



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

**UIT-T**

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

**X.681**

(12/97)

SERIE X: REDES DE DATOS Y COMUNICACIÓN  
ENTRE SISTEMAS ABIERTOS

Gestión de redes de interconexión de sistemas abiertos y  
aspectos de sistemas – Notación de sintaxis abstracta  
uno

---

**Tecnología de la información – Notación de  
sintaxis abstracta uno: Especificación de  
objetos de información**

Recomendación UIT-T X.681

(Anteriormente Recomendación del CCITT)

---

RECOMENDACIONES DE LA SERIE X DEL UIT-T  
**REDES DE DATOS Y COMUNICACIÓN ENTRE SISTEMAS ABIERTOS**

<b>REDES PÚBLICAS DE DATOS</b>	
Servicios y facilidades	X.1–X.19
Interfaces	X.20–X.49
Transmisión, señalización y conmutación	X.50–X.89
Aspectos de redes	X.90–X.149
Mantenimiento	X.150–X.179
Disposiciones administrativas	X.180–X.199
<b>INTERCONEXIÓN DE SISTEMAS ABIERTOS</b>	
Modelo y notación	X.200–X.209
Definiciones de los servicios	X.210–X.219
Especificaciones de los protocolos en modo conexión	X.220–X.229
Especificaciones de los protocolos en modo sin conexión	X.230–X.239
Formularios para declaraciones de conformidad de implementación de protocolo	X.240–X.259
Identificación de protocolos	X.260–X.269
Protocolos de seguridad	X.270–X.279
Objetos gestionados de capa	X.280–X.289
Pruebas de conformidad	X.290–X.299
<b>INTERFUNCIONAMIENTO ENTRE REDES</b>	
Generalidades	X.300–X.349
Sistemas de transmisión de datos por satélite	X.350–X.399
<b>SISTEMAS DE TRATAMIENTO DE MENSAJES</b>	<b>X.400–X.499</b>
<b>DIRECTORIO</b>	<b>X.500–X.599</b>
<b>GESTIÓN DE REDES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS Y ASPECTOS DE SISTEMAS</b>	
Gestión de redes	X.600–X.629
Eficacia	X.630–X.639
Calidad de servicio	X.640–X.649
Denominación, direccionamiento y registro	X.650–X.679
<b>Notación de sintaxis abstracta uno</b>	<b>X.680–X.699</b>
<b>GESTIÓN DE INTERCONEXIÓN DE SISTEMAS ABIERTOS</b>	
Marco y arquitectura de la gestión de sistemas	X.700–X.709
Servicio y protocolo de comunicación de gestión	X.710–X.719
Estructura de la información de gestión	X.720–X.729
Funciones de gestión y funciones de arquitectura de gestión distribuida abierta	X.730–X.799
<b>SEGURIDAD</b>	<b>X.800–X.849</b>
<b>APLICACIONES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS</b>	
Compromiso, concurrencia y recuperación	X.850–X.859
Procesamiento de transacciones	X.860–X.879
Operaciones a distancia	X.880–X.899
<b>PROCESAMIENTO DISTRIBUIDO ABIERTO</b>	<b>X.900–X.999</b>

**NORMA INTERNACIONAL 8824-2**

**RECOMENDACIÓN UIT-T X.681**

**TECNOLOGÍA DE LA INFORMACIÓN –  
NOTACIÓN DE SINTAXIS ABSTRACTA UNO:  
ESPECIFICACIÓN DE OBJETOS DE INFORMACIÓN**

**Resumen**

La presente Recomendación | Norma Internacional proporciona la notación de ASN.1 que permite definir y dar nombres de referencia a clases de objetos de información así como a objetos de información individuales y conjuntos de los mismos. Una clase de objeto de información es una plantilla para una recopilación de información que forma los atributos de cualquier miembro de esa clase.

**Orígenes**

El texto de la Recomendación UIT-T X.681 se aprobó el 12 de diciembre de 1997. Su texto se publica también, en forma idéntica, como Norma Internacional ISO/CEI 8824-2.

## PREFACIO

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la 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.

En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI.

## NOTA

En esta Recomendación, la expresión *empresa de explotación reconocida (EER)* designa a toda persona, compañía, empresa u organización gubernamental que explote un servicio de correspondencia pública. Los términos *Administración, EER y correspondencia pública* están definidos en la *Constitución de la UIT (Ginebra, 1992)*.

## PROPIEDAD INTELECTUAL

La UIT señala a la atención la posibilidad de que la utilización o aplicación de la presente Recomendación suponga el empleo de un derecho de propiedad intelectual reivindicado. La UIT no adopta ninguna posición en cuanto a la demostración, validez o aplicabilidad de los derechos de propiedad intelectual reivindicados, ya sea por los miembros de la UIT o por terceros ajenos al proceso de elaboración de Recomendaciones.

En la fecha de aprobación de la presente Recomendación, la UIT no ha recibido notificación de propiedad intelectual, protegida por patente, que puede ser necesaria para aplicar esta Recomendación. Sin embargo, debe señalarse a los usuarios que puede que esta información no se encuentre totalmente actualizada al respecto, por lo que se les insta encarecidamente a consultar la base de datos sobre patentes de la TSB.

© UIT 1998

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>
Introducción.....	v
1 Alcance.....	1
2 Referencias normativas.....	1
2.1 Recomendaciones   Normas Internacionales idénticas.....	1
3 Definiciones.....	1
3.1 Especificación de la notación básica.....	1
3.2 Especificación de constricciones.....	1
3.3 Parametrización de especificación de ASN.1.....	2
3.4 Definiciones adicionales.....	2
4 Abreviaturas.....	3
5 Convenio.....	3
6 Notación.....	3
6.1 Asignaciones.....	3
6.2 Tipos.....	3
6.3 Valores.....	4
6.4 Elementos.....	4
7 Ítems de ASN.1.....	4
7.1 Referencias de clase de objetos de información.....	4
7.2 Referencias de objetos de información.....	4
7.3 Referencias de conjunto de objetos de información.....	4
7.4 Referencias de campo de tipo.....	4
7.5 Referencias de campo de valor.....	5
7.6 Referencias de campo de conjunto de valores.....	5
7.7 Referencias de campo de objeto.....	5
7.8 Referencias de campo de conjunto de objetos.....	5
7.9 Palabra.....	5
7.10 Ítems de palabras clave adicionales.....	5
8 Definiciones referentes.....	5
9 Definición y asignación de clases de objetos de información.....	6
10 Lista de sintaxis.....	10
11 Definición y asignación de objetos de información.....	12
12 Definición y asignación de conjunto de objetos de información.....	14
13 Cuadros asociados.....	15
14 Notación para el tipo de campo de clase de objeto.....	16
15 Información procedente de objetos.....	18

	<i>Página</i>
Anexo A – Clase de objeto de información TYPE-IDENTIFIER .....	21
Anexo B – Definición de sintaxis abstracta .....	22
Anexo C – El tipo ejemplar de.....	23
Anexo D – Ejemplos.....	25
D.1 Ejemplo de utilización de clase OPERATION simplificada.....	25
D.2 Ejemplo de utilización de "ObjectClassFieldType" .....	26
D.3 Ilustración de utilización de objetos y conjuntos de objetos .....	27
Anexo E – Anexo ilustrativo sobre el modelo ASN.1 de ampliación de conjunto de objetos .....	28
E.1 Conjuntos de objetos ampliables.....	28
Anexo F – Resumen de la notación .....	29

## Introducción

Con frecuencia, un diseñador de aplicaciones tiene que diseñar un protocolo que funcionará con uno cualquiera de un número de casos de alguna clase de objetos de información, cuando los casos de la clase pueden ser definidos por una variedad de otros órganos y pueden ser añadidos con el tiempo. Como ejemplos de estas clases de objetos de información cabe citar las "operaciones" del servicio de operaciones a distancia y los "atributos" del directorio de interconexión de sistemas abiertos.

Esta Recomendación | Norma Internacional proporciona la notación que permite definir clases de objetos de información así como objetos de información individuales y conjuntos de los mismos y darles nombres de referencia.

Una clase de objeto de información se caracteriza por los géneros de campos poseídos por sus casos. Un campo puede contener:

- un tipo arbitrario (un campo de tipo); o
- un valor de un tipo especificado (un campo de valor de tipo fijo); o
- un valor de un tipo especificado en un campo de tipo (denominado) (un campo de valor de tipo variable);
- un conjunto de valores no vacío de un tipo especificado (un campo de conjunto de valores de tipo fijo); o
- un conjunto de valores no vacío de un tipo especificado en un campo de tipo (denominado) (un campo de conjunto de valores de tipo variable); o
- un objeto de información de una clase de objeto de información especificada (un campo de objeto);
- un conjunto de objetos de información de una clase de objeto de información especificada (un campo de conjunto de objetos).

Se puede seleccionar un campo de valor de tipo fijo de una clase de objeto de información para proporcionar identificación única de objetos de información en esa clase. Esto se denomina el campo de identificador para esa clase. Los valores del campo de identificador, si se suministran, tienen que ser únicos dentro de cualquier conjunto de objetos de información que se defina para esa clase. Pueden, pero no tienen que, servir para identificar inequívocamente objetos de información de esa clase dentro de un ámbito más amplio, en particular mediante la utilización del identificador de objeto como el tipo del campo de identificador.

Una clase de objeto de información se define especificando:

- los nombres de los campos;
- para cada campo, la forma de ese campo (tipo, valor de tipo fijo, valor de tipo variable, conjunto de valores de tipo fijo, conjunto de valores de tipo variable, objeto o conjunto de objetos);
- la posibilidad de opción y las fijaciones por defecto de campos;
- qué campo, si lo hubiere, es el campo de identificador.

Un objeto de información individual en la clase se define proporcionando la información necesaria para cada campo.

La notación definida en esta Recomendación | Norma Internacional permite especificar un tipo de ASN.1 por referencia a un campo de alguna clase de objeto de información – el tipo de campo de clase de objeto. En la Rec. UIT-T X.682 | ISO/CEI 8824-3, se proporciona notación para que este tipo pueda ser restringido por referencia a algún conjunto específico de objetos de información.

Puede ser útil considerar que la definición de una clase de objeto de información define la forma de un cuadro conceptual subyacente con una columna para cada campo y con una fila completada que define un objeto de información. La forma del cuadro (determinada por la especificación de la clase de objeto de información) determina el género de información que se ha de recopilar y utilizar para completar alguna especificación de protocolo. El cuadro conceptual subyacente proporciona el enlace entre los que especifican objetos de información de esa clase y el protocolo que necesita esa información para completar su especificación. Normalmente, el conjunto de objetos de información reales utilizados para completar una especificación de protocolo particular será un parámetro de ese protocolo (véase la Rec. UIT-T X.683 | ISO/CEI 8824-4).

La notación "InformationFromObjects" (información de objetos) que hace referencia a un objeto o conjunto de objetos específicos (probablemente un parámetro) se puede utilizar para extraer información de las casillas de los cuadros conceptuales.

La presente Recomendación | Norma Internacional:

- especifica una notación para definir una clase de objeto de información, y para identificarla con un nombre de referencia (véase la cláusula 9);
- especifica una notación por la cual el definidor de una clase de objeto de información puede proporcionar una sintaxis definida para la definición de objetos de información de esa clase; se proporciona una notación por defecto para aquellas clases para las cuales no se ha definido una sintaxis determinada (véase la cláusula 10);
- especifica una notación para definir un objeto de información, y para asignarlo a un nombre de referencia (véase la cláusula 11) y proporciona una notación análoga para un conjunto de objetos (véase la cláusula 12);
- define el "cuadro asociado" para un objeto o conjunto de objetos de una clase (véase la cláusula 13);
- especifica la notación para el tipo de campo de clase de objeto y sus valores (véase la cláusula 14);
  - NOTA – Estos constructivos permiten especificar un tipo de ASN.1 utilizando un campo denominado de una clase de objeto de información denominada. Las constricciones impuestas a ese tipo para restringirlo a valores relacionados con un conjunto específico de objetos de información aparece en la Rec. UIT-T X.682 | ISO/CEI 8824-3;
- especifica la notación para extraer información de objetos (véase la cláusula 15).

El conjunto de objetos de información utilizados al definir un conjunto de objetos puede ser parcial o totalmente desconocido en el momento de la definición de una especificación ASN.1. Tales casos ocurren, por ejemplo, en la gestión de redes en donde el conjunto de objetos gestionados varía mientras está actuando el gestor de red. Esta Recomendación | Norma Internacional especifica las reglas para la inclusión de un **marcador de ampliación** en la definición de conjuntos de objetos para señalar a los implementadores la intención del diseñador de que el contenido del conjunto de objetos no esté totalmente definido en la especificación ASN.1. Cuando un objeto se define con un marcador de ampliación, el implementador debe proporcionar los medios, posiblemente fuera del alcance de la ASN.1, con los que añadir objetos al conjunto de objetos y eliminar de dicho conjunto objetos añadidos con anterioridad dinámicamente.

El anexo A, que es parte integrante de la presente Recomendación | Norma Internacional, especifica la clase de objeto de información cuya referencia de clase de objeto es TYPE-IDENTIFIER (identificador de tipo). Ésta es la clase útil más simple, con sólo dos campos, un campo de identificador de objeto de tipo y un campo de un tipo que define el tipo de ASN.1 para transportar toda la información relativa a cualquier objeto particular en esa clase. Se define aquí debido a la utilización generalizada de objetos de información de esta forma.

El anexo B, que es parte integrante de la presente Recomendación | Norma Internacional, especifica la notación para definir una sintaxis abstracta (compuesta del conjunto de valores de un tipo de ASN.1) mediante la definición de un objeto de información apropiado.

El anexo C, que es parte integrante de la presente Recomendación | Norma Internacional, establece la notación para el tipo ejemplar de (la notación "INSTANCE OF"), que puede transportar cualquier valor de cualquier objeto de información en una clase especificada (que debe haber sido definida en términos de TYPE-IDENTIFIER).

El anexo D, que no es parte integrante de esta Recomendación | Norma Internacional, contiene ejemplos de cómo utilizar la notación descrita en la misma.

El anexo E, que no es parte integrante de esta Recomendación | Norma Internacional, contiene un resumen del modelo ASN.1 de ampliación de conjunto de objetos.

El anexo F, que no es parte integrante de esta Recomendación | Norma Internacional, contiene un resumen de la notación definida en la misma.

## NORMA INTERNACIONAL

## RECOMENDACIÓN UIT-T

**TECNOLOGÍA DE LA INFORMACIÓN –  
NOTACIÓN DE SINTAXIS ABSTRACTA UNO:  
ESPECIFICACIÓN DE OBJETOS DE INFORMACIÓN**

**1 Alcance**

La presente Recomendación | Norma Internacional forma parte de la notación de sintaxis abstracta uno (ASN.1) y proporciona la notación para especificar clases de objetos de información, objetos de información y conjuntos de objetos de información.

**2 Referencias normativas**

Las siguientes Recomendaciones y Normas Internacionales contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación | Norma Internacional. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y Normas son objeto de revisiones, por lo que se preconiza que las participantes en acuerdos basados en la presente Recomendación | Norma Internacional investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones | Normas Internacionales citadas a continuación. Los miembros de la CEI y de la ISO mantienen registros de las Normas Internacionales actualmente vigentes. La Oficina de Normalización de las Telecomunicaciones de la UIT mantiene una lista de las Recomendaciones del UIT-T actualmente vigentes.

**2.1 Recomendaciones | Normas Internacionales idénticas**

- Recomendación UIT-T X.680 (1997) | ISO/CEI 8824-1:1998, Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de la notación básica.
- Recomendación UIT-T X.682 (1997) | ISO/CEI 8824-3:1998, Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de constricciones.
- Recomendación UIT-T X.683 (1997) | ISO/CEI 8824-4:1998, Tecnología de la información – Notación de sintaxis abstracta uno: Parametrización de las especificaciones de la notación de sintaxis abstracta uno.

**3 Definiciones**

A los efectos de la presente Recomendación | Norma Internacional se aplican las siguientes definiciones.

**3.1 Especificación de la notación básica**

La presente Recomendación | Norma Internacional utiliza los términos definidos en la Rec. UIT-T X.680 | ISO/CEI 8824-1.

**3.2 Especificación de constricciones**

La presente Recomendación | Norma Internacional utiliza el siguiente término definido en la Rec. UIT-T X.682 | ISO/CEI 8824-3:

- constricción de tabla.

### 3.3 Parametrización de especificación de ASN.1

La presente Recomendación | Norma Internacional utiliza los siguientes términos definidos en la Rec. UIT-T X.683 | ISO/CEI 8824-4:

- a) tipo parametrizado;
- b) valor parametrizado.

### 3.4 Definiciones adicionales

**3.4.1 tabla asociada; cuadro asociado:** (Para algún objeto de información o conjunto de objetos de información) una tabla abstracta, que puede derivarse del objeto o conjunto de objetos simplificando la estructura jerárquica resultante de la presencia de campos de enlace (véase 3.4.14).

NOTA – Una tabla asociada se puede utilizar para determinar la naturaleza precisa de alguna restricción (véase la Rec. UIT-T X.682 | ISO/CEI 8824-3), que ha sido aplicada utilizando un conjunto de objetos.

**3.4.2 sintaxis por defecto:** La notación que se utilizará para definir objetos de información de clases cuyos definidores no han proporcionado una sintaxis de definición especial (véase el ejemplo en 11.9).

**3.4.3 sintaxis definida:** Una notación, proporcionada por el definidor de una clase, que permite definir objetos de información de esa clase de una manera fácil para el usuario.

NOTA – Por ejemplo, la sintaxis de definición para la clase OPERATION pudiera permitir la definición de casos de la clase mediante la palabra ARGUMENT seguida de &ArgumentType, después la palabra RESULT seguida de &ResultType, después la palabra CODE seguida de &operationCode (véase el ejemplo de 11.10).

**3.4.4 conjunto de objetos ampliable:** Un conjunto de objetos con un marcador de ampliación.

**3.4.5 campo:** Un componente de una clase de objeto de información. Cada campo es un campo de tipo, un campo de valor de tipo fijo, un campo de valor de tipo variable, un campo de conjunto de valores de tipo fijo, un campo de conjunto de valores de tipo variable, un campo de objeto de información o un campo de conjunto de objetos de información.

**3.4.6 nombre de campo:** Nombre que identifica un campo con respecto a alguna clase; la clase que especifica el campo directamente, en cuyo caso el nombre es un nombre de campo primitivo, o una clase que tiene una cadena de campos de enlace a aquélla en la cual el campo está especificado realmente (véanse 9.13 y 9.14).

**3.4.7 campo de identificador:** Un campo de valor de tipo fijo de una clase, seleccionado para proporcionar identificación única de objetos de información en esa clase. Los valores del campo de identificador, si se suministran, tienen que ser inequívocos dentro de cualquier conjunto de objetos de información que se defina para esa clase. Pueden, pero no tienen que servir para identificar inequívocamente objetos de información de esa clase en un ámbito más amplio.

NOTA 1 – El campo del identificador tiene un tipo ASN.1 fijo y se pueden transportar valores de ese tipo en el protocolo para identificar objetos de información dentro de la clase.

NOTA 2 – El ámbito dentro del cual el identificador es inequívoco es el de un conjunto de objetos de información. Sin embargo, podría hacerse inequívoco dentro de cualquier sintaxis abstracta dada, o dentro de un contexto de aplicación completo, o podría incluso ser global a través de todas las clases y todos los contextos de aplicación mediante la utilización del tipo de identificador de objeto para el campo de identificador.

**3.4.8 objeto de información:** Un caso de alguna clase de objeto de información, que se compone de un conjunto de campos conformes a las especificaciones de campo de la clase.

NOTA – Por ejemplo, un caso específico de la clase de objeto de información OPERATION (mencionado en el ejemplo en 3.4.9) pudiera ser invertMatrix que tiene un campo &ArgumentType que contiene el tipo Matrix, un campo &ResultType que contiene también el tipo Matrix y un campo &operationCode que contiene el valor 7 (véase el ejemplo en 10.13).

**3.4.9 clase de objeto de información (clase):** Un conjunto de campos, que forman una plantilla para la definición de una colección de objetos de información potencialmente ilimitados, los casos de la clase.

NOTA – Por ejemplo, se pudiera definir una clase de objeto de información OPERATION para que corresponda con el concepto "operación" de operaciones a distancia. Cada una de las especificaciones de los distintos campos denominados correspondería con algún aspecto que puede variar de una operación a otra. De este modo, podrá haber especificaciones de campos &ArgumentType, &ResultType y &operationCode, las dos primeras que especifican campos de tipo y la tercera que especifica un campo de valor.

**3.4.10 campo de objeto de información:** Un campo que contiene un objeto de información de alguna clase especificada.

**3.4.11 conjunto de objetos de información:** Un conjunto no vacío de objetos de información, todos de la misma clase de objeto de información.

NOTA – Por ejemplo, un conjunto de objetos de información, MatrixOperations, de la clase OPERATION (utilizado en el ejemplo en 3.4.9) pudiera contener invertMatrix (mencionado en 3.4.8) junto con otras operaciones conexas, tales como addMatrices, multiplyMatrices, etc. Este conjunto de objetos se pudiera utilizar para definir una sintaxis abstracta que debe proporcionar la invocación e informe de resultado de todas estas operaciones (véase el ejemplo en 12.7).

**3.4.12 campo de conjunto de objetos de información:** Un campo que contiene un conjunto de objetos de información de alguna clase especificada.

**3.4.13 tipo ejemplar de:** Un tipo definido haciendo referencia a una clase de objeto de información que asocia identificadores de objetos con tipos.

**3.4.14 campo de enlace:** Un campo de objeto o de conjunto de objetos.

**3.4.15 tipo de campo de clase de objeto:** Un tipo especificado por referencia a algún campo de una clase de objeto de información. En la Rec. UIT-T X.682 | ISO/CEI 8824-3, se proporciona la notación para poder restringir este tipo mediante referencia a un conjunto de objetos de información de esa clase.

**3.4.16 nombre de campo primitivo:** El nombre especificado directamente en una definición de clase de objeto de información sin utilizar un campo de enlace.

**3.4.17 campo de tipo:** Un campo que contiene un tipo arbitrario.

**3.4.18 campo de valor:** Un campo que contiene un valor. Este campo es de tipo fijo o de tipo variable. En el primer caso, el tipo del valor es fijado por la especificación del campo. En el segundo caso, el tipo del valor está contenido en algún campo de tipo (específico) del mismo objeto de información.

**3.4.19 campo de conjunto de valores:** Un campo que contiene un conjunto de valores no vacío de algún tipo. Este campo es de tipo fijo o de tipo variable. En el primer caso, el tipo de los valores es fijado por la especificación del campo. En el segundo caso, el tipo de los valores está contenido en algún campo de tipo (específico) del mismo objeto de información.

NOTA – El conjunto de valores en un campo de conjunto de valores para un objeto de información constituye un subtipo del tipo especificado.

## 4 Abreviaturas

A los efectos de esta Recomendación | Norma Internacional se utiliza la siguiente sigla.

ASN.1 Notación de sintaxis abstracta uno (*abstract syntax notation one*).

## 5 Convenio

La presente Recomendación | Norma Internacional emplea el convenio notacional definido en la Rec. UIT-T X.680 | ISO/CEI 8824-1, cláusula 5.

## 6 Notación

En esta cláusula se resume la notación definida en esta Recomendación | Norma Internacional.

### 6.1 Asignaciones

Las siguientes notaciones, que se pueden utilizar como alternativas para "Assignment" (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, cláusula 12) se definen en esta Recomendación | Norma Internacional:

- ObjectClassAssignment (asignación de clase de objeto) (véase 9.1).
- ObjectAssignment (asignación de objeto) (véase 11.1).
- ObjectSetAssignment (asignación de conjunto de objetos) (véase 12.1).

### 6.2 Tipos

**6.2.1** Las siguientes notaciones que se pueden utilizar como alternativas para "BuiltinType" (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 16.2), se definen en esta Recomendación | Norma Internacional:

- ObjectClassFieldType (tipo de campo de clase de objeto) (véase 14.1).
- InstanceOfType (tipo ejemplar de) (véase anexo C).

## ISO/CEI 8824-2 : 1998 (S)

**6.2.2** Las siguientes notaciones, que se pueden utilizar como alternativas para "ReferencedType" (tipo referenciado) (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 16.3) se definen en esta Recomendación | Norma Internacional:

- TypeFromObject (tipo de objeto) (véase la cláusula 15).
- ValueSetFromObjects (conjunto de valores de objetos) (véase la cláusula 15).

### 6.3 Valores

**6.3.1** Las siguientes notaciones, que se pueden utilizar como alternativa para "BuiltinValue" (valor incorporado) (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 16.8) se definen en esta Recomendación | Norma Internacional:

- ObjectClassFieldValue (valor de campo de clase de objetos) (véase 14.6).
- InstanceOfValue (valor de ejemplar de) (véase anexo C).

**6.3.2** Las siguientes notaciones, que se pueden utilizar como alternativas para "ReferencedValue" (valor referenciado) (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 16.9) se definen en esta Recomendación | Norma Internacional:

- ValueFromObject (valor de objeto) (véase la cláusula 15).

### 6.4 Elementos

**6.4.1** La siguiente notación, que se puede utilizar como una alternativa para "Elements" (Elementos) (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 46.3), se define en esta Recomendación | Norma Internacional:

- ObjectSetElements (elementos de conjuntos de objetos) (véase 12.6).

## 7 Ítems de ASN.1

Además de los ítems de ASN.1 especificados en la Rec. UIT-T X.680 | ISO/CEI 8824-1, cláusula 11, la presente Recomendación | Norma Internacional utiliza los ítems de ASN.1 especificados en las siguientes subcláusulas. Las reglas generales aplicables a estos ítems son las definidas en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.1. Estos nuevos ítems de ASN.1 utilizan el conjunto de caracteres ASN.1, especificado en la Rec. UIT-T X.680 | ISO/CEI 8824-1, cláusula 10, y además, el carácter "y comercial" ("&").

NOTA – La nota de la Rec. UIT-T X.680 | ISO/CEI 8824-1, 10.1, se aplica también a los ítems especificados en 7.1 a 7.9.

### 7.1 Referencias de clase de objetos de información

Nombre de ítem – objectclassreference (referencia de clase de objeto).

"objectclassreference" consistirá en una secuencia de caracteres especificados para "typereference" en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.2, salvo que no se incluirán letras minúsculas.

### 7.2 Referencias de objetos de información

Nombre de ítem – objectreference (referencia de objeto).

"objectreference" consistirá en una secuencia de caracteres especificados para "valuereference" en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.4.

### 7.3 Referencias de conjunto de objetos de información

Nombre de ítem – objectsetreference (referencia de conjunto de objetos).

"objectsetreference" consistirá en una secuencia de caracteres especificados para "typereference" en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.2.

### 7.4 Referencias de campo de tipo

Nombre de ítem – typefieldreference (referencia de campo de tipo).

"typefieldreference" consistirá en una "y comercial" ("&") seguida inmediatamente de una secuencia de caracteres como se especifica para "typereference" en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.2.

## 7.5 Referencias de campo de valor

Nombre de ítem – valuefieldreference (referencia de campo de valor).

"valuefieldreference" consistirá en una "y comercial"("&") seguida inmediatamente de una secuencia de caracteres como se especifica para "valuereference" en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.4.

## 7.6 Referencias de campo de conjunto de valores

Nombre de ítem – valuesetfieldreference (referencia de campo de conjunto de valores).

"valuesetfieldreference" consistirá en una "y comercial"("&") seguida inmediatamente de una secuencia de caracteres como se especifica para "typereference" en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.2.

## 7.7 Referencias de campo de objeto

Nombre de ítem – objectfieldreference (referencia de campo de objeto).

"objectfieldreference" consistirá en una "y comercial"("&") seguida inmediatamente de una secuencia de caracteres como se especifica para "objectreference" en 7.2.

## 7.8 Referencias de campo de conjunto de objetos

Nombre de ítem – objectsetfieldreference (referencia de campo de conjunto de objetos).

"objectsetfieldreference" consistirá en una "y comercial"("&") seguida inmediatamente de una secuencia de caracteres como se especifica para "objectsetreference" en 7.3.

## 7.9 Palabra

Nombre de ítem – word (palabra).

"word" consistirá en una secuencia de caracteres como se especifica para "typereference" en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.2, salvo que no se incluirán letras minúsculas ni cifras.

## 7.10 Ítems de palabras clave adicionales

Los nombres CLASS (CLASE), INSTANCE (CASO), SYNTAX (SINTAXIS) y UNIQUE (ÚNICO) se indican en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.18 como palabras reservadas.

## 8 Definiciones referentes

8.1 Los constructivos:

**DefinedObjectClass ::=**

**ExternalObjectClassReference |  
objectclassreference |  
UsefulObjectClassReference**

**DefinedObject ::=**

**ExternalObjectReference |  
objectreference**

**DefinedObjectSet ::=**

**ExternalObjectSetReference |  
objectsetreference**

se utilizan para definiciones de clase de referencia, de objeto de información y de conjunto de objetos de información, respectivamente.

8.2 Salvo como se especifica en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 12.15, las alternativas "objectclassreference", "objectreference" y "objectsetreference" se utilizarán solamente dentro del módulo en el cual se asigna una clase u objeto de información o conjunto de objetos de información a esa referencia (véanse 9.1, 11.1 y 12.1).

Las alternativas "ExternalObjectClassReference", "ExternalObjectReference" y "ExternalObjectSetReference" se definen como sigue:

**ExternalObjectClassReference ::=**

**modulereference**  
"."  
**objectclassreference**

**ExternalObjectReference ::=**

**modulereference**  
"."  
**objectreference**

**ExternalObjectSetReference ::=**

**modulereference**  
"."  
**objectsetreference**

Estas alternativas no se utilizarán a menos que a las correspondientes "objectclassreference", "objectreference" u "objectsetreference" se les haya asignado una clase u objeto de información o conjunto de objetos de información respectivamente (véanse 9.1, 11.1 y 12.1) dentro del módulo (diferente del módulo referente) identificado por el "modulereference" correspondiente. Es a esa clase u objeto de información o conjunto de objetos de información respectivamente que se hace referencia.

8.3 La alternativa "Usefulobjectclassreference" de "DefinedObjectClass" se define como sigue:

**Usefulobjectclassreference ::= TYPE-IDENTIFIER | ABSTRACT-SYNTAX**

de la cual la primera alternativa se especifica en el anexo A y la segunda en el anexo B.

NOTA – Los nombres TYPE-IDENTIFIER y ABSTRACT-SYNTAX se indican en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.18, como palabras reservadas.

## 9 Definición y asignación de clases de objetos de información

9.1 El constructivo "ObjectClassAssignment" se utiliza para asignar una clase de objeto de información a un nombre de referencia ("objectclassreference"). Este constructivo es una de las alternativas para "Assignment" en la Rec. UIT-T X.680 | ISO/CEI 8824-1, cláusula 12, y se define como sigue:

**ObjectClassAssignment ::=**

**objectclassreference**  
"::=" **ObjectClass**

9.2 La clase de objeto de información es la definida por el constructivo "ObjectClass":

**ObjectClass ::=**

**DefinedObjectClass |**  
**ObjectClassDefn |**  
**ParameterizedObjectClass**

si "ObjectClass" es:

- "DefinedObjectClass", la definición de clase es igual que la de la clase referenciada.
- "ObjectClassDefn", la clase se define como se describe en 9.3.
- "ParameterizedObjectClass", la clase se define como se describe en la Rec. UIT-T X.683 | ISO/CEI 8824-4, 9.2.

9.3 Cada clase es definida a la larga por "ObjectClassDefn":

```
ObjectClassDefn ::=
  CLASS
  "{" FieldSpec "," + "}"
  WithSyntaxSpec?

WithSyntaxSpec ::= WITH SYNTAX SyntaxList
```

Esta notación permite al definidor de una clase proporcionar las especificaciones de campo denominadas, cada una de las cuales es una "FieldSpec", definida en 9.4. Facultativamente, el definidor puede proporcionar una sintaxis de definición de objeto de información ("SyntaxList"), según se define en 10.5. El definidor de la clase puede especificar también la semántica asociada con la definición de la clase.

9.4 Cada "FieldSpec" especifica y denomina uno de los campos que estará o puede estar asociado con casos de la clase:

```
FieldSpec ::=
  TypeFieldSpec |
  FixedTypeValueFieldSpec |
  VariableTypeValueFieldSpec |
  FixedTypeValueSetFieldSpec |
  VariableTypeValueSetFieldSpec |
  ObjectFieldSpec |
  ObjectSetFieldSpec
```

Las diversas alternativas para "FieldSpec" se especifican en las subcláusulas siguientes.

9.5 "TypeFieldSpec" especifica que el campo es un campo de tipo (véase 3.4.17):

```
TypeFieldSpec ::=
  typefieldreference
  TypeOptionalitySpec?

TypeOptionalitySpec ::= OPTIONAL | DEFAULT Type
```

El nombre del campo es "typefieldreference". Si "TypeOptionalitySpec" está ausente, todas las definiciones de objetos de información para esa clase tienen que incluir una especificación de un tipo para ese campo. Si "OPTIONAL" está presente, el campo se puede dejar indefinido. Si "DEFAULT" está presente, el siguiente "Type" proporciona la fijación por defecto para el campo si se omite en una definición.

9.6 "FixedTypeValueFieldSpec" especifica que el campo es un campo de valor de tipo fijo (véase 3.4.18):

```
FixedTypeValueFieldSpec ::=
  valuefieldreference
  Type
  UNIQUE?
  ValueOptionalitySpec?

ValueOptionalitySpec ::= OPTIONAL | DEFAULT Value
```

El nombre del campo es "valuefieldreference". El constructivo "Type" especifica el tipo del valor contenido en el campo. Si está presente, "ValueOptionalitySpec" especifica que se puede omitir el valor en una definición de objeto de información, o, en el caso "DEFAULT", esa omisión produce el siguiente "Value" que será de ese tipo. La presencia de la palabra clave "UNIQUE" especifica que este campo es un campo de identificador. Si la palabra clave está presente, "ValueOptionalitySpec" no será "DEFAULT Value".

9.7 Cuando se asigna un valor para un campo de identificador, ese valor tiene que ser inequívoco dentro de cualquier conjunto de objetos de información definido.

9.8 "VariableTypeValueFieldSpec" especifica que el campo es un campo de valor de tipo variable (véase 3.4.18):

```
VariableTypeValueFieldSpec ::=
  valuefieldreference
  FieldName
  ValueOptionalitySpec?
```

El nombre del campo es "valuefieldreference". "FieldName" (véase 9.14), que es relativo con respecto a la clase que se especifica, será el de un campo de tipo; el campo de tipo que está en el mismo objeto de información que el campo de valor, o que está enlazado por la cadena de campos de objeto cuyas referencias aparecen en "FieldName", contendrá el

tipo del valor. (Todos los campos de enlace cuyas referencias de campo aparecen en "FieldName" serán campos de objeto.) Si está presente, "ValueOptionalitySpec" especifica que se puede omitir el valor en una definición de objeto de información o, en el caso de "DEFAULT", esa omisión produce el siguiente "Value". "ValueOptionalitySpec" será tal que:

- a) si el campo de tipo denotado "FieldName" tiene "TypeOptionalitySpec" de "OPTIONAL", "ValueOptionalitySpec" será también "OPTIONAL"; y
- b) si "ValueOptionalitySpec" es "DEFAULT Value", el campo de tipo denotado por "FieldName" tendrá una "TypeOptionalitySpec" de "DEFAULT Type" y "Value" será un valor de ese tipo.

**9.9** "FixedTypeValueSetFieldSpec" especifica que el campo es un campo de conjunto de valores de tipo fijo (véase 3.4.19):

**FixedTypeValueSetFieldSpec ::=**

**valuesetfieldreference**

**Type**

**ValueSetOptionalitySpec?**

**ValueSetOptionalitySpec ::= OPTIONAL | DEFAULT ValueSet**

NOTA – "ValueSet" se define en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 15.4 y 15.5 y permite el listado explícito (entre llaves) del conjunto de valores, o la utilización de "typereference" para un subtipo de "Type".

El nombre del campo es "valuesetfieldreference". El constructivo "Type" especifica el tipo de los valores contenidos en el campo. Si está presente, "ValueSetOptionalitySpec" especifica que el campo puede estar no especificado en la definición de objeto de información o, en el caso de "DEFAULT", esa omisión produce el siguiente "ValueSet" que será el subtipo de ese tipo.

**9.10** "VariableTypeValueSetFieldSpec" especifica que el campo es un campo de conjunto de valores de tipo variable (véase 3.4.19):

**VariableTypeValueSetFieldSpec ::=**

**valuesetfieldreference**

**FieldName**

**ValueSetOptionalitySpec?**

El nombre del campo es "valuesetfieldreference". "FieldName" (véase 9.14) que es relativo a la clase que se especifica, será el de un campo de tipo; el campo de tipo que está en el objeto de información como el campo de conjunto de valores, o está enlazado por la cadena de campo de objeto cuya referencia aparece en "FieldName", contendrá el tipo de los valores. (Todos los campos de enlace, cuyas referencias aparecen en "FieldName" serán campos de objeto.) Si está presente, "ValueSetOptionalitySpec" especifica que se puede omitir el conjunto de valores en una definición de objeto de información o, en el caso de "DEFAULT", esa omisión produce el siguiente "ValueSet". "ValueSetOptionalitySpec" será tal que:

- a) si el campo de tipo denotado por "FieldName" tiene una "TypeOptionalitySpec" de "OPTIONAL", entonces "ValueSetOptionalitySpec" será también "OPTIONAL"; y
- b) si "ValueSetOptionalitySpec" es "DEFAULT ValueSet", el campo de tipo denotado por "FieldName" tendrá una "TypeOptionalitySpec" de "DEFAULT Type" y "ValueSet" será un subtipo de ese tipo.

**9.11** "ObjectFieldSpec" especifica que el campo es un campo de objeto de información (véase 3.4.10):

**ObjectFieldSpec ::=**

**objectfieldreference**

**DefinedObjectClass**

**ObjectOptionalitySpec?**

**ObjectOptionalitySpec ::= OPTIONAL | DEFAULT Object**

El nombre del campo es "objectfieldreference". "DefinedObjectClass" hace referencia a la clase de objeto contenido en el campo (que puede ser la "ObjectClass" que se está definiendo). Si está presente, "ObjectOptionalitySpec" especifica

que el campo puede estar no especificado en una definición de objeto de información o, en el caso de "DEFAULT", esa omisión produce el siguiente "Object" (véase 11.2) que será la "DefinedObjectClass".

**9.12** ObjectSetFieldSpec" especifica que el campo es un campo de conjunto de objetos de información (véase 3.4.12):

```
ObjectSetFieldSpec ::=
    objectsetfieldreference
    DefinedObjectClass
    ObjectSetOptionalitySpec?
```

```
ObjectSetOptionalitySpec ::= OPTIONAL | DEFAULT ObjectSet
```

El nombre de este campo es "objectsetfieldreference". "DefinedObjectClass" hace referencia a la clase de objetos contenidos en el campo. Si está presente, "ObjectSetOptionalitySpec" especifica que el campo puede estar no especificado en una definición de objeto de información o, en el caso de "DEFAULT", esa omisión produce el siguiente "ObjectSet" (véase 12.2), cuyos objetos todos serán de "DefinedObjectClass".

**9.13** El constructivo "PrimitiveFieldName" se utiliza para identificar un campo con respecto a la clase que contiene su especificación:

```
PrimitiveFieldName ::=
    typefieldreference |
    valuefieldreference |
    valuesetfieldreference |
    objectfieldreference |
    objectsetfieldreference
```

Los nombres de todos los campos especificados en la definición de clase serán distintos.

**9.14** El constructivo "FieldName" se utiliza para identificar un campo con respecto a alguna clase que contiene la especificación de campo directamente o que tiene una cadena de campos de enlace con la clase contenedora. La cadena se indica mediante una lista de "PrimitiveFieldName" separados por puntos:

```
FieldName ::= PrimitiveFieldName "." +
```

**9.15** Si hay alguna cadena (de longitud uno o más) de especificaciones de campos de enlace (véase 3.4.14) de modo que:

- el primero está en la clase que se está definiendo; y
- cada uno subsiguiente es un campo de la clase utilizada para definir la anterior; y
- el último se define utilizando la clase que se está definiendo,

entonces por lo menos una de las especificaciones de campo tendrá una "ObjectOptionalitySpec" u "ObjectSetOptionalitySpec" (según proceda).

NOTA – Esto es para evitar definiciones de clases de objetos de información recursivas que no tienen representación finita para un objeto de información de esa clase recursiva.

## 9.16 Ejemplos

Una versión ampliada de la clase de objeto de información descrita informalmente como un ejemplo en 3.4.9 podrá definirse como sigue:

```
OPERATION ::= CLASS
{
    &ArgumentTypeOPTIONAL,
    &ResultType    OPTIONAL,
    &Errors          ERROR OPTIONAL,
    &Linked          OPERATION OPTIONAL,
    &resultReturnedBOOLEAN DEFAULT TRUE,
    &code            INTEGER UNIQUE
}

ERROR ::= CLASS
{
    &ParameterType    OPTIONAL,
    &code              INTEGER UNIQUE
}
```

NOTA 1 – Este ejemplo se basa en los conceptos de operación y error de la norma relativa a operaciones a distancia, pero simplificada para los fines presentes.

NOTA 2 – Los campos especificados para esta clase comprenden dos tipos de campo (&ArgumentType y &ResultType), dos campos de conjunto de objetos (&Errors y &Linked) y dos campos de valor (&resultReturned y &code) y el último es un campo de identificador.

NOTA 3 – Cualquier conjunto de objetos de información formado de OPERATION debe ser tal que ninguno de dos objetos del conjunto tenga el mismo valor para el campo &code. (Lo mismo se aplica a conjuntos de objeto de ERROR.)

NOTA 4 – La clase de objeto de información OPERATION incluye una cadena de campos de enlace como se describe en 9.15. La cadena es de longitud uno y está formada por el campo &Linked, que se especifica (recursivamente) por medio de OPERATION. Sin embargo, esto es bastante válido, porque el campo está designado OPTIONAL.

NOTA 5 – Ninguno de estos ejemplos incluye una "WithSyntaxSpec". No obstante, en 10.13 se proporcionan ejemplos que sí la incluyen.

## 10 Lista de sintaxis

**10.1** Frecuentemente ocurre que una sola especificación define una clase de objeto de información, para la cual muchas otras especificaciones independientes definen separadamente objetos de información. Puede ser conveniente para el definidor de la clase proporcionar una notación fácil para el usuario, para la definición de objetos de información en esa clase.

NOTA – Es para esto que la "notación macro" de ASN.1 (histórica) se utilizaba principalmente, antes de su sustitución por el texto que figura en esta Recomendación | Norma Internacional.

**10.2** Esta cláusula especifica una notación por la cual el especificador de una clase de objeto de información define la sintaxis definida específica de la clase para la especificación de objetos de información de esa clase.

**10.3** La notación es el constructivo sintáctico "SyntaxList", que aparece en el constructivo sintáctico "ObjectClassDefn".

**10.4** "SyntaxList" especifica la sintaxis para la definición de un objeto de información de la clase que se define. La sintaxis aparece como la "DefinedSyntax" en la siguiente subcláusula.

NOTA – Es una propiedad de esta especificación que el final de cualquier constructivo sintáctico definido por una "SyntaxList" (un caso de "DefinedSyntax") puede ser determinado:

- a) pasando por alto comentarios ASN.1;
- b) tratando los valores de cadena de caracteres como testigos léxicos;
- c) previendo una "{" inicial, concordando con "{" and "}" jerarquizados, y terminando en una "}" no concordada.

**10.5** "SyntaxList" especifica la secuencia de "DefinedSyntaxToken" que ha de aparecer en "DefinedSyntax" (véase 11.5):

```
SyntaxList ::= "{" TokenOrGroupSpec empty + "}"
TokenOrGroupSpec ::= RequiredToken | OptionalGroup
OptionalGroup ::= "[" TokenOrGroupSpec empty + "]"
RequiredToken ::=
    Literal |
    PrimitiveFieldName
```

NOTA – Al escritor de "SyntaxList" **no** se le da toda la potencia de BNF. Aproximadamente, la potencia notacional es equivalente a la que se utiliza corrientemente para especificar sintaxis de línea de instrucciones para intérpretes de instrucciones. La lista de posibles "RequiredToken" se da en el orden en que están permitidos; uno o más testigos consecutivos pueden hacerse facultativos colocándolos entre corchetes.

**10.6** Un testigo "word" utilizado como un "Literal" no será uno de los siguientes:

```
BIT
BOOLEAN
CHARACTER
CHOICE
EMBEDDED
END
ENUMERATED
EXTERNAL
FALSE
INSTANCE
INTEGER
INTERSECTION
```

**MINUS-INFINITY**  
**NULL**  
**OBJECT**  
**OCTET**  
**PLUS-INFINITY**  
**REAL**  
**SEQUENCE**  
**SET**  
**TRUE**  
**UNION**

NOTA – Esta lista comprende solamente todas las palabras reservadas de la ASN.1 que pueden aparecer como el primer ítem de un "Type", "Value", "ValueSet", "Object" u "ObjectSet" y también la palabra reservada "END". La utilización de otras palabras de ASN.1 no produce ambigüedad y se permite. Cuando se emplea la sintaxis definida en un entorno en el cual una "word" es también una "typereference" u "objectsetreference", la utilización de una "word" tiene precedencia.

**10.7** "Literal" especifica la inclusión real de ese "Literal", una "word" o una coma (","), en esa posición en la sintaxis definida:

**Literal ::=**  
     **word |**  
     **","**

**10.8** Cada "PrimitiveFieldName" especifica la inclusión (en esa posición en la nueva sintaxis) de "Setting" (véase 11.6) para el campo correspondiente.

**10.9** Cada "PrimitiveFieldName" de la clase de objeto de información aparecerá exactamente una vez.

**10.10** Cuando, en el proceso de análisis se encuentra "OptionalGroup" y el siguiente ítem ASN.1 es sintácticamente aceptable como el primer ítem ASN.1 en el grupo facultativo, se supone que ese grupo está presente. Si no es sintácticamente aceptable como el primer ítem ASN.1 en el grupo facultativo, se supone que el grupo está ausente.

NOTA – Para evitar efectos imprevistos, los diseñadores deben normalmente hacer el primer ítem ASN.1 en un grupo facultativo un "Literal".

**10.11** Un caso de utilización de "DefinedSyntax" no es válido a menos que especifique todos los campos obligatorios para la clase de objeto de información.

**10.12** Para asegurar un análisis fácil de la nueva sintaxis y evitar abusos, se imponen las siguientes restricciones adicionales al definidor de nueva sintaxis:

- a) Cada "OptionalGroup" tiene que tener por lo menos un "PrimitiveFieldName" u "OptionalGroup" dentro de él.

NOTA 1 – Esto ayuda a evitar la recopilación aparente de información que no está reflejada en ningún campo del objeto de información.

- b) La utilización de "OptionalGroup" será tal que en ningún momento del proceso de análisis pueda aparecer "Setting" que pudiera ser posiblemente una fijación para más de un "FieldName".
- c) Si un "OptionalGroup" comienza con un "Literal", el primer testigo que sigue a "OptionalGroup" será también un "Literal" y debe ser diferente del primer "Literal" de todos los "OptionalGroup" inmediatamente precedentes,

mientras que se impone la siguiente restricción al usuario de "DefinedSyntax":

- d) Siempre que un "Literal" esté presente en una "DefinedSyntax", que aparece en un "OptionalGroup", entonces debe estar presente también "Setting" para un "PrimitiveFieldName" en ese "OptionalGroup".

NOTA 2 – Esto ayuda a evitar la recopilación aparente de información que no está reflejada en ningún campo del objeto de información.

NOTA 3 – El siguiente ejemplo es una sintaxis legal pero la restricción d) impide que el usuario escriba "LITERAL" sin que vaya seguido de uno o ambos de los grupos facultativos.

[LITERAL [A &field] [B &field2]]

### 10.13 Ejemplos

Los ejemplos de definiciones de clase de 9.16 anterior pueden estar dotados con sintaxis definida para proporcionar una manera "fácil para el usuario" de definir casos de las clases. (Esta sintaxis definida se utiliza en el ejemplo indicado en 11.10.)

```

OPERATION ::= CLASS
{
    &ArgumentType OPTIONAL,
    &ResultType    OPTIONAL,
    &Errors          ERROR OPTIONAL,
    &Linked          OPERATION OPTIONAL,
    &resultReturned BOOLEAN DEFAULT TRUE,
    &operationCode  INTEGER UNIQUE
}
WITH SYNTAX
{
    [ARGUMENT          &ArgumentType]
    [RESULT            &ResultType]
    [RETURN RESULT    &resultReturned]
    [ERRORS           &Errors]
    [LINKED           &Linked]
    CODE              &operationCode
}
ERROR ::= CLASS
{
    &ParameterType  OPTIONAL,
    &errorCode       INTEGER UNIQUE
}
WITH SYNTAX
{
    [PARAMETER        &ParameterType]
    CODE              &errorCode
}

```

## 11 Definición y asignación de objetos de información

**11.1** El constructivo sintáctico "ObjectAssignment" se utiliza para asignar un objeto de información de una clase especificada a un nombre de referencia ("objectreference"). Este constructivo es una de las alternativas para "Assignment" en la Rec. UIT-T X.680 | ISO/CEI 8824-1, cláusula 12, y se define como sigue:

```

ObjectAssignment ::=
    objectreference
    DefinedObjectClass
    "::="
    Object

```

**11.2** El objeto de información, que será de la clase referenciada por "DefinedObjectClass", es el definido por el constructivo "Object".

```

Object ::=
    DefinedObject |
    ObjectDefn |
    ObjectFromObject |
    ParameterizedObject

```

Si "Object" es:

- "DefinedObject", el objeto es igual al referenciado;
- "ObjectDefn", el objeto es como se especifica en 11.3;
- "ObjectFromObject", el objeto es como se especifica en la cláusula 15;
- "ParameterizedObject", el objeto se define como se especifica en la Rec. UIT-T X.683 | ISO/CEI 8824-4, 9.2.

**11.3** Cada objeto de información a la larga es definido por "ObjectDefn":

**ObjectDefn ::=**  
**DefaultSyntax |**  
**DefinedSyntax**

"ObjectDefn" será "DefaultSyntax" (véase 11.4) si la definición de clase no incluye "WithSyntaxSpec" y será "DefinedSyntax" (véase 11.5) si incluye una.

**11.4** El constructivo "DefaultSyntax" se define como sigue:

**DefaultSyntax ::= "{" FieldSetting "," \* "}"**  
**FieldSetting ::= PrimitiveFieldName Setting**

Habrá exactamente una "FieldSetting" para cada "FieldSpec" en la definición de clase que no es "OPTIONAL" y que no tiene "DEFAULT", y como máximo una "FieldSetting" para cada otra "FieldSpec". "FieldSetting" puede aparecer en cualquier orden. El "PrimitiveFieldName" en cada "FieldSetting" será el nombre de la "FieldSpec" correspondiente. El constructivo "Setting" se especifica en 11.6.

**11.5** El constructivo "DefinedSyntax" se define como sigue:

**DefinedSyntax ::= "{" DefinedSyntaxToken empty \* "}"**  
**DefinedSyntaxToken ::=**  
**Literal |**  
**Setting**

"SyntaxList" en "WithSyntaxSpec" (véase la cláusula 10) determina la secuencia de los "DefinedSyntaxToken" que han de aparecer en "DefinedSyntax". El constructivo "Setting" se especifica en 11.6; cada aparición especifica la fijación para algún campo de objeto de información. El constructivo "Literal" se define en 10.7; los "Literal" están presentes para la legibilidad por seres humanos.

**11.6** "Setting" especifica la fijación de algún campo dentro de un objeto de información que se define:

**Setting ::=**  
**Type |**  
**Value |**  
**ValueSet |**  
**Object |**  
**ObjectSet**

Si el campo es:

- a) un campo de tipo, se seleccionará la alternativa "Type";
- b) un campo de valor, se seleccionará la alternativa "Value";
- c) un campo de conjunto de valores, se seleccionará la alternativa "ValueSet";
- d) un campo de objetos de información, se seleccionará la alternativa "Object";
- e) un campo de conjunto de objetos de información, se seleccionará la alternativa "ObjectSet".

NOTA – La fijación está restringida además como se describe en la subcláusula apropiada de 9.5 a 9.12 anteriores y 11.7 a 11.8.

**11.7** Una fijación de un campo de valor de tipo variable será un valor del tipo especificado por el campo de tipo apropiado del mismo objeto u objeto enlazado (es decir, no se emplea la notación de valor para el tipo abierto).

**11.8** Una fijación de un campo de conjunto de valores de tipo variable será un conjunto de valores del tipo especificado por el campo de tipo apropiado del mismo objeto u objeto enlazado (es decir, no se emplea la notación de valor para el tipo abierto).

## 11.9 Ejemplos (sintaxis por defecto)

Dadas las definiciones de clase de objeto de información de 9.16 anterior (que no incluyen "WithSyntaxSpec"), se definen casos de clases utilizando la "DefaultSyntax". Por ejemplo (una versión ampliada del ejemplo dado en 3.4.8):

```
invertMatrix OPERATION ::=
{
    &ArgumentTypeMatrix,
    &ResultType           Matrix,
    &Errors                {determinantIsZero},
    &operationCode         7
}

determinantIsZero ERROR ::=
{
    &errorCode            1
}
```

## 11.10 Ejemplos (sintaxis definida)

En 10.13 se proporcionan clases de ejemplo "WithSyntaxSpec" y por tanto se definen casos de las clases utilizando la "DefinedSyntax". Por consiguiente, los ejemplos indicados en 11.9 se escribirían:

```
invertMatrix OPERATION ::=
{
    ARGUMENT Matrix
    RESULT      Matrix
    ERRORS      {determinantIsZero}
    CODE        7
}

determinantIsZero ERROR ::=
{
    CODE        1
}
```

## 12 Definición y asignación de conjunto de objetos de información

**12.1** El constructivo sintáctico "ObjectSetAssignment" se utiliza para asignar un conjunto de objetos de información de una clase especificada a un nombre de referencia ("objectsetreference"). Este constructivo es una de las alternativas para "Assignment" en la Rec. UIT-T X.680 | ISO/CEI 8824-1, cláusula 12 que se define como sigue:

```
ObjectSetAssignment ::=
    objectsetreference
    DefinedObjectClass
    "::="
    ObjectSet
```

**12.2** El conjunto de objetos de información será de la clase referenciada por "DefinedObjectClass" y es definido por el constructivo "ObjectSet":

```
ObjectSet ::= "{" ObjectSetSpec "}"
ObjectSetSpec ::= ElementSetSpecs | "..."
```

"ElementSetSpecs" se especifica en la Rec. UIT-T X.680 | ISO/CEI 8824-1 y permite especificar un conjunto de objetos de información en términos de objetos de información o conjunto de los mismos de la clase rectora. Habrá por lo menos un objeto de información en el conjunto a no ser que se especifique la segunda alternativa de "ObjectSetSpec". En este segundo caso, la presencia de puntos suspensivos es una indicación de que el conjunto de objetos está vacío inicialmente pero el programa de aplicación le va añadir objetos dinámicamente.

NOTA – A diferencia de los tipos ampliables tales como SET o SEQUENCE, o de las constricciones de subtipos variables, que son estáticos con respecto al conjunto de valores "comprendidos" que se fijan para cada versión de la especificación ASN.1, un objeto ampliable puede crecer y contraerse dinámicamente en una determinada versión. De hecho, puede ampliarse y reducirse dentro de un ejemplar dado de utilización de programa de aplicación a medida que define o no define objetos dinámicamente.

**12.3** Si un conjunto de objetos ampliable, A, es referenciado en la definición de otro conjunto de objetos, B, un marcador de ampliación es heredado por B.

**12.4** Si un "ValueSetFromObjects" (véase la cláusula 15) se define utilizando un conjunto de objetos ampliable, el conjunto de valores resultante no hereda el marcador de ampliación del conjunto de objetos.

**12.5** Si un tipo está constreñido por una restricción de cuadro (véase 10.3 de la Rec. UIT-T X.682 | ISO/CEI 8824-3) y el conjunto de objetos referenciado en el cuadro es ampliable, el tipo no hereda el marcador de ampliación del conjunto de objetos. Si el tipo ha de ser ampliable, se añadirá explícitamente un marcador de ampliación a su "ElementSetSpecs".

**12.6** La notación para "ObjectSetElements" es la siguiente:

```
ObjectSetElements ::=
    Object |
    DefinedObjectSet |
    ObjectSetFromObjects |
    ParameterizedObjectSet
```

Los elementos especificados por esta notación son determinados por la alternativa que se emplea, como sigue:

- a) si se utiliza la alternativa "Object", sólo se especifica el objeto así designado. Ese objeto será de la clase rectora;
- b) si se utiliza alguna de las alternativas restantes, se especifican todos los objetos de conjunto así designados. Los objetos serán de la clase rectora. Si se utiliza la alternativa "DefinedObjectSet", el conjunto de objetos es el referenciado. Si se utiliza la alternativa "ObjectSetFromObjects", el conjunto de objetos es el especificado en la cláusula 15. Si se utiliza la alternativa "ParameterizedObjectSet", el conjunto de objetos es el que se especifica en la Rec. UIT-T X.683 | ISO/CEI 8824-4, 9.2.

## 12.7 Ejemplo

El conjunto de objetos de información descrito informalmente en la nota de 3.4.11 se puede especificar como sigue:

```
MatrixOperations OPERATION ::=
{
    invertMatrix |
    addMatrices |
    subtractMatrices |
    multiplyMatrices
}
```

## 13 Cuadros asociados

**13.1** Cada objeto de información o conjunto de objetos de información puede ser considerado como un cuadro: su cuadro asociado. Cada casilla del cuadro asociado corresponde a la fijación de algún campo de un objeto de información, o está vacía. El conjunto de columnas del cuadro asociado es determinado por la clase a la cual pertenece el objeto u objetos; sin embargo, el conjunto de filas es determinado por el objeto u objetos en cuestión.

**13.2** Dada la definición de una clase, el conjunto de columnas se determina como sigue:

- a) hay una columna para cada especificación de campo en la definición de clase. Cada una de estas columnas es denominada por el correspondiente "PrimitiveFieldName";
- b) hay un conjunto adicional de columnas correspondientes a cada especificación de campo de enlace. Este conjunto de columnas es el determinado por la aplicación de estas reglas para la clase rectora del campo de enlace, salvo que sus nombres están prefijados por el "PrimitiveFieldName" del campo de enlace, y un punto (".").

NOTA – Estas reglas son recursivas, y son tales que si una clase es directa o indirectamente autorreferente, el conjunto de columnas no es finito. Esto no está prohibido.

**13.3** Dado un objeto de información de alguna clase, el cuadro asociado es el que resultaría de la aplicación de 13.4 al conjunto de objetos que contienen precisamente ese objeto.

**13.4** Dado un conjunto de objetos de información de alguna clase, el conjunto de filas en el cuadro asociado son las que resultarían de la aplicación del siguiente procedimiento recursivo:

- a) Comiencese con una fila para cada objeto en el conjunto de objetos. En cada una de estas filas, las casillas en las columnas denominadas por "PrimitiveFieldName" corresponderán a la fijación del campo apropiado en el objeto, mientras que las otras casillas estarán vacías.
- b) Para cada campo de enlace que aparece en alguna fila del conjunto:
  - 1) Genérese el cuadro asociado (subordinado) del contenido del campo de enlace.
  - 2) Sustitúyase el renglón en el cual aparece el campo de enlace por una colección de filas, una para cada fila del cuadro asociado subordinado. Cada una de las filas de esta colección es igual que la que se sustituye, salvo que las casillas de la fila seleccionada del cuadro asociado subordinado se utilizan para rellenar las casillas correspondientes, hasta ahora vacías, cuyos "FieldName" están prefijados por "PrimitiveFieldName" del campo de enlace.

NOTA – Estas reglas son recursivas y son tales que si un objeto de información es directa o indirectamente autorreferente el procedimiento no terminará. Esto no está prohibido. En la práctica, sólo es necesario conocer el contenido de las casillas con nombres de una longitud finita y para esto puede diseñarse un procedimiento limitado.

### 13.5 Ejemplos de "FieldName" válidos

Los siguientes "FieldName" están entre aquellos que son válidos para el cuadro asociado para objetos o conjuntos de objetos de la clase OPERATION (definida en 10.13):

**&ArgumentType**  
**&Errors.&Parameter**  
**&Errors.&errorCode**  
**&Linked.&ArgumentType**  
**&Linked.&Linked.&operationCode**  
**&Linked.&Linked.&Linked.&Linked.&Linked.&Errors.&errorCode**

Como la clase OPERATION es autorreferente (mediante el campo &Linked) no hay un número finito de columnas.

## 14 Notación para el tipo de campo de clase de objeto

El tipo que es referenciado por esta notación depende de la categoría del nombre de campo. Para las diferentes categorías de nombres de campo, en 14.2 a 14.5 se especifica el tipo que es referenciado.

**14.1** La notación para un tipo de campo de clase de objeto (véase 3.4.15) será "ObjectClassFieldType":

**ObjectClassFieldType ::=**  
**DefinedObjectClass**  
**"."**  
**FieldName**

donde "FieldName" es el especificado en 9.14 con respecto a la clase identificada por "DefinedObjectClass".

**14.2** Para un campo de tipo, la notación define un tipo abierto, es decir, uno cuyo conjunto de valores es el conjunto completo de todos los valores posibles que se pueden especificar utilizando la ASN.1. La especificación de constricciones empleando un conjunto de objetos de información correspondiente (véase la Rec. UIT-T X.682 | ISO/CEI 8824-3) puede restringir este tipo a un tipo específico. Las siguientes constricciones a la utilización de esta notación se aplican cuando "FieldName" hace referencia a un campo de tipo:

- a) Esta notación no se utilizará directa o indirectamente en la definición del tipo de un campo de valor o campo de conjunto de valores de una clase de objeto de información.
- b) Esta notación tiene un rótulo indeterminado y por lo tanto no se puede utilizar cuando se requiere un rótulo distinto del de algún otro tipo.

NOTA 1 – Esta restricción se puede evitar normalmente rotulando (explícitamente) el tipo.

NOTA 2 – Con independencia de lo establecido en 47.7.3 de la Rec. UIT-T X.680 | ISO/CEI 8824-1 que estipula que el elemento añadido conceptualmente para un marcador de extensión tenga un rótulo distinto del de rótulos de todos los tipos ASN.1 conocidos, el tipo abierto no debe utilizarse donde se requiera que tenga un rótulo distinto del elemento añadido conceptualmente.

- c) Esta notación no se rotulará implícitamente.

NOTA 3 – El motivo es que cuando este tipo abierto se restringe a un tipo particular, ese tipo puede ser un tipo de opción.

- d) Se requieren reglas de codificación para codificar el valor asignado a un componente definido de esta manera, de modo que un receptor pueda determinar satisfactoriamente los valores abstractos correspondientes a todas las otras partes de la construcción en la cual está insertado el componente sin ningún conocimiento del tipo real de este componente.

NOTA 4 – Este constructivo "Type" estará constreñido corrientemente por el uso de un conjunto de objetos de información y "AtNotation", como se especifica en la Rec. UIT-T X.682 | ISO/CEI 8824-3, cláusula 10. No obstante, se avisa a los usuarios de la ASN.1 que la utilización de esta notación sin la aplicación de una restricción puede conducir a ambigüedad en requisitos de implementación, por lo que debe evitarse normalmente.

**14.3** Para un campo de valor de tipo fijo o un campo de conjunto de valores, la notación denota el "Type" que aparece en la especificación de ese campo en la definición de la clase objeto de información.

**14.4** Para un campo de valor de tipo variable o un campo de conjunto de valores de tipo variable, la notación define un tipo abierto. Su utilización está sujeta a las mismas restricciones especificadas en 14.2.

**14.5** No se permite esta notación si el campo es un campo de objeto o un campo de conjunto de objetos.

**14.6** La notación para definir un valor de este tipo será "ObjectClassFieldValue":

**ObjectClassFieldValue ::=**

**OpenTypeFieldVal |**

**FixedTypeFieldVal**

**OpenTypeFieldVal ::= Type ":" Value**

**FixedTypeFieldVal ::= BuiltinValue | ReferencedValue**

**14.7** Para un campo de valor de tipo fijo o un campo de conjunto de valores empleado en "ObjectClassFieldType", se utilizará "FixedTypeFieldVal" y será un valor del "Type" especificado en la definición de la clase de objeto de información.

**14.8** Para un campo de tipo o un campo de valor de tipo variable o un campo de conjunto de valores utilizado en "ObjectClassFieldType", se empleará "OpenTypeFieldVal". El "Type" en "OpenTypeFieldVal" será cualquier tipo ASN.1, y "Value" será cualquier valor de ese tipo.

### **14.9 Ejemplo de utilización de "ObjectClassFieldType"**

Cada uno de los ejemplos siguientes se basa en el ejemplo de 10.13 y muestran: (a) un posible "ObjectClassFieldType", (b) el tipo al cual el tipo del ejemplo (a) es equivalente (cuando se utiliza sin constricciones) y (c) la notación para un ejemplo de valor de ese tipo.

- 1 (a) OPERATION.&operationCode  
(b) INTEGER  
(c) 7
- 2 (a) OPERATION.&ArgumentType  
(b) *open type*  
(c) Matrix:  
{ { 1, 0, 0, 0 },  
{ 0, 1, 0, 0 },  
{ 0, 0, 1, 0 },  
{ 0, 0, 0, 1 } }
- 3 (a) OPERATION.&Linked.&Linked.&Errors.&errorCode  
(b) INTEGER  
(c) 1
- 4 (a) OPERATION.&Linked.&ArgumentType  
(b) *open type*  
(c) UniversalString: { planckConstant, " and ", hamiltonOperator }

## 15 Información procedente de objetos

**15.1** La información de la columna del cuadro asociado para un objeto o un conjunto de objetos puede ser referenciada por los distintos casos de la notación "InformationFromObjects":

```

InformationFromObjects ::=
    ValueFromObject           |
    ValueSetFromObjects       |
    TypeFromObject            |
    ObjectFromObject          |
    ObjectSetFromObjects

ValueFromObject ::=
    ReferencedObjects
    "."
    FieldName

ValueSetFromObjects ::=
    ReferencedObjects
    "."
    FieldName

TypeFromObject ::=
    ReferencedObjects
    "."
    FieldName

ObjectFromObject ::=
    ReferencedObjects
    "."
    FieldName

ObjectSetFromObjects ::=
    ReferencedObjects
    "."
    FieldName

ReferencedObjects ::=
    DefinedObject | ParameterizedObject |
    DefinedObjectSet | ParameterizedObjectSet

```

**15.2** Esta notación hace referencia al contenido total de la columna referenciada del cuadro asociado para "ReferencedObjects".

**15.3** Según la forma de "ReferencedObjects" y "FieldName", esta notación puede denotar un valor, un conjunto de valores, un tipo, un objeto o un conjunto de objetos. Estos cinco casos son denotados por los constructivos "ValueFromObject", "ValueSetFromObjects", "TypeFromObject", "ObjectFromObject" y "ObjectSetFromObjects", respectivamente. Cada uno de estos constructivos es un caso especial de "InformationFromObjects".

**15.4** La producción "InformationFromObjects" se puede dividir en dos partes. La primera parte se forma suprimiendo el final (o solamente) "PrimitiveFieldName" y su punto precedente. Si la primera parte denota un objeto o un conjunto de objetos, se aplican las subcláusulas 15.5 a 15.9. En los demás casos, la notación es ilegal. La segunda parte es el final (o solamente) "PrimitiveFieldName".

NOTA – (didáctica) Dada la siguiente definición:  
obj.&a.&b.&c.&d

La primera parte de la definición es obj.&a.&b.&c y la segunda parte es &d.

**15.5** La primera columna indica la primera del cuadro 1 parte definida en 15.4. La segunda columna indica la segunda parte definida en 15.4. La tercera columna indica cuál de los cinco casos (si hubiere alguno) de "InformationFromObjects" (enumerados en 15.3) se aplica.

**Cuadro 1 – Casos admisibles de "InformationFromObjects"**

Primera parte de InformationFromObjects	Segunda parte de InformationFromObjects	Constructivo
objeto	campo de valor de tipo fijo	"ValueFromObject"
	campo de valor de tipo variable	"ValueFromObject"
	campo de conjunto de valores de tipo fijo	"ValueSetFromObjects"
	campo de conjunto de valores de tipo variable	"ValueSetFromObjects"
	campo de tipo	"TypeFromObject"
	campo de objeto	"ObjectFromObject"
	campo de conjunto de objetos	"ObjectSetFromObjects"
conjunto de objetos	campo de valor de tipo fijo	"ValueSetFromObjects"
	campo de valor de tipo variable	no permitido
	campo de conjunto de valores de tipo fijo	"ValueSetFromObjects"
	campo de conjunto de valores de tipo variable	no permitido
	campo de tipo	no permitido
	campo de objeto	"ObjectSetFromObjects"
	campo de conjunto de objetos	"ObjectSetFromObjects"

**15.6** Si hay conjuntos de objetos y el "PrimitiveFieldName" final identifica un campo de conjunto de valores de tipo fijo, "ValueSetFromObjects" es la unión de los conjuntos de los valores seleccionados.

**15.7** Si hay conjuntos de objetos y el "PrimitiveFieldName" final identifica un campo de conjunto de objetos, "ObjectSetFromObjects" es la unión de los conjuntos de objetos seleccionados.

**15.8** Como se muestra en el cuadro 1, no se permite usar la notación si interviene un conjunto de objetos y el "PrimitiveFieldName" final identifica un campo de valor de tipo variable o un campo de conjunto de valores o un campo de tipo.

**15.9** No se permite utilizar esta notación si todas las casillas en la columna referenciada están vacías, salvo cuando se utiliza para definir directamente un campo de un objeto de información que es "OPTIONAL" (o "DEFAULT"), lo que da como resultado que el campo se convierte en vacío (o por defecto).

### 15.10 Ejemplo de información de objetos

Habida cuenta de las definiciones dadas en los ejemplos de 11.9, 11.10 y 12.7, los siguientes constructivos (en la columna de la izquierda) son válidos y se pueden utilizar como equivalentes a la expresión que figura en la columna de la derecha.

**"ValueFromObject"**

invertMatrix.&operationCode	7
determinantIsZero.&errorCode	1

**"TypeFromObject"**

invertMatrix.&ArgumentType	Matrix
----------------------------	--------

**"ValueSetFromObjects"**

invertMatrix.&Errors.&errorCode	{ 1 }
MatrixOperations.&operationCode	{7   <i>and others</i> }

**"ObjectSetFromObjects"**

invertMatrix.&Errors	{ determinantIsZero }
MatrixOperations.&Errors	{ determinantIsZero   <i>and others</i> }

## Anexo A

### Clase de objeto de información TYPE-IDENTIFIER

(Este anexo es parte integrante de la presente Recomendación | Norma Internacional)

**A.1** Este anexo especifica una clase de objeto de información útil, con la referencia de clase TYPE-IDENTIFIER (identificador de tipo).

NOTA – Esta clase de objeto de información es la clase útil más simple, que sólo tiene dos campos: un campo de identificador de tipo OBJECT IDENTIFIER y un campo de tipo que define el tipo ASN.1 para transportar toda la información relativa a cualquier objeto particular de la clase. Se define en esta Recomendación | Norma Internacional debido a la utilización generalizada de objetos de información de esta forma.

**A.2** La clase de objeto de información TYPE-IDENTIFIER se define como:

```
TYPE-IDENTIFIER ::= CLASS
{
    &id OBJECT IDENTIFIER UNIQUE,
    &Type
}
WITH SYNTAX {&Type IDENTIFIED BY &id}
```

**A.3** Esta clase se define como una clase de objeto de información "útil" y está disponible en cualquier módulo sin necesidad de importarla.

#### A.4 Ejemplo

El cuerpo de una comunicación del sistema de tratamiento de mensajes se puede definir como:

```
MHS-BODY-CLASS ::= TYPE-IDENTIFIER
g4FaxBody MHS-BODY-CLASS ::=
{BIT STRING IDENTIFIED BY {mhsbody 3}}
```

Un diseñador de protocolo definiría típicamente un componente para transportar una MHS-BODY-CLASS especificando el tipo "INSTANCE OF MHS-BODY-CLASS" definido en C.9.

## Anexo B

### Definición de sintaxis abstracta

(Este anexo es parte integrante de la presente Recomendación | Norma Internacional)

**B.1** Este anexo especifica una clase de objeto de información ABSTRACT-SYNTAX, para definir sintaxis abstractas.

NOTA – Se recomienda que se defina un caso de esta clase de objeto de información siempre que se defina una sintaxis abstracta como los valores de un solo tipo ASN.1.

**B.2** La clase de objeto de información ABSTRACT-SYNTAX se define como:

```

ABSTRACT-SYNTAX ::= CLASS
{
    &id          OBJECT IDENTIFIER UNIQUE,
    &Type,
    &property    BIT STRING {handles-invalid-encodings(0)} DEFAULT {}
}
WITH SYNTAX {
    &Type IDENTIFIED BY &id [HAS PROPERTY &property]
}

```

El campo &id de cada ABSTRACT-SYNTAX es el nombre de sintaxis abstracta, mientras que el campo &Type contiene el tipo ASN.1 cuyos valores forman la sintaxis abstracta. La propiedad "handles-invalid-encodings" indica que las codificaciones no válidas no se han de tratar como un error durante el proceso de decodificación, y la decisión sobre cómo tratar esas codificaciones no válidas se deja a criterio de la aplicación.

**B.3** Esta clase se define como una clase de objeto de información "útil", y está disponible en cualquier módulo sin necesidad de importarla.

### B.4 Ejemplo

Si se ha definido un tipo ASN.1 denominado XXX-PDU, se puede especificar una sintaxis abstracta que contenga todos los valores de XXX-PDU mediante la notación:

```

xxx-Abstract-Syntax ABSTRACT-SYNTAX ::=
    { XXX-PDU IDENTIFIED BY {xxx 5} }

```

Para un ejemplo detallado de la utilización de la clase de objeto de información ABSTRACT-SYNTAX, véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, C.3.

**B.5** Ocurrirá frecuentemente que una sintaxis abstracta se definirá en términos de un tipo parametrizado (como se define en la Rec. UIT-T X.683 | ISO/CEI 8824-4), por ejemplo, con parámetros que proporcionan límites en algunos componentes del protocolo. Estos parámetros, sujetos a las restricciones especificadas en la Rec. UIT-T X.683 | ISO/CEI 8824-4, cláusula 10, se pueden resolver en el momento de la definición de la sintaxis abstracta, o se pueden transportar como parámetros de la sintaxis abstracta.

## Anexo C

### El tipo ejemplar de

(Este anexo es parte integrante de la presente Recomendación | Norma Internacional)

**C.1** Este anexo especifica la notación de tipo y de valor para los tipos ejemplar de (véase 3.4.13). Estos tipos son capaces de transportar cualquier valor de cualquier objeto de información en una clase de objeto de información definida como la clase TYPE-IDENTIFIER (véase el anexo A) utilizando una asignación de clase de objeto de información (la referencia de clase de objeto de información se especifica como parte de esta notación).

**C.2** En la Rec. UIT-T X.680 | ISO/CEI 8824-1, 16.2, se hace referencia a la notación "InstanceOfType" (tipo ejemplar de) como una de las notaciones que producen un "Type", y se define como:

**InstanceOfType ::= INSTANCE OF DefinedObjectClass**

NOTA – La Rec. UIT-T X.682 | ISO/CEI 8824-3, cláusula 10, especifica la manera en que este tipo puede ser constreñido aplicando una "constricción de tabla", que restringe los valores del tipo a los que representan algún conjunto de objeto de información específicos de la clase.

**C.3** Esta notación especifica un tipo que transporta el campo &id (un OBJECT IDENTIFIER) y un valor del campo &Type de cualquier caso de "DefinedObjectClass".

NOTA – Este constructivo estará constreñido normalmente por un conjunto de objetos que en general será (pero no es necesariamente) un nombre de referencia ficticio definido en la Rec. UIT-T X.683 | ISO/CEI 8824-4, 8.3 a 8.11, y el conjunto de objetos reales se define en otra parte.

**C.4** Todos los tipos ejemplar de tienen un rótulo, que es clase universal, número 8.

NOTA – Éste es el mismo rótulo universal que para el tipo externo y la utilización del tipo ejemplar de puede ser compatible bit por bit con el tipo externo cuando se usan las reglas de codificación básicas para ASN.1.

**C.5** El tipo ejemplar de tiene un tipo de secuencia asociada que se utiliza para definir valores y subtipos del tipo ejemplar de.

NOTA – Cuando este tipo está constreñido por la notación de constricción de la Rec. UIT-T X.682 | ISO/CEI 8824-3, el tipo de secuencia asociado está constreñido también. Las constricciones del tipo de secuencia asociado resultantes de una constricción del tipo ejemplar de se especifican en la Rec. UIT-T X.682 | ISO/CEI 8824-3, anexo A.

**C.6** Se supone que el tipo de secuencia asociado se define dentro de un entorno en el cual está vigente "EXPLICIT TAGS" (rótulos explícitos).

**C.7** El tipo de secuencia asociado será:

**SEQUENCE**

```
{
    type-id    <DefinedObjectClass>.&id,
    value      [0] <DefinedObjectClass>.&Type
}
```

donde "<DefinedObjectClass>" se sustituye por la "DefinedObjectClass" particular utilizada en la notación "InstanceOfType".

**C.8** La notación de valor "InstanceOfValue" para una notación "InstanceOfType" será la notación de valor para el tipo de secuencia asociado:

**InstanceOfValue ::= Value**

## C.9 Ejemplo

El siguiente ejemplo se basa en el ejemplo mostrado en A.4:

El tipo:

**INSTANCE OF MHS-BODY-CLASS**

tiene un tipo de secuencia asociado de:

**SEQUENCE**

```
{  
  type-id MHS-BODY-CLASS.&id,  
  value [0] MHS-BODY-CLASS.&Type  
}
```

En la Rec. UIT-T X.682 | ISO/CEI 8824-3, anexo A, figura un ejemplo de la aplicación de una construcción de tabla a este tipo.

## Anexo D

### Ejemplos

(Este anexo no es parte integrante de la presente Recomendación | Norma Internacional)

#### D.1 Ejemplo de utilización de clase OPERATION simplificada

Dada la siguiente definición simple de clases de objetos de información OPERATION y ERROR:

```

OPERATION ::= CLASS
{
    &ArgumentType      OPTIONAL,
    &ResultType        OPTIONAL,
    &Errors             ERROR OPTIONAL,
    &Linked            OPERATION OPTIONAL,
    &resultReturned    BOOLEAN DEFAULT TRUE,
    &operationCode     INTEGER UNIQUE
}

WITH SYNTAX
{
    [ARGUMENT          &ArgumentType]
    [RESULT            &ResultType]
    [RETURN RESULT    &resultReturned]
    [ERRORS           &Errors]
    [LINKED           &Linked]
    CODE              &operationCode
}

ERROR ::= CLASS
{
    &ParameterType    OPTIONAL,
    &errorCode         INTEGER UNIQUE
}

WITH SYNTAX
{
    [PARAMETER        &ParameterType]
    CODE              &errorCode
}

```

Se puede definir el siguiente conjunto de objetos que contiene dos objetos OPERATION:

```

My-Operations OPERATION ::= { operationA | operationB }

operationA OPERATION ::= {
    ARGUMENT          INTEGER
    ERRORS            { { PARAMETER INTEGER CODE 1000 } | { CODE 1001 } }
    CODE              1
}

operationB OPERATION ::= {
    ARGUMENT          IA5String
    RESULT            BOOLEAN
    ERRORS            { { CODE 1002 } | { PARAMETER IA5String CODE 1003 } }
    CODE              2
}

```

La extracción del conjunto de objetos ERROR a partir del conjunto de objetos anterior se hace de la siguiente manera:

```
My-OperationErrors ERROR ::= { My-Operations.&Errors }
```

El conjunto de objetos resultante es:

```
My-OperationErrors ERROR ::= {
  { PARAMETER INTEGER CODE 1000 } |
  { CODE 1001 } |
  { CODE 1002 } |
  { PARAMETER IA5String CODE 1003 }
}
```

La extracción del conjunto de códigos de error de los errores de las operaciones se hace de la siguiente manera:

```
My-OperationErrorCodes INTEGER ::= { My-Operations.&Errors.&errorCode }
```

El conjunto de valores resultante es:

```
My-OperationErrorCodes INTEGER ::= { 1000 | 1001 | 1002 | 1003 }
```

## D.2 Ejemplo de utilización de "ObjectClassFieldType"

El "ObjectClassFieldType" puede utilizarse en la especificación de tipos, por ejemplo:

```
-- "ObjectClassFieldType"s are extracted from this class.
-- Only the first five fields can be used in the extraction.
```

```
EXAMPLE-CLASS ::= CLASS {
  &TypeField                OPTIONAL,
  &fixedTypeValueField      INTEGER OPTIONAL,
  &variableTypeValueField   &TypeField OPTIONAL,
  &FixedTypeValueSetField   INTEGER OPTIONAL,
  &VariableTypeValueSetField &TypeField OPTIONAL,
  &objectField              SIMPLE-CLASS OPTIONAL,
  &ObjectSetField           SIMPLE-CLASS OPTIONAL
}
```

```
WITH SYNTAX {
  [TYPE-FIELD                &TypeField]
  [FIXED-TYPE-VALUE-FIELD    &fixedTypeValueField]
  [VARIABLE-TYPE-VALUE-FIELD &variableTypeValueField]
  [FIXED-TYPE-VALUE-SET-FIELD &FixedTypeValueSetField]
  [VARIABLE-TYPE-VALUE-SET-FIELD &VariableTypeValueSetField]
  [OBJECT-FIELD              &objectField]
  [OBJECT-SET-FIELD          &ObjectSetField]
}
```

```
SIMPLE-CLASS ::= CLASS {
  &value INTEGER
}
```

```
WITH SYNTAX {
  &value
}
```

```
-- This type contains components which are specified using "ObjectClassFieldType" notation.
-- In case of type fields and variable-type value and value set fields the resulting
-- component type is an open type. In case of fixed-type value and value set fields the
-- resulting component type is INTEGER.
-- NOTE – Constraints are omitted from all the following uses of "ObjectClassFieldType";
-- you normally will use constraints when referencing an "ObjectClassFieldType".
```

```
ExampleType ::= SEQUENCE {
  openTypeComponent1      EXAMPLE-CLASS.&TypeField,
  integerComponent1       EXAMPLE-CLASS.&fixedTypeValueField,
  openTypeComponent2      EXAMPLE-CLASS.&variableTypeValueField,
  integerComponent2       EXAMPLE-CLASS.&FixedTypeValueSetField,
  openTypeComponent3      EXAMPLE-CLASS.&VariableTypeValueSetField
}
```

```

exampleValue ExampleType ::= {
    openTypeComponent1          BOOLEAN : TRUE,
    integerComponent1           123,
    openTypeComponent2          IA5String : "abcdef",
    integerComponent2           456,
    openTypeComponent3          BIT STRING : '01010101'B
}

```

### D.3 Ilustración de utilización de objetos y conjuntos de objetos

En lo que sigue se utiliza la clase de objeto definida en D.2:

```

objectA EXAMPLE-CLASS ::= {
    FIXED-TYPE-VALUE-FIELD      123
    FIXED-TYPE-VALUE-SET-FIELD { 1 | 2 | 3 }
    OBJECT-FIELD                { 1 }
    OBJECT-SET-FIELD            { { 2 } | { 3 } }
}

```

```

objectB EXAMPLE-CLASS ::= {
    TYPE-FIELD                  IA5String
    FIXED-TYPE-VALUE-FIELD      456
    VARIABLE-TYPE-VALUE-FIELD   "abc"
    VARIABLE-TYPE-VALUE-SET-FIELD { "d" | "e" | "f" }
}

```

*-- The following object set contains two defined objects and one builtin object.*

```

ObjectSet EXAMPLE-CLASS ::= {
    objectA |
    objectB |
    {
        TYPE-FIELD              INTEGER
        FIXED-TYPE-VALUE-FIELD  789
        VARIABLE-TYPE-VALUE-SET-FIELD { 4 | 5 | 6 }
    }
}

```

*-- The following definitions extract information from the objects and the object set.*

```

integerValue INTEGER ::= objectA.&fixedTypeValueField

stringValue IA5String ::= objectB.&variableTypeValueField

IntegerValueSetFromObjectA INTEGER ::= { objectA.&FixedTypeValueSetField }

StringValueSet IA5String ::= { objectB.&VariableTypeValueSetField }

StringType ::= objectB.&TypeField

objectFromObjectA SIMPLE-CLASS ::= objectA.&objectField

ObjectSetFromObjectA SIMPLE-CLASS ::= { objectA.&ObjectSetField }

SetOfValuesInObjectSet INTEGER ::= { ObjectSet.&fixedTypeValueField }

SetOfValueSetsInObjectSet INTEGER ::= { ObjectSet.&FixedTypeValueSetField }

SetOfObjectsInObjectSet SIMPLE-CLASS ::= { ObjectSet.&objectField }

SetOfObjectSetsInObjectSet SIMPLE-CLASS ::= { ObjectSet.&ObjectSetField }

```

## Anexo E

### Anexo ilustrativo sobre el modelo ASN.1 de ampliación de conjunto de objetos

(Este anexo no es parte integrante de la presente Recomendación | Norma Internacional)

#### E.1 Conjuntos de objetos ampliables

Una especificación ASN.1 puede definir conjuntos de objetos de información y esos conjuntos de objetos se pueden marcar como ampliables mediante un marcador de ampliación. La utilización de un marcador de ampliación con conjuntos de objetos difiere de esa utilización con tipos en que a menudo indica que se requiere una aplicación para añadir/suprimir objetos a/de un conjunto de objetos. Las constricciones de cuadro y de relación de componentes no satisfechas no se consideran errores por sí mismas si el conjunto de objetos es ampliable. En tales casos, no es un error el hecho de que el valor del tipo referenciado no se encuentre en el conjunto de objetos, pero si se encuentra, debe satisfacerse la restricción impuesta al tipo referenciador.

## Anexo F

### Resumen de la notación

(Este anexo no es parte integrante de la presente Recomendación | Norma Internacional)

Los siguientes ítems se definen en la cláusula 7:

**objectclassreference**  
**objectreference**  
**objectsetreference**  
**typefieldreference**  
**valuefieldreference**  
**valuesetfieldreference**  
**objectsetfieldreference**  
**word**  
**CLASS**  
**INSTANCE**  
**SYNTAX**  
**UNIQUE**

Los siguientes ítems se definen en la Rec. UIT-T X.680 | ISO/CEI 8824-1 y se utilizan en esta Recomendación | Norma Internacional:

**empty**  
**modulereference**  
**::="**  
**{**  
**}**  
**"**  
**'**  
**:"**  
**["**  
**]"**  
**:"**  
**DEFAULT**  
**OF**  
**OPTIONAL**  
**WITH**

Las siguientes producciones se definen en la Rec. UIT-T X.680 | ISO/CEI 8824-1 y se utilizan en esta Recomendación | Norma Internacional:

**ElementSetSpec**  
**Type**  
**Value**  
**ValueSet**

Las siguientes producciones se definen en la Rec. UIT-T X.683 | ISO/CEI 8824-4 y se utilizan en esta Recomendación | Norma Internacional:

**ParameterizedObjectClass**  
**ParameterizedObjectSet**  
**ParameterizedObject**

Las siguientes producciones se definen en esta Recomendación | Norma Internacional:

**DefinedObjectClass ::=**  
     **ExternalObjectClassReference | objectclassreference | UsefulObjectClassReference**  
**ExternalObjectClassReference ::= modulereference "." objectclassreference**  
**UsefulObjectClassReference ::=**  
     **TYPE-IDENTIFIER |**  
     **ABSTRACT-SYNTAX**  
**ObjectClassAssignment ::= objectclassreference "::=" ObjectClass**  
**ObjectClass ::= DefinedObjectClass | ObjectClassDefn | ParameterizedObjectClass**  
**ObjectClassDefn ::= CLASS "{" FieldSpec "," + "}" WithSyntaxSpec?**

**FieldSpec ::=**  
 TypeFieldSpec |  
 FixedTypeValueFieldSpec |  
 VariableTypeValueFieldSpec |  
 FixedTypeValueSetFieldSpec |  
 VariableTypeValueSetFieldSpec |  
 ObjectFieldSpec |  
 ObjectSetFieldSpec

**PrimitiveFieldName ::=**  
 typefieldreference |  
 valuefieldreference |  
 valuesetfieldreference |  
 objectfieldreference |  
 objectsetfieldreference

**FieldName ::= PrimitiveFieldName "." +**

**TypeFieldSpec ::= typefieldreference TypeOptionalitySpec?**

**TypeOptionalitySpec ::= OPTIONAL | DEFAULT Type**

**FixedTypeValueFieldSpec ::= valuefieldreference Type UNIQUE? ValueOptionalitySpec?**

**ValueOptionalitySpec ::= OPTIONAL | DEFAULT Value**

**VariableTypeValueFieldSpec ::= valuefieldreference FieldName ValueOptionalitySpec?**

**FixedTypeValueSetFieldSpec ::= valuesetfieldreference Type ValueSetOptionalitySpec?**

**ValueSetOptionalitySpec ::= OPTIONAL | DEFAULT ValueSet**

**VariableTypeValueSetFieldSpec ::= valuesetfieldreference FieldName ValueSetOptionalitySpec?**

**ObjectFieldSpec ::= objectfieldreference DefinedObjectClass ObjectOptionalitySpec?**

**ObjectOptionalitySpec ::= OPTIONAL | DEFAULT Object**

**ObjectSetFieldSpec ::= objectsetfieldreference DefinedObjectClass ObjectSetOptionalitySpec?**

**ObjectSetOptionalitySpec ::= OPTIONAL | DEFAULT ObjectSet**

**WithSyntaxSpec ::= WITH SYNTAX SyntaxList**

**SyntaxList ::= "{" TokenOrGroupSpec empty \* "}"**

**TokenOrGroupSpec ::= RequiredToken | OptionalGroup**

**OptionalGroup ::= "[" TokenOrGroupSpec empty + "]"**

**RequiredToken ::= Literal | PrimitiveFieldName**

**Literal ::= word | ","**

**DefinedObject ::= ExternalObjectReference | objectreference**

**ExternalObjectReference ::= modulereference "." objectreference**

**ObjectAssignment ::= objectreference DefinedObjectClass "==" Object**

**Object ::= DefinedObject | ObjectDefn | ObjectFromObject | ParameterizedObject**

**ObjectDefn ::= DefaultSyntax | DefinedSyntax**

**DefaultSyntax ::= "{" FieldSetting "," \* "}"**

**FieldSetting ::= PrimitiveFieldName Setting**

**DefinedSyntax ::= "{" DefinedSyntaxToken empty \* "}"**

**DefinedSyntaxToken ::= Literal | Setting**

**Setting ::= Type | Value | ValueSet | Object | ObjectSet**

**DefinedObjectSet ::= ExternalObjectSetReference | objectsetreference**

**ExternalObjectSetReference ::= modulereference "." objectsetreference**

**ObjectSetAssignment ::= objectsetreference DefinedObjectClass "==" ObjectSet**

**ObjectSet ::= "{" ObjectSetSpec "}"**

**ObjectSetSpec ::= ElementSetSpecs | "..."**  
**ObjectSetElements ::=**  
     **Object | DefinedObjectSet | ObjectSetFromObjects | ParameterizedObjectSet**  
**ObjectClassFieldType ::= DefinedObjectClass "." FieldName**  
**ObjectClassFieldValue ::= OpenTypeFieldVal | FixedTypeFieldVal**  
**OpenTypeFieldVal ::= Type ":" Value**  
**FixedTypeFieldVal ::= Value**  
**InformationFromObjects ::= ValueFromObject | ValueSetFromObjects |**  
     **TypeFromObject | ObjectFromObject | ObjectSetFromObjects**  
**ReferencedObjects ::=**  
     **DefinedObject | ParameterizedObject |**  
     **DefinedObjectSet | ParameterizedObjectSet**  
**ValueFromObject ::= ReferencedObjects "." FieldName**  
**ValueSetFromObjects ::= ReferencedObjects "." FieldName**  
**TypeFromObject ::= ReferencedObjects "." FieldName**  
**ObjectFromObject ::= ReferencedObjects "." FieldName**  
**ObjectSetFromObjects ::= ReferencedObjects "." FieldName**  
**InstanceOfType ::= INSTANCE OF DefinedObjectClass**  
**InstanceOfValue ::= Value**



## SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie B	Medios de expresión: definiciones, símbolos, clasificación
Serie C	Estadísticas generales de telecomunicaciones
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedios
Serie I	Red digital de servicios integrados
Serie J	Transmisiones de señales radiofónicas, de televisión y de otras señales multimedios
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	RGT y mantenimiento de redes: sistemas de transmisión, circuitos telefónicos, telegrafía, facsímil y circuitos arrendados internacionales
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de transmisión telefónica, instalaciones telefónicas y redes locales
Serie Q	Conmutación y señalización
Serie R	Transmisión telegráfica
Serie S	Equipos terminales para servicios de telegrafía
Serie T	Terminales para servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
<b>Serie X</b>	<b>Redes de datos y comunicación entre sistemas abiertos</b>
Serie Y	Infraestructura mundial de la información
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación