"
```

### 16.2.2.10 Parámetro de importancia (Importance)

El parámetro de importancia (importance-parameter) se codifica como una NUMERIC-STRING fijada a uno de los siguientes valores:

"0"	baja
"1"	normal (por defecto)
"2"	alta

*Sintaxis:*

```
<importance-parameter> :=    "0" | ... | "2"
```

### 16.2.2.11 Parámetro identificador de IPM (Ipm-id)

El parámetro Ipm-id-parameter se codifica como STRING.

En los sistemas basados en MHS consta de dos componentes:

- un identificador único, relativo al usuario (identificador local);
- una dirección O/R facultativa que identifica al usuario.

*Sintaxis:*

<Ipm-id-parameter> := <Free-Ipm-id> | <Mhs-Ipm-id>

-- Para la definición del <Mhs-or-descriptor>, véase 16.2.2.4

<Free-ipm-id> :=  
STRING  
-- El contenido será el requerido por el sistema de  
-- correo electrónico (no MHS) subyacente  
-- El string no incluirá caracteres ", " o " ;"

<Mhs-ipm-id> :=  
<Local-ipm-id> [", " <Mhs-or-address>]  
-- para uso en sistemas MHS  
-- para la definición del <Mhs-or-address>, véase 6.2.2.4

<Local-ipm-id> :=  
STRING (SIZE(1..64))  
-- El string no incluirá caracteres ", " o " ;"

#### 16.2.2.12 Parámetro identificador de idioma (Language-id)

El parámetro Language-id-parameter se codifica como STRING con una longitud de 2 a 5 caracteres. Para la asignación del código de idioma para el MHS, véase la Norma ISO 639.2

*Sintaxis:*

<Language-id-parameter> := STRING (SIZE(2..5))

#### 16.2.2.13 Parámetro identificador de depósito de mensaje (Msg-sub-id)

El parámetro Msg-sub-id-parameter se codifica como STRING. Este parámetro identifica única e inequívocamente el depósito de un mensaje. Es generado por el sistema de correo electrónico<sup>14</sup>.

*Sintaxis:*

<Msg-sub-id-parameter> := <Free-sub-id> | <Mhs-sub-id>

<Free-sub-id> :=  
STRING  
-- Para uso en sistemas distintos del MHS

<Mhs-sub-id> :=  
<Global-id> ", " <Local-id>  
-- Para uso en sistemas MHS

<Global-id> :=  
"/" "C" "=" STRING (SIZE(1..3)) -- CountryName  
"/" "A" "=" STRING (SIZE(1..16)) -- AdministrationDomainName  
["/" "P" "=" STRING (SIZE(1..16))] -- PrivateDomainName  
-- Véase también la definición de <Mhs-or-address>, 16.2.2.4

<Local-id> :=  
STRING (SIZE(1..32))

#### 16.2.2.14 Parámetro de prioridad (Priority)

El parámetro Priority-parameter se codifica como una NUMERIC-STRING fijada a uno de los siguientes valores:

- "0" normal (por defecto)
- "1" no urgente
- "2" urgente

<sup>14</sup> En el caso del MHS, el Msg-sub-id-parameter corresponde al MPDUIdentifier (Recomendación UIT-T X.411, versión de 1984), llamado MTSIdentifier en la versión 1988 de la mencionada Recomendación UIT-T.

*Sintaxis:*

<Priority-parameter> := "0" | "1" | "2"

#### **16.2.2.15 Parámetro de asunto (Subject)**

El parámetro Subject-parameter se codifica como una STRING.

*Sintaxis:*

<Subject-parameter> := STRING (Size(1..128))

#### **16.2.2.16 Parámetro de sensibilidad (Sensitivity)**

El parámetro Sensitivity-parameter se codifica como una NUMERIC-STRING fijada a uno de los valores siguientes:

"0"	no se especifica sensibilidad (por defecto)
"1"	personal
"2"	privado
"3"	confidencial de la compañía

*Sintaxis:*

<Sensitivity-parameter> := "0" | ... | "3"

#### **16.2.2.17 Parámetro de información de usuario (Userinfo)**

El parámetro Userinfo-parameter se codifica como STRING.

*Sintaxis:*

<Userinfo-parameter> := STRING (SIZE(1..16))

### **16.3 Mensajería interpersonal**

#### **16.3.1 Función: Envío y acuse de envío (Send y SendAck)**

La TDD de respuesta de la función SendAck (acuse de recibo de envío) la genera la CA tan pronto como se haya depositado el documento (el mensaje) en el proveedor de correo electrónico del originador y el proveedor correo electrónico le haya retornado (a la CA) el mensaje <MsgSubId>. Obsérvese que de lo que se acusa recibo es del depósito, no de la transmisión del mensaje. Véanse los Cuadros 103 y 104.

#### **16.3.2 Función: Recibir (Receive)**

Véase el Cuadro 105.

#### **16.3.3 Función: Codificar IPM y decodificar IPM (EncodeIPM y DecodeIPM)**

Las TDD de EncodeIPM y de DecodeIPM permiten a la LA codificar o decodificar un IPM completo (que contiene encabezamiento y partes de cuerpo) para formar una parte de cuerpo de tipo "MESSAGE", y viceversa (véase también 16.2.2.2).

El procedimiento de codificación/decodificación requiere que la LA proporcione una TDD que contenga elementos de sintaxis referentes al encabezamiento del IPM que se va a codificar/decodificar, así como elementos de sintaxis que lancen la operación de codificación/decodificación propiamente dicha. El Cuadro 106 muestra los elementos de sintaxis adicionales para la <EncodeIPMExtension>, y el Cuadro 107 muestra los elementos de sintaxis adicionales para la <DecodeIPMExtension>, de la <ExtendTDD>.

CUADRO 103/T.611

**Funcionalidad adicional de la <SendTDD> para acceso a servicios de correo electrónico**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<IpmlId>	B	m	I	IPM-ID	<Ipml-id-parameter>	–	Identificador de mensaje
<S-RecipientSpec>	B	m	I	ADDRESS	"@" <File-of-s-recipient-spec>	–	Trayecto del fichero que contiene las Recipientspec de todos los recibientes deseados
<S-OriginatorSpec>	B	m	I	ORIGINATOR	"@" <File-of-originator-spec>	–	Nombre de trayecto del fichero que contiene los usuarios originador y autorizante
<Alternate>	B	o	I	ALTERNATE	<Boolean-parameter>	"No"	Recibiente o recibientes alternativos autorizados
<ContType>	B	o	I	CONT-TYPE	<Content-type-parameter>	"mhsIPM84"	Tipo de contenido
<DiscloRec>	B	o	I	DISCLO-REC	<Boolean-parameter>	"No"	Revelación del recibiente o recibientes
<ExpiryTime>	B	o	I	EXPIRYTIME	<Date-time-parameter>	–	Expiración del mensaje (tiempo de expiración)
<ImplicitConv>	B	o	I	IMPLICIT-CONV	<Boolean-parameter>	"Yes" (Sí)	Conversión implícita
<Importance>	B	o	I	IMPORTANCE	<Importance-parameter>	"1"	Importancia del contenido
<Language>	B	o	I	LANGUAGE	<Language-id-parameter>	–	Identificación del idioma utilizado
<Priority>	B	o	I	PRIORITY	<Priority-parameter>	"0"	Prioridad del contenido

**Funcionalidad adicional de la <SendTDD> para acceso a servicios de correo electrónico**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<RelatedSpec>	B	o	I	RELATED	"@" <File-of-Relatedspec>	–	Nombre de trayecto del fichero que contiene las Relatedspec de todos los mensajes relacionados con el presente mensaje
<ReplyId>	B	o	I	REPLYID	<Ipm-id-parameter>	–	ID del mensaje a que se responde con éste
<ReplyTime>	B	o	I	REPLYTIME	<Date-time-parameter>	–	Fecha y hora a la que el recipiente o recipientes del mensaje deberán haber respondido a los usuarios autorizantes (tiempo de respuesta)
<Sensitivity>	B	o	I	SENSITIVITY	<Sensitivity-parameter>	"0"	Sensibilidad del contenido
<Subject>	B	o	I	SUBJECT	<Subject-parameter>	–	
<Userinfo>	+	o	I	USERINFO	<Userinfo-parameter>	–	Identificador de sobre facultativo proporcionado por el usuario
Envío de un solo fichero							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Formato de transferencia del fichero saliente
<Document>	B	m	I	FILENAME	<File-parameter>	–	Fichero individual a transmitir y entregar de acuerdo con el formato de transferencia identificado por la palabra clave Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Especifica el tipo de documento a enviar
Envío de uno o más ficheros							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Especifica una lista de ficheros; utiliza una sintaxis especial

**Funcionalidad adicional de la <SendAckTDD> para acceso a servicios de correo electrónico**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<IpmId>	B	m	I	IPM-ID	<Ipm-id-parameter>	–	Identificador de mensaje
<S-RecipientSpec>	B	m	I	ADDRESS	"@" <File-of-s-recipient-spec>	–	Trayecto del fichero que contiene las Recipientspec de todos los destinatarios deseados
<S-OriginatorSpec>	B	m	I	ORIGINATOR	"@" <File-of-originator-spec>	–	Nombre de trayecto del fichero que contiene los usuarios originador y autorizador
<Alternate>	B	o	I	ALTERNATE	<Boolean-parameter>	"No"	Destinatario o destinatarios alternativos autorizados
<ContType>	B	o	I	CONT-TYPE	<Content-type-parameter>	"mhsIPM84"	Tipo de contenido
<DiscloRec>	B	o	I	DISCLO-REC	<Boolean-parameter>	"No"	Revelación del destinatario o destinatarios
<ExpiryTime>	B	o	I	EXPIRYTIME	<Date-time-parameter>	–	Expiración del mensaje (tiempo de expiración)
<ImplicitConv>	B	o	I	IMPLICIT-CONV	<Boolean-parameter>	"Yes" (Sí)	Conversión implícita
<Importance>	B	o	I	IMPORTANCE	<Importance-parameter>	"1"	Importancia del contenido
<Language>	B	o	I	LANGUAGE	<Language-id-parameter>	–	Identificación del idioma utilizado
<MsgSubId>	B	o	O	MSG-SUB-ID	<Msg-sub-id-parameter>	–	Identificador de depósito de mensaje. Retornado por el proveedor de servicio de correo electrónico después de aceptar la petición
<Priority>	B	o	I	PRIORITY	<Priority-parameter>	"0"	Prioridad del contenido
<RelatedSpec>	B	o	I	RELATED	"@" <File-of-Related-spec>	–	Nombre de trayecto del fichero que contiene las Relatedspec de todos los mensajes relacionados con el presente mensaje
<ReplyId>	B	o	I	REPLYID	<Ipm-id-parameter>	–	ID del mensaje a que se responde con éste

**Funcionalidad adicional de la <SendAckTDD> para acceso a servicios de correo electrónico**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<ReplyTime>	B	o	I	REPLYTIME	<Date-time-parameter>	-	Fecha y hora para la cual el recipiente o recipientes del mensaje deberán haber respondido a los usuarios autorizantes (tiempo de respuesta)
<Sensitivity>	B	o	I	SENSITIVITY	<Sensitivity-parameter>	"0"	Sensibilidad del contenido
<Subject>	B	o	I	SUBJECT	<Subject-parameter>	-	
<SubmitTime>	B	o	O	SUBMITTIME	<Date-time-parameter>	-	Fecha y hora para la cual el proveedor de servicio de correo electrónico deberá haber procesado la petición
<Userinfo>	+	o	I	USERINFO	<Userinfo-parameter>	-	Identificador de sobre facultativo proporcionado por el usuario
Envío de un solo fichero							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	-	Formato de transferencia del fichero saliente
<Document>	B	m	I	FILENAME	<File-parameter>	-	Fichero individual a transmitir y entregar de acuerdo con el formato de transferencia identificado por la palabra clave Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Especifica el tipo de documento a enviar
Envío de uno o más ficheros							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	-	Especifica una lista de ficheros; utiliza una sintaxis especial

**Funcionalidad adicional de la <ReceiveTDD> para acceso a servicios de correo electrónico**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<IpmId>	B	m	O	IPM-ID	<Ipm-id-parameter>	–	Identificador de mensaje
<R-RecipientSpec>	B	m	I	RECIPIENT	"@" <File-of-r-recipient-spec>	–	Trayecto del fichero que contiene las Recipientspec de todos los destinatarios deseados
<R-OriginatorSpec>	B	m	I	ADDRESS	"@" <File-of-originator-spec>	–	Nombre de trayecto del fichero que contiene los usuarios originador y autorizante
<ContType>	B	o	O	CONT-TYPE	<Content-type-parameter>	–	Tipo de contenido
<ExpiryTime>	B	o	O	EXPIRYTIME	<Date-time-parameter>	–	Expiración del mensaje (tiempo de expiración)
<Forwarded>	B	o	O	FORWARDED	<Boolean-parameter>	–	Indicación de reenvío automático
<Importance>	B	o	O	IMPORTANCE	<Importance-parameter>	–	Importancia del contenido
<Language>	B	o	O	LANGUAGE	<Language-id-parameter>	–	Identificación del idioma utilizado
<Priority>	B	o	O	PRIORITY	<Priority-parameter>	–	Prioridad del contenido
<RelatedSpec>	B	o	I	RELATED	"@" <File-of-related-spec>	–	Nombre de trayecto del fichero que contiene las Relatedspec de todos los mensajes relacionados con el presente mensaje
<ReplyId>	B	o	O	REPLYID	<Ipm-id-parameter>	–	ID del mensaje a que se responde con éste
<ReplyTime>	B	o	O	REPLYTIME	<Date-time-parameter>	–	Fecha y hora el destinatario o destinatarios del mensaje deberán haber respondido a los usuarios autorizantes (tiempo de respuesta)
<Sensitivity>	B	o	O	SENSITIVITY	<Sensitivity-parameter>	"0"	Sensibilidad del contenido
<Subject>	B	o	O	SUBJECT	<Subject-parameter>	–	

**Funcionalidad adicional de la <ReceiveTDD> para acceso a servicios de correo electrónico**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<SubmitTime>	B	o	O	SUBMITTIME	<Date-time-parameter>	–	Fecha y hora para la cual el proveedor de servicio de correo electrónico deberá haber procesado la petición
<IncCopy>	+	o	O	INC-COPY	<Boolean-parameter>	–	Indicador de copia incompleta
<Userinfo>	+	o	O	USERINFO	<Userinfo-parameter>	–	Identificador de sobre facultativo proporcionado por el usuario
Envío de un solo fichero							
<Convert>	B	m	O	CONVERT	<Convert-id-parameter>	–	Indica el formato de transferencia del fichero recibido
<Document>	B	m	I/O	FILENAME	<File-parameter>	–	El nombre del fichero puede haber sido fijado previamente por la LA en la petición. Si se recibe un solo fichero, la CA retendrá el nombre. Si se reciben varios ficheros, la CA puede escribir "aplastando" el nombre anteriormente escrito
<Type>	B	o	O	TYPE	<Type-id-parameter>	–	Especifica el tipo de documento recibido
Envío de varios ficheros							
<DocumentSpec>	+	m	O	FILENAME	"@" <File-of-filespec>	–	Especifica una lista de ficheros; utiliza una sintaxis especial

CUADRO 106/T.611

**Elementos de sintaxis adicionales para la <EncodeIPMExtension> de servicios de correo electrónico**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
Elementos de sintaxis pertenecientes al propio procedimiento de codificación (descripción de "job")							
<EncodeIPM-SubFunction>	B	m	I	SUBFUNC	"EncodeIPM"	–	
<Message>	B	m	I	MESSAGE	<Path-parameter>	–	Trayecto del fichero que recibirá el IPM codificado
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<Warning>	+	o	O	WARNING	<Error-parameter>	–	
Deberá codificarse una sola parte de cuerpo							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Formato de transferencia de la parte de cuerpo a codificar
<Document>	B	m	I	FILENAME	<File-parameter>	–	Parte de cuerpo individual a transmitir y entregar de acuerdo con el formato de transferencia identificado por la palabra clave Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Especifica el tipo de documento a codificar
Deberá codificarse una o varias partes de cuerpo							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Especifica una lista de ficheros; utiliza una sintaxis especial
Elementos de sintaxis que contienen la información de encabezamiento para el IPM que se ha de codificar							
<IpmId>	B	m	I	IPM-ID	<Ipm-id-parameter>	–	Identificador de mensaje
<S-RecipientSpec>	B	m	I	ADDRESS	"@" <File-of-s-recipientspec>	–	Trayecto del fichero que contiene las Recipientspec de todos los recibientes deseados
<S-OriginatorSpec>	B	m	I	ORIGINATOR	"@" <File-of-originatorspec>	–	Nombre de trayecto del fichero que contiene los usuarios originador y autorizante
<ContType>	B	o	I	CONT-TYPE	<Content-type-parameter>	"mhsIPM84"	Tipo de contenido
<ExpiryTime>	B	o	I	EXPIRYTIME	<Date-time-parameter>	–	Expiración del mensaje

**Elementos de sintaxis adicionales para la <EncodeIPMExtension> de servicios de correo electrónico**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<Importance>	B	o	I	IMPORTANCE	<Importance-parameter>	"1"	Importancia del contenido
<Priority>	B	o	I	PRIORITY	<Priority-parameter>	"0"	Prioridad del contenido
<RelatedSpec>	B	o	I	RELATED	"@" <File-of-Relatedspec>	–	Nombre de trayecto del fichero que contiene las Relatedspec de todos los mensajes relacionados con el presente mensaje
<ReplyId>	B	o	I	REPLYID	<Ipm-id-parameter>	–	ID del mensaje a que se responde con éste
<ReplyTime>	B	o	I	REPLYTIME	<Date-time-parameter>	–	Fecha y hora a la que el recipiente o recibientes del mensaje deberán haber respondido a los usuarios autorizantes (tiempo de respuesta)
<Sensitivity>	B	o	I	SENSITIVITY	<Sensitivity-parameter>	"0"	Sensibilidad del contenido
<Subject>	B	o	I	SUBJECT	<Subject-parameter>	–	
<Userinfo>	+	o	I	USERINFO	<Userinfo-parameter>	–	Identificador de sobre facultativo proporcionado por el usuario

CUADRO 107/T.611

**Elementos de sintaxis adicionales para la <DecodeIPMExtension>  
de servicios de correo electrónico**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
Elementos de sintaxis pertenecientes al propio procedimiento de decodificación (descripción de "job")							
<DecodeIPM-SubFunction>	B	m	I	SUBFUNC	"DecodeIPM"	–	
<Message>	B	m	I	MESSAGE	<Path-parameter>	–	Trayecto del fichero que contiene el IPM a codificar
<Minor>	+	o	O	MINOR	<Error-parameter>	–	
<Warning>	+	o	O	WARNING	<Error-parameter>	–	
Si sólo una parte de cuerpo está presente							
<Convert>	B	m	O	CONVERT	<Convert-id-parameter>	–	Indica el formato de transferencia de la parte de cuerpo decodificada
<Document>	B	m	I/O	FILENAME	<File-parameter>	–	El nombre de fichero puede haber sido fijado previamente por la LA en la petición. Si se recibe una sola parte de cuerpo, la CA retendrá el nombre. Si se reciben varias partes de cuerpo, la CA puede escribir "aplastando" el nombre anteriormente escrito
<Type>	B	o	O	TYPE	<Type-id-parameter>	–	Especifica el tipo de documento a recibir
Si muchas partes de cuerpo están presentes							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Especifica una lista de ficheros; utiliza una sintaxis especial
Elementos de sintaxis que contienen la información de encabezamiento para el IPM que se va a decodificar							
<IpmId>	B	m	O	IPM-ID	<Ipm-id-parameter>	–	Identificador de mensaje
<R-RecipientSpec>	B	m	I	RECIPIENT	"@" <File-of-r-recipient-spec>	–	Trayecto del fichero que contiene las Recipientspec de todos los recibientes deseados

**Elementos de sintaxis adicionales para la <DecodeIPMExtension>  
de servicios de correo electrónico**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<R-OriginatorSpec>	B	m	I	ADDRESS	"@" <File-of-originatorspec>	–	Nombre de trayecto del fichero que contiene los usuarios originador y autorizante
<ContType>	B	o	O	CONT-TYPE	<Content-type-parameter>	–	Tipo de contenido
<ExpiryTime>	B	o	O	EXPIRYTIME	<Date-time-parameter>	–	Expiración del mensaje (tiempo de expiración)
<Forwarded>	B	o	O	FORWARDED	<Boolean-parameter>	–	Indicación de reenvío automático
<Importance>	B	o	O	IMPORTANCE	<Importance-parameter>	–	Importancia del contenido
<Language>	B	o	O	LANGUAGE	<Language-id-parameter>	–	Identificación del idioma utilizado
<Priority>	B	o	O	PRIORITY	<Priority-parameter>	–	Prioridad del contenido
<RelatedSpec>	B	o	O	RELATED	"@" <File-of-Relatedspec>	–	Nombre de trayecto del fichero que contiene las Relatedspec de todos los mensajes relacionados con el presente mensaje
<ReplyId>	B	o	O	REPLYID	<Ipm-id-parameter>	–	ID del mensaje a que se responde con éste
<ReplyTime>	B	o	O	REPLYTIME	<Date-time-parameter>	–	Fecha y hora para la el recipiente o recipientes del mensaje deberán haber respondido a los usuarios autorizantes (tiempo de respuesta)
<Sensitivity>	B	o	O	SENSITIVITY	<Sensitivity-parameter>	"0"	Sensibilidad del contenido
<Subject>	B	o	O	SUBJECT	<Subject-parameter>	–	
<Userinfo>	+	o	O	USERINFO	<Userinfo-parameter>	–	Proporcionado por el usuario

## 16.4 Notificación interpersonal

Una LA puede transmitir una notificación interpersonal (IPN) con una TDD de envío y puede recibir una IPN con una TDD de recepción.

La CA trata la IPN como un documento especial denominado documento IPN, esto es, la CA espera de la LA un fichero especialmente preparado que contenga los parámetros IPN, cuando hay que enviar una IPN. La CA convertirá el documento IPN en el formato requerido por el sistema de correo electrónico y lo transmitirá al recipiente. Por el contrario, la CA presentará un documento IPN a la LA cuando reciba una IPN a través de la red.

Para distinguir entre IPM e IPN (y DR, véase 16.5) se utiliza el parámetro Content-type-parameter del elemento de sintaxis <ContType> (véase también 16.2.2.9).

El documento IPN se ajustará a las mismas reglas sintácticas aplicables a las TDD.

En las cláusulas siguientes se describe la sintaxis, y la codificación basada en texto, del documento IPN.

### 16.4.1 Sintaxis del documento IPN

El documento IPN se define para el MHS y se ajusta a la siguiente sintaxis de estilo BNF (véase el Cuadro 108).

```
<IPN-Document> ::=      <NotifyType> <SubjectIPM> <ReceiptTime>
                        [<IPNOrigin>] [<PrefRecipient>] [<Reason>]
                        [<AutoComment>] [<SupplmInfo>]
```

CUADRO 108/T.611

#### Elementos de sintaxis del documento IPN

Elemento de sintaxis	Finalidad
<AutoComment>	Comentario del originador de esta notificación, si el correspondiente IPM fue reenviado automáticamente; sólo se utiliza para notificaciones de tipo NRN-FORW
<Reason>	Identifica el motivo por el cual se descartó el IPM relacionado con esta notificación: 0: ipm-expired (expiración del IPM) 1: ipm-obsolete (IPM obsoleto) 2: user-subscription-terminated (terminación del abono del usuario) sólo se utiliza para notificaciones de no recepción de tipo NRN-DISC
<IPNOrigin>	Identifica el originador de esta notificación
<NotifyType>	Tipo de la notificación: RN: notificación de recepción NRN-DISC: notificación de recepción (descartado) NRN-FORW: notificación de no recepción (reenviado automáticamente) OTHER: otros tipos de notificación
<PrefRecipient>	El recipiente preferido del IPM relacionado con esta notificación
<ReceiptTime>	Identifica la hora (tiempo) a que el originador de esta notificación recibió el correspondiente IPM; sólo se utiliza para notificación de recepción (RN)
<SubjectIPM>	Identifica el IPM relacionado con esta notificación
<SupplmInfo>	Información suplementaria del MTA de origen; sólo se utiliza para notificación de recepción (RN)

## 16.4.2 Codificación basada en texto

### 16.4.2.1 Correspondencia de las palabras clave

Véase el Cuadro 109.

CUADRO 109/T.611

#### Codificación basada en texto de los elementos de sintaxis IPN para servicios de correo electrónico

(El símbolo ↵ significa nueva línea)

Elemento de sintaxis	Par palabra clave/parámetro
<AutoComment>	"AUTO-COMMENT" ":" STRING (SIZE(1..256)) ↵
<Reason>	"REASON" ":" "0"   "1"   "2" ↵
<IPNOrigin>	"IPN-ORIGIN" ":" <Mhs-or-descriptor> ↵
<NotifyType>	"NOTIFY-TYPE" ":" "RN"   "NRN-DISC"   "NRN-FORW"   "OTHER" ↵
<PrefRecipient>	"DISCLO-REC" ":" <Mhs-or-descriptor> ↵
<ReceiptTime>	"RECEIPT-TIME" ":" <Date-time-parameter> ↵
<SubjectIPM>	"SUBJECT-IPM" ":" "@" <Ipm-id-parameter> ↵
<SupplmInfo>	"SUPPLM-INFO" ":" STRING (SIZE(1..256)) ↵

### 16.4.2.2 Codificación de parámetros

Para la codificación basada en texto de:

- <Mhs-or-descriptor> véase 16.2.2.4
- <Date-time-parameter> véase 6.4.4.4
- <Ipm-id-parameter> véase 16.2.2.11.

## 16.5 Informe de entrega

Una LA puede recibir un informe de entrega (DR) con una TDD.

La CA trata el DR como un documento especial denominado documento DR. La CA presentará ese documento DR a la LA cuando reciba un DR a través de la red.

Para distinguir entre IPM y DR (e IPN, véase 16.4) se utiliza el parámetro Content-type-parameter del elemento de sintaxis <ContType> (véase también 16.2.2.9).

El documento DR entregado a la CA se ajustará exactamente a las mismas reglas sintácticas aplicadas a las TDD.

En las cláusulas siguientes se describe la sintaxis, y la codificación basada en texto, del documento DR.

### 16.5.1 Sintaxis del documento DR

El documento DR se define para el MHS y se ajusta a la siguiente sintaxis de estilo BNF (véase el Cuadro 110).

```
<DR-Document> := <ReportType> <ActRecipient> <MsgSubId> <ReceiveTime>]  
[<Reason>] [<Diagnostic>] [<SupplmInfo>]
```

CUADRO 110/T.611

**Elementos de sintaxis del documento DR**

Elemento de sintaxis	Finalidad
<ActRecipient>	Identifica el recipiente efectivo del IPM relacionado con este informe
<Diagnostic>	Información de diagnóstico adicional al informe, definida en la Recomendación UIT-T X.411; sólo se utiliza para informes de no entrega (NDR)
<MsgSubId>	Identifica el IPM relacionado con este informe (MTSIdentifier)
<Reason>	Identifica el motivo (o la razón) por el cual el IPM relacionado con este informe no fue entregado, como prescribe la Recomendación UIT-T X.411; sólo se utiliza para informes de no entrega (NDR)
<ReceiveTime>	Hora (tiempo) a la que se entregó el mensaje (DeliveryTime); sólo se utiliza para informes de entrega (DR)
<ReportType>	Tipo del informe de entrega: DR: informe de entrega NDR: informe de no entrega
<SupplmInfo>	Información suplementaria del MTA de origen

**16.5.2 Codificación basada en texto**

**16.5.2.1 Correspondencia de las palabras clave**

Véase el Cuadro 111.

CUADRO 111/T.611

**Codificación basada en texto de los elementos de sintaxis DR para servicios de correo electrónico**

(El símbolo ↵ significa nueva línea)

Elemento de sintaxis	Par palabra clave/parámetro
<ActRecipient>	"ACT-RECIPIENT" ":" <Mhs-or-name> ↵
<Diagnostic>	"DIAGNOSTIC" ":" NUMERIC-STRING ↵
<MsgSubId>	"MSG-SUB-ID" ":" <Msg-sub-id-parameter> ↵
<Reason>	"REASON" ":" NUMERIC-STRING ↵
<ReceiveTime>	"RCVTIME" ":" <Date-time-parameter> ↵
<ReportType>	"REPORT-TYPE" ":" "DR"   "NDR" ↵
<SupplmInfo>	"SUPPLM-INFO" ":" STRING (SIZE(1..256)) ↵

**16.5.2.2 Codificación de parámetros**

Para la codificación basada en texto de:

- <Mhs-or-name> véase 16.2.2.4
- <Msg-sub-id-parameter> véase 16.2.2.13
- <Date-time-parameter> véase 6.4.4.4.

## 16.6 Correspondencia de los elementos de servicio MHS

El Cuadro 112 muestra la correspondencia de los elementos de servicio MHS definidos en las Recomendaciones UIT-T de la serie X.400 con los elementos de sintaxis definidos en la presente Recomendación. Se indican las palabras clave correspondientes utilizadas en la presentación basada en texto de las TDD, y si los elementos de servicio están o no soportados para una LA emisora y/o receptora.

CUADRO 112/T.611

### Correspondencia de los elementos de servicio MHS

Elemento de servicio MHS	Elemento de sintaxis	Palabra clave	LA emisora	LA receptora
Indicación de destinatario alternativo autorizado	<Alternate>	ALTERNATE	Soportado	No soportado
Indicación de usuario autorizante	<R-OriginatorSpec> <S-OriginatorSpec>	ADDRESS ORIGINATOR	– Soportado	Soportado –
Indicación de mensaje reenviado automáticamente	<Forwarded>	FORWARDED	No soportado	Soportado
Indicación de destinatarios de copia ciega	<R-RecipientSpec> <S-RecipientSpec>	RECIPIENT ADDRESS	– Soportado	Soportado –
Indicación de tipo de contenido	<ContType>	CONT-TYPE	Soportado	Soportado
Prohibición de conversión	<ImplicitConv>	IMPLICIT-CONV	Soportado	No soportado
Indicación de referencias cruzadas	<RelatedSpec>	RELATED	Soportado	Soportado
Indicación de entrega diferida	<SendTime>	SENDTIME	Soportado	No soportado
Peticion de informe de entrega	<R-RecipientSpec> <S-RecipientSpec>	RECIPIENT ADDRESS	– Soportado	Soportado –
Indicación de hora de entrega	<ReceiveTime>	RCVTIME	No soportado	Soportado
Revelación de otros destinatarios	<DiscloRec>	DISCLO-REC	Soportado	No soportado
Indicación de fecha de expiración	<ExpiryTime>	EXPIRYTIME	Soportado	Soportado
Mensaje IP reenviado	<Type>	TYPE	Soportado	Soportado
Selección de grado de entrega	<Priority>	PRIORITY	Soportado	Soportado
Indicación de importancia	<Importance>	IMPORTANCE	Soportado	Soportado
Indicación de copia incompleta <sup>a)</sup>	<IncompleteCopy>	INC-Copy	No soportado	Soportado
Indicación de destinatario deseado	<R-RecipientSpec>	RECIPIENT	No soportado	Soportado
Indicación de mensaje IP	<IpmId>	IPM-ID	Soportado	Soportado

CUADRO 112/T.611 (fin)

Correspondencia de los elementos de servicio MHS

Elemento de servicio MHS	Elemento de sintaxis	Palabra clave	LA emisora	LA receptora
Indicación de idioma	<Language>	LANGUAGE	No soportado	Soportado
Indicación de tiempo límite para la entrega <sup>a)</sup>	<LastTime>	LASTTIME	Soportado	No soportado
Identificación de mensaje <sup>a)</sup>	<Userinfo>	USERINFO	Soportado	Soportado
Identificación de depósito de mensaje	<MsgSubId>	MSG-SUB-ID	Soportado <sup>b)</sup>	No soportado
Entrega multidespacho	<R-RecipientSpec> <S-RecipientSpec>	RECIPIENT ADDRESS	– Soportado	Soportado –
Múltiples partes de cuerpo	implícito	n/a <sup>c)</sup>	Soportado	Soportado
Notificación de no recepción	<R-RecipientSpec> <S-RecipientSpec>	RECIPIENT ADDRESS	– Soportado	Soportado –
Informe de no entrega	<R-RecipientSpec> <S-RecipientSpec>	RECIPIENT ADDRESS	– Soportado	Soportado –
Indicación de obsolescencia de IPM	<RelatedSpec>	RELATED	Soportado	Soportado
Indicación de originador	<R-OriginatorSpec> <S-OriginatorSpec>	ADDRESS ORIGINATOR	– Soportado	Soportado –
Indicación de destinatarios primarios y de copia	<R-RecipientSpec> <S-RecipientSpec>	RECIPIENT ADDRESS	– Soportado	Soportado –
Indicación de sonda	<Convert>	CONVERT	Soportado	No soportado
Notificación de recepción	<R-RecipientSpec> <S-RecipientSpec>	RECIPIENT ADDRESS	– Soportado	Soportado –
Indicación de mensaje IP de respuesta	<ReplyId>	REPLYID	Soportado	Soportado
Indicación de destinatarios de respuesta	<R-RecipientSpec> <S-RecipientSpec>	RECIPIENT ADDRESS	– Soportado	Soportado –
Indicación de IPM de respuesta	<RelatedSpec>	RELATED	Soportado	Soportado
Indicación de tiempo de respuesta	<ReplyTime>	REPLYTIME	Soportado	Soportado
Indicación de sensibilidad	<Sensitivity>	SENSITIVITY	Soportado	Soportado
Indicación de asunto	<Subject>	SUBJECT	Soportado	Soportado
Indicación de tiempo de depósito	<SubmitTime>	SUBMITTIME	Soportado <sup>b)</sup>	Soportado

a) Los elementos de servicio IncompleteCopyIndication, LatestDeliveryIndication y MessageIdentification son adicionales. Por consiguiente, no se tratan en la definición de perfil.

b) Los elementos de servicio MessageSubmissionIdentification y SubmissionTimeStamp son originados y mantenidos por el sistema de transferencia de mensajes (MTS). Ambas palabras clave se utilizan en la <SendAckTDD> como parámetro de salida.

c) La asignación de una palabra clave no es aplicable.

El Cuadro 113 recapitula los sellos de tiempo que se utilizan en esta Recomendación en relación con elementos de servicio del sistema de transferencia de mensajes y el sistema de mensajería interpersonal.

Se muestran estos elementos de servicio, su significado y las correspondientes palabras clave de la codificación basada en texto. En otra columna de este cuadro se indica si los sellos de tiempo pueden ser especificados por la LA emisora (TDD de envío) y si pueden ser recibidos por la LA receptora (TDD de recepción).

CUADRO 113/T.611

**Resumen de los sellos de tiempo del MHS**

Elemento de servicio	Palabra clave	Emisión/recepción	Significado
Indicación de fecha de expiración	EXPIRYTIME	em/rec	Este campo de encabezamiento facultativo de un IPM especifica la fecha (tiempo) en la que el usuario o usuarios autorizantes consideran que el IPM pierde su validez
Indicación de hora de respuesta	REPLYTIME	em/rec	Este campo de encabezamiento facultativo de un IPM especifica la hora (tiempo) para la cual el usuario o usuarios autorizantes solicitan que las eventuales respuestas al IPM asunto hayan sido originadas
Indicación de entrega diferida	SENDTIME	em/-	Especifica la hora (tiempo) antes de la cual el mensaje no deberá entregarse al recipiente o recipientes; puede ser generado por el originador del mensaje
Indicación de tiempo límite para la entrega	LASTTIME	em/-	Especifica la hora (tiempo) después de la cual el mensaje no deberá entregarse al recipiente o recipientes; puede ser generado por el originador del mensaje
Indicación de tiempo de entrega	RCVTIME	-/rec	Especifica la hora (tiempo) a la que el mensaje se entregó al sistema de transferencia de mensajes del recipiente; deberá ser generado por el sistema de transferencia de mensajes, si el mensaje fue debidamente entregado
Indicación de tiempo de depósito	SUBMITTIME	em/rec	Especifica la hora (tiempo) a la que el sistema de transferencia de mensajes acepta la responsabilidad del mensaje; será generado por el sistema de transferencia de mensajes
Indicación de tiempo de recepción	RECEIPT-TIME	em/rec	Sólo se utiliza para notificación de recepción (RN); identifica la hora (tiempo) a la que el originador de una notificación recibió el IPM correspondiente

**16.7 Perfiles de correo electrónico**

Esta Recomendación prevé el acceso a los servicios de correo electrónico descritos en las Recomendaciones UIT-T de la serie X.400, así como a otros entornos de correo electrónico.

Los potentes servicios de telecomunicaciones basados en almacenamiento y transmisión ("store-and-forward") imponen elevadas exigencias a la especificación de una interfaz. Las TDD de envío, acuse de recibo de envío, y recepción son enriquecidas con un gran número de elementos específicos del correo electrónico básico que se ajustan a los principios de esta Recomendación. La mayor parte de estos elementos de servicio, aquí tratados, son facultativos.

Las características enriquecidas proporcionadas a nivel de interfaz para el acceso al correo electrónico son:

- transmisión y recepción de notificaciones;
- recepción de informes de entrega;
- transmisión y recepción de mensaje o mensajes de correo electrónico insertados en un mensaje de correo electrónico (por ejemplo, IPM como parte de cuerpo).

Se requiere que una implementación de CA se ajuste a esta Recomendación, aunque no admita todas las características posibles. Para mejorar la interoperabilidad entre las LA y las CA de diferentes vendedores, se introduce el concepto de "perfil".

Los perfiles:

- X400-IPM,
- BasicX400-IPM,
- SimpleEMail

definen subconjuntos de servicio funcionales ofrecidos en el nivel de interfaz descrito en esta Recomendación.<sup>15)</sup>

Junto con el ICE y las clases funcionales, el concepto de perfil identifica los elementos de servicio de correo electrónico admitidos por la CA.

Una CA puede admitir uno o más perfiles. Además, los realizadores pueden describir y denominar sus propios perfiles.

#### **16.7.1 Perfil de servicio X400-IPM**

El perfil X400-IPM comprende todos los elementos de servicio X.400 proporcionados en el nivel de interfaz de esta Recomendación para el soporte del transporte de los IPM<sup>16)</sup> (véase el Cuadro 114).

#### **16.7.2 Perfil de servicio BasicX400-IPM**

El perfil BasicX400-IPM especifica el subconjunto funcional básico de elementos de servicio X.400 proporcionados en el nivel de interfaz de esta Recomendación para el soporte del transporte de IPM (véase el Cuadro 115).

#### **16.7.3 Perfil de servicio SimpleEMail**

El perfil SimpleEMail se introdujo para permitir el acceso a servicios de correo electrónico que no se ajustan a las Recomendaciones UIT-T de la serie X.400 sobre la comunicación de sistemas abiertos. Proporciona un subconjunto funcional mínimo de elementos de servicio de correo electrónico ("E-Mail") en el nivel de interfaz de esta Recomendación (véase el Cuadro 116).

### **16.8 Fijaciones del descriptor de CA**

Una CA que soporte servicios de correo electrónico ("E-Mail") especificará en el descriptor de CA las partes de cuerpo (formato de transmisión de documento), el perfil o perfiles, y los tipos de contenido soportados (véase también 9.5).

Para el soporte de las TDD EncodeIPM y DecodeIPM hay que dar los valores correspondientes a la palabra clave EXTEND del descriptor de CA. Esto se muestra en el Cuadro 117.

## **17 Servicio: transferencia de ficheros**

Esta Recomendación permite enviar y recibir información a través de servicios de transferencia de ficheros. Sin embargo, no se pretende un control exhaustivo de los servicios de transferencia de ficheros a través de la interfaz.

---

<sup>15)</sup> Otros perfiles como EDIM y EDIN quedan en estudio.

<sup>16)</sup> El perfil X400-IPM está especificado de acuerdo con el European Procurement Handbook for Open Systems (EPHOS), es decir, con los perfiles ENV 41201 y ENV 41202.

CUADRO 114/T.611

Perfil de servicio X400-IPM

Elemento de servicio	Elemento de sintaxis	Soporte de CA
Indicación de destinatario alternativo autorizado	<Alternate>	Obligatorio
Indicación del usuario autorizante	<R-OriginatorSpec>, <S-OriginatorSpec>	Obligatorio
Indicación de mensaje reenviado automáticamente	<Forwarded>	Obligatorio
Indicación de destinatarios de copia ciega	<R-RecipientSpec>, <S-RecipientSpec>	Obligatorio
Indicación de tipo de contenido	<ContType>	Obligatorio
Prohibición de conversión	<ImplicitConv>	Obligatorio
Indicación de referencias de remisión (o "cruzadas")	<RelatedSpec>	Obligatorio
Indicación de entrega diferida	<SendTime>	Obligatorio
Informe de entrega	<R-RecipientSpec>, <S-RecipientSpec>	Obligatorio
Indicación de hora de entrega	<ReceiveTime>	Obligatorio
Revelación de otros destinatarios	<DisclRec>	Obligatorio
Indicación de fecha de expiración	<ExpiryTime>	Obligatorio
Mensaje IP reenviado	<Type>	Obligatorio
Selección de grado de entrega	<Priority>	Obligatorio
Indicación de importancia	<Importance>	Obligatorio
Indicación de destinatario deseado	<R-RecipientSpec>	Obligatorio
Indicación de mensaje IP	<IpmlId>	Obligatorio
Indicación de idioma	<Language>	Obligatorio
Indicación de depósito de mensaje	<MsgSubId>	Obligatorio
Entrega multidestino	<R-RecipientSpec>, <S-RecipientSpec>	Obligatorio
Múltiples partes de cuerpo	implicit	Obligatorio
Notificación de no recepción	<R-RecipientSpec>, <S-RecipientSpec>	Obligatorio
Informe de no entrega	<R-RecipientSpec>, <S-RecipientSpec>	Obligatorio
Indicación de obsolescencia de IPM	<RelatedSpec>	Obligatorio
Indicación de originador	<R-OriginatorSpec>, <S-OriginatorSpec>	Obligatorio
Indicación de destinatarios primarios y de copia	<R-RecipientSpec>, <S-RecipientSpec>	Obligatorio
Indicación de sonda	<Convert>	Obligatorio
Notificación de recepción	<R-RecipientSpec>, <S-RecipientSpec>	Obligatorio
Indicación de mensaje IP de respuesta	<ReplyId>	Obligatorio
Indicación de destinatarios de respuesta	<R-RecipientSpec>, <S-RecipientSpec>	Obligatorio
Indicación de IPM de respuesta	<RelatedSpec>	Obligatorio
Indicación de hora de respuesta	<ReplyTime>	Obligatorio
Indicación de sensibilidad	<Sensitivity>	Obligatorio
Indicación de asunto	<Subject>	Obligatorio
Indicación de hora de depósito	<SubmitTime>	Obligatorio

CUADRO 115/T.611

**Perfil de servicio BasicX400-IPM**

Elemento de servicio	Elemento de sintaxis	Soporte de CA
Indicación de recipiente alternativo autorizado	<Alternate>	–
Indicación del usuario autorizante	<R-OriginatorSpec>, <S-OriginatorSpec>	Obligatorio
Indicación de mensaje reenviado automáticamente	<Forwarded>	–
Indicación de recipientes de copia ciega	<R-RecipientSpec>, <S-RecipientSpec>	Obligatorio
Indicación de tipo de contenido	<ContType>	Obligatorio
Prohibición de conversión	<ImplicitConv>	–
Indicación de referencias de remisión (o "cruzadas")	<RelatedSpec>	Obligatorio
Indicación de entrega diferida	<SendTime>	Obligatorio
Informe de entrega	<R-RecipientSpec>, <S-RecipientSpec>	–
Indicación de hora de entrega	<ReceiveTime>	–
Revelación de otros recipientes	<DisclRec>	–
Indicación de fecha de expiración	<ExpiryTime>	Obligatorio
Mensaje IP reenviado	<Type>	Obligatorio
Selección de grado de entrega	<Priority>	Obligatorio
Indicación de importancia	<Importance>	Obligatorio
Indicación de recipiente deseado	<R-RecipientSpec>	–
Indicación de mensaje IP	<IpmId>	Obligatorio
Indicación de idioma	<Language>	Obligatorio
Indicación de depósito de mensaje	<MsgSubId>	–
Entrega multidesfino	<R-RecipientSpec>, <S-RecipientSpec>	Obligatorio
Múltiples partes de cuerpo	implicit	Obligatorio
Notificación de no recepción	<R-RecipientSpec>, <S-RecipientSpec>	–
Informe de no entrega	<R-RecipientSpec>, <S-RecipientSpec>	–
Indicación de obsoletización de IPM	<RelatedSpec>	Obligatorio
Indicación de originador	<R-OriginatorSpec>, <S-OriginatorSpec>	Obligatorio
Indicación de recipientes primarios y de copia	<R-RecipientSpec>, <S-RecipientSpec>	Obligatorio
Indicación de sonda	<Convert>	–
Notificación de recepción	<R-RecipientSpec>, <S-RecipientSpec>	–
Indicación de mensaje IP de respuesta	<ReplyId>	Obligatorio
Indicación de recipientes de respuesta	<R-RecipientSpec>, <S-RecipientSpec>	Obligatorio
Indicación de IPM de respuesta	<RelatedSpec>	Obligatorio
Indicación de hora de respuesta	<ReplyTime>	Obligatorio
Indicación de sensibilidad	<Sensitivity>	Obligatorio
Indicación de asunto	<Subject>	Obligatorio
Indicación de hora de depósito	<SubmitTime>	–

CUADRO 116/T.611

**Perfil de servicio SimpleEMail**

Elemento de servicio	Elemento de sintaxis	Soporte de CA
Indicación de destinatario alternativo autorizado	<Alternate>	–
Indicación del usuario autorizante	<R-OriginatorSpec>, <S-OriginatorSpec>	Obligatorio
Indicación de mensaje reenviado automáticamente	<Forwarded>	–
Indicación de destinatarios de copia ciega	<R-RecipientSpec>, <S-RecipientSpec>	Obligatorio
Indicación de tipo de contenido	<ContType>	Obligatorio
Prohibición de conversión	<ImplicitConv>	–
Indicación de referencias de remisión (o "cruzadas")	<RelatedSpec>	Obligatorio
Indicación de entrega diferida	<SendTime>	–
Informe de entrega	<R-RecipientSpec>, <S-RecipientSpec>	–
Indicación de hora de entrega	<ReceiveTime>	–
Revelación de otros destinatarios	<DisclóRec>	–
Indicación de fecha de expiración	<ExpiryTime>	–
Mensaje IP reenviado	<Type>	–
Selección de grado de entrega	<Priority>	Obligatorio
Indicación de importancia	<Importance>	–
Indicación de destinatario deseado	<R-RecipientSpec>	–
Indicación de mensaje IP	<IpmId>	Obligatorio
Indicación de idioma	<Language>	–
Indicación de depósito de mensaje	<MsgSubId>	–
Entrega multidestino	<R-RecipientSpec>, <S-RecipientSpec>	Obligatorio
Múltiples partes de cuerpo	implicit	Obligatorio
Notificación de no recepción	<R-RecipientSpec>, <S-RecipientSpec>	–
Informe de no entrega	<R-RecipientSpec>, <S-RecipientSpec>	–
Indicación de obsolescencia de IPM	<RelatedSpec>	–
Indicación de originador	<R-OriginatorSpec>, <S-OriginatorSpec>	Obligatorio
Indicación de destinatarios primarios y de copia	<R-RecipientSpec>, <S-RecipientSpec>	Obligatorio
Indicación de sonda	<Convert>	–
Notificación de recepción	<R-RecipientSpec>, <S-RecipientSpec>	–
Indicación de mensaje IP de respuesta	<ReplyId>	Obligatorio
Indicación de destinatarios de respuesta	<R-RecipientSpec>, <S-RecipientSpec>	Obligatorio
Indicación de IPM de respuesta	<RelatedSpec>	Obligatorio
Indicación de hora de respuesta	<ReplyTime>	Obligatorio
Indicación de sensibilidad	<Sensitivity>	Obligatorio
Indicación de asunto	<Subject>	Obligatorio
Indicación de hora de depósito	<SubmitTime>	–

CUADRO 117/T.611

Valores adicionales fijados en el descriptor de CA para servicios de correo electrónico

Palabra clave	Parámetro	CA declara
EMAIL	"STD"	Soporte de partes de cuerpo de texto; "STD" da la correspondencia con el alfabeto básico apropiado del servicio de correo electrónico seleccionado
EMAIL	"TELETEX"	Soporte de partes de cuerpo del teletex
EMAIL	"VIDEOTEX"	Soporte de partes de cuerpo del videotex
EMAIL	"G3FAX"	Soporte de partes de cuerpo del telefax grupo 3
EMAIL	"G4CLASS1"	Soporte de partes de cuerpo del telefax grupo 4
EMAIL	"MIXEDMODE"	Soporte de partes de cuerpo del modo mixto
EMAIL	"MESSAGE"	Soporte de partes de cuerpo de IPM
EMAIL	"BILATERAL" <sup>a)</sup>	Soporte de partes de cuerpo de definidas por acuerdo bilateral
EMAIL	"NATIONAL"	Soporte de partes de cuerpo de definidas en el plano nacional
EMAIL	"ODA"	Soporte de partes de cuerpo de ODA
CONT-TYPE	"mhsIPMxx"	Soporte del protocolo de contenido IPM del MHS (xx representa el año de la versión de la Recomendación correspondiente, por ejemplo 84 para 1984)
CONT-TYPE	"mhsIPNxx"	Soporte del protocolo de contenido IPN del MHS (la CA entiende los documentos IPN como se describe en 16.4)
CONT-TYPE	"mhsDR"	Soporte del protocolo de contenido DR del MHS (la CA entiende los documentos DR como se describe en 16.5)
PROFILE	"X400-IPM"	Soporte del perfil X400
PROFILE	"BasicX400-IPM"	Soporte del perfil BasicX400
PROFILE	"SimpleEMail"	Soporte del perfil SimpleEMail
EXTEND	"EncodeIPM"	Soporte de la funcionalidad EncodeIPM
EXTEND	"DecodeIPM"	Soporte de la funcionalidad DecodeIPM
<p><sup>a)</sup> "Bilateral" se utiliza para otros tipos de partes de cuerpo no mencionados (en esta lista).</p> <p>NOTA – Una CA puede también soportar otros sistemas de correo electrónico, o sistemas de correo electrónico privados. En este caso, el tipo de sistema de correo electrónico debe indicarse en los parámetros CONT-TYPE y PROFILE (por ejemplo, "CONT-TYPE : cmcIPM" para el soporte de interfaces CMC, véase 16.2.2.9).</p>		

### 17.1 Elementos de sintaxis específicos del servicio

<ServiceDependentKeywordsSend> :=  
 (<Recipient> | <RecipientSpec>)  
 (((<Document> <Convert> [<Type>]) | <DocumentSpec>)  
 <Compress> <Environ> [<Password>] [<Name>])

<ServiceDependentKeywordsSendAck> :=  
 <Recipient>  
 (((<Document> <Convert> [<Type>]) | <DocumentSpec>)  
 <Compress> <Environ> [<Password>] [<Name>])

<ServiceDependentKeywordsReceive> :=  
 [<Originator>]  
 ((<Document> <Convert> <Type>) | <DocumentSpec>)  
 <Compress> <Environ> [<Password>] [<Name>]

Véase el Cuadro 118.

CUADRO 118/T.611

**Elementos de sintaxis adicionales para servicios de transferencia de ficheros**

Elemento de sintaxis	Finalidad
<Compress>	Especifica el formato de compresión de datos del fichero o ficheros transferidos
<Convert>	Especifica el formato de transferencia que habrá de utilizarse
<Document>, <DocumentSpec>	Especifica el documento o documentos que se enviarán o que se están recibiendo
<Environ>	Especifica el sistema operativo utilizado por el originador para crear el fichero
<Name>	Nombre del fichero en el sistema operativo del originador
<Originator>	Especifica la dirección de comunicaciones del originador
<Password>	Contraseña necesaria para el acceso a una estación distante
<Recipient>, <RecipientSpec>	Especifica la dirección o direcciones de comunicaciones del recipiente o recipientes
<Type>	Especifica el formato de transmisión utilizado

**17.2 Codificación basada en texto**

**17.2.1 Correspondencia de palabras clave**

Véase el Cuadro 119.

CUADRO 119/T.611

**Codificación basada en texto de elementos adicionales para servicios de transferencia de ficheros**

(El símbolo ↵ representa nueva línea)

Elemento de sintaxis	Par palabra clave/parámetro
<Compress>	"COMPRESS" ":" <Compress-parameter> ↵
<Convert>	"CONVERT" ":" <Convert-id-parameter> ↵
<Document>	"FILENAME" ":" <File-parameter> ↵
<DocumentSpec>	"FILENAME" ":" "@" <File-of-filespec> ↵
<Environ>	"ENVIRON" ":" <Environ-parameter> ↵
<Name>	"NAME" ":" <Name-parameter> ↵
<Originator>	"ADDRESS" ":" <Address-parameter> ↵
<Password>	"PASSWORD" ":" <Password-parameter> ↵
<Recipient>	"ADDRESS" ":" <Address-parameter> ↵
<RecipientSpec>	"ADDRESS" ":" "@" <File-of-addrspec> ↵
<Type>	"TYPE" ":" <Type-id-parameter> ↵

## 17.2.2 Codificación de parámetros

Véase también 6.4.4 para los parámetros que no dependen del servicio.

### 17.2.2.1 Parámetro identificador de servicio

El parámetro service-id-parameter se codifica como una STRING fijada al valor constante "FT".

*Sintaxis:*

```
<Service-id-parameter> := "FT"
```

### 17.2.2.2 Parámetro identificador de tipo

El parámetro Type-id-parameter se codifica como una STRING fijada al valor constante "STD".

*Sintaxis:*

```
<Type-id-parameter> := "STD"
```

### 17.2.2.3 Parámetro identificador de conversión

El parámetro Convert-id-parameter se codifica como una STRING fijada al valor constante "VOID".

*Sintaxis:*

```
<Convert-id-parameter> := "VOID"
```

### 17.2.2.4 Parámetros File-of-addrspec y Address-parameter

El parámetro File-of-addrspec se codifica como un PATH (trayecto) que apunta a un fichero que contiene parámetros Addrspec-parameter, que a su vez contienen el Address-parameter.

El Address-parameter se codifica como STRING. La STRING contendrá el número de teléfono. Si el número de teléfono comienza con un carácter "!", puede contener caracteres especiales que se tratan como operadores (o modificadores) y no como dígitos marcados. Los caracteres autorizados quedan en estudio.

Como otra posibilidad, puede darse un nombre de alias en lugar del número de teléfono, a condición de que el alias se introduzca con un carácter "&". Se supone que la CA sabe cómo decodificar el alias especificado.

*Sintaxis:*

```
<File-of-addrspec> := PATH
-- El trayecto apunta a un fichero que contiene uno o más
-- parámetros <Addrsec-parameter>s

<Addrsec-parameter> := <Address-parameter>

<Address-parameter> := <Phone-number> | ("!" <Dial-command>) | ("&" <Alias>)
-- contiene número de teléfono, o secuencia marcada, o alias

<Phone-number> := NUMERIC-STRING

<Dial-command> := STRING
-- en estudio

<Alias> := STRING
```

### 17.2.2.5 Parámetros File-of-filespec y File-parameter

El parámetro File-of-filespec se codifica como un PATH (trayecto) que apunta a un fichero que contiene el Filespec-parameter, el cual a su vez contiene el File-parameter. El propio File-parameter se codifica también como un PATH, que apunta al fichero transferido.

*Sintaxis:*

<File-of-filespec> := PATH  
-- El trayecto apunta a un fichero que contiene uno o más  
-- <Filespec-parameter>

<Filespec-parameter> := <File-parameter> "," <File-conv>  
["," [<File-type>] ["," [<File-compress>] ["," <File-name>] ] ]

<File-parameter> := PATH  
-- Trayecto del fichero transferido

<File-conv> := <Convert-id-parameter>  
-- Especifica el formato de transferencia de un fichero

<File-type> := <Type-id-parameter>  
-- Especifica el formato de transmisión de un fichero

<File-compress> := <Compress-parameter>  
-- Especifica la compresión

<File-name> := <Name-parameter>  
-- Especifica el nombre del fichero

#### 17.2.2.6 Parámetro de compresión (Compress-parameter)

Compress-parameter se codifica como una STRING fijada a uno de los siguientes valores:

"VOID" No hay compresión (por defecto)  
Compresión AZJ  
cualquier otro Identificador de compresión

*Sintaxis:*

<Compress-parameter> := "VOID" | <Compress-identifier>  
<Compress-identifier> := STRING  
-- por ejemplo "ZIP", "AZJ", "ARJ", "LZH" o "V42"

#### 17.2.2.7 Parámetro de entorno (Environ-parameter)

Environ-parameter se codifica como una STRING fijada a uno de los siguientes valores:

"MSDOS" MSDOS  
"WINDOWS" Windows  
"UNIX" Unix  
"OS2" OS/2  
"MacOS" Mac  
cualquier otro Reservado para ulterior estudio

*Sintaxis:*

<Environ-parameter> := "MSDOS" | "WINDOWS" | "UNIX" | "OS2" | "MacOS"

#### 17.2.2.8 Parámetro de contraseña (Password-parameter)

Password-parameter se codifica como una STRING.

*Sintaxis:*

<Password-parameter> := STRING

#### 17.2.2.9 Parámetro de nombre (Name-parameter)

Name-parameter se codifica como una STRING con una longitud de 1 a 16 caracteres.

*Sintaxis:*

<Name-parameter> := STRING (SIZE(1..16))

## 17.3 Funcionalidad adicional

### 17.3.1 Función: Send y SendAck

Véanse los Cuadros 120 y 121.

CUADRO 120/T.611

#### Funcionalidad adicional de la <SendTDD> para transferencia de ficheros

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<Environ>	B	m	I	ENVIRON	<Environ-parameter>	–	Sistema operativo del emisor
<Password>	B	o	I	PASSWORD	<Password-parameter>	–	Contraseña para el acceso al sistema de ficheros distante
Envío a un solo destinatario							
<Recipient>	B	m	I	ADDRESS	<Address-parameter>	–	Especifica el número de llamada de un recipiente
Envío a uno o varios destinatarios							
<RecipientSpec>	+	m	I	ADDRESS	"@" <File-of-addrspec>	–	Especifica una lista de recipientes
Envío de un solo fichero							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	–	Indica el formato de transferencia del fichero saliente
<Document>	B	m	I	FILENAME	<File-parameter>	–	El fichero saliente individual que habrá de transmitirse, y entregarse de acuerdo con el formato de transferencia especificado por la palabra clave Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Especifica el tipo de documento a enviar
<Compress>	B	o	I	COMPRESS	<Compress-parameter>	"VOID"	Compresión
<Name>	B	o	I	NAME	<Name-parameter>	–	Nombre del fichero en el entorno original
Envío de uno o varios ficheros							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	–	Especifica una lista de ficheros; utiliza una sintaxis especial

CUADRO 121/T.611

**Funcionalidad adicional de la <SendAckTDD> para transferencia de ficheros**

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<Recipient>	B	m	I/O	ADDRESS	<Address-parameter>	-	Especifica un número de llamada de un recipiente
<Environ>	B	m	I	ENVIRON	<Environ-parameter>	-	Sistema operativo del emisor
<Password>	B	o	I	PASSWORD	<Password-parameter>	-	Contraseña para el acceso al sistema de ficheros distante
Envío de un solo fichero							
<Convert>	B	m	I	CONVERT	<Convert-id-parameter>	-	Indica el formato de transferencia del fichero saliente
<Document>	B	m	I	FILENAME	<File-parameter>	-	El fichero saliente individual que habrá de transmitirse, y entregarse de acuerdo con el formato de transferencia especificado por la palabra clave Convert
<Type>	B	o	I	TYPE	<Type-id-parameter>	"STD"	Especifica el tipo de documento a transmitir
<Compress>	B	o	I	COMPRESS	<Compress-parameter>	"VOID"	Compresión
<Name>	B	o	I	NAME	<Name-parameter>	-	Nombre del fichero en el entorno original
Envío de uno o varios ficheros							
<DocumentSpec>	+	m	I	FILENAME	"@" <File-of-filespec>	-	Especifica una lista de ficheros; utiliza una sintaxis especial

### 17.3.2 Función: Recibir (Receive)

Véase el Cuadro 122.

CUADRO 122/T.611

#### Funcionalidad adicional de la <ReceiveTDD> para transferencia de ficheros

Elemento de sintaxis	C	T	I/O	Codificación basada en texto			Observaciones
				Palabra clave	Parámetro	Por defecto	
<Environ>	B	m	O	ENVIRON	<Environ-parameter>	–	Sistema operativo del emisor
<Password>	B	o	I	PASSWORD	<Password-parameter>	–	Contraseña para el acceso al sistema de ficheros distante
<Originator>	B	o	O	ADDRESS	<Address-parameter>	–	Especifica un número de teléfono del originador
Recepción de un solo fichero							
<Convert>	B	m	O	CONVERT	<Convert-id-parameter>	–	Indica el formato de transferencia del fichero recibido
<Document>	B	m	I/O	FILENAME	<File-parameter>	–	El nombre del fichero puede haber sido fijado previamente por la LA en la petición. Si se recibe un solo fichero, la CA retendrá el nombre. Si se reciben varios ficheros, la CA puede escribir "aplastando" el nombre anteriormente escrito
<Type>	B	o	O	TYPE	<Type-id-parameter>	–	Especifica el tipo de documento recibido
<Compress>	B	o	O	COMPRESS	<Compress-parameter>	"VOID"	Compresión
<Name>	B	o	O	NAME	<Name-parameter>	–	Nombre del fichero en el entorno original
Recepción de varios ficheros							
<DocumentSpec>	+	m	O	FILENAME	"@" <File-of-filespec>	–	Especifica una lista de ficheros; utiliza una sintaxis especial

### 17.4 Fijaciones del descriptor de CA

Una CA que soporte el servicio de transferencia de ficheros especificará el parámetro Type-id-parameter "STD" (véase el Cuadro 123).

CUADRO 123/T.611

#### Valores adicionales del descriptor de CA para transferencia de ficheros

Palabra clave	Parámetro	CA declara
FT	"STD"	Soporte del servicio básico de transferencia de ficheros

## PARTE III – ESQUEMA DE CODIFICACIÓN BINARIA

### 18 Descripción del lenguaje C genérico

En esta cláusula se describe el esquema de codificación binaria para los TDD que utilizan una descripción del lenguaje C «genérico» (véase también A.2). El esquema de codificación binaria requiere que el identificador de código esté puesto a «C» (véase 6.3). El propósito de esta descripción es proporcionar compatibilidad binaria de las TDD entre las diferentes aplicaciones de vendedor que utilizan el esquema de codificación binaria.

Las descripciones suministradas tienen que adaptarse a la plataforma de soporte para que funcionen adecuadamente (véase también la Parte IV de esta Recomendación).

Se utilizan los siguientes tipos de datos C genéricos para distribuir las TDD. Estos tipos de datos deberán corresponder con sus contrapartes específicas de tipos de datos C que dependen de la plataforma (véase el Cuadro 124).

CUADRO 124/T.611

#### Tipos de datos C genéricos

Tipo de datos C genéricos	Comentario
INT16	Valor entero con signo, codificado en 16 bits
UINT32	Valor entero sin signo, codificado en 32 bits
CHAR	Carácter, codificado en 8 bits
BYTE	Octeto, codificado en 8 bits

#### 18.1 Codificación binaria de las TDD

La TDD de petición y las respuestas se representan por estructuras C. Los componentes de esas estructuras corresponden a los elementos de sintaxis utilizados en codificaciones basadas en texto, como se define en la cláusula 6 y en las cláusulas pertinentes de la Parte II de esta Recomendación.

Se ha elegido la disposición binaria, descrita con estructuras de lenguaje C, de modo que los elementos de diferentes tipos de datos estén alineados a una frontera de 4 bytes.

Cada TDD consta, en orden de:

- un encabezamiento binario de 4 octetos;
- un descriptor de TDD de 16 octetos, que contiene los desplazamientos a cuatro secciones específicas en la disposición TDD; los desplazamientos se calculan a partir de la base del encabezamiento binario.

El resto de la disposición binaria está formada por cuatro secciones específicas, las cuales pueden aparecer en cualquier orden:

- *Sección 1 de TDD* – Contiene los parámetros de entrada comunes, es decir los descritos en la cláusula 6.
- *Sección 2 de TDD* – Contiene los parámetros de entrada que dependen del servicio, es decir, los descritos en la sección pertinente de la Parte II de esta Recomendación.
- *Sección 3 de TDD* – Contiene los parámetros de salida comunes, es decir los que se describen en la cláusula 6.
- *Sección 4 de TDD* – Contiene los parámetros de salida que dependen del servicio, es decir, los descritos en la sección pertinente de la Parte II de esta Recomendación.

Debido a esta estructura es posible que las aplicaciones añadan o mejoren parámetros de una TDD con parámetros privados.

En la Figura 16 se muestra esta estructura.

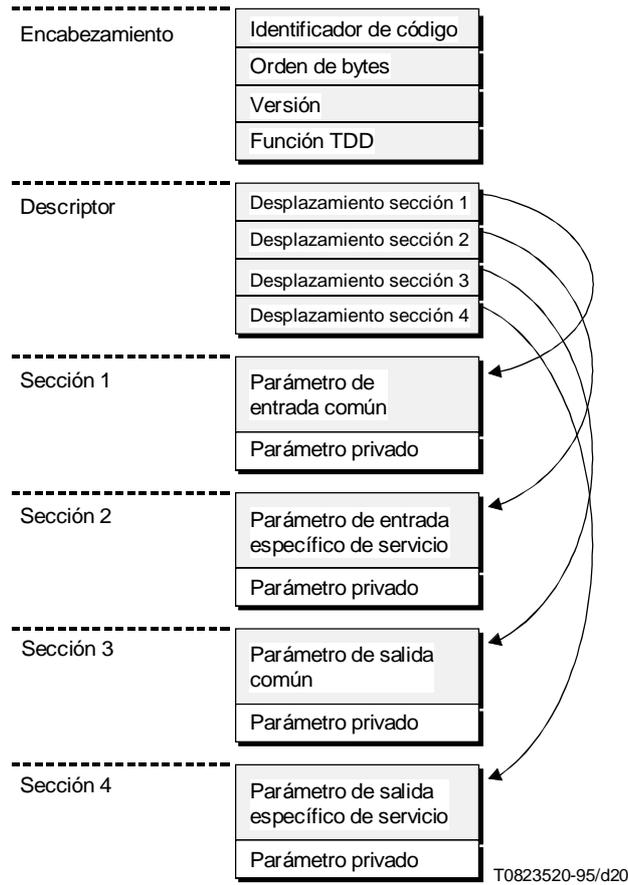


FIGURA 16/T.611  
Estructura binaria de TDD

### 18.1.1 Encabezamiento TDD

```
/**
 * <TDD> syntax element
 ***/
```

```
struct tdd {
    struct tdd_header    hdr;        /* <TDD Header> syntax element, see below */
    struct tdd_descriptor ofs;      /* Descriptor containing offsets, see below */
};                                  /* Function dependent TDD structures ... */
```

```
/**
 * <TDD Header> syntax element
 ***/
```

```
struct tdd_header {
    BYTE_code_id;          /* Code-ID, constant set to 0x43 ('C') */
    BYTE_byte_order;      /* Optional constant value indicating byte order */
    BYTE_itu_version;     /* Constant indicating version of this Recommendation */
    BYTE_function;        /* Function code */
};
```

```

/**
 *      Offset descriptor
 ***/

```

```

struct tdd_descriptor {
    UINT32 cinp;           /* offset to section 1 (common input parameter) */
    UINT32 sinp;           /* offset to section 2 (service dependent input parameter) */
    UINT32 coup;          /* offset to section 3 (common output parameter) */
    UINT32 soup;          /* offset to section 4 (service dependent output parameter) */
};

```

### 18.1.2 Estructuras TDD independientes del servicio

```

/**
 *      <SendTDD> structure
 ***/

```

```

struct send_tdd_inp {
    La_id_type           la_id;           /* common input parms */
    Req_id_type          req_id;          /* <Lald> */
    Service_type         service;         /* <ReqID> */
    Date_time_type       sendtime;        /* <Service> */
    Date_time_type       lasttime;        /* <SendTime> */
    Comment_type         comment;         /* <LastTime> */
    Userkey_type         userkey;         /* <Comment> */
};

```

```

/**
 *      <SendackTDD> structure
 ***/

```

```

struct sendack_tdd_inp {
    La_id_type           la_id;           /* common input parms */
    Req_id_type          req_id;          /* <Lald> */
    Service_type         service;         /* <ReqID> */
    Date_time_type       sendtime;        /* <Service> */
    Date_time_type       lasttime;        /* <SendTime> */
    Comment_type         comment;         /* <LastTime> */
    Userkey_type         userkey;         /* <Comment> */
};

```

```

struct sendack_tdd_out {
    Comid_type          comid;            /* common output parms */
    Status_type         status;           /* <ComId> */
    Error_type          error;            /* <Status> */
    Error_type          minor;            /* <Error> */
    Error_type          warning;          /* <Minor> */
};

```

```

/**
 *      <ReceiveTDD> structure
 ***/

```

```

struct receive_tdd_inp {
    La_id_type           la_id;           /* common input parms */
    Req_id_type          req_id;          /* <Lald> */
    Service_type         service;         /* <ReqID> */
    Boolean_type         deletedoc;       /* <Service> */
    Comid_type          comid;            /* <Delete > */
};

```

```

struct receive_tdd_out {
    Comid_type      comid;      /* common output parms */
    Status_type     status;     /* <ComId> */
    Error_type      error;      /* <Status> */
    Error_type      minor;      /* <Error> */
    Error_type      warning;    /* <Minor> */
    Service_type    service;    /* <Warning> */
    Date_time_type  rcvtime;    /* <Service> */
    };                          /* <ReceiveTime> */

```

```

/****
*      <CopyTDD> structure
****/

```

```

struct copy_tdd_inp {
    La_id_type      la_id;      /* common input parms */
    Req_id_type     req_id;     /* <Lald> */
    Comid_type      comid;     /* <ReqID> */
    Req_id_type     reqref;    /* <ComId> */
    State_type      state;     /* <ReqRef> */
    Path_type       target;    /* <State> */
    Layout_id_type  layout;    /* <Target> */
    };                          /* <Layout> */

```

```

struct copy_tdd_out {
    Error_type      error;      /* common output parms */
    Error_type      minor;     /* <Error> */
    Error_type      warning;   /* <Minor> */
    };                          /* <Warning> */

```

```

/****
*      <DeleteTDD> structure
****/

```

```

struct delete_tdd_inp {
    La_id_type      la_id;      /* common input parms */
    Req_id_type     req_id;     /* <Lald> */
    Comid_type      comid;     /* <ReqID> */
    Req_id_type     reqref;    /* <ComId> */
    };                          /* <ReqRef> */

```

```

struct delete_tdd_out {
    Error_type      error;      /* common output parms */
    Error_type      minor;     /* <Error> */
    Error_type      warning;   /* <Minor> */
    };                          /* <Warning> */

```

```

/****
*      <CancelTDD> structure
****/

```

```

struct cancel_tdd_inp {
    La_id_type      la_id;      /* common input parms */
    Req_id_type     req_id;     /* <Lald> */
    Comid_type      comid;     /* <ReqID> */
    Req_id_type     reqref;    /* <ComId> */
    };                          /* <ReqRef> */

```

```

struct cancel_tdd_out {
    Error_type    error;
    Error_type    minor;
    Error_type    warning;
};

```

```

/**
 * <PurgeTDD> structure
 */

```

```

struct purge_tdd_inp {
    La_id_type    la_id;
    Req_id_type   req_id;
    Comid_type    comid;
    Req_id_type   reqref;
    State_type    state;
};

```

```

struct purge_tdd_out {
    Error_type    error;
    Error_type    minor;
    Error_type    warning;
};

```

```

/**
 * <RescheduleTDD> structure
 */

```

```

struct reschedule_tdd_inp {
    La_id_type    la_id;
    Req_id_type   req_id;
    Comid_type    comid;
    Req_id_type   reqref;
    Address_type  address;
    Date_time_type sendtime;
    Date_time_type lasttime;
};

```

```

struct reschedule_tdd_out {
    Error_type    error;
    Error_type    minor;
    Error_type    warning;
};

```

```

/**
 * <DispatchTDD> structure
 */

```

```

struct dispatch_tdd_inp {
    La_id_type    la_id;
    Req_id_type   req_id;
    Comid_type    comid;
    La_id_type    newla;
};

```

```

struct dispatch_tdd_out {
    Error_type    error;
    Error_type    minor;
    Error_type    warning;
};

```

```

/**
 * <PreviewTDD> structure
 ***/

struct preview_tdd_inp {
    La_id_type      la_id;      /* common input parms */
    Req_id_type     req_id;     /* <Lald> */
    Comid_type      comid;     /* <ReqID> */
    Path_type       target;    /* <ComId> */
    Convert_id_type convert;   /* <Target> */
};

struct preview_tdd_out {
    Error_type      error;     /* common output parms */
    Error_type      minor;    /* <Error> */
    Error_type      warning;  /* <Minor> */
};

/**
 * <PrintTDD> structure
 ***/

struct print_tdd_inp {
    La_id_type      la_id;     /* common input parms */
    Req_id_type     req_id;    /* <Lald> */
    Path_type       filename;  /* <ReqID> */
    Convert_id_type informat;  /* <FileName> */
    Printer_id_type printer;   /* <InFormat> */
};

struct print_tdd_out {
    Error_type      error;     /* common output parms */
    Error_type      minor;    /* <Error> */
    Error_type      warning;  /* <Minor> */
};

/**
 * <ConvertTDD> structure
 ***/

struct convert_tdd_inp {
    La_id_type      la_id;     /* common input parms */
    Req_id_type     req_id;    /* <Lald> */
    Path_type       filename;  /* <ReqID> */
    Path_type       target;    /* <FileName> */
    Convert_id_type informat;  /* <Target> */
    Convert_id_type outformat; /* <InFormat> */
};

struct convert_tdd_out {
    Error_type      error;     /* common output parms */
    Error_type      minor;    /* <Error> */
    Error_type      warning;  /* <Minor> */
};

/**
 * <CheckTDD> structure
 ***/

```

```

struct check_tdd_inp {
    La_id_type      la_id;
    Req_id_type     req_id;
    Path_type       filename;
    Convert_id_type check;
};

/* common input parms */
/* <Lald> */
/* <ReqID> */
/* <FileName> */
/* <Check> */

struct check_tdd_out {
    Error_type      error;
    Error_type      minor;
    Error_type      warning;
};

/* common output parms */
/* <Error> */
/* <Minor> */
/* <Warning> */

/**
 * <ExtendTDD> structure
 ***/

struct extend_tdd_inp {
    Subfunc_type    subfunc;
    La_id_type      la_id;
    Req_id_type     req_id;
};

/* common input parms */
/* <SubFunction> */
/* <Lald> */
/* <ReqID> */

struct extend_tdd_out {
    Error_type      error;
    Error_type      minor;
    Error_type      warning;
};

/* common output parms */
/* <Error> */
/* <Minor> */
/* <Warning> */

/**
 * <NationalTDD> structure
 ***/

struct national_tdd_inp {
    subfunc;
    La_id_type      la_id;
    Req_id_type     req_id;
    /*      ...
    */
};

/* common input parms */
/* Subfunc_type
   /* <NationalFunction>, for further study */
/* <Lald> */
/* <ReqID> */
/* <--- others are added here, for further study */

struct national_c_out {
    Error_type      error;
    Error_type      minor;
    Error_type      warning;
    /*      ...
    */
};

/* common output parms */
/* <Error> */
/* <Minor> */
/* <Warning> */
/* <--- others are added here, for further study */

/**
 * <PrivateTDD> structure
 ***/

struct private_tdd_inp {
    Subfunc_type    subfunc;
    La_id_type      la_id;
    Req_id_type     req_id;
    /*      ...
    */
};

/* common input parms */
/* <PrivateFunction>, for further study */
/* <Lald> */
/* <ReqID> */
/* <--- others are added here, for further study */

```

```

struct private_tdd_out {
    Error_type      error;      /* common output parms */
    Error_type      minor;     /* <Error> */
    Error_type      warning;   /* <Minor> */
    /* ...           /* <Warning> */
    /* --- others are added here, for further study */
};

```

### 18.1.3 Constantes independientes del servicio y definiciones de tipo

```

/**
 * Constants used for fields of struct tdd_header
 */

/* code_id field */
#define CODE_ID          0x43          /* Code-ID, set to 'C' */

/* byte_order field */
#define L_BYTEORDER      0x4C          /* Little Endian (Intel, low-high) byte order */
#define B_BYTEORDER      0x42          /* Big endian byte order */

/* itu_version field */
#define ITU_VERSION      94           /* Current version (1994) */

/* function field */
#define SENDACK_TDD      0x10          /* <SendAckFunction> */
#define SEND_TDD         0x11          /* <SendFunction> */
#define RECEIVE_TDD     0x20          /* <ReceiveFunction> */
#define COPY_TDD         0x30          /* <CopyFunction> */
#define DELETE_TDD      0x31          /* <DeleteFunction> */
#define CANCEL_TDD      0x32          /* <CancelFunction> */
#define PURGE_TDD       0x33          /* <PurgeFunction> */
#define RESCHEDULE_TDD  0x34          /* <RescheduleFunction> */
#define DISPATCH_TDD    0x35          /* <DispatchFunction> */
#define PREVIEW_TDD     0x36          /* <PreviewFunction> */
#define PRINT_TDD       0x40          /* <PrintFunction> */
#define CONVERT_TDD     0x41          /* <ConvertFunction> */
#define CHECK_TDD       0x42          /* <CheckFunction> */
#define EXTEND_TDD      0x50          /* <ExtendFunction> */
#define NATIONAL_TDD    0x60          /* <NationalFunction> */
#define PRIVATE_TDD     0x70          /* <PrivateFunction> */

/**
 * Constants used for struct extend_tdd_inp
 */

/* subfunction field */
#define POLL_FX3         0x0101       /* FX3: Poll */

#define ENCODE_IPM      0x2010        /* EMAIL: EncodeIPM */
#define DECODE_IPM      0x2011        /* EMAIL: DecodeIPM */

/**
 * Type definitions
 * Some types are service dependent. Their values might be restricted or expanded upon further.
 */

```

```

/*
**      Address_type
**      Some telecommunication services may constrain or expand this type furthermore.
*/

typedef CHAR Address_type[127+1];

/*
**      Boolean_type
**      Takes the values true or false
*/

typedef enum { false=0, true } Boolean_type;

/*
**      Comid type
*/

typedef CHAR Comid_type[31+1];

/*
**      Comment type
*/

typedef CHAR Comment_type[127+1];

/*
**      Convert_id_type
**      The (asciiz) strings assigned for text based encoding shall be used.
*/

typedef CHAR Convert_id_type[15+1];

/*
**      Date_time_type and Send_time_type
**      The date and time specification follows closely the ANSI specification
**      The values "IMMEDIATE" and "URGENT" are set by specifying the seconds
**      part of the struct below (tm_sec) with (-1) or (-2)
*/

typedef struct {
    INT16          tm_sec;          /* seconds; used for "IMMEDIATE" and "URGENT"
    specification */
    INT16          tm_min;          /* minutes */
    INT16          tm_hour;         /* hours */
    INT16          tm_mday;         /* day of month */
    INT16          tm_mon;          /* month */
    INT16          tm_year;         /* full year */
} Date_time_type;

#define Send_time_type      Date_time_type
#define IMMEDIATE          (-1)      /* Set into tm_sec of above struct */
#define URGENT              (-2)

/*
**      Error_type
**      The definition of the Error type does not include the text part of the error message; If the text
**      part is needed, it shall be treated as a private extension!
*/

typedef UINT32 Error_type;

```

```

/*
**      La_id_type
**      Reference of an LA-ID. Presented as a string.
**      The purpose of the parameter is to identify the "owning" LA of a request.
*/

typedef CHAR La_id_type[15+1];

/*
**      Layout_id_type
**      Defines the layout of the <CopyTDD> target file.
*/

typedef enum {std=0, csv, tab } Layout_id_type;

/*
**      Path_type
**      Full path addressing a file of a directory. Stands for full path to document, file or directory.
**      Full path means: path given absolute, without relative components.
*/

typedef CHAR Path_type[255+1];

/*
**      Printer_id_type
**      ID of selected printer. Represented as a string. Depends on the supporting operating system.
**      The CA manufacturer shall state in documentation how to address printers.
*/

typedef CHAR Printer_id_type[127+1];

/*
**      Req_id_type
**      Reference of a Request ID. The parameter value is represented as a string encoded as implied
**      by the APPLI/COM header ID of the TDD. The purpose of the parameter is to identify the
**      relation of a response to a previous request. So the REQ-ID shall be unique within an LA. It is the
**      responsibility of the LA to ensure the REQ-ID is unique.
*/

typedef CHAR Req_id_type[31+1];

/*
**      Service_type
**      Specifies the ITU-T Service to be used.
*/

typedef enum { fx3=1, fx4, ttx, tlx, tx, email, ft } Service_type;

/*
**      State_type
**      Specifies the state of the CA-Record. See also 5.6.
*/

typedef enum { delayed=1, sending, sent, send_failed, reception, retrieved, receive_failed } State_type;

/*
**      Status_type
*/

typedef enum { unknown, positive, partial, negative } Status_type;

```

```

/*
**      Subfunc_type
*/

```

```
typedef UINT32 Subfunc_type;
```

```

/*
**      Type_id_type
**      Specifies the Type (Subtype) of a telecommunications service. Each telecommunications
**      service has its own set of Type-ids. See appropriate clause of Part II of this Recommendation.
*/

```

```
typedef enum{ std=0, btm, dtm, bft, edi, opd, md, ctl, teletex, g3fax, g4class1, videotex, message, bilateral,
national, oda } Type_id_type;
```

```

/*
**      Userkey_type
*/

```

```
typedef CHAR_Userkey_type[31+1];
```

#### 18.1.4 Estructuras de TDD dependientes de FX3, constantes y definiciones

```

/****
*      FX3: Service specific structure for <SendTDD>
****/

```

```

struct send_fx3_inp {
    G3_speed_type           g3speed;           /* <G3Speed> */
    Boolean_type            gencil;            /* <GenCil> */
    G3_high_res_type       highres;           /* <HighRes> */
    G3_sub_address_type    subaddr;           /* <SubAddress> */
    Boolean_type            useecm;           /* <UseEcm> */
    Address_type            address;           /* <Recipient> or <RecipientSpec> */
    Boolean_type            dopoll;           /* <DoPoll> */
    Poll_password_type     password;          /* <PollPassword> */
    Poll_select_type       select;            /* <PollSelector> */
    Path_type              filename;          /* <Document> or <DocumentSpec> */
    Convert_id_type        convert;           /* <Convert> */
    Type_id_type            type;             /* <Type> */
    INT16                  from;             /* <From> */
    INT16                  to;               /* <To> */
};

```

```

/****
*      FX3: Service specific structures for <SendAckTDD>
****/

```

```

struct sendack_fx3_inp {
    G3_speed_type           g3speed;           /* <G3Speed> */
    Boolean_type            gencil;            /* <GenCil> */
    G3_high_res_type       highres;           /* <HighRes> */
    G3_sub_address_type    subaddr;           /* <SubAddress> */
    Boolean_type            useecm;           /* <UseEcm> */
    Address_type            address;           /* <Recipient> */
    Boolean_type            dopoll;           /* <DoPoll> */
    Poll_password_type     password;          /* <PollPassword> */
    Poll_select_type       select;            /* <PollSelector> */
    Path_type              filename;          /* <Document> or <DocumentSpec> */
};

```

```

Convert_id_type      convert;      /* <Convert> */
Type_id_type        type;          /* <Type> */
INT16               from;         /* <From> */
INT16               to;          /* <To> */
};

struct sendack_fx3_out {
    Address_type      address;      /* <Recipient> */
    G3_speed_type     g3speed;      /* <G3Speed> */
    Boolean_type       useecm;      /* <UseEcm> */
};

/**
 *   FX3: Service specific structures for <ReceiveTDD>
 ***/

struct receive_fx3_inp {
    Convert_id_type   cvfax3;       /* <CvFax3> */
    G3_sub_address_type subaddr;    /* <SubAddress> */
    Path_type         filename;     /* <Document> */
};

struct receive_fx3_out {
    Address_type      address;      /* <Originator> */
    G3_speed_type     g3speed;      /* <G3Speed> */
    G3_sub_address_type subaddr;    /* <SubAddress> */
    Path_type         filename;     /* <Document> or <DocumentSpec> */
    Convert_id_type   convert;      /* <Convert> */
    Type_id_type      type;         /* <Type> */
};

/**
 *   FX3: Structures for poll <ExtendTDD>
 ***/

struct poll_fx3_inp {
    Address_type      address;      /* <Address> */
    Poll_password_type password;    /* <PollPassword> */
    Poll_select_type  select;       /* <PollSelector> */
    Date_time_type    sendtime;     /* <SendTime> (optional) */
};

struct poll_fx3_out {
    Comid_type        comid;        /* <ComId> */
};

/**
 *   Constants to be used for specific fields of the Send, Sendack and Receive TDD.
 *   Use of some definitions made in 18.1.3 are restricted. This applies to the following
 *   definitions:
 *
 *   Service_type is restricted to:    fx3.
 *   Type_id_type is restricted to:    std, btm, dtm, bft and edi.
 *   Convert_id_type:                  See Table 78 for further information.
 ***/

/**
 **
 *   G3_speed_type
 */
typedef enum {bps2400=1, bps4800, bps7200, bps9600, bps12200, bps14400} G3_speed_type;

```

```

/*
**      G3_sub_address_type
*/

typedef CHAR G3_sub_address_type[23+1];

/*
**      G3_high_res_type
*/

typedef enum {g3dpi98=0, g3dpi196, g3dpi200, g3dpi300, g3dpi400, g3dpi392_8, g3dpi392_16 }
G3_high_res_type;

/*
**      Poll_password_type
*/

typedef CHAR Poll_password_type[23+1]; /* only first 20 digits are valid */

/*
**      Poll_select_type
*/

typedef CHAR Poll_password_type[23+1]; /* only first 20 digits are valid */

```

#### 18.1.5 Estructuras de TDD dependientes de FX4, constantes y definiciones

```

/**
 *      FX4: Service specific structure for <SendTDD>
 ***/

struct send_fx4_inp {
    G4_high_res_type      highres;      /* <HighRes> */
    G4_sub_address_type  subaddr;      /* <SubAddress> */
    Address_type         address;      /* <Recipient> or <RecipientSpec> */
    Path_type           filename;     /* <Document> or <DocumentSpec> */
    Convert_id_type     convert;      /* <Convert> */
    Type_id_type        type;         /* <Type> */
    String12_type       name;         /* <Name> */
    String12_type       userinfo;     /* <UserInfo> */
    Path_type           prolog;       /* <Prolog> */
    INT16               from;         /* <From> */
    INT16               to;           /* <To> */
};

/**
 *      FX4: Service specific structures for <SendAckTDD>
 ***/

struct sendack_fx4_inp {
    G4_high_res_type      highres;      /* <HighRes> */
    G4_sub_address_type  subaddr;      /* <SubAddress> */
    Address_type         address;      /* <Recipient> */
    Path_type           filename;     /* <Document> or <DocumentSpec> */
    Convert_id_type     convert;      /* <Convert> */
    Type_id_type        type;         /* <Type> */
    String12_type       name;         /* <Name> */
    String12_type       userinfo;     /* <UserInfo> */
};

```

```

    Path_type          prolog;          /* <Prolog> */
    INT16              from;           /* <From> */
    INT16              to;            /* <To> */
};

struct sendack_fx4_out {
    Cil_type          cil;          /* <Cil> */
};

/**
 *   FX4: Service specific structures for <ReceiveTDD>
 ***/

struct receive_fx4_inp {
    Convert_id_type   cvfax4;        /* <CvFax4> */
    G4_sub_address_type subaddr;     /* <SubAddress> */
    Path_type         filename;      /* <Document> */
    Path_type         prolog;        /* <Prolog> */
};

struct receive_fx4_out {
    Address_type      address;        /* <Originator> */
    G4_sub_address_type subaddr;     /* <SubAddress> */
    Path_type         filename;      /* <Document> or <DocumentSpec> */
    Convert_id_type   convert;       /* <Convert> */
    Type_id_type      type;          /* <Type> */
    String12_type     name;          /* <Name> */
    String12_type     userinfo;     /* <UserInfo> */
    Cil_type          cil;          /* <Cil> */
    INT16             firstpg;       /* <FirstPg> */
};

/**
 *   Constants to be used for specific fields of the Send, Sendack and Receive TDD.
 *   Use of some definitions made in 18.1.3 are restricted. This applies to the following
 *   definitions:
 *
 *       Service_type is restricted to:   fx4.
 *       Type_id_type is restricted to:   std, dtm, bft, edi, opd, md, and ctl.
 *       Convert_id_type:                 See Table 87 for further information.
 ***/

/*
 **   String12_type
 */

typedef CHAR String12_type[15+1];

/*
 **   G4_sub_address_type
 */

typedef CHAR G4_sub_address_type[7+1];

/*
 **   G4_high_res_type
 */

typedef enum {g4dpi200=1, g4dpi240, g4dpi300, g4dpi400} G4_high_res_type;

```

```

/*
**      Cil_type
*/

typedef struct {
    CHAR  receiver_tid[24];
    CHAR  sep1;           /* = '/' (slash) */
    CHAR  sender_tid[24];
    CHAR  sep2;           /* = '/' (slash) */
    CHAR  datetime[14];
    CHAR  sep3;           /* = '/' (slash) */
    CHAR  refinfo[7];
} Cil_type;             /* There is no terminating 0x00; Total length = 72 */

```

### 18.1.6 Estructuras de TDD dependientes de TTX, constantes y definiciones

```

/**
 *      TTX: Service specific structure for <SendTDD>
 ***/

struct send_ttx_inp {
    Ttx_sub_address_type  subaddr;      /* <SubAddress> */
    Address_type          address;      /* <Recipient> or <RecipientSpec> */
    Path_type             filename;     /* <Document> or <DocumentSpec> */
    Convert_id_type       convert;      /* <Convert> */
    Type_id_type          type;         /* <Type> */
    String12_type         name;         /* <Name> */
    String12_type         userinfo;     /* <UserInfo> */
    Path_type             prolog;       /* <Prolog> */
    INT16                 from;        /* <From> */
    INT16                 to;          /* <To> */
    T61_options_type     t61options;    /* <T61Options> */
};

/**
 *      TTX: Service specific structures for <SendAckTDD>
 ***/

struct sendack_ttx_inp {
    Ttx_sub_address_type  subaddr;      /* <SubAddress> */
    Address_type          address;      /* <Recipient> */
    Path_type             filename;     /* <Document> or <DocumentSpec> */
    Convert_id_type       convert;      /* <Convert> */
    Type_id_type          type;         /* <Type> */
    String12_type         name;         /* <Name> */
    String12_type         userinfo;     /* <UserInfo> */
    Path_type             prolog;       /* <Prolog> */
    INT16                 from;        /* <From> */
    INT16                 to;          /* <To> */
    T61_options_type     t61options;    /* <T61Options> */
};

struct sendack_ttx_out {
    Cil_type              cil;          /* <Cil> */
};

```

```

/****
*      TTX: Service specific structures for <ReceiveTDD>
****/

struct receive_ttx_inp {
    Convert_id_type      cvttx;          /* <CvTtx> */
    Ttx_sub_address_type subaddr;       /* <SubAddress> */
    Path_type           filename;       /* <Document> */
    Path_type           prolog;         /* <Prolog> */
};

struct receive_ttx_out {
    Address_type        address;        /* <Originator> */
    Ttx_sub_address_type subaddr;       /* <SubAddress> */
    Path_type           filename;       /* <Document> or <DocumentSpec> */
    Convert_id_type     convert;        /* <Convert> */
    Type_id_type        type;           /* <Type> */
    String12_type       name;           /* <Name> */
    String12_type       userinfo;      /* <UserInfo> */
    Cil_type            cil;            /* <Cil> */
    INT16               firstpg;       /* <FirstPg> */
};

/****
*      Constants to be used for specific fields of the Send, Sendack and Receive TDD.
*      Use of some definitions made in 18.1.3 are restricted. This applies to the following
*      definitions:
*
*      Service_type is restricted to:      ttx.
*      Type_id_type is restricted to:     std, dtm, bft and edi.
*      Convert_id_type:                   See Table 94 for further information.
****/

/*
**      String12_type
**      See definition of FX4!
*/

/*
typedef CHAR String12_type[15+1];
*/

/*
**      Ttx_sub_address_type
*/

typedef CHAR Ttx_sub_address_type[7+1];

/*
**      Cil_type
**      See definition of FX4!
*/

/*
typedef struct {
    CHAR  receiver_tid[24];
    CHAR  sep1          /* = '/' (slash) */
    CHAR  sender_tid[24];
    CHAR  sep2          /* = '/' (slash) */
    CHAR  datetime[14];
    CHAR  sep3          /* = '/' (slash) */
    CHAR  refinfo[7];
} Cil_type;
*/

```

```

/*
**      T61_options_type
*/

```

```
typedef CHAR T61_options_type[15+1];
```

### 18.1.7 Estructuras TDD dependientes de TLX, constantes y definiciones

```

/****
*      TX: Service specific structure for <SendTDD>
****/

```

```

struct send_tx_inp {
    Boolean_type          notify;          /* <Notify> */
    Tx_sub_address_type  subaddr;         /* <SubAddress> */
    Address_type         address;         /* <Recipient> or <RecipientSpec> */
    Path_type           filename;         /* <Document> or <DocumentSpec> */
    Convert_id_type     convert;          /* <Convert> */
    Type_id_type        type;             /* <Type> */
    INT16               from;             /* <From> */
    INT16               to;               /* <To> */
};

```

```

/****
*      TX: Service specific structures for <SendAckTDD>
****/

```

```

struct sendack_tx_inp {
    Boolean_type          notify;          /* <Notify> */
    Tx_sub_address_type  subaddr;         /* <SubAddress> */
    Address_type         address;         /* <Recipient> */
    Path_type           filename;         /* <Document> or <DocumentSpec> */
    Convert_id_type     convert;          /* <Convert> */
    Type_id_type        type;             /* <Type> */
    INT16               from;             /* <From> */
    INT16               to;               /* <To> */
};

```

```
struct sendack_tx_out {};
```

```

/****
*      TX: Service specific structures for <ReceiveTDD>
****/

```

```

struct receive_tx_inp {
    Convert_id_type     cvtx;             /* <CvTx> */
    Tx_sub_address_type  subaddr;         /* <SubAddress> */
    Path_type           filename;         /* <Document> */
};

```

```

struct receive_tx_out {
    Address_type        address;          /* <Originator> */
    Tx_sub_address_type  subaddr;         /* <SubAddress> */
    Path_type           filename;         /* <Document> or <DocumentSpec> */
    Convert_id_type     convert;          /* <Convert> */
    Type_id_type        type;             /* <Type> */
};

```

```

/****
*
* Constants to be used for specific fields of the Send, Sendack and Receive TDD.
* Use of some definitions made in 18.1.3 are restricted. This applies to the following
* definitions:
*
* Service_type is restricted to: tx.
* Type_id_type is restricted to: std
* Convert_id_type is restricted to: "ASCII", "ASCIIxxx", "T.50"
****/

```

```

/*
** Tx_sub_address_type
*/

```

```
typedef CHAR Tx_sub_address_type[7+1];
```

### 18.1.8 Estructuras TDD dependientes de TLX, constantes y definiciones

```

/****
* TLX: Service specific structure for <SendTDD>
****/

```

```

struct send_tlx_inp {
    Tx_sub_address_type    subaddr;    /* <SubAddress> */
    Address_type           address;    /* <Recipient> or <RecipientSpec> */
    Path_type              filename;   /* <Document> or <DocumentSpec> */
    Convert_id_type        convert;    /* <Convert> */
    Type_id_type           type;       /* <Type> */
    INT16                  from;       /* <From> */
    INT16                  to;         /* <To> */
};

```

```

/****
* TLX: Service specific structures for <SendAckTDD>
****/

```

```

struct sendack_tlx_inp {
    Tx_sub_address_type    subaddr;    /* <SubAddress> */
    Address_type           address;    /* <Recipient> */
    Path_type              filename;   /* <Document> or <DocumentSpec> */
    Convert_id_type        convert;    /* <Convert> */
    Type_id_type           type;       /* <Type> */
    INT16                  from;       /* <From> */
    INT16                  to;         /* <To> */
};

```

```
struct sendack_tlx_out {};
```

```

/****
* TLX: Service specific structures for <ReceiveTDD>
****/

```

```

struct receive_tlx_inp {
    Convert_id_type        cvtlx;      /* <CvTx> */
    Tx_sub_address_type    subaddr;    /* <SubAddress> */
    Path_type              filename;   /* <Document> */
};

```

```

struct receive_tl原因_out {
    Address_type      address;      /* <Originator> */
    Tlx_sub_address_type subaddr;    /* <SubAddress> */
    Path_type         filename;     /* <Document> or <DocumentSpec> */
    Convert_id_type   convert;      /* <Convert> */
    Type_id_type      type;         /* <Type> */
};

```

```

/****
*
* Constants to be used for specific fields of the Send, Sendack and Receive TDD.
* Use of some definitions made in 18.1.3 are restricted. This applies to the following
* definitions:

```

```

*
*     Service_type is restricted to:    tlx.
*     Type_id_type is restricted to:   std
*     Convert_id_type is restricted to: "ASCII", "ASCIIxx", "T.50"
****/

```

```

/*
** Tlx_sub_address_type
*/

```

```

typedef CHAR Tlx_sub_address_type[7+1];

```

### 18.1.9 Estructuras TDD dependientes de EMAIL, constantes y definiciones

```

/****
*
* EMAIL: Service specific structure for <SendTDD>
****/

```

```

struct send_email_inp {
    Ipm_id_type      ipm_id;        /* <IpmId> */
    Path_type        s_recipient;   /* <S-Recipient> */
    Boolean_type     alternate;     /* <Alternate> (optional) */
    Cont_type_type   cont_type;     /* <ContType> (optional) */
    Boolean_type     disclo_rec;    /* <DiscloRec> (optional) */
    Date_time_type   expirytime;   /* <ExpiryTime> (optional) */
    Boolean_type     implicit_conv; /* <ImplicitConv> (optional) */
    Importance_type  importance;    /* <Importance> (optional) */
    Language_id_type language;     /* <Language> (optional) */
    Priority_type     priority;     /* <Priority> (optional) */
    Path_type        related;      /* <Related> (optional) */
    Ipm_id_type      reply_id;     /* <ReplyId> (optional) */
    Date_time_type   replytime;    /* <ReplyTime> (optional) */
    Path_type        s_originator; /* <S-Originator> (optional) */
    Sensitivity_type sensitivity;  /* <Sensitivity> (optional) */
    Subject_type     subject;      /* <Subject> (optional) */
    Userinfo_type    userinfo;     /* <UserInfo> (optional) */
};

```

```

/****
*
* EMAIL:Service specific structures for <SendAckTDD>
****/

```

```

struct sendack_email_inp {
    Ipm_id_type      ipm_id;        /* <IpmId> */
    Path_type        s_recipient;   /* <S-Recipient> */
    Boolean_type     alternate;     /* <Alternate> (optional) */
    Cont_type_type   cont_type;     /* <ContType> (optional) */
};

```

```

Boolean_type      disclo_rec;      /* <DiscloRec> (optional) */
Date_time_type   expirytime;    /* <ExpiryTime> (optional) */
Boolean_type     implicit_conv; /* <ImplicitConv> (optional) */
Importance_type  importance;    /* <Importance> (optional) */
Language_id_type language;      /* <Language> (optional) */
Priority_type     priority;      /* <Priority> (optional) */
Path_type        related;       /* <Related> (optional) */
Ipm_id_type      reply_id;      /* <ReplyId> (optional) */
Date_time_type   replytime;     /* <ReplyTime> (optional) */
Path_type        s_originator;  /* <S-Originator> (optional) */
Sensitivity_type sensitivity;   /* <Sensitivity> (optional) */
Subject_type     subject;       /* <Subject> (optional) */
Userinfo_type    userinfo;      /* <UserInfo> (optional) */
};

```

```

struct sendack_email_out {
    Msg_sub_id_type  msg_sub_id; /* <MsgSubId> (optional) */
    Date_time_type  submittime;  /* <SubmitTime> (optional) */
};

```

```

/****
*      EMAIL:Service specific structures for <ReceiveTDD>
****/

```

```

struct receive_email_inp {
    Path_type        r_originator; /* <R-Originator> */
    Path_type        r_recipient;  /* <R-Recipient> */
    Path_type        related;      /* <Related> (optional) */
};

```

```

struct receive_email_out {
    Ipm_id_type      ipm_id;       /* <IpmId> */
    Cont_type_type   cont_type;    /* <ContType> */
    Date_time_type   expirytime;   /* <ExpiryTime> */
    Boolean_type     forwarded;    /* <Forwarded> */
    Importance_type  importance;   /* <Importance> */
    Boolean_type     inc_copy;     /* <IncCopy> */
    Language_id_type language;     /* <Language> */
    Priority_type     priority;     /* <Priority> */
    Ipm_id_type      reply_id;     /* <ReplyId> */
    Date_time_type   replytime;    /* <ReplyTime> */
    Sensitivity_type sensitivity;  /* <Sensitivity> */
    Subject_type     subject;      /* <Subject> */
    Date_time_type   submittime;   /* <SubmitTime> */
    Userinfo_type    userinfo;     /* <UserInfo> */
};

```

```

/****
*      EMAIL: Structure of file-record of R-Recipientspec
****/

```

```

struct r_recipientspec {
    X_name_type      x_name;
    R_recipient_type r_type;
    Boolean_type     reply;
    Notify_type     notify;
    Report_type     report;
};

```

```
/**
 * EMAIL: Structure of file-record of S-Recipientspec
 ***/
```

```
struct s_recipientspec {
    X_name_type          x_name;
    S_recipient_type     s_type;
    Boolean_type         reply;
    Notify_type          notify;
    Report_type          report;
};
```

```
/**
 * EMAIL: Structure of file-record of Originatorspec
 ***/
```

```
struct originatorspec {
    X_name_type          x_name;
    Originator_type     o_type;
};
```

```
/**
 * EMAIL: Structure of file-record of Relatedspec
 ***/
```

```
struct relatedspec {
    lpm_id_type          ipm_id;
    Relation_type        relation;
};
```

```
/**
 * Constants to be used for specific fields of the Send, Sendack and Receive TDD.
 * Use of some definitions made in 18.1.3 are restricted. This applies to the following
 * definitions:
 *
 * Service_type is restricted to:    email.
 * Type_id_type is restricted to:   std, teletex, g3fax, g4class1, videotex, message,
 *                                  bilateral, national
 * Convert_id_type:                 See Table 113 for further information.
 ***/
```

```
/*
 ** Cont_type_type
 */
```

```
typedef enum { mhsIPM84=0, mhsIPM88, mhsIPN84, mhsIPN88, mhsDR, cmcIPM, cmcIPN, cmcDR }
Cont_type_type;
```

```
/*
** Importance_type
*/
```

```
typedef enum { low = 0, normal, high } Importance_type;
```

```
/*
** lpm_id_type
*/
```

```
typedef struct {
    BYTE id[63+1];
    BYTE oraddress[511+1];
} lpm_id_type;
```

```
/*  
**      Language_id_type  
*/
```

```
typedef CHAR Language_id_type[5+1];
```

```
/*  
**      Msg_sub_id_type  
*/
```

```
typedef CHAR Msg_sub_id_type[67+1];
```

```
/*  
**      Priority_type  
*/
```

```
typedef enum { standard=0, nonurgent, urgent } Priority_type;
```

```
/*  
**      Relation_type  
*/
```

```
typedef enum { reference=0, obsolete } Relation_type;
```

```
/*  
**      Subject_type  
*/
```

```
typedef CHAR Subject_type[127+1];
```

```
/*  
**      Sensitivity_type  
*/
```

```
typedef enum { none=0, personal, privateonly, companyconfidential } Sensitivity_type;
```

```
/*  
**      Userinfo_type  
*/
```

```
typedef CHAR Userinfo_type[15+1];
```

```
/*  
**      X_name_type  
*/
```

```
typedef struct {  
    Boolean_type      is_alias;  
    CHAR              addr_or_alias[511+1];  
} X_name_type;
```

```
/*  
**      S_recipient_type  
*/
```

```
typedef enum { sndprimary=1, sndcopy, sndblind } S_recipient_type;
```

```

/*
**      R_recipient_type
*/

typedef enum { recprimary=1, reccopy, recblind, recintended } R_recipient_type;

```

```

/*
**      Notify_type
*/

typedef enum { nonotify=1, notreceived, received, explicit } Notify_type;

```

```

/*
**      Report_type
*/

typedef enum { noreport=1, basic, confirmed } Report_type;

```

```

/*
**      Originator_type
*/

typedef enum { authorizing=1, originator, replyto } Originator_type;

```

#### 18.1.10 Estructuras TDD dependientes de FT, constantes y definiciones

```

/**
 *      FT: Service specific structure for <SendTDD>
 */

struct send_ft_inp {
    Environ_type      environ;      /* <Environ> */
    Password_type     password;     /* <Password> (optional) */
    Address_type      address;      /* <Recipient> or <RecipientSpec> */
    Path_type         filename;     /* <Document> or <DocumentSpec> */
    Convert_id_type   convert;      /* <Convert> */
    Type_id_type      type;         /* <Type> */
    Compress_type     compress;     /* <Compress> */
    Name_type         name;         /* <Name> (optional) */
};

```

```

/**
 *      FT: Service specific structures for <SendAckTDD>
 */

struct sendack_ft_inp {
    Environ_type      environ;      /* <Environ> */
    Password_type     password;     /* <Password> (optional) */
    Address_type      address;      /* <Recipient> */
    Path_type         filename;     /* <Document> or <DocumentSpec> */
    Convert_id_type   convert;      /* <Convert> */
    Type_id_type      type;         /* <Type> */
    Compress_type     compress;     /* <Compress> */
    Name_type         name;         /* <Name> (optional) */
};

```

```

struct sendack_ft_out {};

```

```

/****
* FT: Service specific structures for <ReceiveTDD>
****/

```

```

struct receive_ft_inp {
    Password_type      password;    /* <Password> (optional) */
    Path_type          filename;    /* <Document> (optional) */
};

```

```

struct receive_ft_out {
    Environ_type       environ;     /* <Environ> */
    Address_type       address;     /* <Originator> */
    Path_type          filename;    /* <Document> or <DocumentSpec> */
    Convert_id_type    convert;     /* <Convert> */
    Type_id_type       type;        /* <Type> */
    Compress_type      compress;    /* <Compress> */
    Name_type          name;        /* <Name> */
};

```

```

/****
* Constants to be used for specific fields of the Send, Sendack and Receive TDD.
* Use of some definitions made in 18.1.3 are restricted. This applies to the following
* definitions:
*
* Service_type is restricted to: ft
* Type_id_type is restricted to: std
* Convert_id_type: std, void.
****/

```

```

/*
** Compress_type
* See also 17.2.2.6
*/

```

```

typedef CHAR Compress_type[7+1];

```

```

/*
** Environ_type
*/

```

```

typedef enum { msdos=1, windows, unix, os2, macos } Environ_type;

```

```

/*
** Password_type
*/

```

```

typedef CHAR Password_type[15+1];

```

```

/*
** Name_type
*/

```

```

typedef CHAR Name_type[15+1];

```

## PARTE IV – DEPENDENCIA DE PLATAFORMA

### 19 Aspectos que dependen de la realización

Esta cláusula describe los aspectos que dependen de la realización para diversas plataformas.

El único punto de esta Recomendación que depende de la plataforma es la aplicación del método de intercambio de primitivas (véase 7.3.1). El método de intercambio de primitivas suministra el acceso directo a las funciones del método de intercambio básico descrito en 7.1. La subcláusula 19.3 describe la aplicación del método de intercambio de primitivas para cada plataforma.

La aplicación del método de intercambio de primitivas es facultativa para algunas plataformas, pues se podría elegir el método de intercambio de ficheros más independiente de plataforma como método de intercambio por defecto para una plataforma específica. En la subcláusula 19.2 se enumeran las asignaciones por defecto para las diferentes plataformas.

El esquema de codificación binaria suministrado en la Parte III de esta Recomendación *no* depende de la plataforma. Sin embargo, dado que para la descripción de esta codificación se ha utilizado un lenguaje C genérico, se da primero en 19.1 la correspondencia de los tipos de datos C genéricos con los tipos de datos reales utilizados para una plataforma específica.

#### 19.1 Correspondencia de tipos de datos TDD codificados en binario

Véase el Cuadro 125.

CUADRO 125/T.611

##### Correspondencia de tipos de datos C genéricos para las diversas plataformas

Plataforma	Tipo C real que corresponde al tipo de datos C genérico			
	INT16	UINT32	CHAR	BYTE
MS-DOS	short int	unsigned long int	char	unsigned char
WINDOWS	short int	dword	char	byte
Unix	short int	unsigned long int	char	unsigned char
OS/2	short int	unsigned long int	char	unsigned char
MacOS	En estudio			

#### 19.2 Método de intercambio por defecto

Véase el Cuadro 126.

CUADRO 126/T.611

##### Asignaciones del método de intercambio por defecto para las diversas plataformas

Plataforma	Método de intercambio por defecto
MS-DOS	Método de intercambio de ficheros
WINDOWS	Método de intercambio de primitivas
Unix	En estudio
OS/2	En estudio
MacOS	En estudio

### 19.3 Aplicación del método de intercambio de primitivas

El método de intercambio de primitivas se aplica proporcionando el acceso a las funciones del método de intercambio básico, que se describen en general en 7.1. Para garantizar la compatibilidad binaria a través de una plataforma determinada, se deberá respetar estrictamente las reglas, tipos de datos y prototipos de función especificados en las siguientes subcláusulas para cada plataforma.

Para funcionar correctamente, la realización debe contar con ciertas estructuras de datos y constantes. Las estructuras de datos y las constantes comunes a todas las realizaciones se describen a continuación en lenguaje C.

```
/*
** This generic C language description uses certain generic types which are mapped to
** real data types in the platform dependent descriptions further below.
** The generic types used are: NUMBER, BUFPOINTER, LISTPOINTER and WORD.
*/

/**
* The structure bem_data_buffer describes a buffer.
* The types NUMBER and BUFPOINTER have to be mapped to each platform
***/

struct bem_data_buffer {
    NUMBER          buffer_size;
    BUFPOINTER      buffer;
};

/**
* The structure bem_data_files describes a list of filenames.
* The types NUMBER and LISTPOINTER have to be mapped to each platform
***/

struct bem_data_files {
    NUMBER          number_of_files;
    LISTPOINTER     file_name[ ];
};

/**
* The structure bem_data_descriptor describes the data conveyed between LAs and CAs
* (data files, address lists, etc.). The data may be in files and/or in memory. The structure
* bem_data_descriptor accounts for these situations.
*
* When data are in memory, the structure bem_data_descriptor allows to actually convey that
* data through the EPutData or EGetData functions. When data are in files, bem_data_descriptor
* describes the actual file names that contain that data, i.e., the structure does not contain the data
* itself.
*
* The type WORD denotes a 16 bit unsigned integer, which has to be mapped to each platform
***/

struct bem_data_descriptor {
    WORD            key;          /* key = 0 for data in buffer */
                                /* key = 1 for data in file */
    WORD            type;        /* type = 1 for "document lists" */
                                /* type = 2 for "recipient lists", etc. */
                                /* used only in SEND, SENDACK and RECEIVE */
                                /* ignored otherwise */
    union {
        struct bem_data_buffer buffer; /* data in memory buffer */
        struct bem_data_files file;    /* data in file */
    } bof; /* buffer or file */
};
```

```

#define BEM_DATA      struct bem_data_descriptor

/* Constants used for the "key" field of BEM_DATA */
#define BUFFER_KEY    0      /* key: data in buffer */
#define FILE_KEY      1      /* key: data in file */

/* Constants used for the "type" field of BEM_DATA */
#define DOCUMENT_TYPE 1      /* type: document list */
#define RECIPIENT_TYPE 2     /* type: recipient list (= address list) */
#define ORIGINATOR_TYPE 3    /* type: originator list */
#define RELATED_TYPE  4      /* type: related list */

/* Constants used for TDD Types, see also 7.2.4.2 */
#define NO_TDD_RESPONSE 0x00
#define SENDACK_RESPONSE 0x10
#define RECEIVE_RESPONSE 0x20
#define COPY_RESPONSE 0x30
#define DELETE_RESPONSE 0x31
#define CANCEL_RESPONSE 0x32
#define PURGE_RESPONSE 0x33
#define RESCHEDULE_RESPONSE 0x34
#define DISPATCH_RESPONSE 0x35
#define PREVIEW_RESPONSE 0x36
#define PRINT_RESPONSE 0x40
#define CONVERT_RESPONSE 0x41
#define CHECK_RESPONSE 0x42
#define EXTEND_RESPONSE 0x50
#define NATIONAL_RESPONSE 0x60
#define PRIVATE_RESPONSE 0x70

/* Constants used for Alarm_types (may be or'ed), see also 7.2.7.1 */
#define ASYNC_RESPONSES 0x0001
#define QUEUE_FULL 0x0002
#define DOCUMENT_RECEIVED 0x0004
#define CONNECTION_LOST 0x0008
#define SEND_SUCCESS 0x0010
#define SEND_FAILED 0x0020
#define CORRUPTED_TDD 0x0040
#define SEND_EVENT 0x0080
#define RECEIVE_EVENT 0x0100
#define CA_WILL_STOP 0x0200
#define ALARMS_UNAVAILABLE 0x0400
#define TDD_RESP_AVAILABLE 0x0800

```

NOTA – Tratamiento del parámetro Data-ID en la función de EPutTDD para entrada y salida. A la entrada: la LA fija una dirección no-NULL si la LA necesita transferir ficheros de datos en llamadas de función EPutData subsiguientes. Si la LA no necesita transferir ficheros, este parámetro se pone a NULL. En salida: si el parámetro de entrada se puso a NULL, la CA no cambia ese valor. En los demás casos, la CA calcula un Data-ID y lo devuelve a la dirección especificada en la entrada.

### 19.3.1 MSDOS

El acceso al método de intercambio básico se producirá a través de un mecanismo de interrupción DOS. La LA llamante respetará las siguientes reglas:

- el número de interrupción utilizado por la CA se tomará del ICE (descriptor de CA);
- si se utiliza un esquema múltiplex, el número del múltiplex ha de ser tomado del ICE (descriptor de CA) y se cargará en un registro AH antes de la llamada;

- todos los parámetros se transferirán a la CA introduciéndolos en la pila del llamante. La pila tendrá 256 bytes libres como mínimo cuando se efectúa la llamada;
- los parámetros se introducirán utilizando convenios de llamada C (de derecha a izquierda), los punteros se introducirán como direcciones de palabra doble (punteros distantes);
- el direccionamiento de una función deseada se efectuará a través de un código de función, también introducido en la pila del llamante como parámetro más a la izquierda;
- el valor de retorno se pasará en el registro AX;
- la CA preservará los registros de segmento (así como el puntero de pila) durante la llamada; todos los otros registros pueden ser destruidos por la CA;
- el llamante vaciará la pila (expulsión de los parámetros llamantes) después del retorno de función (convenio llamante de lenguaje C).

Para direccionar las diversas funciones se utilizarán los códigos de funciones definidos en el Cuadro 127.

CUADRO 127/T.611  
Códigos de función DOS

Función del método de intercambio básico	Código de función
ELogin ( )	1
EPutTDD ( )	2
EPutData ( )	3
EPollTDD ( )	4
EGetTDD ( )	5
EGetData ( )	6
ESetAlarm ( )	7
EAbortData ( )	8
ELogout ( )	9

### 19.3.1.1 Prototipos de función C y definiciones

```
#define STATUS      short int
```

```
#define BEM_DATA   struct bem_data_descriptor
```

```
struct bem_data_buffer {
    unsigned long      buffer_size;      /* = generic type NUMBER */
    unsigned char far * buffer;          /* = generic type BUFPOINTER */
};
```

```
struct bem_data_files {
    unsigned long      number_of_files; /* = generic type NUMBER */
    char far *        file_name[ ];     /* = generic type LISTPOINTER */
};
```

```

struct bem_data_descriptor {
    unsigned short int      key;                /* = generic type WORD */
    unsigned short int      type;              /* = generic type WORD */
    union {
        struct bem_data_buffer buffer;
        struct bem_data_files file;
    } bof;
};

```

\*\*\*/

\* ELogin ( )

\*

* Stack: SP+18	Connection_ID	far pointer
* SP+14	CA_ID	far pointer
* SP+10	Selector	far pointer
* SP+6	Password	far pointer
* SP+2	Login_name	far pointer
* SP -->	0x0001	word value; function code to identify ELogin ( )

\*\*\*/

```

STATUS ELogin (
    short int      function_code,          /* ELogin = 0x0001 */
    char far *     Login_name,
    char far *     Password,
    char far *     Selector,
    short int far * CA_ID,
    short int far * Connection_ID
);

```

\*\*\*/

\* EPutTDD ( )

\*

* Stack: SP+12	Data_ID	far pointer
* SP+10	TDD_size	word value
* SP+6	TDD_location	far pointer
* SP+4	CA_ID	word value
* SP+2	Connection_ID	word value
* SP -->	0x0002	word value; function code to identify EPutTDD ( )

\*

\* If the address of Data\_ID is set to NULL on call, CA will not provide the Data\_ID and process the TDD immediately, assuming the pathes given in the TDD are correct

\*\*\*/

```

STATUS EPutTDD (
    short int      function_code,          /* EPutTDD = 0x0002 */
    short int      Connection_ID,
    short int      CA_ID,
    unsigned char far * TDD_location,
    short int      TDD_size,
    unsigned long far * Data_ID
);

```



```

STATUS      EGetTDD (
short int    function_code,          /* EGetTDD = 0x0005 */
short int    Connection_ID,
short int    CA_ID,
unsigned char far * TDD_location,
short int far * TDD_size,
unsigned long far * Data_ID
);

***/

*      EGetData (
*
*      Stack:  SP+14      Next      far pointer
*                SP+10      Data      far pointer
*                SP+6       Data_ID   dword value
*                SP+4       CA_ID     word value
*                SP+2       Connection_ID word value
*                SP -->    0x0006    word value; function code to identify EGetData ( )
***/

```

```

STATUS      EGetData (
short int    function_code,          /* EGetData = 0x0006 */
short int    Connection_ID,
short int    CA_ID,
unsigned long Data_ID,
BEM_DATA far * Data,
short int far * Next
);

***/

*      ESetAlarm (
*
*      Stack:  SP+8      Alarm_handler far pointer
*                SP+6      Alarm_event  word value
*                SP+4      CA_ID        word value
*                SP+2      Connection_ID word value
*                SP -->    0x0007      word value; function code to identify ESetAlarm ( )
***/

```

```

void      FAR AlarmHandler (
short int    Connection_ID,
short int    CA_ID,
unsigned short Alarm_type,
unsigned char far * Parameter,      /* additional parameter buffer */
short int    Length                /* length of parameter buffer */
);

```

```

STATUS      ESetAlarm (
short int    function_code,          /* ESetAlarm = 0x0007 */
short int    Connection_ID,
short int    CA_ID,
unsigned short Alarm_event,
void (* far Alarm_handler) ( )
);

```

```

***/
*      EAbortData ( )
*
*      Stack:  SP+6      Data_ID      dword value
*                SP+4      CA_ID        word value
*                SP+2      Connection_ID  word value
*                SP -->    0x0008      word value; function code to identify EAbortData ( )
***/

```

```

STATUS      EAbortData (
short int      function_code,      /* EAbortData = 0x0008 */
short int      Connection_ID,
short int      CA_ID,
unsigned long   Data_ID
);

```

```

***/
*      ELogout ( )
*
*      Stack:  SP+4      CA_ID        word value
*                SP+2      Connection_ID  word value
*                SP -->    0x0009      word value; function code to identify ELogout ( )
***/

```

```

STATUS      ELogout (
short int      function_code,      /* ELogout = 0x0009 */
short int      Connection_ID,
short int      CA_ID
);

```

### 19.3.2 WINDOWS

Para acceder a las funciones del método de intercambio básico, el proveedor de CA ofrecerá una biblioteca de vinculación dinámica (DLL, *dynamic link library*) que opone las funciones del método de intercambio, descritas en la siguiente subcláusula, a la LA.

El configurador de la plataforma deseada indicará el nombre de la DLL en el lugar adecuado de ICE. Para acceder a las funciones DLL, la LA emitirá primero una llamada de la función de sistema Windows "LoadLibrary", utilizando el nombre del parámetro DLL escogido.

#### 19.3.2.1 Prototipos y definiciones de la función C

```

#include <windows.h>

#define BEM_DATA struct bem_data_descriptor

struct bem_data_buffer {
int      buffer_size;      /* = generic type NUMBER */
HANDLE   buffer;          /* = generic type BUFPOINTER */
};

struct bem_data_files {
int      number_of_files;  /* = generic type NUMBER */
LPSTR    file_name[1];    /* = generic type LISTPOINTER */
};

```

```

struct bem_data_descriptor {
    word                key;                /* = generic type WORD */
    word                type;              /* = generic type WORD */
    union {
        struct bem_data_buffer buffer;
        struct bem_data_files file;
    } bof;
};

/**
 *      ELogin ( )
 ***/

void FAR PASCAL ELogin (
    LPSTR                Login_name,        /* User name */
    LPSTR                Password,         /* User's password */
    LPSTR                Selector,         /* CA Selector */
    int far *            CA_ID,           /* CA Identifier */
    int far *            Connection_ID,    /* Connection ID */
    int far *            Status           /* 0 success, 1 fail */
);

/**
 *      EPutTDD ( )
 *
 *      If the address of Data_ID is set to NULL on call, CA will not provide the Data_ID and process
 *      the TDD immediately, assuming the pathes given in the TDD are correct
 ***/

void FAR PASCAL EPutTDD (
    int                Connection_ID,      /* Connection ID */
    int                CA_ID,             /* CA Identifier */
    LPSTR              TDD_location,      /* TDD buffer text */
    int                TDD_size,          /* size of the TDD text */
    int far *          Data_ID,           /* Data identifier */
    int far *          Status             /* 0 success, 1 fail */
);

/**
 *      EPutData ( )
 ***/

void FAR PASCAL EPutData (
    int                Connection_ID,      /* Connection ID */
    int                CA_ID,             /* CA Identifier */
    int                Data_ID,           /* Data identifier */
    BEM_DATA far *    Data,               /* Data descriptor */
    int                Next,              /* next boolean */
    int far *          Status             /* 0 success, 1 fail */
);

/**
 *      EPollTDD ( )
 ***/

```

```

void FAR PASCAL EPollTDD (
    int          Connection_ID, /* Connection ID */
    int          CA_ID,        /* CA Identifier */
    word far *   TDD_type,     /* Type of next TDD */
    int far *   TDD_size,     /* Size of next TDD */
    int far *   TDD_count,    /* Count of waiting TDDs */
    int far *   Status        /* 0 success, 1 fail */
);

/**
 *      EGetTDD ( )
 ***/

void FAR PASCAL EGetTDD (
    int          Connection_ID, /* Connection ID */
    int          CA_ID,        /* CA Identifier */
    LPSTR       TDD_location,  /* TDD buffer text */
    int far *   TDD_size,     /* size of the TDD text */
    int far *   Data_ID,     /* Data identifier */
    int far *   Status        /* 0 success, 1 fail */
);

/**
 *      EGetData ( )
 ***/

void FAR PASCAL EGetData (
    int          Connection_ID, /* Connection ID */
    int          CA_ID,        /* CA Identifier */
    int          Data_ID,      /* Data identifier */
    BEM_DATA far * Data,      /* Data descriptor */
    int far *   Next,         /* next boolean */
    int far *   Status        /* 0 success, 1 fail */
);

/**
 *      ESetAlarm ( )
 ***/

void FAR PASCAL AlarmHandler (
    int          Connection_ID, /* Connection ID */
    int          CA_ID,        /* CA Identifier */
    word        Alarm_type,    /* Alarm type */
    HANDLE      Parameter,    /* additional parameter buffer */
    int        Length         /* length of parameter buffer */
);

void FAR PASCAL ESetAlarm (
    int          Connection_ID, /* Connection ID */
    int          CA_ID,        /* CA Identifier */
    word        Alarm_type,    /* Alarm event */
    FARPROC     Alarm_handler, /* Alarm handler entry */
    int far *   Status        /* 0 success, 1 fail */
);

```

```

/**
 *      EAbortData ( )
 ***/

void FAR PASCAL EAbortData (
    int          Connection_ID, /* Connection ID */
    int          CA_ID,        /* CA Identifier */
    int          Data_ID,      /* Data identifier */
    int far *    Status        /* 0 success, 1 fail */
);

/**
 *      ELogout ( )
 ***/

void FAR PASCAL ELogout (
    int          Connection_ID, /* Connection ID */
    int          CA_ID,        /* CA Identifier */
    int far *    Status        /* 0 success, 1 fail */
);

```

### 19.3.3 UNIX

Queda en estudio.

### 19.3.4 OS/2

Queda en estudio.

### 19.3.5 MacOS

Queda en estudio.

## Anexo A

### Sintaxis para presentación y codificación

(Este anexo es parte integrante de la presente Recomendación)

#### A.1 Sintaxis de estilo BNF

Para proporcionar una descripción genérica se utiliza una sintaxis basada en BNF. En esta Recomendación se siguen las reglas generales de la BNF.

- Un testigo terminal (hoja) se designa por un literal. Si el literal no puede distinguirse de los literales utilizados para describir la sintaxis, el testigo se escribirá entre caracteres de comillas ("").
- Las cadenas de caracteres encerradas entre caracteres de comillas denotan testigos terminales que consisten en el texto constante formado por esas cadenas.
- Un testigo no terminal (nodo) se designa por un literal delimitado por los caracteres "<" (menor que) y ">" (mayor que).
- Un testigo (o grupo de testigos) facultativo está delimitado por los caracteres "[" y "]".
- Un grupo de testigos está delimitado por los caracteres "(" y ")". Los grupos pueden estar anidados.
- Un grupo de testigos encerrados entre "{" y "}" puede repetirse 0, 1 o más veces.
- El carácter "|" se utiliza para separar testigos (o grupos de testigos) alternativos.

- La cadena ":@" se utiliza como delimitador de producción.
- Los testigos van separados por caracteres de espacio o de tabulación (o una combinación de ambos).
- La cadena "--" se utiliza para introducir un comentario en la descripción BNF. El comentario termina al final de la línea. La BNF no autoriza comentarios anidados.

Conceptos relativos a las codificaciones basadas en texto:

- El testigo terminal STRING denota una cadena de caracteres codificados según el valor del identificador de código (Code-ID) (véase el Cuadro 14). La cadena de caracteres puede ser de un tamaño limitado, como se indica por el testigo STRING (SIZE (xxx..yyy)). En este caso xxx designa la longitud mínima de la cadena, yyy designa la longitud máxima de la cadena.
- La notación especial "STRING (xxxx + yyyy)", donde "xxxx" e "yyyy" son cadenas de caracteres que producen una cadena formada por la concatenación de "xxxx" e "yyyy".
- El testigo terminal PATH denota una STRING que contiene solamente caracteres válidos como la especificación de trayecto absoluto para el sistema operativo subyacente.
- El testigo terminal NUMERIC-STRING denota una cadena de caracteres que representan un número. Sólo pueden usarse los dígitos 0 a 9.
- El testigo ↵ indica que se ha producido un corte de línea. Sin embargo, es posible incluir un comentario en una TDD terminando la línea TDD con un carácter ";", el comentario propiamente dicho y el corte de línea.
- El orden implicado por las reglas de producción en la descripción BNF puede no ser el mismo en que deben codificarse los testigos. Las reglas adicionales para los esquemas de codificación especifican el orden que ha de seguirse, si es que se estableciese alguno.

## A.2 Notación del lenguaje C

La notación genérica del lenguaje C utilizada se basa en la actual norma ANSI. De esta forma, si las descripciones se registran en un fichero, pueden compilarse con los compiladores ANSI existentes. Sin embargo, en las secciones que dependen del sistema operativo, es necesario aplicar los compiladores correspondientes. En general, las descripciones se basan en los siguientes convenios:

- En las definiciones y en los nombres de las variables sólo son significativos los primeros 16 caracteres.
- Se distingue entre las mayúsculas y las minúsculas.

## Anexo B

### Ubicación del entorno de configuración de interfaz (ICE)

(Este anexo es parte integrante de la presente Recomendación)

El entorno de configuración de interfaz (ICE, *interface configuration environment*) representa una fuente "global" para todas las aplicaciones locales (LA) conformes a esta Recomendación.

En los sistemas operativos UNIX y MacOS y en los sistemas operativos pertenecientes a la familia Microsoft (OS/2, MS-DOS, etc.), el ICE maestro está representado por un fichero<sup>17)</sup>. En un entorno informático LAN, como Netware, Vines o LAN Manager, el ICE maestro se presenta también como un fichero. Este fichero será abierto y leído por la aplicación para extraer información sobre las CA accesibles dentro del sistema. El nombre del fichero en los sistemas antes mencionados es "APPLICOM.ICE".

<sup>17)</sup> Un "fichero" en este contexto significa un fichero real de un sistema operativo, o un programa de control de dispositivo de un sistema operativo que se comporta exactamente como un fichero.

El fichero maestro ICE está situado en lugares diferentes en los mencionados sistemas. En algunos sistemas existe una "variable de entorno" ("environment variable")<sup>18)</sup>, que puede contener el trayecto del ICE. En esos casos la variable se llama "APPLICOM" (escrita con mayúsculas). Deberán utilizarse los siguientes algoritmos para determinar la ubicación del ICE (véase el Cuadro B.1).

CUADRO B.1/T.611

**Determinación de la ubicación del ICE maestro**

Sistema operativo	Algoritmo
UNIX	Primeramente se examina la variable de entorno "APPLICOM". Si esta variable no existe, el ICE está ubicado en el subdirectorio /dev y se llama APPLICOM_ICE"
MacOS	El ICE está ubicado en el "folder System"
MS-DOS	Examina la variable de entorno "APPLICOM". Si esta variable no existe, se debe considerar que se ha producido un error
Windows	El fichero se llama "ICE.INI" y está situado en el directorio windows local del sistema de la LA. Si este fichero no existe, se examina el fichero WIN.INI en el directorio windows, sección [APPLICOM_ICE]

En un entorno informático LAN, el fichero ICE maestro debe ser instalado por el administrador del sistema en un directorio compartido que sea accesible por todas las LA posibles.

Los ficheros o drivers de descriptor de CA individuales para las CA en un entorno de usuario único o de LAN no tienen que ser instalados en el mismo directorio que el ICE maestro, pero deben colocarse en directorios accesibles a todas las LA.

Deberá utilizarse el siguiente convenio de denominación para los ficheros o drivers de descriptor de CA:

CAnnnnAC.<ext>

donde:

nnnn es un número en la gama 1..9999

y:

<ext> := ("ini" | "exe" | "dll" | <other>)

"ini" se utiliza en el caso de un fichero de texto, y <other> puede ser definido por el fabricante de CA basándose en los convenios de denominación utilizados en el entorno informático de destino.

El acceso al descriptor o descriptors de CA para una CA individual se definirá en el ICE maestro en forma de un parámetro de trayecto (trayecto + nombre de fichero). Si no hay un trayecto definido, la ubicación por defecto del descriptor de CA será en el mismo trayecto que el ICE maestro.

<sup>18)</sup> Los sistemas operativos como UNIX o MS-DOS proporcionan un denominado entorno que consiste en un conjunto de cadenas ASCII aplicadas a una variable representada en ASCII.

## Anexo C

### Lista de los códigos de error de APPLI/COM (Este anexo es parte integrante de la presente Recomendación)

El Cuadro C.1 indica los códigos de error que comúnmente utilizan las realizaciones de CA conformes a esta Recomendación.

CUADRO C.1/T.611  
Lista de códigos de error

Código de error <sup>a)</sup>	Texto de error	Comentario
0000	Exito	Sin errores – Operación satisfactoria
0001-4999	Uso privado	Reservado para uso privado de CA
5000	Error de interfaz de CA	
5001	Servicio no soportado especificado	
5002	Ya existe fichero "FILENAME"	La CA no puede crear o escribir el fichero "FILENAME"
5003	Error de creación "TARGET"	La CA no puede crear o escribir el fichero "TARGET"
5004	Error de lectura lista de direcciones	
5005	Error de lectura lista de documentos	
5006	Error de lectura documento	
5007	Error de escritura documento	
5008	"Code-ID" no admitido	
5009	"Connection-ID" no válido	
5010	"CA-ID" no válido	
5011-5499	Reservado	Reservado para ulterior estudio
5500-5999	Uso privado	Reservado para uso privado
6000	Error de sintaxis	
6001	Función desconocida solicitada	
6002	TDD vacío	
6003	Encabezamiento APPLI/COM no válido	Longitud o contenido no válidos
6004	Error de analizador (línea TDD demasiado larga)	
6005	Error de analizador (línea TDD demasiado corta)	
6006	Múltiples apariciones de palabras clave	
6007	El parámetro con tamaño equivocado	
6008	Palabra clave ERROR faltante	
6009	Palabra clave obligatoria faltante	
6010	Palabra clave no definida	
6011	Palabras clave contradictorias	
6012	Parámetro fuera de gama	
6013	Palabras clave fuera de secuencia	Orden: FUNCTION, SUBFUNC, otros ...
6014	Separador de palabra clave faltante	Dos puntos (":") faltante
6015	"LA-ID" no válido	

CUADRO C.1/T.611 (fin)

Lista de códigos de error

Código de error <sup>a)</sup>	Texto de error	Comentario
6016	"REQ-ID" no válido	
6017	"COM-ID" no válido	
6018	"Service-ID" no válido	
6019	"Type-ID" no adaptado al "Service-ID"	
6020	"Conversion-ID" no adaptado al "Type-ID"	
6021	Fichero "TARGET" faltante	Aplicable a COPY TDD
6022	Identificador de código no válido	Permitidos: "A", "B", "C", "E" "I" o "P"
6023	Versión de interfaz incompatible	Versión del interfaz APPLI/COM no adaptada
6024-6499	Reservado	Reservado para ulterior estudio
6500-6999	Uso privado	Reservado para uso privado de CA
7000	Error conexo de soporte físico/sistema	
7001-7499	Reservado	Reservado para ulterior estudio
7500-7999	Uso privado	Reservado para uso privado de CA
8000	Error durante conversión de documento	
8001	Formato de transferencia no válido	
8002	Fin de documento inesperado	
8003	Error de acceso a documento inesperado	
8004	Error durante conversión T.61	
8005	Error durante conversión ASCII	
8006	Error durante conversión TIFF	
8007	Error durante conversión T.4	
8008	Error durante conversión T.6	
8009-8499	Reservado	Reservado para ulterior estudio
8500-8999	Uso privado	Reservado para uso privado de CA
9000	Error de transmisión	
9001	Conexión interrumpida	
9002	Destino distante ocupado	
9003	Establecimiento de conexión fracasado	
9006	Error de transmisión (capa de transporte)	
9007	Error de transmisión (capa de sesión)	
9008	Conexión cerrada o destino distante	
9009	Llamada denegada por destino distante	
9010	Destino distante no compatible	
9011	Fallo de transmisión - error de documento	
9012-9499	Reservado	Reservado para ulterior estudio
9500-9999	Uso privado	Reservado para uso privado de CA
<p>a) Los códigos de error figuran en decimal.</p>		

## Anexo D

### Ejemplos de intercambios de TDD

(Este anexo no es parte integrante de la presente Recomendación)

#### D.1 Ejemplo de sesión de envío

Una LA que funciona en un sistema basado en MS-DOS (o Windows) desea enviar un documento (que contiene información gráfica), por ejemplo el documento «c:\dtp\graphic1.tif» a un recipiente, por medio de los servicios facsímil grupo 3. La LA debe hacer, en secuencia, lo siguiente:

- buscar una CA capaz de proporcionar el servicio facsímil grupo 3 inspeccionando el ICE y, si la encuentra, abrir la sesión con esa CA mediante una llamada de función Login (si aún no lo ha hecho);
- preparar el documento como fichero TIFF de APPLI/COM (si aún no lo ha hecho);
- crear una TDD ENVÍO;
- entregar la TDD a la CA;
- interrogar repetidamente a la CA (o esperar a ser informada por la función de alarma) hasta que la respuesta TDD esté disponible;
- recuperar la respuesta TDD para conocer la situación de la transmisión;
- cerrar la sesión con la CEA (logout) o realizar otras funciones con la misma CA.

Supóngase que la LA ya ha establecido la comunicación con la CA y que el fichero «c:\dtp\graphic1.tif» ya está preparado en formato TIFF. Seguidamente, la LA debe preparar la TDD ENVÍO en su memoria. La LA utiliza la codificación por defecto (Rec. T.50) para la preparación de la TDD. La TDD puede presentarse del modo indicado en la Figura D.1.

!*APPLI/COM*1994*UIT-T*		
; Envío de un documento gráfico por facsímil grupo 3		
; Los campos en los que puede devolverse un valor están prefijados con caracteres de subrayado (5F <sub>HEX</sub> )		
FUNCTION	: SendAck	; Envío con respuesta
LA-ID	: myLA	; Nombre de la LA
REQ-ID	: g_0815	; Id de petición, generado por la LA
SERVICE	: FX3	; Servicio facsímil G3
ADDRESS	: 08154711	; Recibiente
FILENAME	: c:\dtp\graphic1.tif	; Trayecto completo al documento
CONVERT	: TIFF	; Formato de transferencia
COMID	: _____	; ID de CA exclusivo (respuesta)
STATUS	: _____	; Situación de la transmisión (respuesta)
ERROR	: _____	; ¿Hubo error? (respuesta)

FIGURA D.1/T.611

Después de preparada la TDD, la LA llama internamente a la función del mecanismo de intercambio básico PutTDD, que transfiere la TDD a la CA.

Seguidamente, la LA interroga repetidamente a la CA utilizando la función del mecanismo de intercambio básico PollTDD hasta que la respuesta TDD esté disponible. Utilizando la función del mecanismo de intercambio básico GetTDD, la LA recupera la TDD de la CA, la coloca en su propia memoria e inspecciona los resultados. La respuesta TDD puede presentarse del modo indicado en la Figura D.2.

```
I*APPLI/COM*1994*UIT-T*

; Envío de un documento gráfico por facsímil grupo 3
; Los campos en los que puede devolverse un valor están prefijados con caracteres de subrayado (5FHEX)

FUNCTION      :   SendAck                               ; Envío con respuesta
LA-ID         :   myLA                                  ; Nombre de la LA
REQ-ID        :   g_0815                               ; Id de petición, generado por la LA
SERVICE      :   FX3                                  ; Servicio facsímil G3
ADDRESS       :   08154711                             ; Recibiente
FILENAME      :   c:\dtp\graphic1.tif                  ; Trayecto completo al documento
CONVERT       :   TIFF                                  ; Formato de transferencia
COMID         :   1234546                              ; ID de CA exclusivo (respuesta)
STATUS        :   +                                    ; Situación de la transmisión (respuesta)
ERROR         :   0000/Success                         ; ¿Hubo error? (respuesta)
```

FIGURA D.2/T.611

Como puede verse en los campos STATUS y ERROR, el envío fue realizado satisfactoriamente.

## D.2 Ejemplo de sesión de recepción

Una LA que funciona en cualquier entorno operativo desea saber si tiene documentos por recibir en una CA de facsímil. La LA debe hacer, en secuencia, lo siguiente:

- buscar una CA capaz de proporcionar el servicio facsímil inspeccionando el ICE y, si la encuentra, abrir la sesión con esa CA mediante una llamada de función Login;
- preparar una TDD de recepción y transferirla a la CA utilizando la función PutTDD;
- interrogar repetidamente a la CA (utilizando la función PollTDD) hasta que la respuesta TDD esté disponible (o esperar a ser informada por una función de alarma de retorno);
- recuperar la respuesta TDD (utilizando la función GetTDD) para conocer la situación de la recepción;
- cerrar la sesión con la CA (logout) o realizar otras funciones.

La LA utiliza la codificación por defecto (Rec. T.50) para la preparación de la TDD de recepción. La TDD puede presentarse del modo indicado en la Figura D.3.

```

!*APPLI/COM*1994*UIT-T*

; Recepción de un documento T.4 por facsímil grupo 3

; Los campos en los que puede devolverse un valor están prefijados con caracteres de subrayado (5FHEX).

FUNCTION      :   Receive                               ; Envío con respuesta
LA-ID        :   myLA                                 ; Nombre de la LA
REQ-ID       :   g_0816                               ; Id de petición, generado por la LA
SERVICE     :   FX3                                  ; Servicio facsímil G3
FILENAME     :   c:\file.ext                          ; Trayecto completo al documento
Cvfax3      :   TIFF                                  ; Formato de transferencia deseado
TypeID      :   _____                            ; Situación del fichero recibido
ADDRESS     :   _____                            ; Destinatario del expedidor,
; rellenado por la CA
CONVERT     :   _____                            ; Formato de transferencia
COMID       :   _____                            ; ID de CA exclusivo (respuesta)
STATUS      :   _____                            ; Situación de la transmisión (respuesta)
ERROR       :   _____                            ; ¿Hubo error? (respuesta)

```

FIGURA D.3/T.611

Tras preparar la TDD, la LA llama internamente a la función del mecanismo de intercambio básico PutTDD y transfiere la TDD a la CA.

Seguidamente, la LA interroga repetidamente a la CA utilizando la función del mecanismo de intercambio básico PollTDD hasta que la respuesta TDD esté disponible. Utilizando la función del mecanismo de intercambio básico GetTDD, la LA recupera la TDD de la CA, la coloca en su propia memoria e inspecciona los resultados. La respuesta TDD puede presentarse del modo indicado en la Figura D.4.

!*APPLI/COM*1994*UIT-T*		
; Recepción de un documento T.4 por facsímil grupo 3		
; Los campos en los que puede devolverse un valor están prefijados con caracteres de subrayado (5F <sub>HEX</sub> ).		
FUNCTION	: Receive	; Envío con respuesta
LA-ID	: myLA	; Nombre de la LA
REQ-ID	: g_0816	; Id de petición, generado por la LA
SERVICE	: FX3	; Servicio facsímil G3
FILENAME	: c:\file.ext	; Trayecto completo al documento
Cvfax3	: TIFF	; Formato de transferencia deseado
TypeID	: STD	; Situación del fichero recibido
ADDRESS	: 033145782762	; Destinatario del expedidor, rellenado por la CA
CONVERT	: T.4	; Formato de transferencia
COMID	: 000001	; ID de CA exclusivo (respuesta)
STATUS	: +	; Situación de la transmisión (respuesta)
ERROR	: 0000/Success	; ¿Hubo error? (respuesta)

FIGURA D.4/T.611

Como puede verse en los campos STATUS y ERROR, la recepción se realizó satisfactoriamente.

### D.3 Ejemplo de sesión de rastreo

Una LA que funcione en cualquier entorno operativo puede obtener información acerca del estado transitorio o definitivo de un registro de comunicación (registro de CA). La LA debe hacer, en secuencia, lo siguiente:

- buscar una CA capaz de proporcionar cualquier servicio de telecomunicación inspeccionando el ICE y, si la encuentra, abrir la sesión con la CA mediante una llamada de función Login;
- preparar una TDD de rastreo y enviarla a la CA utilizando la función PutTDD;
- interrogar repetidamente a la CA (utilizando la función PollTDD) hasta que la respuesta TDD esté disponible (o esperar a ser informada mediante la función de alarma de retorno);
- recuperar la respuesta TDD (utilizando la función GetTDD) para conocer la situación de la recepción;
- cerrar la sesión con la CA (logout) o realizar otras funciones.

La LA utiliza la codificación por defecto (Rec. T.50) para la preparación de la TDD de recepción. La TDD puede presentarse del modo indicado en la Figura D.5.

```

I*APPLI/COM*1994*UIT-T*

; Recuperación de una lista de todos los registros de CA en el estado «en envío» (sending)
; Los campos en los que puede devolverse un valor están prefijados con caracteres de subrayado (5FHEX).

FUNCTION      : Copy                               ; Envío con respuesta
LA-ID        : myLA                               ; Nombre de la LA
REQ-ID       : g_0816                             ; Id de petición, generada por la LA
State        : sending                             ; Registros de CA que se están procesando
Target       : c:\file.ext                         ; Trayecto completo al documento
ERROR        : _____                         ; ¿Hubo error? (respuesta)

```

FIGURA D.5/T.611

Tras preparar la TDD, la LA llama internamente a la función del mecanismo de intercambio básico PutTDD y transfiere la TDD a la CA.

Seguidamente, la LA interroga repetidamente a la CA utilizando la función del mecanismo de intercambio básico PollTDD hasta que la respuesta TDD esté disponible. Utilizando la función del mecanismo de intercambio básico GetTDD, la LA recupera de la CA todos los registros de CA que se encuentran en un estado determinado. La respuesta TDD puede presentarse del modo indicado en la Figura D.6.

```

I*APPLI/COM*1994*UIT-T*

; Recuperación de una lista de todos los registros de CA en el estado «en envío» (sending)
; Los campos en los que puede devolverse un valor están prefijados con caracteres de subrayado (5FHEX).

FUNCTION      : Copy                               ; Envío con respuesta
LA-ID        : myLA                               ; Nombre de la LA
REQ-ID       : g_0816                             ; Id de petición, generada por la LA
State        : sending                             ; Registros de CA que se están procesando
Target       : c:\file.ext                         ; Trayecto completo al documento
ERROR        : 0000/Success                       ; ¿Hubo error? (respuesta)

```

FIGURA D.6/T.611

Como puede verse en el campo ERROR, la copia fue recibida por la CA.

## Anexo E

### Ejemplo de entorno de configuración de interfaz (ICE)

(Este anexo no es parte integrante de la presente Recomendación)

En la Figura E.1 se da un ejemplo de una configuración de ICE maestro (ubicado en una máquina basada en MS-DOS como fichero). En el ejemplo se muestra un ICE maestro que proporciona acceso a múltiples CA que admiten múltiples servicios y métodos de acceso.

```
!*APPLI/COM*1994*UIT-T*MASTER_ICE

; Se señala que puede haber una o varias CA referenciadas en el ICE maestro y que cada una debe incluir
; las palabras clave APPLICOM; SERVICE; EM y ACCESS.
; Una nueva entrada CA (CA-Entry) se indica por "#" (signo de número) como se muestra más abajo.

#                               ; Comienzo de una nueva CA-Entry
APPLICOM:      Product1 (c) by DonaldDuck   ; Producto y fabricante de CA
SERVICE:      FX3                          ; Admite el servicio facsímil grupo 3
SERVICE:      EMAIL                        ; Admite el servicio EMAIL
EM:            file                          ; Método de intercambio de ficheros
ACCESS:        CA1AC.INI                    ; Descriptor de CA basado en fichero

#                               ; Comienzo de un nuevo CA-Header
APPLICOM:      Product2 (c) by MickeyMouse   ; Producto y fabricante de CA
SERVICE:      FX3                          ; Admite el servicio facsímil grupo 3
SERVICE:      TLX                          ; Admite el servicio télex
SERVICE:      FT                            ; Admite el servicio de transferencia de ficheros
EM:            primitive                     ; Método de intercambio de primitivas
CA-ID:         007                          ; Identificación de CA
ACCESS:        CA2AC.INI                    ; Descriptor de CA basado en fichero
ACCESS:        CA2AC.EXE                    ; Descriptor de CA ejecutable
ACCESS:        CA2AC.DLL                    ; Descriptor de CA DLL
```

FIGURA E.1/T.611

**Ejemplo de ICE maestro**

En la Figura E.2 se da un ejemplo de descriptores de CA (ubicados en una máquina basada en MS-DOS en un fichero).

```

I*APPLI/COM*1994*UIT-T*ICE
; Se señala que aquí puede almacenarse cualquier tipo de información de configuración. Una nueva información
; de configuración va siempre precedida de «#» (signo de número), como se ve seguidamente.

# ; Comienzo de una nueva configuración
APPLICOM: Product1 (c) by Company XYZ ; Producto CA y fabricante
FC: A ; Clase funcional APPLI/COM
EM: FILE ; Método de intercambio de TDD
SYNC: No ; No accionado por «sync»
CODING: I ; Codificación de TDD
F_JOB_Q: c:\applicom\job ; Fila de "jobs"
F_ACK_Q: c:\applicom\ack ; Fila de acuses de recibo (respuesta)
ERROR_Q: c:\applicom\err ; Fila de errores (respuesta)
TLX: STD ; Servicio télex (sin facilidad de diálogo) admitido
TX: STD ; Servicio télex vía teletex admitido
TTX: STD ; Servicio teletex y opciones de tipo
TTX: OPD ; Servicio teletex y opciones de tipo
TTX: CTL ; Servicio teletex y opciones de tipo
TTX: DTM ; Servicio teletex y opciones de tipo
TTX: EDI ; Servicio teletex y opciones de tipo
FX3: STD ; Servicio telefax grupo 3
ADDKEYS: LASTTIME ; Palabras clave aditivas
ADDKEYS: SUBADDR ; Palabras clave aditivas

# ; Comienzo de una nueva configuración
APPLICOM: Product2 (c) by Company ABC ; Producto CA y fabricante
FC: B ; Clase funcional APPLI/COM
EM: primitive ; Método de intercambio de TDD
ALARM: yes ; Llamada de retorno posible
CODING: I ; Codificación de TDD
DRIVER: applicom ; Controlador ("driver") suministrado
FX3: STD ; Servicio fax G3 y opciones de tipo
FX3: DTM ; Servicio fax G3 y opciones de tipo
FX3: BFT ; Servicio fax G3 y opciones de tipo
SUBMIT: CONVERT ; Admite depósitos de conversiones
CONVCHK: TIFF2 ; Pueden pedirse conversiones a/de TIFF2
CONVCHK: PCX ; Pueden pedirse conversiones a/de PCX
ADDKEYS: LASTTIME ; Palabras clave aditivas
ADDKEYS: SPEED ; Palabras clave aditivas
ADDKEYS: COMMENT ; Palabras clave aditivas
...

```

FIGURA E.2/T.611

**Ejemplo de descriptores de CA**

## Anexo F

### Método de intercambio de la versión 1992

(Este anexo no es parte integrante de la presente Recomendación)

En este anexo se describen las funciones del método de intercambio de la versión de 1992 de esta Recomendación, que han sido remplazadas por las nuevas funciones del método de intercambio básico. En el Cuadro F.1 se muestra la correspondencia entre las funciones de ambas versiones.

CUADRO F.1/T.611

#### Comparación de los métodos de intercambio

Funciones del método de intercambio (versión de 1992)	Funciones del método de intercambio básico (esta Recomendación)
Login ( )	ELogin ( )
PutTDD ( )	EPutTDD ( )
–	EPutData ( )
PollTDD ( )	EPollTDD ( )
GetTDD ( )	EGetTDD ( )
–	EGetData ( )
SetAlarm ( )	ESetAlarm ( )
–	EAbortData ( )
Logout ( )	ELogout ( )

### F.1 Funciones del método de intercambio de la versión de 1992

#### F.1.1 Función Login

La función Login deberá ser soportada por la CA. Deberá ser invocada por la LA antes de cualquier intercambio LA-CA de peticiones y respuestas TDD.

##### F.1.1.1 Finalidad

La función Login retorna a la LA un identificador de conexión que se utilizará a lo largo de toda la interacción LA-CA hasta que la LA pone fin a la sesión (o "desengancha", efectuando un "logout").

La función login es el lugar en que una CA puede controlar el acceso a ella por una LA. Esto se realiza comprobando el nombre de login (Login-name) y la contraseña (Password) dados por la LA. Sin embargo, la medida en que se efectúa el control de los derechos de acceso depende de la implementación de la CA.

##### F.1.1.2 Comportamiento

La CA comprueba los parámetros de la llamada a Login. Si concuerdan, genera un identificador de conexión que la LA utilizará seguidamente en las llamadas a las funciones PutTDD, PollTDD, GetTDD y Logout. La LA esperará a recibir la indicación de la situación para continuar el procedimiento: si el identificador de conexión transmitido tiene el valor Null (cero), ello significa que la CA no ha conectado con la LA por defectos de identificación.

##### F.1.1.3 Parámetros

Se requieren los siguientes parámetros (véase el Cuadro F.2).

CUADRO F.2/T.611

**Parámetros de la función Login**

Nombre del parámetro	Estructura	Comentario	Sentido
Nombre de Login (Login-Name)	Cadena (string)	Indica el nombre del usuario LA utilizado para conectar una LA a una CA [difiere del identificador de LA (LA-ID)]	Parámetro de entrada
Contraseña (Password)	Cadena (string)	La LA transmite su contraseña para que la CA pueda identificarla	Parámetro de entrada
Identificador de conexión (Connection-ID)	Entero (integer)	Transmitido por la CA si la función Login tiene éxito. En caso contrario (por ejemplo, si la identificación falla) la CA fija el valor a Null (cero) y el parámetro Situación (Status) proporciona el código de error (véase más adelante)	Parámetro de salida
Situación (Status)	Entero (integer)	Devuelve código de error (0000 significa éxito)	Parámetro de salida

**F.1.2 Función PutTDD**

La función PutTDD deberá ser admitida por la CA y podrá ser invocada por una LA.

**F.1.2.1 Finalidad**

La finalidad de la función PutTDD es transmitir una petición TDD de una LA a una CA.

**F.1.2.2 Comportamiento**

La CA copia la petición TDD cursada por la función PutTDD en sus estructuras internas. De ello resulta una situación notificada de inmediato a la LA solicitante, y una referencia REQ-ID (identificador de petición).

Seguidamente, la CA examina la TDD y sigue procesándola de acuerdo con su naturaleza.

**F.1.2.3 Parámetros**

Se requieren los siguientes parámetros (véase el Cuadro F.3).

CUADRO F.3/T.611

**Parámetros de la función PutTDD**

Nombre del parámetro	Estructura	Comentario	Sentido
Identificador de conexión (connection-ID)	Entero (integer)	El identificador de conexión devuelto por la función Login	Parámetro de entrada
Ubicación de la TDD (TDD location)	Dirección memoria (memory address)	Especifica la ubicación la TDD de la LA, de modo que la CA pueda copiarla en su propia estructura interna. Cuando la función esté completa, la TDD de la LA podrá suprimirse o usarse para otros fines	Parámetro de entrada
Tamaño de la TDD (TDD size)	Entero (integer)	Indica el tamaño de la TDD, de modo que la CA pueda asignar suficientes recursos internos para tratarla	Parámetro de entrada
Situación (Status)	Entero (integer)	Acusa recibo de la función PutTDD. Devuelve código de error (0000 significa éxito)	Parámetro de salida

### F.1.3 Función PollTDD

La función PollTDD pregunta a la CA cuántas respuestas TDD están en espera de ser tratadas por la LA solicitante. En la contestación a la PollTDD se comunica el número de respuestas TDD en espera y el tipo y tamaño de la primera respuesta TDD que será devuelta con la siguiente llamada a la función GetTDD.

#### F.1.3.1 Finalidad

La finalidad de la función PollTDD es preparar la LA para el tratamiento de una posible respuesta TDD procedente de una CA. También da una indicación del número de respuestas TDD que están en espera de tratamiento por la LA.

#### F.1.3.2 Comportamiento

Cuando la CA tiene muchas respuestas TDD para transmitir, elige la que transmitirá primero. Esta respuesta TDD es la TDD que se transmitirá a la LA en la siguiente llamada a la función GetTDD emitida por la misma LA.

Cuando no hay una respuesta TDD para la LA solicitante, se pone a cero la situación ("status") de retorno, en cuyo caso el tamaño de TDD retornado se pone a cero. Si la TDD depositada era errónea o desconocida, por ejemplo, le faltaba el <TDD Header> (encabezamiento de la TDD), la función retorna la petición TDD original y pone la situación de retorno al valor cero.

Cuando está disponible una respuesta TDD, la LA asignará una TDD vacía que contendrá la copia de la respuesta TDD que todavía está bajo el control de la CA. El Cuadro F.4 define los diversos tipos de TDD.

CUADRO F.4/T.611

#### Asignación de tipos de TDD

Número	Tipo de TDD	Número	Tipo de TDD
0	Se depositó una TDD desconocida o errónea, o no hay respuesta TDD disponible	4	Respuesta SUBMIT
1	Respuesta SEND	5	Respuesta EXTEND
2	Respuesta RECEIVE	6	Respuesta NATIONAL
3	Respuesta TRACE	7	Respuesta PRIVATE

#### F.1.3.3 Parámetros

Se requieren los siguientes parámetros (véase el Cuadro F.5).

### F.1.4 Función GetTDD

La función GetTDD deberá ser admitida por la CA, y podrá ser invocada por una LA.

#### F.1.4.1 Finalidad

La finalidad de la función GetTDD es recuperar una respuesta TDD procedente de una CA. La CA copia la respuesta TDD en una estructura interna de la LA.

CUADRO F.5/T.611

**Parámetros de la función PollTDD**

Nombre del parámetro	Estructura	Comentario	Sentido
Identificador de conexión (Connection-ID)	Entero (integer)	El identificador de conexión devuelto por la función Login	Parámetro de entrada
Tamaño de la TDD (TDD size)	Entero (integer)	Indica el tamaño de la siguiente respuesta TDD para que la LA pueda preparar recursos suficientes para tratarla	Parámetro de salida
Tipo de TDD (TDD type)	Entero (integer)	Indica el tipo de la TDD que va a recibir a continuación la LA. Véanse los valores más arriba	Parámetro de salida
Cuenta de TDD (TDD count)	Entero (integer)	Indica el número de respuestas TDD que están en espera de ser recuperadas por la LA. El valor Null significa que no hay TDD en espera	Parámetro de salida
Situación (Status)	Entero (integer)	Acusa recibo de la función PollTDD. Devuelve código de error (0000 significa éxito)	Parámetro de salida

**F.1.4.2 Comportamiento**

La LA indica la ubicación de una TDD vacía donde la CA copiará la respuesta TDD que está disponible para la LA.

La CA devolverá a la LA la respuesta TDD designada en la función PollTDD anterior emitida por la misma LA. La LA tendrá preparada en sus estructuras internas una zona receptora de respuestas TDD. La invocación de la función GetTDD por una LA será precedida siempre por una llamada a la función PollTDD.

Si la LA invoca dos o más funciones GetTDD consecutivas (sin una llamada a función PollTDD intermedia) siempre se devolverá la misma TDD (la que está indicada en el retorno de la última función PollTDD).

**F.1.4.3 Parámetros**

Se requieren los siguientes parámetros (véase el Cuadro F.6).

CUADRO F.6/T.611

**Parámetros de la función GetTDD**

Nombre del parámetro	Estructura	Comentario	Sentido
Identificador de conexión (Connection-ID)	Entero (integer)	El identificador de conexión devuelto por la función Login	Parámetro de entrada
Ubicación de la TDD (TDD location)	Dirección memoria (memory address)	Especifica el lugar de la estructura interna de la LA en que la CA puede copiar la respuesta TDD. Cuando la función se ha completado, la respuesta TDD existente en la CA puede ser suprimida o utilizada para otros fines	Parámetro de entrada/salida
Tamaño de la TDD (TDD size)	Entero (integer)	Indica el tamaño de la TDD vacía transferida a la CA	Parámetro de entrada
Situación (Status)	Entero (integer)	Acusa recibo de la función GetTDD. Devuelve código de error (0000 significa éxito)	Parámetro de salida

### F.1.5 Función SetAlarm

La CA puede admitir, con carácter optativo, la función SetAlarm, que puede ser invocada opcionalmente por una LA. Si la LA utiliza la función SetAlarm, deberá admitir la función CallbackRoutine. El soporte de la función SetAlarm se declara en el ICE.

#### F.1.5.1 Finalidad

La finalidad de la función SetAlarm es declarar a la CA el punto de entrada de la función CallbackRoutine. La función SetAlarm indica a la CA que puede "despertar" a la LA invocando la función CallbackRoutine. Esta función puede ser invocada una sola vez en cada sesión de diálogo LA/CA.

#### F.1.5.2 Comportamiento

La CA registrará la ubicación de la función CallbackRoutine asignada por la LA. La CA puede registrar tantas ubicaciones CallbackRoutine como LA haya con sesión abierta. La CA podrá así alertar a una LA determinada invocando su CallbackRoutine.

#### F.1.5.3 Parámetros

Se requieren los siguientes parámetros (véase el Cuadro F.7).

CUADRO F.7/T.611

#### Parámetros de la función SetAlarm

Nombre del parámetro	Estructura	Comentario	Sentido
Identificador de conexión (Connection-ID)	Entero (integer)	El identificador de conexión devuelto por la función Login	Parámetro de entrada
Ubicación de la CallbackRoutine (CallbackRoutine location)	Dirección memoria (memory address)	Especifica el punto de entrada de una función CallbackRoutine de una determinada LA	Parámetro de entrada
Situación (Status)	Entero (integer)	Acusa recibo de la función SetAlarm. Devuelve código de error (0000 significa éxito)	Parámetro de salida

### F.1.6 Función CallbackRoutine

La LA puede admitir, con carácter optativo, la función CallbackRoutine, que puede ser invocada opcionalmente por una CA. Para que la CA pueda invocarla, la LA tendrá que haberla declarado a la CA por medio de la función SetAlarm.

#### F.1.6.1 Finalidad

La función CallbackRoutine define un mecanismo que permite a una CA alertar a la LA de que existen algunas respuestas TDD disponibles. El uso de este mecanismo optativo puede mejorar el control de flujo entre las LA y las CA en sistemas muy cargados.

La invocación de CallbackRoutine por una CA no garantiza que la LA la interrogará dentro de un plazo determinado. Sólo asegura que la LA recibirá una alarma de una CA específica.

La CA puede invocar repetidamente la función CallbackRoutine de una LA determinada, si ésta no interroga a la CA con suficiente rapidez.

Sólo se permite una dirección de llamada de retorno ("Callback address") por LA. No obstante, si una LA desea que se le llame con relación a un evento determinado, tiene que implementar esta característica por sí misma dentro de la función de llamada de retorno.

### F.1.6.2 Comportamiento

La CallbackRoutine está destinada a permitir que una CA con la que una LA tenga abierta una sesión, advierta a esta última que debe interrogarla por medio de la función PollTDD. Así pues, la LA deberá interrogar a la CA lo más pronto posible para que ésta no pase a una condición de error por desbordamiento.

### F.1.6.3 Parámetros

Se requiere el siguiente parámetro (véase el Cuadro F.8).

CUADRO F.8/T.611

#### Parámetros de la función CallbackRoutine

Nombre del parámetro	Estructura	Comentario	Sentido
Identificador de conexión (Connection-ID)	Entero (integer)	El identificador de conexión devuelto por la función Login	Parámetro de entrada

### F.1.7 Función Logout

La función Logout deberá ser admitida por la CA y deberá ser invocada por la LA al completarse el intercambio de peticiones y respuestas TDD entre la LA y la CA.

#### F.1.7.1 Finalidad

La función Logout devuelve a la LA una situación ("status") que indica si la interacción LA-CA ha concluido de manera ordenada.

#### F.1.7.2 Comportamiento

Antes de concluir el diálogo LA-CA, la CA puede procesar todas las peticiones TDD pendientes enviadas por esa LA (aunque no está obligada a ello).

#### F.1.7.3 Parámetros

Se requieren los siguientes parámetros (véase el Cuadro F.9).

CUADRO F.9/T.611

#### Parámetros de la función Logout

Nombre del parámetro	Estructura	Comentario	Sentido
Identificador de conexión (Connection-ID)	Entero (integer)	El identificador de conexión devuelto por la función Login	Parámetro de entrada
Situación (Status)	Entero (integer)	Devuelto por la CA. Indica si el logout se ha procesado de manera ordenada. Devolver código de error (0000 significa éxito)	Parámetro de salida

## Anexo G

### Información específica del servicio

(Este anexo no es parte integrante de la presente Recomendación)

#### G.1 Renglón de identificación de la llamada del servicio telefax G4 y del servicio teletex

El renglón (o línea) de identificación de la llamada (o comunicación) se define en la Recomendación UIT-T F.200 y su composición se indica en la Figura G.1.

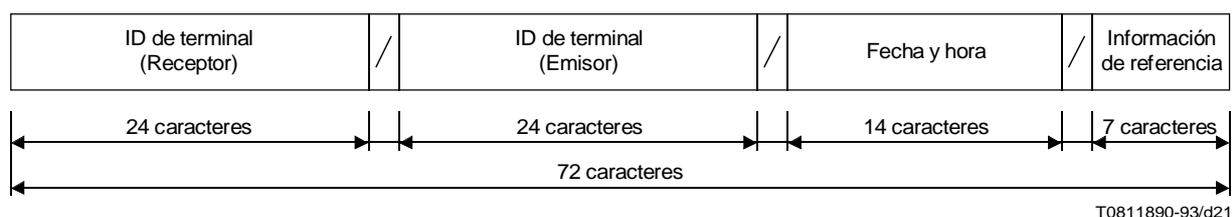


FIGURA G.1/T.611

Composición del renglón de identificación de la llamada; "/" es el carácter 2F<sub>HEX</sub>

#### G.2 Identificador de terminal

El identificador de terminal para el servicio teletex se define en la Recomendación UIT-T F.200, y para el servicio telefax del grupo 4 en la Recomendación UIT-T F.184. Su composición se muestra en la Figura G.2. DNIC significa código de identificación de red de datos (*data network identification code*).

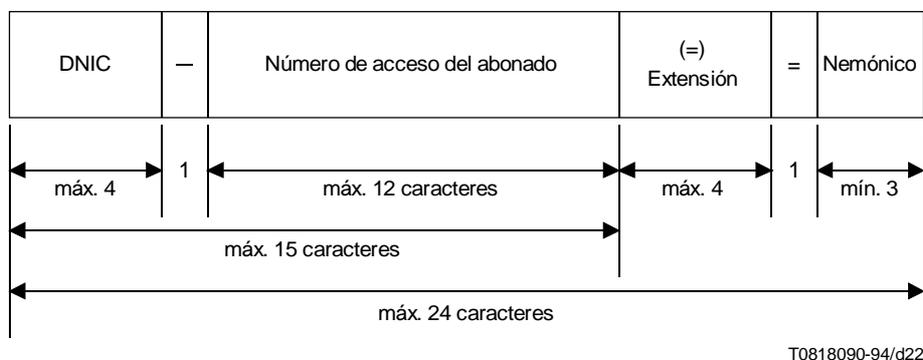


FIGURA G.2/T.611

Identificación de terminal telefax G4/teletex

## Anexo H

### Resumen de los formatos de transferencia y de transmisión

(Este anexo no es parte integrante de la presente Recomendación)

#### H.1 Formato de transferencia relacionado con los formatos de transmisión

Véase el Cuadro H.1

CUADRO H.1/T.611

#### Resumen de los formatos de transferencia

Parámetro identificador de servicio	Parámetro identificador de tipo	Parámetro identificador de conversión autorizado para	
		documentos salientes	documentos entrantes
FX3	STD	ASCII, ASCIIxxx <sup>a)</sup> , T.50, TIFF	TIFF, TIFFx <sup>b)</sup>
	BTM, DTM, BFT, EDI	VOID	
FX4	STD	ASCII, ASCIIxxx, T.50, TIFF	TIFF, TIFFx
	DTM, BFT, EDI	VOID	
TLX	STD	ASCII, ASCIIxxx, T.50, T.61	
TX	STD	ASCII, ASCIIxxx, T.50, T.61	
TTX	STD	ASCII, ASCIIxxx, T.50, T.61	
	OPD, MD, CTL	ASCII, ASCIIxxx, T.50, T.61	
	DTM, BFT, EDI	VOID	
EMAIL	STD	T.50	
	TELETEX	ASCII, ASCIIxxx, T.50, T.61	
	G3FAX, G4CLASS1	ASCII, ASCIIxxx, T.50, TIFF	TIFFx
	VIDEOTEX, MESSAGE, BILATERAL, NATIONAL. ODA	VOID	
FT	STD	VOID	

a) "xxx" en "ASCIIxxx" representa la página de códigos declarada en el ICE (por ejemplo, "ASCII437").

b) "x" representa la clase TIFF que habrá de leerse; su valor está comprendido entre 2 y 4.

#### H.2 Formatos de transmisión relacionados con el servicio

Véase el Cuadro H.2

CUADRO H.2/T.611

**Resumen de los formatos de transmisión (tipos de documento)**

Id. de servicio	Id. de tipo	Significa
TLX	STD	Servicio télex (sin facilidad de diálogo)
TX	STD	Télex por facilidad de conversión TELETEX
TTX	STD <sup>a)</sup>	TELETEX básico (T.61)
	OPD <sup>a)</sup>	TELETEX básico (T.61): Documento de operador
	MD <sup>a)</sup>	TELETEX básico (T.61): Documento de supervisor
	CTL <sup>a)</sup>	TELETEX básico (T.61): Documento de control
	DTM <sup>b)</sup>	Transferencia de ficheros telemáticos (TFT) del servicio TELETEX: Modo transparente al documento
	BFT <sup>c)</sup>	Transferencia de ficheros binarios del servicio TELETEX
	EDI <sup>b)</sup>	Transferencia de ficheros telemáticos (TFT) del servicio TELETEX: Edifact
FX3	STD	Servicio telefax G3 básico (MH)
	BTM <sup>b)</sup>	Transferencia de ficheros telemáticos (TFT) del servicio telefax G3: Modo transparente básico
	DTM <sup>b)</sup>	Transferencia de ficheros telemáticos (TFT) del servicio telefax G3: Modo transparente al documento
	BFT <sup>d)</sup>	Fichero de transferencia binario del servicio telefax G3
	EDI <sup>b)</sup>	Transferencia de ficheros telemáticos (TFT) del servicio telefax G3: Edifact
FX4	STD	Servicio telefax G4 básico (MR)
	DTM <sup>b)</sup>	Transferencia de ficheros telemáticos (TFT) del servicio telefax G4: Modo transparente al documento
	BFT <sup>c)</sup>	Transferencia de ficheros binarios del servicio telefax G4
	EDI <sup>b)</sup>	Transferencia de ficheros telemáticos (TFT) del servicio telefax G4: Edifact
EMAIL	STD	Parte de cuerpo texto IA5
	TELETEX	Parte de cuerpo teletex
	G3FAX	Parte de cuerpo telefax grupo 3
	G4CLASS1	Parte de cuerpo telefax grupo 4
	VIDEOTEX	Parte de cuerpo videotex
	MESSAGE	Parte de cuerpo que contiene un IPM (que consta de encabezamiento y partes de cuerpo) transferido por el sistema E-Mail
	BILATERAL	Contenido de parte de cuerpo definido bilateral
	NATIONAL	Contenido de parte de cuerpo definido nacional
	ODA	Parte de cuerpo ODA
FT	STD	Transferencia de fichero básico
<p>a) Tipo de documento de acuerdo con Anexo E/T.62.                      b) Transferencia de ficheros telemáticos (TFT) de acuerdo con la Recomendación T.571.                      c) Transferencia de ficheros binarios de acuerdo con la Recomendación T.434.                      d) Transferencia de ficheros binarios de acuerdo con las Recomendaciones T.434 y T.30 (anexos).</p>		