

Reemplazada por una versión más reciente



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

CCITT

X.711

COMITÉ CONSULTIVO
INTERNACIONAL
TELEGRÁFICO Y TELEFÓNICO

(03/91)

**REDES DE COMUNICACIÓN DE DATOS:
INTERCONEXIÓN DE SISTEMAS ABIERTOS (ISA);
GESTIÓN**

**ESPECIFICACIÓN DEL PROTOCOLO COMÚN
DE INFORMACIÓN DE GESTIÓN PARA
APLICACIONES DEL CCITT**

Recomendación X.711

Reemplazada por una versión más reciente



Ginebra, 1991

Reemplazada por una versión más reciente

PREFACIO

El CCITT (Comité Consultivo Internacional Telegráfico y Telefónico) 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 Asamblea Plenaria del CCITT, que se celebra cada cuatro años, establece los temas que han de estudiarse y aprueba las Recomendaciones preparadas por sus Comisiones de Estudio. La aprobación de Recomendaciones por los miembros del CCITT entre las Asambleas Plenarias de éste es el objeto del procedimiento establecido en la Resolución N.º 2 del CCITT (Melbourne, 1988).

La Recomendación X.711 ha sido preparada por la Comisión de Estudio VII y fue aprobada por el procedimiento de la Resolución N.º 2 el 22 de marzo de 1991.

NOTA DEL CCITT

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

© UIT 1991

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.

Reemplazada por una versión más reciente

Recomendación X.711

ESPECIFICACIÓN DEL PROTOCOLO COMÚN DE INFORMACIÓN DE GESTIÓN PARA APLICACIONES DEL CCITT¹⁾

ÍNDICE

- 1 *Campo de aplicación*
- 2 *Referencias*
- 3 *Definiciones*
- 4 *Símbolos y abreviaturas*
- 5 *Visión de conjunto*
- 6 *Elementos de procedimiento*
- 7 *Sintaxis abstracta*
- 8 *Conformidad*

Anexo A – Reglas de asociación para el ESCIG

Anexo B – Sintaxis NSA.1 ampliada

Anexo C – Ejemplos de UDPA de ESOD del ESCIG

1 **Campo de aplicación**

Esta Recomendación especifica un protocolo utilizado por las entidades de capa de aplicación para intercambiar información de gestión.

Esta Recomendación especifica:

- los procedimientos para la transmisión de información de gestión entre entidades de aplicación;
- la sintaxis abstracta del protocolo común de información de gestión (PCIG) y las reglas de codificación asociadas que han de aplicarse;
- los procedimientos para la interpretación correcta de información de control de protocolo;
- los requisitos de conformidad que ha de satisfacer la realización de esta Recomendación.

Esta Recomendación no especifica:

- la estructura o significado de la información de gestión transmitida por medio del PCIG;
- la manera en que se realiza la gestión como resultado de intercambios de PCIG;
- las interacciones que resultan de la utilización del PCIG.

¹⁾ La Recomendación X.711 y la Norma ISO/CEI 9596: *Information Technology – Open Systems Interconnection – Common management information protocol specification*, se han redactado en estrecha colaboración y son técnicamente idénticas.

Reemplazada por una versión más reciente

2 Referencias

- [1] Recomendación del CCITT *Modelo de referencia de interconexión de sistemas abiertos para aplicaciones del CCITT*, Libro Azul, fascículo VIII.4, Rec. X.200, UIT, Ginebra, 1988.
- [2] ISO/IEC 7498-4 *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 4: Management Framework*, 1989.
- [3] Recomendación del CCITT *Especificación de la notación de sintaxis abstracta uno (NSA.1)*, Libro Azul, fascículo VIII.4, Rec. X.208, UIT, Ginebra, 1988.
- [4] Recomendación del CCITT *Especificación de las reglas básicas de codificación de la notación de sintaxis abstracta uno (NSA.1)*, Libro Azul, fascículo VIII.4, Rec. X.209, UIT, Ginebra, 1988.
- [5] Recomendación del CCITT *Definición del servicio de sesión para la interconexión de sistemas abiertos para aplicaciones del CCITT*, Libro Azul, fascículo VIII.4, Rec. X.215, UIT, Ginebra, 1988.
- [6] Recomendación del CCITT *Definición del servicio de presentación para la interconexión de sistemas abiertos para aplicaciones del CCITT*, Libro Azul, fascículo VIII.4, Rec. X.216, UIT, Ginebra, 1988.
- [7] Recomendación del CCITT *Definición del servicio de control de asociación para la interconexión de sistemas abiertos para aplicaciones del CCITT*, Libro Azul, fascículo VIII.4, Rec. X.217, UIT, Ginebra, 1988.
- [8] Recomendación del CCITT *Operaciones a distancia: Modelo, notación y definición del servicio*, Libro Azul, fascículo VIII.4, Rec. X.219, UIT, Ginebra, 1988.
- [9] Recomendación del CCITT *Especificación del protocolo de presentación para la interconexión de sistemas abiertos para aplicaciones del CCITT*, Libro Azul, fascículo VIII.5, Rec. X.226, UIT, Ginebra, 1988.
- [10] Recomendación del CCITT *Especificación del protocolo de control de asociación para la interconexión de sistemas abiertos para aplicaciones del CCITT*, Libro Azul, fascículo VIII.5, Rec. X.227, UIT, Ginebra, 1988.
- [11] Recomendación del CCITT *Operaciones a distancia: Especificación del protocolo*, Libro Azul, fascículo VIII.5, Rec. X.229, UIT, Ginebra, 1988.
- [12] Recomendación del CCITT *Definición del servicio común de información de gestión para aplicaciones del CCITT*, Rec. X.710, UIT, Ginebra, 1991.

3 Definiciones

A los efectos de esta Recomendación se emplean las definiciones siguientes.

3.1 *Definiciones del modelo de referencia básico*

En esta Recomendación se utilizan los siguientes términos definidos en la Recomendación X.200 [1]:

- a) elemento de servicio de aplicación;
- b) proceso de aplicación;
- c) sistema abierto real;
- d) gestión de sistemas.

3.2 *Definiciones del marco de gestión (management framework)*

En esta Recomendación se utilizan los siguientes términos definidos en la norma ISO/CEI 7498-4 [2]:

- a) objeto gestionado;
- b) información de gestión;

Reemplazada por una versión más reciente

- c) base de información de gestión;
- d) entidad de aplicación de gestión de sistemas.

3.3 *Definiciones de operaciones a distancia*

En esta Recomendación se utilizan los siguientes términos definidos en la Recomendación X.219 [8]:

- a) iniciador de asociación;
- b) respondedor de asociación;
- c) operaciones ligadas;
- d) operaciones a distancia;
- e) elemento de servicio de operaciones a distancia;
- f) invocador;
- g) realizador;
- h) clase de asociación;
- i) clase de operación.

3.4 *Definiciones del servicio común de información de gestión (SCIG)*

En esta Recomendación se utilizan los siguientes términos definidos en la Recomendación X.710 [12]:

- a) atributo;
- b) elemento de servicio común de información de gestión;
- c) servicio común de información de gestión;
- d) proveedor del servicio ESCIG;
- e) usuario del servicio ESCIG;
- f) usuario invocador del servicio ESCIG;
- g) usuario realizador del servicio ESCIG.

3.5 *Definiciones de elemento de servicio de control de asociación (ESCA)*

En esta Recomendación se utilizan los siguientes términos definidos en la Recomendación X.217 [7]:

- a) contexto de aplicación;
- b) asociación de aplicación;
- c) asociación.

3.6 Definiciones de presentación

En esta Recomendación se utilizan los siguientes términos definidos en la Recomendación X.216 [6]:

- a) sintaxis abstracta;
- b) sintaxis de transferencia.

4 **Símbolos y abreviaturas**

ACSE	ESCA	Elemento de servicio de control de asociación
APDU	UDPA	Unidad de datos de protocolo de aplicación
ASE	ESA	Elemento de servicio de aplicación
ASN.1	NSA.1	Notación de sintaxis abstracta uno

Reemplazada por una versión más reciente

CMIP	PCIG	Protocolo común de información de gestión
CMIPM	MPCIG	Máquina de protocolo común de información de gestión
CMIS	SCIG	Servicio común de información de gestión
CMISE	ESCIG	Elemento de servicio común de información de gestión
DCS	CCD	Conjunto de contexto definido
PCI	ICP	Información de control de protocolo
PDU	UDP	Unidad de datos de protocolo
PICS	ECRP	Enunciado de conformidad de realización de protocolo
RO	OD	Operaciones a distancia
ROSE	ESOD	Elemento de servicio de operaciones a distancia
SMAE	EAGS	Entidad de aplicación de gestión de sistemas

5 Visión de conjunto

El protocolo común de información de gestión (PCIG) especifica elementos de protocolo que pueden utilizarse para proporcionar los servicios de operaciones y notificación descritos en la Recomendación X.710 [12], que define los servicios comunes de información de gestión (SCIG).

5.1 Servicio prestado

El protocolo especificado en esta Recomendación soporta los servicios especificados en la Recomendación X.710 [12]. Estos servicios se resumen en el cuadro 1/X.711.

CUADRO 1/X.711

Servicios comunes de información de gestión

Servicio	Tipo
G-CANCELACIÓN-OBTENCIÓN	Confirmado
G-INFORME-EVENTO	Confirmado/no confirmado
G-OBTENCIÓN	Confirmado
G-FIJACIÓN	Confirmado/no confirmado
G-ACCIÓN	Confirmado/no confirmado
G-CREACIÓN	Confirmado
G-SUPRESIÓN	Confirmado

Reemplazada por una versión más reciente

5.2 *Servicios subyacentes*

En esta Recomendación se utilizan los servicios OD-INVOCACIÓN, OD-RESULTADO, OD-ERROR y OD-RECHAZO-U del elemento de servicio de operaciones a distancia (ESOD) definido en la Recomendación X.219 [8]. El ESOD conlleva la utilización del servicio de presentación definido en la Recomendación X.216 [6]. Las operaciones confirmadas del PCIG son operaciones de clase 2 (asíncronas) u operaciones de clase 1 (síncronas), según lo requiera la aplicación. La elección de la clase de operaciones es un asunto local. Las operaciones no confirmadas del PCIG son las operaciones de clase 5 (asíncronas, resultado no comunicado). El PCIG utiliza asociaciones de clase 3.

Si la unidad funcional servicio ampliado se negocia satisfactoriamente, las unidades de datos de protocolo de aplicación del ESOD pueden hacerse corresponder con otros servicios de presentación distintos del servicio P-DATOS.

Nota – Por ejemplo, puede ser necesario modificar el CCD definido por la presentación cuando la operación PCIG se envía al usuario del servicio ESCIG. En este caso, la UDPA de ESOD que transporta la operación PCIG se hará corresponder con el servicio P-MODIFICACIÓN-CONTEXTO que también se utiliza para realizar los cambios del CCD.

En la descripción del contexto de aplicación utilizado en la asociación se dan detalles de los otros servicios de presentación que se necesitan y de la manera de utilizarlos.

5.2.1 *Servicios que se supone prestará el ESCA*

En esta Recomendación se supone la utilización de los servicios A-ASOCIACIÓN, A-LIBERACIÓN, A-ABORTO, y A-P-ABORTO del elemento de servicio de control de asociación.

5.2.2 *Servicios de la capa de presentación cuya utilización se supone*

En la Recomendación X.229 [11] se supone la utilización del servicio P-DATOS de la capa de presentación para la transferencia de las UDP OD-INVOCACIÓN, OD-RESULTADO, OD-ERROR y OD-RECHAZO.

5.3 *Definiciones de información de gestión*

En esta Recomendación se define la sintaxis abstracta del protocolo común de información de gestión. Los atributos específicos de un determinado objeto gestionado se indican en la Recomendación en que se define dicho objeto.

6 **Elementos de procedimiento**

En este punto se definen los elementos de procedimiento del PCIG. Los procedimientos definen la transferencia de las UDP de PCIG cuya estructura, codificación y relación con las primitivas de servicio del SCIG se especifican en el § 7.

La máquina de protocolo común de información de gestión (MPCIG) acepta primitivas de servicio petición y respuesta del SCIG, y emite UDP de PCIG que inician elementos de procedimiento específicos descritos en este punto.

Una MPCIG aceptará cualquier UDP de PCIG bien formada y la pasará al usuario realizador del servicio ESCIG para su procesamiento, por medio de las primitivas de servicio indicación y confirmación del SCIG. Si la UDP recibida no está bien formada, o no contiene una notificación u operación soportada, se devuelve una UDP que indica que la UDP recibida ha sido rechazada.

Reemplazada por una versión más reciente

Los procedimientos indican solamente cómo deben interpretarse los diversos campos de la UDP del PCIG, y no lo que un usuario invocador del servicio ESCIG debe hacer con la información que pide, ni cómo un usuario realizador del servicio ESCIG debe procesar la invocación.

6.1 *Establecimiento de la asociación*

El establecimiento de una asociación comprende dos usuarios del servicio ESCIG, uno que es el iniciador de la asociación y el otro que es el respondedor de la asociación.

Un usuario del servicio ESCIG puede iniciar un establecimiento de asociación utilizando el servicio A-ASOCIACIÓN de la Recomendación X.217 [7].

El contexto de aplicación específica, entre otras cosas, las reglas requeridas para la coordinación de la información de iniciación correspondiente a distintos ESA. Las reglas de asociación del ESCIG se especifican en el anexo A.

6.2 *Operaciones a distancia*

6.2.1 *Elementos de procedimiento de operaciones a distancia*

Los elementos de procedimiento del PCIG dependen de los siguientes elementos de procedimiento de operaciones a distancia subyacentes:

- a) invocación;
- b) retorno de resultado;
- c) retorno de error;
- d) rechazo de usuario;
- e) rechazo de proveedor.

Estos elementos de procedimiento se describen detalladamente en la Recomendación X.229 [11].

En el cuadro 2/X.711 se especifica la correspondencia entre los parámetros SCIG y ESOD.

CUADRO 2/X.711

Correspondencia entre los parámetros SCIG y ESOD

Parámetro SCIG	Parámetro ESOD
Identificador de invocación	ID de invocación
Identificador de ligazón	ID de ligazón

La correspondencia entre otros parámetros SCIG y ESOD se especifica en el § 7.

Reemplazada por una versión más reciente

6.2.2 *Parámetros de problema OD-RECHAZO*

Los parámetros de problema OD-RECHAZO se hacen corresponder o se procesan como sigue:

6.2.2.1 La correspondencia de OD-rechazo por usuario.Problema de invocación (RO-Reject-User.Invoke-problem) con códigos de error SCIG se indican en el cuadro 3/711.

CUADRO 3/X.711

Correspondencia de OD-Rechazo-usuario. Invocación-problema con códigos de error ESCIG

Parámetro OD- RECHAZO	Código de error ESCIG
duplicate-invocation	invocación duplicada
mistyped-argument	argumento mal tipificado
resource-limitation	limitación de recurso
unrecognized-operation	operación no reconocida

Los demás parámetros de problema de invocación son un asunto local.

6.2.2.2 Los demás parámetros OD-RECHAZO se tratarán como un asunto local.

6.3 *Procedimiento de informe de evento*

6.3.1 *Invocación*

Los procedimientos de informe de evento son iniciados por la primitiva Petición G-INFORME-EVENTO.

Al recibir la primitiva Petición G-INFORME-EVENTO, la MPCIG:

- a) construirá en el modo confirmado, una UDPA que solicite la operación, g-Informe-Evento-Confirmado (m-EventReport-Confirmed), y en caso contrario, construirá una UDPA que solicite la operación g-Informe-Evento (m-EventReport);
- b) enviará la UDPA empleando el procedimiento OD-INVOCACIÓN.

6.3.2 *Recepción*

Al recibir una UDPA que solicita la operación m-EventReport o m-EventReport-Confirmed, la MPCIG emitirá, si la UDPA está bien formada, una primitiva Indicación G-INFORME-EVENTO al usuario, servicio-ESCIG, con el parámetro modo indicando si se solicita o no confirmación, y en caso contrario, construirá una UDPA que contenga una notificación del error y la enviará empleando el procedimiento OD-RECHAZO-U.

Reemplazada por una versión más reciente

6.3.3 *Respuesta*

En el modo confirmado, la MPCIG aceptará una primitiva Respuesta G-INFORME-EVENTO, y:

- a) construirá una UDPA que confirme la notificación G-INFORME-EVENTO;
- b) si los parámetros de la primitiva Respuesta G-INFORME-EVENTO indican que la notificación ha sido aceptada, enviará la UDPA empleando el procedimiento OD-RESULTADO, y en caso contrario, enviará la UDPA empleando el procedimiento OD-ERROR.

6.3.4 *Recepción de respuesta*

Al recibir una UDPA que responde a una notificación G-INFORME-EVENTO, la MPCIG emitirá, si la UDPA está bien formada, una primitiva Confirmación G-INFORME-EVENTO al usuario del servicio ESCIG, completando así el procedimiento de notificación, y en caso contrario, construirá una UDPA que contenga una notificación del error y la enviará empleando el procedimiento OD- RECHAZO-U.

6.4 *Procedimiento de obtención*

6.4.1 *Invocación*

Los procedimientos de obtención son iniciados por la primitiva Petición G-OBTENCIÓN.

Al recibir la primitiva Petición G-OBTENCIÓN, la MPCIG:

- a) construirá una UDPA que solicite la operación g-Obtención (m-Get);
- b) enviará la UDPA empleando el procedimiento OD-INVOCACIÓN.

6.4.2 *Recepción*

Al recibir una UDPA que solicita la operación g-Obtención, la MPCIG enviará, si la UDPA está bien formada, una primitiva Indicación G-OBTENCIÓN al usuario del servicio ESCIG, y en caso contrario, construirá una UDPA que contiene una notificación del error y la enviará empleando el procedimiento OD-RECHAZO-U.

6.4.3 *Respuesta*

La MPCIG:

- a) aceptará cero o más primitivas Respuesta G-OBTENCIÓN que contengan un ID de ligazón seguido por una sola primitiva Respuesta G-OBTENCIÓN sin ID de ligazón;
- b) para cada primitiva Respuesta G-OBTENCIÓN que contiene un ID de ligazón:
 - construirá una UDPA que solicite la operación g-Respuesta-Ligada (m-Linked-Reply) con LinkedReplyArgument fijado de manera apropiada a getListError, a getResult o a processingFailure;
 - enviará cada UDPA empleando el procedimiento OD-INVOCACIÓN;
- c) para la primitiva Respuesta G-OBTENCIÓN que no contenga un ID de ligazón, la MPCIG:
 - construirá una UDPA que confirme la operación g-Obtención;
 - si los parámetros de la primitiva Respuesta G-OBTENCIÓN indican que la operación ha sido realizada correctamente, enviará la UDPA empleando el procedimiento OD-RESULTADO. Si los parámetros de la primitiva Respuesta G-OBTENCIÓN indican que la operación ha obtenido un éxito parcial, o no se ha realizado a causa de un error, enviará la UDPA empleando el procedimiento OD-ERROR.

Reemplazada por una versión más reciente

6.4.4 *Recepción de respuesta*

Al recibir una UDPA que responde a una operación g-Obtención, la MPCIG:

- a) si la UDPA incluye un ID de ligazón y, está bien formada, enviará una primitiva Confirmación G-OBTENCIÓN al usuario del servicio ESCIG;
- b) si la UDPA es la última respuesta (es decir que no contiene un ID de ligazón) y está bien formada, enviará una primitiva Confirmación G-OBTENCIÓN al usuario-servicio-ESCIG, completando así el procedimiento G-OBTENCIÓN;
- c) si la UDPA no está bien formada, construirá una UDPA que contiene una notificación del error y la enviará empleando el procedimiento OD-RECHAZO-U.

6.4.5 *Procedimiento de cancelación de obtención*

6.4.5.1 *Invocación*

Los procedimientos de cancelación de obtención (CancelGet) son iniciados por la primitiva Petición G-CANCELACIÓN-OBTENCIÓN.

Al recibir la primitiva Petición G-CANCELACIÓN-OBTENCIÓN, la MPCIG:

- a) construirá una UDPA que solicite la operación g-Cancelación de Obtención;
- b) enviará la UDPA empleando el procedimiento OD-INVOCACIÓN.

6.4.5.2 *Recepción*

Al recibir una UDPA que solicita la operación g-Cancelación de Obtención, la MPCIG enviará, si la UDPA está bien formada una primitiva Indicación G-CANCELACIÓN-OBTENCIÓN al usuario-servicio-ESCIG, y en caso contrario, construirá una UDPA que contenga una notificación del error y la enviará empleando el procedimiento OD-RECHAZO-U.

6.4.5.3 *Respuesta*

La MPCIG:

- a) construirá una UDPA que confirme la operación g-Cancelación de Obtención;
- b) si los parámetros de la primitiva Respuesta G-CANCELACIÓN- OBTENCIÓN indican que la operación se ejecutó correctamente, enviará la UDPA utilizando el procedimiento OD-RESULTADO; en otro caso, enviará la UDPA utilizando el procedimiento OD-ERROR. Si la operación G-OBTENCIÓN tiene éxito, el usuario realizador del servicio ESCIG dejará de enviar respuestas ligadas a la operación G-OBTENCIÓN y enviará una primitiva Respuesta G-OBTENCIÓN que contendrá el error «operación cancelada».

6.4.5.4 *Recepción de respuesta*

Al recibir una UDPA que responde a una operación g-Cancelación de Obtención, la MPCIG enviará, si la UDPA está bien formada, una primitiva Confirmación G-CANCELACIÓN-OBTENCIÓN al usuario del servicio ESCIG, y en caso contrario, construirá una UDPA que contenga una notificación del error y la enviará empleando el procedimiento OD-RECHAZO-U.

6.5 *Procedimiento de fijación*

6.5.1 *Invocación*

Los procedimientos de fijación son iniciados por la primitiva Petición G-FIJACIÓN.

Al recibir la primitiva Petición G-FIJACIÓN, la MPCIG:

Reemplazada por una versión más reciente

- a) en el modo confirmado, construirá una UDPA que solicite la operación g-Fijación-Confirmado (m-Set-Confirmed), y en caso contrario, construirá una UDPA que solicita la operación g-Fijación;
- b) enviará la UDPA empleando el procedimiento OD-INVOCACIÓN.

6.5.2 *Recepción*

Al recibir una UDPA que solicita la operación g-Fijación o g-Fijación-Confirmado, la MPCIG enviará, si la UDPA está bien formada, una primitiva Indicación G-FIJACIÓN al usuario del servicio ESCIG, con el parámetro modo que indica si se solicita o no confirmación, y en caso contrario, construirá una UDPA que contiene una notificación del error y la enviará empleando el procedimiento OD-RECHAZO-U.

6.5.3 *Respuesta*

En el modo confirmado, la MPCIG:

- a) aceptará cero o más primitivas Respuesta G-FIJACIÓN que contengan un ID de ligazón seguido por una sola primitiva Respuesta G-FIJACIÓN sin ID de ligazón;
- b) para cada primitiva Respuesta G-FIJACIÓN que contiene un ID de ligazón:
 - construirá una UDPA que solicita la operación g-Respuesta-Ligada (mLinkedReply) con LinkedReplyArgument fijado de manera apropiada a setListError, setResult o processingFailure;
 - enviará las UDPA empleando el procedimiento OD-INVOCACIÓN;
- c) para la primitiva Respuesta G-FIJACIÓN que no contiene ID de ligazón:
 - construirá una UDPA que confirma la operación g-Fijación;
 - si los parámetros de la primitiva Respuesta G-FIJACIÓN indican que la operación ha sido realizada correctamente, enviará la UDPA empleando el procedimiento OD-RESULTADO. Si los parámetros de la primitiva Respuesta G-FIJACIÓN indican que la operación ha obtenido un éxito parcial, o no se ha realizado a causa de un error, enviará la UDPA empleando el procedimiento OD-ERROR.

6.5.4 *Recepción de respuesta*

Al recibir una UDPA que responde a una operación G-Fijación-Confirmado, la MPCIG:

- a) si la UDPA incluye un ID de ligazón y está bien formada, enviará una primitiva Confirmación G-FIJACIÓN al usuario del servicio ESCIG;
- b) si la UDPA es la última respuesta (es decir, no contiene un ID de ligazón) y está bien formada, enviará una primitiva Confirmación G-FIJACIÓN al usuario del servicio ESCIG, completando así el procedimiento G-FIJACIÓN;
- c) si la UDPA no está bien formada, construirá una UDPA que contiene una notificación del error y la enviará empleando el procedimiento OD-RECHAZO-U.

6.6 *Procedimiento de acción*

6.6.1 *Invocación*

Los procedimientos de acción son iniciados por la primitiva Petición G-ACCIÓN.

Al recibir la primitiva Petición G-ACCIÓN, la MPCIG:

Reemplazada por una versión más reciente

- a) en el modo confirmado, construirá una UDPA que solicita la operación g-Acción-Confirmada (m-ActionConfirmed), y en caso contrario, construirá una UDPA que solicita la operación g-Acción (m-Action);
- b) enviará la UDPA empleando el procedimiento OD-INVOCACIÓN.

6.6.2 *Recepción*

Al recibir una UDPA que solicita la operación g-Acción o g-Acción-Confirmada (m-Action-Confirmed), la MPCIG enviará, si la UDPA está bien formada, una primitiva Indicación G-ACCIÓN al usuario del servicio ESCIG, con el parámetro modo que indica si se solicita o no confirmación, y en caso contrario, construirá una UDPA que contiene la notificación del error y la enviará empleando el procedimiento OD-RECHAZO-U.

6.6.3 *Respuesta*

En el modo confirmado, la MPCIG:

- a) aceptará cero o más primitivas Respuesta G-ACCIÓN que contengan un ID de ligazón seguido por una sola primitiva Respuesta G-ACCIÓN sin ID de ligazón;
- b) para cada primitiva Respuesta G-ACCIÓN que contiene un ID de ligazón:
 - construirá una UDPA que solicite la operación g-Respuesta-Ligada con LinkedReplyArgument fijado de manera apropiada a actionError, actionResult o processingFailure;
 - enviará las UDPA empleando el procedimiento OD-INVOCACIÓN;
- c) para la primitiva Respuesta G-ACCIÓN que no contiene ID de ligazón:
 - construirá una UDPA que confirme la operación g-Acción;
 - si los parámetros de la primitiva Respuesta G-ACCIÓN indican que la operación ha sido realizada correctamente, enviará la UDPA empleando el procedimiento OD-RESULTADO, y en caso contrario, enviará la UDPA empleando el procedimiento OD-ERROR.

6.6.4 *Recepción de respuesta*

Al recibir una UDPA que responde a una operación g-Acción-Confirmada, la MPCIG:

- a) si la UDPA incluye un ID-enlazado y está bien formada, enviará una primitiva Confirmación G-ACCIÓN al usuario del servicio ESCIG;
- b) si la UDPA es la última respuesta (es decir, no contiene un ID de ligazón) y está bien formada, enviará una primitiva Confirmación G-ACCIÓN al usuario del servicio ESCIG, completando así el procedimiento G-ACCIÓN;
- c) si la UDPA no está bien formada, construirá una UDPA que contiene una notificación del error y la enviará empleando el procedimiento OD-RECHAZO-U.

6.7 *Procedimiento de creación*

6.7.1 *Invocación*

Los procedimientos de creación son iniciados por la primitiva Petición G-CREACIÓN.

Al recibir la primitiva Petición G-CREACIÓN, la MPCIG:

- a) construirá una UDPA que solicita la operación M-CREATE;
- b) enviará la UDPA empleando el procedimiento OD-INVOCACIÓN.

Reemplazada por una versión más reciente

6.7.2 *Recepción*

Al recibir una UDPA que solicita la operación g-Creación (m-Create), la MPCIG, si la UDPA está bien formada, enviará una primitiva Indicación G-CREACIÓN al usuario del servicio ESCIG, y en caso contrario, construirá una UDPA que contenga una notificación del error y la enviará empleando el procedimiento OD-RECHAZO-U.

6.7.3 *Respuesta*

La MPCIG aceptará una primitiva Respuesta G-CREACIÓN y:

- a) construirá una UDPA que confirma la operación g-Creación;
- b) si los parámetros de la primitiva Respuesta G-CREACIÓN indican que la operación ha sido realizada correctamente, enviará la UDPA empleando el procedimiento OD-RESULTADO, y en caso contrario, enviará la UDPA empleando el procedimiento OD-ERROR.

6.7.4 *Recepción de respuesta*

Al recibir una UDPA que responde a una operación g-Creación, la MPCIG enviará, si la UDPA está bien formada, una primitiva Confirmación G-CREACIÓN al usuario del servicio ESCIG, completando así el procedimiento G-CREACIÓN, y en caso contrario, construirá una UDPA que contiene una notificación del error y la enviará empleando el procedimiento OD-RECHAZO-U.

6.8 *Procedimiento de supresión*

6.8.1 *Invocación*

Los procedimientos de supresión son iniciados por la primitiva Petición G-SUPRESIÓN.

Al recibir la primitiva Petición G-SUPRESIÓN, la MPCIG:

- a) construirá una UDPA que solicita la operación g-Supresión (m-Delete);
- b) enviará la UDPA empleando el procedimiento OD-INVOCACIÓN.

6.8.2 *Recepción*

Al recibir una UDPA que solicita la operación g-Supresión, la MPCIG enviará, si la UDPA está bien formada, una primitiva Indicación G-SUPRESIÓN al usuario del servicio ESCIG, y en caso contrario, construirá una UDPA que contenga una notificación del error y la enviará empleando el procedimiento OD-RECHAZO-U.

6.8.3 *Respuesta*

La MPCIG:

- a) aceptará cero o más primitivas Respuesta G-SUPRESIÓN que contengan un ID de ligazón seguido por una sola primitiva Respuesta G-SUPRESIÓN sin ID de ligazón;
- b) para cada primitiva Respuesta G-SUPRESIÓN que contiene un ID de ligazón:
 - construirá una UDPA que solicita la operación g-Respuesta-Ligada con Linked-Reply con LinkedReplyArgument fijado de manera apropiada a deleteError, deleteResult o processingFailure;
 - enviará las UDPA empleando el procedimiento OD-INVOCACIÓN.
- c) para la primitiva Respuesta G-SUPRESIÓN que no contiene ID de ligazón:
 - construirá una UDPA que confirme la operación g-Supresión;
 - si los parámetros de la primitiva Respuesta G-SUPRESIÓN indican que la operación ha sido realizada correctamente, enviará la UDPA empleando el procedimiento OD-RESULTADO, y en caso contrario, enviará la UDPA empleando el procedimiento OD-ERROR.

Reemplazada por una versión más reciente

6.8.4 *Recepción de respuesta*

Al recibir una UDPA que responde a una operación g-Supresión, la MPCIG:

- a) si la UDPA incluye un ID-enlazado, emitirá una primitiva Confirmación G-SUPRESIÓN al usuario del servicio ESCIG;
- b) si la UDPA es la última respuesta (es decir, no contiene un ID de ligazón) y está bien formada, enviará una primitiva Confirmación G-SUPRESIÓN al usuario del servicio ESCIG completando así el procedimiento G-SUPRESIÓN;
- c) si la UDPA no está bien formada, construirá una UDPA que contiene una notificación del error y la enviará empleando el procedimiento OD-RECHAZO-U.

6.9 *Liberación ordenada de asociación*

Cualquiera de los dos usuarios del servicio ESCIG puede iniciar una liberación ordenada de la asociación utilizando el servicio A-LIBERACIÓN de la Recomendación X.217 [7].

Nota – Esta especificación es diferente de la relativa a la utilización por el ESOD de la operación VINCULACIÓN, en la cual únicamente el iniciador de la asociación puede utilizar el procedimiento A-LIBERACIÓN.

6.10 *Liberación abrupta de asociación*

Cualquiera de los dos usuarios del servicio ESCIG puede iniciar una liberación abrupta de la asociación utilizando el servicio A-ABORTO de la Recomendación X.217 [7].

El proveedor del servicio ESCIG puede iniciar una liberación abrupta de la asociación utilizando el servicio A-ABORTO de la Recomendación X.217 [7].

7 **Sintaxis abstracta**

En este punto se especifica la sintaxis abstracta para las UDP de PCIG.

7.1 *Convenios*

La sintaxis abstracta se define utilizando la notación especificada en la Recomendación X.208 [3]. Las producciones MACRO de NSA.1 utilizadas en esta Recomendación, o citadas en la misma, no emplean los aspectos ambiguos de la gramática.

Para cada uno de los parámetros de servicio ESCIG que han de ser transferidos por una UDP de PCIG, hay un campo de UDP (un ASN.1 NamedType) con el mismo nombre que el parámetro de servicio correspondiente (véase la Recomendación X.710 [12]), salvo las diferencias requeridas por la utilización de la NSA.1, que consisten en suprimir los espacios entre las palabras y escribir con mayúscula la primera letra de la palabra siguiente, por ejemplo, «managed object class» (clase de objeto gestionado) se convierte en «managedObjectClass». Algunos de los nombres se abrevian de la manera siguiente:

ack acknowledgement (acuse de recibo)

arg argument (argumento)

id identifier (identificador)

info information (información)

sync synchronization (sincronización).

Reemplazada por una versión más reciente

7.2 Correspondencia entre primitivas de ESCIG y operaciones de PCIG

CUADRO 4/X.711

Correspondencia entre primitivas de ESCIG y operaciones de PCIG

Primitiva SCIG	Modo	ID de ligazón	Operación PCIG
Pet/Ind G-CANCELACIÓN-OBTENCIÓN	Confirmado	No aplicable	g-Cancelación-Obtención-Confirmado
Rsp/Conf G-CANCELACIÓN-OBTENCIÓN	No aplicable	No aplicable	g-Cancelación-Obtención-Confirmado
Pet/Ind G-INFORME-EVENTO	No confirmado	No aplicable	g-Informe-Evento
Pet/Ind G-INFORME-EVENTO	Confirmado	No aplicable	g-Informe-Evento-Confirmado
Rsp/Conf G-INFORME-EVENTO	No aplicable	No aplicable	g-Informe-Evento-Confirmado
Pet/Ind G-OBTENCIÓN	Confirmado	No aplicable	g-Obtención
Rsp/Conf G-OBTENCIÓN	No aplicable	Ausente	g-Obtención
Rsp/Conf G-OBTENCIÓN	No aplicable	Presente	g-Respuesta-Ligada
Pet/Ind G-FIJACIÓN	No confirmado	No aplicable	g-Fijación
Pet/Ind G-FIJACIÓN	Confirmado	No aplicable	g-Fijación-Confirmado
Rsp/Conf G-FIJACIÓN	No aplicable	Ausente	g-Fijación-Confirmado
Rsp/Conf G-FIJACIÓN	No aplicable	Presente	g-Respuesta-Ligada
Pet/Ind G-ACCIÓN	No confirmado	No aplicable	g-Acción
Pet/Ind G-ACCIÓN	Confirmado	No aplicable	g-Acción-Confirmado
Rsp/Conf G-ACCIÓN	No aplicable	Ausente	g-Acción-Confirmado
Rsp/Conf G-ACCIÓN	No aplicable	Presente	g-Respuesta-Ligada
Pet/Ind G-CREACIÓN	Confirmado	No aplicable	g-Creación
Rsp/Conf G-CREACIÓN	No aplicable	No aplicable	g-Creación
Pet/Ind G-SUPRESIÓN	Confirmado	No aplicable	g-Supresión
Rsp/Conf G-SUPRESIÓN	No aplicable	Ausente	g-Supresión
Rsp/Conf G-SUPRESIÓN	No aplicable	Presente	g-Respuesta-Ligada

Nota – La correspondencia de las macros OPERATION (Operación) y ERROR con el ESOD es la definida en la Recomendación X.219 [8].

Reemplazada por una versión más reciente

7.3 Datos de usuario de ESCA

El protocolo de ESCA (Recomendación X.227 [10]) se describe utilizando la NSA.1. La «información de usuario» se define utilizando el tipo de datos EXTERNAL.

7.3.1 Datos de usuario de A-ASOCIACIÓN

La codificación de la información de usuario PCIG que ha de pasarse a A-ASOCIACIÓN en el parámetro «información de usuario» se define de la manera siguiente:

CMIP-A-ASSOCIATE-Information {joint-iso-ccitt ms(9) cmip(1) modules(0) aAssociateUserInfo(1)}

DEFINITIONS ::= BEGIN

FunctionalUnits ::= BIT STRING {
 multipleObjectSelection (0),
 filter (1),
 multipleReply (2),
 extendedService (3),
 cancelGet (4)
}

-- La unidad funcional *i* es soportada única y exclusivamente si el bit *i* es uno

-- Información transportada en el parámetro información de usuario de A-ASOCIACIÓN

CMIPUserInfo ::= SEQUENCE {
 protocolVersion [0] IMPLICIT ProtocolVersion DEFAULT { version1 },
 functionalUnits [1] IMPLICIT FunctionalUnits DEFAULT {},
 accessControl [2] EXTERNAL OPTIONAL,
 userInfo [3] EXTERNAL OPTIONAL
}

ProtocolVersion ::= BIT STRING {
 version1 (0),
 version2 (1)
}

Nota – La presente Recomendación especifica la versión de protocolo 2 para compatibilidad técnica con ISO/CEI 9596-1: 1991.

END

La codificación de otra «información de usuario» suministrada por el usuario del servicio ESCIG no se define en esta Recomendación.

7.3.2 Datos de usuario de A-ABORTO

La codificación de la información de usuario PCIG que ha de pasarse a A-ABORTO en el parámetro «información de usuario» se define de la manera siguiente:

CMIP-A-ABORT-Information {joint-iso-ccitt-ms(9) cmip(1) modules(0) aAbortUserInfo(2)}

DEFINITIONS ::= BEGIN

-- Información transportada en el parámetro «información de usuario» de A-ABORTO

CMIPAbortInfo ::= SEQUENCE {
 abortSource [0] IMPLICIT CMIPAbortSource,
 userInfo [1] EXTERNAL OPTIONAL
}

Reemplazada por una versión más reciente

```
CMIPAbortSource ::= ENUMERATED {  
    cmiseServiceUser          (0),  
    cmiseServiceProvider      (1)  
}
```

END

La codificación de otra «información de usuario» suministrada por el usuario del servicio ESCIG no se define en esta Recomendación.

7.4 Unidades de datos PCIG

El protocolo se describe en términos de unidades de datos de protocolo común de información de gestión intercambiadas entre entidades pares del ESCIG. Las UDP se especifican utilizando la NSA.1 y las macros externas OPERATION y ERROR del Protocolo de Operaciones a Distancia, definidas en la Recomendación X.219 [8].

-- *Protocolo común de información de gestión (PCIG)*

```
CMIP-1 {joint-iso-ccitt-ms(9) cmip(1) modules(0) protocol(3)}
```

```
DEFINITIONS ::= BEGIN
```

-- *Definiciones de operaciones a distancia*

```
IMPORTS OPERATION, ERROR FROM Remote-Operation-Notation {joint-iso-ccitt remoteOperations(4)  
notation(0)}
```

-- *Definiciones del servicio de operaciones a distancia*

```
InvokeIDType FROM Remote-Operations-APDUs {joint-iso-ccitt remoteOperations(4) apdus(1)}
```

-- *Definiciones del servicio de guía*

```
DistinguishedName, RDNSequence FROM InformationFramework {joint-iso-ccitt ds(5) modules(1)  
informationFramework(1)};
```

-- *Operaciones ESCIG*

-- *en las siguientes operaciones, el tipo de argumento es obligatorio en la ADPU del ESOD (ROSE APDU) correspondientes*

-- *Operaciones de acción (M-ACTION)*

```
m-Action OPERATION  
    ARGUMENT ActionArgument  
    ::= localValue 6
```

```
m-Action-Confirmed OPERATION  
    ARGUMENT      ActionArgument  
    RESULT        ActionResult
```

-- *Este resultado es condicional; para las condiciones, véase el § 8.3.3.2.9 de la Recomendación X.710*

```
ERRORS {  
    accessDenied, classInstanceConflict, complexityLimitation, invalidScope, invalidArgumentValue,  
    invalidFilter, noSuchAction, noSuchArgument, noSuchObjectClass, noSuchObjectInstance,  
    processingFailure, syncNotSupported  
}
```

```
LINKED { m-Linked-Reply }  
    ::= localValue 7
```

```
m-CancelGet OPERATION  
    ARGUMENT  
    getInvokeld      InvokeIDType
```

Reemplazada por una versión más reciente

RESULT

```
ERRORS {  mistypedOperation, noSuchInvokeld, processingFailure }  
::= localValue 10
```

-- *Operación de creación (M-CREATE)*

m-Create OPERATION

```
ARGUMENT      CreateArgument  
RESULT        CreateResult
```

-- *este resultado es condicional; para las condiciones véase el § 8.3.4.1.3 de la Recomendación X.710*

ERRORS {

```
accessDenied, classInstanceConflict, duplicateManagedObjectInstance, invalidAttributeValue,  
invalidObjectInstance, missingAttributeValue, noSuchAttribute, noSuchObjectClass,  
noSuchObjectInstance, noSuchReferenceObject, processingFailure  
}  
::= localValue 8
```

-- *Operación de supresión (M-DELETE)*

m-Delete OPERATION

```
Argument      DeleteArgument  
RESULT        DeleteResult
```

-- *este resultado es condicional; para las condiciones, véase el § 8.3.3.5.2.8 de la Recomendación X.710*

ERRORS {

```
accessDenied, classInstanceConflict, complexityLimitation, invalidFilter,  
InvalidScope, noSuchObjectClass, noSuchObjectInstance, processingFailure, syncNotSupported  
}
```

LINKED { m-Linked-Reply }

```
::= localValue 9
```

-- *Operaciones de informe de evento (M-EVENT-REPORT)*

m-EventReport OPERATION

```
ARGUMENT      EventReportArgument  
::= localValue 0
```

m-EventReport-Confirmed OPERATION

```
ARGUMENT      EventReportArgument  
RESULT        EventReportResult -- optional
```

ERRORS {

```
invalidArgumentValue, noSuchArgument, noSuchEventType, noSuchObjectClass,  
noSuchObjectInstance, processingFailure  
}  
::= localValue 1
```

-- *Operación de obtención (M-GET)*

m-Get OPERATION

```
ARGUMENT      GetArgument  
RESULT        GetResult
```

-- *este resultado es condicional; para las condiciones, véase el § 3.3.1.2.8 de la Recomendación X.710*

ERRORS {

```
accessDenied, classInstanceConflict, complexityLimitation, getListError, invalidFilter, invalidScope,  
noSuchObjectClass, noSuchObjectInstance, operationCancelled, processingFailure,  
syncNotSupported  
}
```

Reemplazada por una versión más reciente

LINKED { m-Linked-Reply }
::= localValue 3

-- Operaciones ligadas a M-GET, M-SET (Confirmed), M-ACTION (Confirmed), y M-DELETE

m-Linked-Reply OPERATION
ARGUMENT LinkedReplyArgument
::= localValue 2

-- Operaciones de fijación (M-SET)

m-Set OPERATION
ARGUMENT SetArgument
::= localValue 4

m-Set-Confirmed OPERATION
ARGUMENT SetArgument
RESULT SetResult

-- este resultado es condicional; para las condiciones, véase el § 8.3.2.2.9 de la Recomendación X.710

ERRORS {
accessDenied, classInstanceConflict, complexityLimitation, invalidFilter, invalidScope,
noSuchObjectClass, noSuchObjectInstance, processingFailure, setListError, syncNotSupported
}

LINKED { m-Linked-Reply }
::= localValue 5

-- Definiciones de error del SCIG (CMIS)

-- en los errores siguientes, a menos que se indique otra cosa, el tipo de parámetro es obligatorio

-- en la UDPA ESOD (ROSE APDU)

accessDenied ERROR
::= localValue 2

classInstanceConflict ERROR
PARAMETER BaseManagedObjectId
::= localValue 19

complexityLimitation ERROR
PARAMETER ComplexityLimitation -- optional
::= localValue 20

duplicateManagedObjectInstance ERROR
PARAMETER ObjectInstance
::= localValue 11

getListError ERROR
PARAMETER GetListError
::= localValue 7

invalidArgumentValue ERROR
PARAMETER InvalidArgumentValue
::= localValue 15

invalidAttributeValue ERROR
PARAMETER Attribute
::= localValue 6

invalidFilter ERROR
PARAMETER CMISFilter
::= localValue 4

Reemplazada por una versión más reciente

invalidObjectInstance ERROR
PARAMETER ObjectInstance
::= localValue 17

invalidScope ERROR
PARAMETER Scope
::= localValue 16

missingAttributeValue ERROR
PARAMETER SET OF Attributeld
::= localValue 18

mistypedOperation ERROR
::= localValue 21

noSuchAction ERROR
PARAMETER NoSuchAction
::= localValue 9

noSuchArgument ERROR
PARAMETER NoSuchArgument
::= localValue 14

noSuchAttribute ERROR
PARAMETER Attributeld
::= localValue 5

noSuchEventType ERROR
PARAMETER NoSuchEventType
::= localValue 13

noSuchInvokeld ERROR
PARAMETER InvokeldType
::= localValue 22

noSuchObjectClass ERROR
PARAMETER ObjectClass
::= localValue 0

noSuchObjectInstance ERROR
PARAMETER ObjectInstance
::= localValue 1

noSuchReferenceObject ERROR
PARAMETER ObjectInstance
::= localValue 12

operationCancelled ERROR
::= localValue 23

processingFailure ERROR
PARAMETER ProcessingFailure -- optional
::= localValue 10

setListError ERROR
PARAMETER SetListError
::= localValue 8

syncNotSupported ERROR
PARAMETER CMISync
::= localValue 3

Reemplazada por una versión más reciente

-- *Definiciones de soporte de tipo*

```
AccessControl ::= EXTERNAL

ActionArgument ::= SEQUENCE {
    COMPONENTS OF
    accessControl
    synchronization
    scope
    filter
    actionInfo
}

ActionError ::= SEQUENCE {
    managedObjectClass
    managedObjectInstance
    currentTime
    actionErrorInfo
}

ActionErrorInfo ::= SEQUENCE {
    errorStatus
    errorInfo
}

errorStatus ::= ENUMERATED {
    accessDenied
    noSuchAction
    noSuchArgument
    invalidArgumentValue
},

errorInfo ::= CHOICE {
    actionType
    actionArgument
    argumentValue
}

ActionInfo ::= SEQUENCE {
    actionType
    actionInfoArg
}

ActionReply ::= SEQUENCE {
    actionType
    actionReplyInfo
}

ActionResult ::= SEQUENCE {
    managedObjectClass
    managedObjectInstance
    currentTime
    actionReply
}

ActionType ::= CHOICE {
    globalForm
    localForm
}
```

-- *Esta Recomendación no asigna valores para localForm. Cuando se utilice esta alternativa, los valores admisibles para los enteros y sus significados se definirán como parte del contexto de aplicación en el que se utilizan*

```
Attribute ::= SEQUENCE {
    attributeld
    attributeValue
}
```

Reemplazada por una versión más reciente

AttributeError	::= SEQUENCE {
errorStatus	ENUMERATED {
accessDenied	(2),
noSuchAttribute	(5),
invalidAttributeValue	(6),
invalidOperation	(24),
invalidOperator	(25)
},	
modifyOperator	[2] IMPLICIT ModifyOperator OPTIONAL,
	-- presente para invalidOperator & invalidOperation
attributeld	Attributeld,
attributeValue	ANY DEFINED BY attributeld OPTIONAL
	-- absent for setToDefault
}	
Attributeld	::= CHOICE {
globalForm	[0] IMPLICIT OBJECT IDENTIFIER,
localForm	[1] IMPLICIT INTEGER
}	

-- Esta Recomendación no asigna valores para *localForm*. Cuando se utiliza esta alternativa, los valores admisibles para los enteros y sus significados se definirán como parte del contexto de aplicación en que se utilizan

AttributeldError	::= SEQUENCE {
errorStatus	ENUMERATED {
accessDenied	(2),
noSuchAttribute	(5) },
attributeld	Attributeld
}	
BaseManagedObjectId	::= SEQUENCE {
baseManagedObjectClass	ObjectClass,
baseManagedObjectInstance	ObjectInstance
}	
CMISFilter	::= CHOICE {
item	[8] FilterItem,
and	[9] IMPLICIT SET OF CMISFilter,
or	[10] IMPLICIT SET OF CMISFilter,
not	[11] CMISFilter
}	
CMISync	::= ENUMERATED {
bestEffort	(0),
atomic	(1)
}	
ComplexityLimitation	::= SET {
scope	[0] Scope OPTIONAL,
filter	[1] CMISFilter OPTIONAL,
sync	[2] CMISync OPTIONAL
}	
CreateArgument	::= SEQUENCE {
managedObjectClass	ObjectClass,

Reemplazada por una versión más reciente

CHOICE {	
managedObjectInstance	ObjectInstance,
superiorObjectInstance	[8] ObjectInstance } OPTIONAL,
accessControl	[5] AccessControl OPTIONAL,
referenceObjectInstance	[6] ObjectInstance OPTIONAL,
attributeList	[7] IMPLICIT SET OF Attribute OPTIONAL
}	
CreateResult	::= SEQUENCE {
managedObjectClass	ObjectClass OPTIONAL,
managedObjectInstance	ObjectInstance OPTIONAL,
	-- shall be returned if omitted from CreateArgument
currentTime	[5] IMPLICIT GeneralizedTime OPTIONAL,
attributeList	[6] IMPLICIT SET OF Attribute OPTIONAL
}	
DeleteArgument	::= SEQUENCE {
COMPONENTS OF	BaseManagedObjectId,
accessControl	[5] AccessControl OPTIONAL,
synchronization	[6] IMPLICIT CMISSync DEFAULT bestEffort,
scope	[7] Scope DEFAULT baseObject,
filter	[8] CMISFilter DEFAULT and {}
}	
DeleteError	::= SEQUENCE {
managedObjectClass	ObjectClass OPTIONAL,
managedObjectInstance	ObjectInstance OPTIONAL,
currentTime	[5] IMPLICIT GeneralizedTime OPTIONAL,
deleteErrorInfo	[6] ENUMERATED { accessDenied (2)
}}	
DeleteResult	::= SEQUENCE {
managedObjectClass	ObjectClass OPTIONAL,
managedObjectInstance	ObjectInstance OPTIONAL,
currentTime	[5] IMPLICIT GeneralizedTime OPTIONAL
}	
EventReply	::= SEQUENCE {
eventType	EventTypeId,
eventReplyInfo	[8] ANY DEFINED BY eventType OPTIONAL
}	
EventReportArgument	::= SEQUENCE {
managedObjectClass	ObjectClass,
managedObjectInstance	ObjectInstance,
eventTime	[5] IMPLICIT GeneralizedTime OPTIONAL,
eventType	EventTypeId,
eventInfo	[8] ANY DEFINED BY eventType OPTIONAL
}	
EventReportResult	::= SEQUENCE {
managedObjectClass	ObjectClass OPTIONAL,
managedObjectInstance	ObjectInstance OPTIONAL,
currentTime	[5] IMPLICIT GeneralizedTime OPTIONAL,
eventReply	EventReply OPTIONAL
}	
EventTypeId	::= CHOICE {
globalForm	[6] IMPLICIT OBJECT IDENTIFIER,
localForm	[7] IMPLICIT INTEGER
}	

Reemplazada por una versión más reciente

-- Esta Recomendación no asigna valores para localForm. Cuando se utilice esta alternativa, los valores admisibles para los enteros y sus significados se definirán como parte del contexto de aplicación en que se utilizan

FilterItem	::= CHOICE {
equality	[0] IMPLICIT Attribute,
substrings	[1] IMPLICIT SEQUENCE OF CHOICE {
initialString	[0] IMPLICIT SEQUENCE {
	attributeld Attributeld,
	string ANY DEFINED BY attributeld },
anyString	[1] IMPLICIT SEQUENCE {
	attributeld Attributeld,
	string ANY DEFINED BY attributeld },
finalString	[2] IMPLICIT SEQUENCE {
	attributeld Attributeld,
	string ANY DEFINED BY attributeld} },
greaterOrEqual	[2] IMPLICIT Attribute, -- <i>asserted value ≥ attribute value</i>
lessOrEqual	[3] IMPLICIT Attribute, -- <i>asserted value ≥ attribute value</i>
present	[4] Attributeld,
subsetOf	[5] IMPLICIT Attribute, -- <i>asserted values es un subconjunto de attribute value</i>
supersetOf	[6]IMPLICIT Attribute, -- <i>asserted value es un superconjunto de attribute value</i>
nonNullSetIntersection	[7] IMPLICIT Attribute
}	
GetArgument	::= SEQUENCE {
COMPONENTS OF	BaseManagedObjectId,
accessControl	[5] AccessControl OPTIONAL,
synchronization	[6] IMPLICIT CMISSync DEFAULT bestEffort,
scope	[7] Scope DEFAULT baseObject,
filter	CMISFilter DEFAULT and {},
attributeldList	[12] IMPLICIT SET OF Attributeld OPTIONAL
}	
GetInfoStatus	::= CHOICE {
attributeldError	[0] IMPLICIT AttributeldError,
attribute	[1] IMPLICIT Attribute
}	
GetListError	::= SEQUENCE {
managedObjectClass	ObjectClass OPTIONAL,
managedObjectInstance	ObjectInstance OPTIONAL,
currentTime	[5] IMPLICIT GeneralizedTime OPTIONAL,
getInfoList	[6] IMPLICIT SET OF GetInfoStatus
}	
GetResult	::= SEQUENCE {
managedObjectClass	ObjectClass OPTIONAL,
managedObjectInstance	ObjectInstance OPTIONAL,
currentTime	[5] IMPLICIT GeneralizedTime OPTIONAL,
attributeList	[6] IMPLICIT SET OF Attribute OPTIONAL
}	

Reemplazada por una versión más reciente

InvalidArgumentValue actionValue eventValue }	::= CHOICE { [0] IMPLICIT ActionInfo, [1] IMPLICIT SEQUENCE { eventType EventTyped, eventInfo [8] ANY DEFINED BY eventType OPTIONAL } }
LinkedReplyArgument getResult getListError setResult setListError actionResult processingFailure deleteResult actionError deleteError }	::= CHOICE { [0] IMPLICIT GetResult, [1] IMPLICIT GetListError, [2] IMPLICIT SetResult, [3] IMPLICIT SetListError, [4] IMPLICIT ActionResult, [5] IMPLICIT ProcessingFailure, [6] IMPLICIT DeleteResult, [7] IMPLICIT ActionError, [8] IMPLICIT DeleteError }
ModifyOperator replace addValues removeValues setToDefault }	::= INTEGER { (0), (1), (2), (3) }
NoSuchAction managedObjectClass actionType }	::= SEQUENCE { ObjectClass, ActionTypeld }
NoSuchArgument actionId eventId }	::= CHOICE { [0] IMPLICIT SEQUENCE { managedObjectClass ObjectClass OPTIONAL, actionType ActionTypeld }, [1] IMPLICIT SEQUENCE { managedObjectClass ObjectClass OPTIONAL, eventType EventTyped } }
NoSuchEventType managedObjectClass eventType }	::= SEQUENCE { ObjectClass, EventTypeld }
ObjectClass globalForm localForm }	::= CHOICE { [0] IMPLICIT OBJECT IDENTIFIER, [1] IMPLICIT INTEGER }

-- *Esta Recomendación no asigna valores para forma local. Cuando se utiliza esta alternativa, los valores admisibles para los enteros y sus significados se definirán como parte del contexto de aplicación en que se utilizan*

ObjectInstance distinguishedName nonSpecificForm localDistinguishedName }	::= CHOICE { [2] IMPLICIT DistinguishedName, [3] IMPLICIT OCTET STRING, [4] IMPLICIT RDNSSequence }
--	---

Reemplazada por una versión más reciente

-- *localDistinguishedName* es la porción del nombre distinguido que es necesaria para identificar inequívocamente el objeto gestionado dentro del contexto de comunicación entre los sistemas abiertos

```
ProcessingFailure ::= SEQUENCE {
    managedObjectClass      ObjectClass,
    managedObjectInstance  ObjectInstance OPTIONAL,
    specificErrorInfo      [5] SpecificErrorInfo
}

Scope ::= CHOICE { INTEGER {
    baseObject              (0),
    firstLevelOnly         (1),
    wholeSubtree           (2) },
    [1] IMPLICIT INTEGER, -- POSITIVE entero indica el nivel
                        -- que ha de seleccionarse
    [2] IMPLICIT INTEGER } -- POSITIVE entero N indica que
                        -- hay que seleccionar la gama de
                        -- niveles (0 - N)
    baseToNthLevel
```

-- con *individualLevels* y *baseToNthLevel*, un valor de 0 tiene la misma semántica que *baseObject*

-- con *individualLevels*, un valor de 1 tiene la misma semántica que *firstLevelOnly*

```
SetArgument ::= SEQUENCE {
    COMPONENTS OF
    accessControl          [5] AccessControl OPTIONAL,
    synchronization       [6] IMPLICIT CMISSync DEFAULT bestEffort,
    scope                  [7] Scope DEFAULT baseObject,
    filter                  CMISFilter DEFAULT and { },
    modificationList      [12] IMPLICIT SET OF SEQUENCE {
    modifyOperator         [2] IMPLICIT ModifyOperator DEFAULT replace,
    attributeld            Attributeld,
    attributeValue        ANY DEFINED BY attributeld OPTIONAL -- absent for
                        setToDefault
    }
}

SetInfoStatus ::= CHOICE {
    attributeError        [0] IMPLICIT AttributeError,
    attribute              [1] IMPLICIT Attribute
}

ListError ::= SEQUENCE {
    managedObjectClass    ObjectClass OPTIONAL,
    managedObjectInstance ObjectInstance OPTIONAL,
    currentTime           [5] IMPLICIT GeneralizedTime OPTIONAL,
    setInfoList           [6] IMPLICIT SET OF SetInfoStatus
}

SetResult ::= SEQUENCE {
    managedObjectClass    ObjectClass OPTIONAL,
    managedObjectInstance ObjectInstance OPTIONAL,
    currentTime           [5] IMPLICIT GeneralizedTime OPTIONAL,
    attributeList         [6] IMPLICIT SET OF Attribute OPTIONAL
}

SpecificErrorInfo ::= SEQUENCE {
    errorId               OBJECT IDENTIFIER,
    errorInfo             ANY DEFINED BY errorId
}

END -- Fin de definiciones de sintaxis PCIG
```

Reemplazada por una versión más reciente

7.5 *Definición de sintaxis abstracta para PCIG*

Esta Recomendación asigna el valor de identificador de objeto de NSA.1

{joint-iso-ccitt ms(9) cmip(1) cmip-pci(1) abstractSyntax(4)}

como un nombre de sintaxis abstracta para el conjunto de valores de datos de presentación, cada uno de los cuales es un valor del tipo NSA.1

Remote-Operations-APDUs.ROSEapdus

definido en la Recomendación X.229 [11] con el componente de argumento relleno de acuerdo con las definiciones en PCIG-1, o un valor de uno de los tipos NSA.1

- **CMIP-A-ASSOCIATE-Information.CMIPUserInfo;**

- **CMIP-A-ABORT-Information.CMIPAbortInfo.**

El correspondiente valor de descriptor de objeto será

“CMIP-PCI”.

Esta sintaxis abstracta se define para incluir todos los tipos de datos resueltos por las producciones ANY DEFINED BY X, donde X es de tipo OBJECT IDENTIFIER.

Los valores de identificador de objeto y de descriptor de objeto

{joint-iso-ccitt asn1(1) basic-encoding(1)} and “Basic Encoding of single ASN.1 type”

(asignados a un objeto en la Recomendación X.229 [4]) pueden utilizarse como un nombre de sintaxis de transferencia con esta sintaxis abstracta.

7.5.1 *Reglas de extensibilidad*

7.5.1.1 *Cuando se procesa Información A-ASOCIACIÓN de PCIG entrante, la MPCIG*

- ignorará todos los valores rotulados («tagged») que no están definidos en la sintaxis abstracta de esta Recomendación; e
- ignorará todas las asignaciones de nombre de bit desconocidas dentro de una BIT STRING.

7.5.1.2 El nombre de sintaxis abstracta puede utilizarse cuando se modifican los valores de datos de presentación para incluir

- operaciones de gestión de un nuevo sistema;
- nuevos elementos rotulados dentro de un SET o SEQUENCE;
- nuevas asignaciones de nombre de bit dentro de una BIT STRING;
- nuevos números denominados para un INTEGER; y
- nuevas enumeraciones denominadas dentro de un ENUMERATED.

8 **Conformidad**

Un sistema que afirma aplicar los procedimientos especificados en esta Recomendación cumplirá los requisitos indicados en los § 8.1 y 8.2.

8.1 *Requisitos estáticos*

El sistema:

- a) soportará la unidad funcional kernel (o de núcleo) definida en la Recomendación X.710 [12], y las facilidades implicadas por dicha unidad funcional;
- b) soportará la sintaxis de transferencia derivada de las reglas de codificación especificadas en la Recomendación X.209 [4] y denominada:

Reemplazada por una versión más reciente

{joint-iso-ccitt asn1(1) basic-encoding(1)}

a los efectos de generar e interpretar información de protocolo ESCIG definida por la sintaxis abstracta

“CMIP-PCI”

para las unidades funcionales soportadas;

- c) soportará el protocolo ESCA definido en la Recomendación X.227 [10], para establecer y liberar una asociación;
- d) soportará las reglas especificadas en el anexo A en cualquier contexto de aplicación que incluya ESCIG como uno de los ESA;
- e) soportará la clase 3 de protocolo ESOD definida en la Recomendación X.229 [11];
- f) soportará la unidad funcional de respuestas múltiples si se soporta la unidad funcional de selección de objetos múltiples.

8.2 *Requisitos dinámicos*

El sistema:

- a) seguirá los procedimientos correspondientes a cada unidad funcional que el sistema afirma aplicar;
- b) verificará los parámetros de seguridad facultativos definidos en las UDP de PCIG, cuando los utilice;
- c) cuando se soporta la unidad funcional de servicio ampliada, soportará el protocolo de presentación definido en la Recomendación X.226 [9], según lo requiera el contexto de aplicación;
- d) cuando se proporcione determinación de ámbito («scoping»), soportará la unidad funcional de respuestas múltiples.

ANEXO A

(Este anexo no forma parte de esta Recomendación)

Reglas de asociación para el ESCIG

A.1 *ESCA, requisitos de sesión y presentación*

A.1.1 El ESCIG requiere la unidad funcional de presentación kernel definida en la Recomendación X.216 [6].

A.1.2 El ESCIG requiere las unidades funcionales kernel y de sesión dúplex definidas en la Recomendación X.215 [5].

A.1.3 El ESCIG requiere el modo normal de ESCA y los servicios de presentación definidos en las Recomendaciones X.227 [10] y X.216 [6].

A.2 *Reglas de iniciación de asociación*

A.2.1 *Petición*

El usuario del servicio ESCIG que inicia el establecimiento de asociación proporcionará la información de usuario A-ASOCIACIÓN definida en la Recomendación X.710 [12]. La información de usuario PCIG se proporcionará a la MPCIG, la cual:

Reemplazada por una versión más reciente

- a) construirá CMIPUserInfo (información de usuario PCIG) a partir de la información suministrada;
- b) fijará el parámetro de versión de protocolo de la información de usuario PCIG fijando el bit correspondiente a cada versión soportada;
- c) incluirá la información de usuario PCIG como un EXTERNAL separado en el parámetro información de usuario de la primitiva Petición A-ASOCIACIÓN;
- d) esperará a que la información de usuario específica del SCIG sea devuelta en la primitiva Confirmación A-ASOCIACIÓN.

A.2.2 *Indicación*

Al recibir una primitiva Indicación A-ASOCIACIÓN, se proporcionará el parámetro información de usuario PCIG a la MPCIG, la cual:

- a) comprobará que se soporta por lo menos una de las versiones de protocolo propuestas;
- b) verificará que el parámetro de control de acceso facultativo es válido;
- c) si fracasa cualquiera de las comprobaciones, se rechazará la asociación fijando el motivo del parámetro de fallo de la primitiva Respuesta A-ASOCIACIÓN a «rechazado por respondedor(permanente)». La asociación no se establece y esa instancia (o manifestación) de la MPCIG dejará de existir;
- d) si estas comprobaciones tienen éxito, la información siguiente, si está presente, se facilitará al usuario de servicio ESCIG: unidades funcionales soportadas por el proveedor del servicio ESCIG, control de acceso e información de usuario. La MPCIG esperará la respuesta del usuario del servicio ESCIG.

A.2.3 *Respuesta*

La primitiva Respuesta A-ASOCIACIÓN que indica «aceptado» o «rechazado», y que, si es aceptada, incluye las unidades funcionales y los parámetros control de acceso e información de usuario, se proporcionará a la MPCIG, la cual:

- a) construirá la CMIPUserInfo (información de usuario del PCIG) requerida para la respuesta. La información de usuario del PCIG incluirá el parámetro de versión que indica todas las versiones de PCIG que son soportadas;
- b) incluirá la información de usuario del PCIG como un EXTERNAL separado en el parámetro de información de usuario de la primitiva Respuesta A-ASOCIACIÓN;
- c) si la respuesta de asociación indica «aceptada», la versión de protocolo acordada es la correspondiente al número más alto soportado por ambas MPCIG. La MPCIG estará preparada entonces para aceptar primitivas Indicación ESCIG;
- d) si la respuesta de asociación indica «rechazada», la instancia de la MPCIG dejará de existir.

A.2.4 *Confirmación*

Al recibir la primitiva Confirmación A-ASOCIACIÓN, se proporcionará el parámetro información de usuario del PCIG a la MPCIG, la cual:

- a) si la confirmación de asociación indica éxito, la asociación se establece y se proporcionan al iniciador de la asociación las unidades funcionales, y los parámetros control de acceso e información de usuario. Las unidades funcionales acordadas corresponden a aquellas que ambos usuarios del servicio ESCIG han indicado que soportan y la versión de protocolo es el número de versión más alto soportado por ambas MPCIG;
- b) si la confirmación de asociación indica fallo, no se establece la asociación y la instancia de la MPCIG dejará de existir.

A.3 *Reglas de liberación de asociación*

Cualquiera de los dos usuarios del servicio ESCIG puede iniciar una liberación de asociación.

Reemplazada por una versión más reciente

A.3.1 *Petición*

Al recibir una petición de liberación de asociación, se proporcionarán los parámetros de A-LIBERACIÓN necesarios a la MPCIG, la cual dejará de aceptar peticiones de servicio y esperará confirmación de la liberación de la asociación.

A.3.2 *Indicación*

Al recibir una primitiva Indicación A-LIBERACIÓN, se proporcionarán los parámetros de Indicación A-LIBERACIÓN necesarios a la MPCIG que responde, la cual esperará la respuesta de la liberación de asociación.

A.3.3 *Respuesta*

Al recibir una respuesta de liberación de asociación del usuario del servicio ESCIG que responde, se proporcionarán los parámetros de Respuesta A-LIBERACIÓN necesarios a la MPCIG que responde. Después de esto, esta instancia de la MPCIG dejará de existir.

A.3.4 *Confirmación*

Al recibir una primitiva Confirmación A-LIBERACIÓN, se proporcionarán los parámetros de Confirmación de A-LIBERACIÓN necesarios a la MPCIG iniciadora. Después de esto, esta instancia de la MPCIG dejará de existir.

A.4 *Reglas de aborto de asociación*

Cualquiera de los dos usuarios del servicio ESCIG puede iniciar una terminación abrupta de la asociación.

En función de la información local, si los servicios subyacentes no disponen de la capacidad de transportar información de usuario ilimitada por A-ABORTO, el parámetro CMIPAbortInfo puede no estar incluido en las primitivas de servicio A-ABORTO.

A.4.1 *Petición A-ABORTO*

Al recibir una petición de aborto de asociación, se proporcionarán los parámetros de Petición A-ABORTO necesarios, que incluyen la información de usuario A-ABORTO definida en la Recomendación X.710 [12], a la MPCIG, la cual:

- a) construirá CMIPAbortInfo (información de aborto del PCIG) a partir de la información proporcionada;
- b) fijará el parámetro fuente de aborto de la información de aborto del PCIG a usuario del servicio ESCIG;
- c) incluirá información de aborto del PCIG como un campo separado en el parámetro Información de usuario de la primitiva Petición A-ABORTO;
- d) después de esto, esta instancia de la MPCIG dejará de existir.

A.4.2 *Indicación A-ABORTO*

Al recibir una primitiva Indicación A-ABORTO, se proporcionarán a la MPCIG los parámetros Indicación A-ABORTO necesarios que incluyen CMIPAbortInfo (información de aborto del PCIG). Después de esto, esta instancia de la MPCIG dejará de existir.

A.4.3 *Indicación A-P-ABORTO*

Al recibir una primitiva Indicación A-P-ABORTO, se proporcionarán a la MPCIG los parámetros de Indicación A-P-ABORTO necesarios. Después de esto, esta instancia de la MPCIG dejará de existir.

Reemplazada por una versión más reciente

A.4.4 Error de protocolo PCIG

Al detectar un error de protocolo, la MPCIG:

- a) construirá la información de aborto del PCIG con el parámetro fuente de aborto fijado a proveedor del servicio ESCIG;
- b) indicará al usuario del servicio ESCIG que se ha producido un error de protocolo;
- c) incluirá la información de aborto del PCIG como un campo separado en el parámetro información de usuario de la primitiva Petición A-ABORTO;
- d) después de esto, esta instancia de la MPCIG dejará de existir.

ANEXO B

(Este anexo no forma parte de esta Recomendación)

Sintaxis NSA.1 ampliada

En este anexo se describe cómo las macros OPERATION y ERROR de la Recomendación X.219 [8] se amplían a tipos y subtipos de datos NSA.1.

Si existen incoherencias entre estas definiciones y las del § 7, tendrán precedencia estas últimas.

-- *Protocolo común de información de gestión (PCIG)*

CMIP-1 {joint-iso-ccitt ms(9) cmip(1) modules(0) protocol(3)}

DEFINITIONS ::= BEGIN

-- *Definiciones de operaciones a distancia*

IMPORTS OPERATION, ERROR FROM Remote-Operation-Notation {joint-iso-ccitt remoteOperations(4) notation(0)}

-- *Definiciones de servicio de guía*

DistinguishedName, RDNSSequence FROM InformationFramework {joint-iso-ccitt ds(5) modules(1) informationFramework(1)};

-- *Operaciones ESCIG*

```
ROSEapdu                               := CHOICE {
    roiv-apdu                             [1] IMPLICIT ROIVapdu,
    rors-apdu                             [2] IMPLICIT RORSapdu,
    roer-apdu                             [3] IMPLICIT ROERapdu,
    rorj-apdu                             [4] IMPLICIT RORJapdu
}

ROIVapdu                                ::= SEQUENCE {
    invokeID                               InvokeIDType,
    linked-ID                             [0] IMPLICIT InvokeIDType OPTIONAL,
    operation-value                       OPERATION,
    argument                             ANY DEFINED BY operation-value OPTIONAL
}

RORSapdu                                ::= SEQUENCE {
    invokeID                               InvokeIDType,
    SEQUENCE {                             operation-value     OPERATION,
                                           result               ANY DEFINED BY
                                                         operation-value } OPTIONAL
}
}
```

Reemplazada por una versión más reciente

```
ROERapdu ::= SEQUENCE {
  invokeID      InvokeIDType,
  error-value   ERROR,
  parameter     ANY DEFINED BY error-value OPTIONAL
}
```

```
RORJapdu ::= SEQUENCE {
  invokeID      CHOICE {
    problem     CHOICE {
      [0] IMPLICIT GeneralProblem,
      [1] IMPLICIT InvokeProblem,
      [2] IMPLICIT ReturnResultProblem,
      [3] IMPLICIT ReturnErrorProblem }
    }
}
```

```
InvokeIDType ::= INTEGER
```

-- La utilización de los códigos *GeneralProblem*, *ReturnResultProblem* y *ReturnErrorProblem* es un asunto local

```
GeneralProblem ::= INTEGER {
  unrecognisedAPDU      (0), -- ROSE-provider detected
  mistypedAPDU          (1),
  badlyStructuredAPDU   (2)
}
```

```
InvokeProblem ::= INTEGER {
  duplicateInvocation      (0), -- ROSE-user detected
  unrecognisedOperation   (1),
  mistypedArgument        (2),
  resourceLimitation      (3),
  initiatorReleasing      (4),
  unrecognisedLinkedID    (5),
  linkedResponseUnexpected (6),
  unexpectedChildOperation (7)
}
```

```
ReturnResultProblem ::= INTEGER {
  unrecognisedInvocation (0) -- ROSE-user detected
  resultResponseUnexpected (1),
  mistypedResult         (2)
}
```

```
ReturnErrorProblem ::= INTEGER {
  unrecognisedInvocation (0), -- ROSE-user detected
  errorResponseUnexpected (1),
  unrecognisedError      (2),
  unexpectedError        (3),
  mistypedParameter      (4)
}
```

-- Esta parte de la especificación NSA.1 ofrece una definición del subtipo *InvokeProblem* utilizado por el PCIG

```
InvokeProblem-CMIPUser ::= InvokeProblem (
  duplicateInvocation |
  unrecognisedOperation |
  mistypedArgument |
  resourceLimitation
)
```

Reemplazada por una versión más reciente

- Esta parte de la especificación de NSA.1 ofrece una definición de los subtipos de ROIVapdu y RORSapdu utilizados por el PCIG. Los subtipos de la ROIVapdu (udpaODIV) definen los valores permitidos del valor de operación y el argumento definido por ese valor de operación para todas las notificaciones y operaciones del PCIG. Los subtipos de la RORSapdu (udpaODRS) definen los valores permitidos del valor de operación y el resultado definido por ese valor de operación para todas las notificaciones y operaciones del PCIG

m-Action OPERATION	::= localValue 6
ROIV-m-Action { invokeID linked-ID operation-value argument	::= ROIVapdu (WITH COMPONENTS PRESENT, ABSENT, (m-Action), (INCLUDES ActionArgument) })
m-Action-Confirmed OPERATION	::= localValue 7
ROIV-m-Action-Confirmed { invokeID linked-ID operation-value argument	::= ROIVapdu (WITH COMPONENTS PRESENT, ABSENT, (m-Action-Confirmed), (INCLUDES ActionArgument) })
RORS-m-Action-Confirmed { ... , invokeID <i>-- secuencia de resultado -- (WITH COMPONENTS</i> { operation-value result <i>-- requerido únicamente si hay una sola respuesta a la ROIV-m-Action-Confirmed ROIVapdu y deben devolverse datos en la RORSapdu</i> })	::= RORSapdu (WITH COMPONENTS PRESENT, (m-Action-Confirmed), (INCLUDES ActionResult) })
m-Cancel-Get OPERATION	::= localValue 10
ROIV-m-Cancel-Get { invokeID linked-ID operation-value argument })	::= ROIVapdu (WITH COMPONENTS PRESENT, ABSENT, (m-Cancel-Get), (INCLUDES InvokeIDType)
RORS-m-Cancel-Get { invokeID <i>-- no hay secuencia de resultado para RORS-m-Cancel-Get</i> })	::= RORSapdu (WITH COMPONENTS PRESENT,
m-Create OPERATION	::= localValue 8
ROIV-m-Create { invokeID linked-ID operation-value argument	::= ROIVapdu (WITH COMPONENTS PRESENT, ABSENT, (m-Create), (INCLUDES CreateArgument) })
RORS-m-Create { ... , invoke-ID <i>-- secuencia de resultado -- (WITH COMPONENTS</i> { operation-value result })	::= RORSapdu (WITH COMPONENTS PRESENT, (m-Create), (INCLUDES CreateResult) })

Reemplazada por una versión más reciente

m-Delete OPERATION	::= localValue 9
ROIV-m-Delete { invokeID linked-ID operation-value argument	::= ROIVapdu (WITH COMPONENTS PRESENT, ABSENT, (m-Delete), (INCLUDES DeleteArgument) })
RORS-m-Delete { ... , invokeID <i>-- secuencia de resultado -- (WITH COMPONENTS</i> { operation-value result <i>-- requerido únicamente si hay una sola respuesta a la ROIV-m-DeleteROIVapdu y deben devolverse datos en la RORSapdu</i> })	::= RORSapdu (WITH COMPONENTS PRESENT, (m-Delete), (INCLUDES DeleteResult) })
m-EventReport OPERATION	::= localValue 0
ROIV-m-EventReport { invokeID linked-ID operation-value argument	::= ROIVapdu (WITH COMPONENTS PRESENT, ABSENT, (m-EventReport), (INCLUDES EventReportArgument) })
m-EventReport-Confirmed OPERATION	::= localValue 1
ROIV-m-EventReport-Confirmed { invokeID linked-ID operation-value argument	::= ROIVapdu (WITH COMPONENTS PRESENT, ABSENT, (m-EventReport-Confirmed), (INCLUDES EventReportArgument) })
RORS-m-EventReport-Confirmed::= RORSapdu (WITH COMPONENTS { ... , invokeID <i>-- secuencia de resultado -- (WITH COMPONENTS</i> { operation-value result <i>-- requerido únicamente si deben devolverse datos en la RORSapdu</i> })	PRESENT, (m-EventReport-Confirmed), (INCLUDES EventReportResult) })
m-Get OPERATION	::= localValue 3
ROIV-m-Get { invokeID linked-ID operation-value argument	::= ROIVapdu (WITH COMPONENTS PRESENT, ABSENT, (m-Get), (INCLUDES GetArgument) })
RORS-m-Get { ... , invokeID <i>-- secuencia de resultado -- (WITH COMPONENTS</i> { operation-value result <i>-- requerido únicamente si hay una sola respuesta a la ROIV-m-Get ROIVapdu</i> })	::= RORSapdu (WITH COMPONENTS PRESENT, (m-Get), (INCLUDES GetResult) })

Reemplazada por una versión más reciente

m-Linked-Reply OPERATION	::= localValue 2
ROIV-m-Linked-Reply	::= ROIVapdu (WITH COMPONENTS
{ invokeID	PRESENT,
linked-ID	PRESENT,
operation-value	(m-Linked-Reply),
argument	(INCLUDES LinkedReplyArgument) })

-- Esta parte de la especificación NSA.1 ofrece una definición de los subtipos ROIV-m-Linked-Reply utilizados por el PCIG. Los subtipos de la ROIV-m-Linked-Reply ROIVapdu definen los valores permitidos del argumento definido por el valor de operación para las operaciones de respuesta ligadas PCIG específicas.

ROIV-m-Linked-Reply-Action	::= ROIV-m-Linked-Reply (WITH COMPONENTS
{ invokeID	PRESENT,
linked-ID	PRESENT,
operation-value	(m-Linked-Reply),
argument	(INCLUDES LinkedReplyArgument (WITH COMPONENTS
	getResult ABSENT,
	getListError ABSENT,
	setResult ABSENT,
	setListError ABSENT,
	actionResult PRESENT,
	processingFailure PRESENT,
	deleteResult ABSENT,
	actionError PRESENT,
	deleteError ABSENT)
))

ROIV-m-Linked-Reply-Delete	::= ROIV-m-Linked-Reply (WITH COMPONENTS
{ invokeID	PRESENT,
linked-ID	PRESENT,
operation-value	(m-Linked-Reply),
argument	(INCLUDES LinkedReplyArgument (WITH COMPONENTS
	getResult ABSENT,
	getListError ABSENT,
	setResult ABSENT,
	setListError ABSENT,
	actionResult ABSENT,
	processingFailure PRESENT,
	deleteResult PRESENT,
	actionError ABSENT,
	deleteError PRESENT)
))

ROIV-m-Linked-Reply-Get	::= ROIV-m-Linked-Reply (WITH COMPONENTS
{ invokeID	PRESENT,
linked-ID	PRESENT,
operation-value	(m-Linked-Reply),
argument	(INCLUDES LinkedReplyArgument (WITH COMPONENTS
	getResult PRESENT,
	getListError PRESENT,
	setResult ABSENT,
	setListError ABSENT,
	actionResult ABSENT,
	processingFailure PRESENT,
	deleteResult ABSENT,
	actionError ABSENT,
	deleteError ABSENT)
))

Reemplazada por una versión más reciente

```

ROIV-m-Linked-Reply-Set ::= ROIV-m-Linked-Reply (WITH COMPONENTS
{ invokeID
linked-ID
operation-value
argument
getResult ABSENT,
getListError ABSENT,
setResult PRESENT
setListError PRESENT,
actionResult ABSENT,
processingFailure PRESENT,
deleteResult ABSENT,
actionError ABSENT,
deleteError ABSENT )
})

```

```

m-Set OPERATION ::= localValue 4

```

```

ROIV-m-Set ::= ROIVapdu (WITH COMPONENTS
{ invokeID
linked-ID
operation-value
argument
ABSENT,
(m-Set),
(INCLUDES SetArgument) } )

```

```

m-Set-Confirmed OPERATION ::= localValue 5

```

```

ROIV-m-Set-Confirmed ::= ROIVapdu (WITH COMPONENTS
{ invokeID
linked-ID
operation-value
argument
ABSENT,
(m-Set-Confirmed),
(INCLUDES SetArgument) } )

```

```

RORS-m-Set-Confirmed ::= RORSapdu (WITH COMPONENTS
{ ... ,
invokeID PRESENT,
-- secuencia de resultado -- (WITH COMPONENTS
{ operation-value (m-Set-Confirmed),
result (INCLUDES SetResult) } )
-- requerido únicamente si hay una sola respuesta a la ROIV-m-Set-Confirmed ROIVapdu y hay que
devolver datos en la RORSapdu
} )

```

-- Esta parte de la especificación NSA.1 ofrece una definición de los subtipos de la ROERapdu utilizados por el PCIG. Los subtipos de la ROERapdu definen los valores permitidos del valor y del parámetro de error definidos por ese valor de error para todas las notificaciones y operaciones del PCIG.

```

accessDenied ERROR ::= localValue 2

```

```

ROER-accessDenied ::= ROERapdu (WITH COMPONENTS
{ invokeID
error-value
PRESENT,
(accessDenied) } )

```

-- Esta ROERapd sólo puede devolverse en respuesta a las ROIV-m-Get,,ROIV-m-Set-Confirmed, ROIV-m-Action-Confirmed, ROIV-m-Create y ROIV-m-Delete ROIVapdus

```

classInstanceConflict ERROR ::= localValue 19

```

```

ROER-classInstanceConflict ::= ROERapdu (WITH COMPONENTS
{ invokeID
error-value
parameter
PRESENT,
(classInstanceConflict),
(INCLUDES BaseManagedObjectId) } )

```

Reemplazada por una versión más reciente

-- Esta ROERapdu sólo puede devolverse en respuesta a las ROIV-m-Get, ROIV-m-Set-Confirmed, ROIV-m-Action-Confirmed, ROIV-m-Create y ROIV-m-Delete ROIVapdu

complexityLimitation ERROR	::= localValue 20
ROER-complexityLimitation	::= ROERapdu (WITH COMPONENTS
{ invokeID	PRESENT,
error-value	(complexityLimitation),
parameter	(INCLUDES ComplexityLimitation) OPTIONAL })

-- Esta ROERapdu sólo puede devolverse en respuesta a las ROIV-m-Get, ROIV-m-Set-Confirmed, ROIV-m-Action-Confirmed y ROIV-m-Delete ROIVapdu

duplicateManagedObjectInstance ERROR	::= localValue 11
ROER-duplicateManagedObjectInstance	::= ROERapdu (WITH COMPONENTS
{ invokeID	PRESENT,
error-value	(duplicateManagedObjectInstance),
parameter	(INCLUDES ObjectInstance) })

-- Esta ROERapdu sólo puede devolverse en respuesta a la ROIV-m-Create ROIVapdu

getListError ERROR	::= localValue 7
ROER-getListError	::= ROERapdu (WITH COMPONENTS
{ invokeID	PRESENT,
error-value	(getListError),
parameter	(INCLUDES GetListError) })

-- Esta ROERapdu sólo puede devolverse en respuesta a la ROIV-m-Get ROIVapdu

invalidArgumentValue ERROR	::= localValue 15
ROER-invalidArgumentValue	::= ROERapdu (WITH COMPONENTS
{ invokeID	PRESENT,
error-value	(invalidArgumentValue),
parameter	(INCLUDES InvalidArgumentValue) })

-- Esta ROERapdu sólo puede devolverse en respuesta a las ROIV-m-EventReport-Confirmed y ROIV-m-Action-Confirmed ROIVapdu

invalidAttributeValue ERROR	::= localValue 6
ROER-invalidAttributeValue	::= ROERapdu (WITH COMPONENTS
{ invokeID	PRESENT,
error-value	(invalidAttributeValue),
parameter	(INCLUDES Attribute) })

-- Esta ROERapdu sólo puede devolverse en respuesta a la ROIV-m-Create ROIVapdu

invalidFilter ERROR	::= localValue 4
ROER-invalidFilter	::= ROERapdu (WITH COMPONENTS
{ invokeID	PRESENT,
error-value	(invalidFilter),
parameter	(INCLUDES CMISFilter) })

Reemplazada por una versión más reciente

-- Esta ROERapdu sólo puede devolverse en respuesta a las ROIV-m-Get, ROIV-m-Set-Confirmed, ROIV-m-Action-Confirmed y ROIV-m-Delete ROIVapdus

invalidObjectInstance ERROR	::= localValue 17
ROER-invalidObjectInstance	::= ROERapdu (WITH COMPONENTS
{ invokeID	PRESENT,
error-value	(invalidObjectInstance),
parameter	(INCLUDES ObjectInstance) })

-- Esta ROERapdu sólo puede devolverse en respuesta a la ROIV-m-Create ROIVapdu

invalidScope ERROR	::= localValue 16
ROER-invalidScope	::= ROERapdu (WITH COMPONENTS
{ invokeID	PRESENT,
error-value	(invalidScope),
parameter	(INCLUDES Scope) })

-- Esta ROERapdu sólo puede devolverse en respuesta a las ROIV-m-Get, ROIV-m-Set-Confirmed, ROIV-m-Action-Confirmed y ROIV-m-Delete ROIVapdus

missingAttributeValue ERROR	::= localValue 18
ROER-missingAttributeValue	::= ROERapdu (WITH COMPONENTS
{ invokeID	PRESENT,
error-value	(missingAttributeValue),
parameter	(INCLUDES SET OF AttributeId) })

-- Esta ROERapdu sólo puede devolverse en respuesta a la ROIV-m-Create ROIVapdu

mistypedOperation ERROR	::= localValue 21
ROER-mistypedOperation	::= ROERapdu (WITH COMPONENTS
{ invokeID	PRESENT,
error-value	(mistypedOperation) })

-- Esta ROERapdu sólo puede devolverse en respuesta a la ROIV-m-Cancel-Get ROIVapdu

noSuchAction ERROR	::= localValue 9
ROER-noSuchAction	::= ROERapdu (WITH COMPONENTS
{ invokeID	PRESENT,
error-value	(noSuchAction),
parameter	(INCLUDES NoSuchAction) })

-- Esta ROERapdu sólo puede devolverse en respuesta a la ROIV-m-Action-Confirmed ROIVapdu

noSuchArgument ERROR	::= localValue 14
ROER-noSuchArgument	::= ROERapdu (WITH COMPONENTS
{ invokeID	PRESENT,
error-value	(noSuchArgument),
parameter	(INCLUDES NoSuchArgument) })

-- Esta ROERapdu sólo puede devolverse en respuesta a las ROIV-m-EventReport-Confirmed y ROIV-m-Action-Confirmed ROIVapdus

noSuchAttribute ERROR	::= localValue 5
ROER-noSuchAttribute	::= ROERapdu (WITH COMPONENTS
{ invokeID	PRESENT,
error-value	(noSuchAttribute),
parameter	(INCLUDES AttributeId) })

Reemplazada por una versión más reciente

-- Esta ROERapdu sólo puede devolverse en respuesta a la ROIV-m-Create ROIVapdu

noSuchEventType ERROR ::= localValue 13

ROER-noSuchEventType ::= ROERapdu (WITH COMPONENTS PRESENT, { invokeID error-value parameter (noSuchEventType), (INCLUDES NoSuchEventType) })

-- Esta ROERapdu sólo puede devolverse en respuesta a la ROIV-m-EventReport-Confirmed ROIVapdu

noSuchInvokeId ERROR ::= localValue 22

ROER-noSuchInvokeId ::= ROERapdu (WITH COMPONENTS PRESENT, { invokeID error-value parameter (noSuchInvokeId), (INCLUDES InvokeIdType) })

-- Esta ROERapdu sólo puede devolverse en respuesta a la ROIV-m-Cancel-Get ROIVapdu

noSuchObjectClass ERROR ::= localValue 0

ROER-noSuchObjectClass ::= ROERapdu (WITH COMPONENTS PRESENT, { invokeID error-value parameter (noSuchObjectClass), (INCLUDES ObjectClass) })

-- Esta ROERapdu sólo puede devolverse en respuesta a las ROIV-m-EventReport-Confirmed, ROIV-m-Get, ROIV-m-Set-Confirmed, ROIV-m-Action-Confirmed, ROIV-m-Create, y ROIV-m-Delete ROIVapdus

noSuchObjectInstance ERROR ::= localValue 1

ROER-noSuchObjectInstance ::= ROERapdu (WITH COMPONENTS PRESENT, { invokeID error-value parameter (noSuchObjectInstance), (INCLUDES ObjectInstance) })

-- Esta ROERapdu sólo puede devolverse en respuesta a las ROIV-m-EventReport-Confirmed, ROIV-m-Get, ROIV-m-Set-Confirmed, ROIV-m-Action-Confirmed, ROIV-m-Create, y ROIV-m-Delete ROIVapdus

noSuchReferenceObject ERROR ::= localValue 12

ROER-noSuchReferenceObject ::= ROERapdu (WITH COMPONENTS PRESENT, { invokeID error-value parameter (noSuchReferenceObject), (INCLUDES ObjectInstance) })

-- Esta ROERapdu sólo puede devolverse en respuesta a la ROIV-m-Create ROIVapdu

operationCancelled ERROR ::= localValue 23

ROER-operationCancelled ::= ROERapdu (WITH COMPONENTS PRESENT, { invokeID error-value (operationCancelled) })

-- Esta ROERapdu sólo puede devolverse en respuesta a la ROIV-m-Get ROIVapdu

processingFailure ERROR ::= localValue 10

ROER-processingFailure ::= ROERapdu (WITH COMPONENTS PRESENT, { invokeID error-value parameter (processingFailure), (INCLUDES ProcessingFailure) OPTIONAL })

Reemplazada por una versión más reciente

-- Esta ROERapdu sólo puede devolverse en respuesta a las ROIV-m-EventReport-Confirmed, ROIV-m-Get, ROIV-m-Set-Confirmed, ROIV-m-Action-Confirmed, ROIV-m-Create, y ROIV-m-Delete ROIVapdus

```
setListError ERROR ::= localValue 8

ROER-setListError ::= ROERapdu (WITH COMPONENTS
  { invokeID PRESENT,
    error-value (setListError),
    parameter (INCLUDES SetListError) } )
```

-- Esta ROERapdu sólo puede devolverse en respuesta a la ROIV-m-Set-Confirmed ROIVapdu

```
syncNotSupported ERROR ::= localValue 3

ROER-syncNotSupported ::= ROERapdu (WITH COMPONENTS
  { invokeID PRESENT,
    error-value (syncNotSupported),
    parameter (INCLUDES CMISync) } )
```

-- Esta ROERapdu sólo puede devolverse en respuesta a las ROIV-m-Get, ROIV-m-Set-Confirmed, ROIV-m-Action-Confirmed y ROIV-m-Delete ROIVapdus

-- Para completar la especificación de sintaxis abstracta proporcionada en este anexo se incorpora como referencia las definiciones de los tipos de soporte que figuran en el § 7.4

END-- Fin de definiciones de sintaxis PCIG

ANEXO C

(Este anexo no forma parte de esta Recomendación)

Ejemplos de UDPA de ESOD del ESCIG

En este anexo se dan algunos ejemplos de la expansión completa de las UDPA de ESOD que transportan información PCIG.

Estos ejemplos servirán de orientación a los usuarios de esta Recomendación.

--- UDPA ODIV (ROIVapdu) para la operación de acción confirmada de ESCIG

```
ROIVapdu-example ::= [1] IMPLICIT SEQUENCE {
  invokeID InvokeIDType,
  operation-value INTEGER {m-Action-Confirmed (7)},
  argument SEQUENCE {
    COMPONENTS OF
    accessControl [5] AccessControl OPTIONAL,
    synchronization [6] IMPLICIT CMISync OPTIONAL,
    scope [7] Scope DEFAULT baseObject,
    filter CMISFilter DEFAULT and {},
  }
  actionInfo [12] IMPLICIT SEQUENCE (
  actionType ActionType,
  actionInfoArg [4] ANY DEFINED BY actionType OPTIONAL
  } }
```

Reemplazada por una versión más reciente

-- UDPA ODRS (RORSapdu) para la operación de acción confirmada de ESCIG

```
RORSapdu-example ::= [2] IMPLICIT SEQUENCE {
    invokeID InvokeIDType,
    SEQUENCE {
        operation-value INTEGER {m-Action-Confirmed (7)},
        result SEQUENCE {
            managedObjectClass ObjectClass OPTIONAL,
            managedObjectInstance ObjectInstance OPTIONAL,
            currentTime [5] IMPLICIT GeneralizedTime OPTIONAL,
            actionReply [6] IMPLICIT SEQUENCE (
                actionType ActionTypeId,
                actionReplyInfo [4] ANY DEFINED BY actionType OPTIONAL } OPTIONAL
        }
    }
}
```

-- UDPA ODIV (ROIVapdu) para la respuesta ligada ESCIG para una operación de acción confirmada

```
ROIVapdu-linked-example ::= [1] IMPLICIT SEQUENCE {
    invokeID InvokeIDType,
    linked-ID [0] IMPLICIT InvokeIDType,
    operation-value INTEGER {m-Action-Confirmed (7)},
    argument CHOICE {
        actionResult [4] IMPLICIT ActionResult,
        processingFailure [5] IMPLICIT ProcessingFailure,
        actionError [7] IMPLICIT ActionError
    }
}
```

-- UDPA ODER (ROERapdu) para la operación de acción confirmada ESCIG cuando se produce un error noSuchAction

```
ROERapdu-example ::= [3] IMPLICIT SEQUENCE {
    invokeID InvokeIDType,
    error-value INTEGER {noSuchAction (9)},
    parameter SEQUENCE { managedObjectClass ObjectClass OPTIONAL,
        actionId ActionTypeId
    }
}
```