



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

X.693

Enmienda 1
(10/2003)

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 – Reglas de
codificación de notación de sintaxis abstracta uno:
Reglas de codificación del lenguaje de marcaje
extensible

**Enmienda 1: Instrucciones de codificación XER
y EXTENDED-XER**

Recomendación UIT-T X.693 (2001) – Enmienda 1

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

REDES PÚBLICAS DE DATOS	
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
INTERCONEXIÓN DE SISTEMAS ABIERTOS	
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
INTERFUNCIONAMIENTO ENTRE REDES	
Generalidades	X.300–X.349
Sistemas de transmisión de datos por satélite	X.350–X.369
Redes basadas en el protocolo Internet	X.370–X.399
SISTEMAS DE TRATAMIENTO DE MENSAJES	X.400–X.499
DIRECTORIO	X.500–X.599
GESTIÓN DE REDES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS Y ASPECTOS DE SISTEMAS	
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
Notación de sintaxis abstracta uno	X.680–X.699
GESTIÓN DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	
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
SEGURIDAD	X.800–X.849
APLICACIONES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	
Compromiso, concurrencia y recuperación	X.850–X.859
Procesamiento de transacciones	X.860–X.879
Operaciones a distancia	X.880–X.899
PROCESAMIENTO DISTRIBUIDO ABIERTO	X.900–X.999
SEGURIDAD DE LAS TELECOMUNICACIONES	X.1000–

Para más información, véase la Lista de Recomendaciones del UIT-T.

**Tecnología de la información – Reglas de codificación de notación de sintaxis
abstracta uno: Reglas de codificación del lenguaje de marcaje extensible**

Enmienda 1

Instrucciones de codificación XER y EXTENDED-XER

Resumen

Con esta enmienda, se dispone ya de una enmienda 1 para cada una de las Recomendaciones siguientes: Rec. UIT-T X.680 | ISO/CEI 8824-1, Rec. UIT-T X.681 | ISO/CEI 8824-2, Rec. UIT-T X.690 | ISO/CEI 8825-1, Rec. UIT-T X.691 | ISO/CEI 8825-2 y Rec. UIT-T X.693 | ISO/CEI 8825-4. Dichas enmiendas permiten lo siguiente:

- La corrección de un error en CXER consecuencia de permitir espacios blancos entre un signo menos y el siguiente valor **INTEGER** o **REAL** (CXER no era canónica). Esta posibilidad ya no se permite en la notación de un valor, ya sea XML, XER o CXER. **Constituye un cambio** (que se introduce mediante texto en la enmienda 1 a la Rec. UIT-T X.680 | ISO/CEI 8824-1) **y no una adición**.
- La adición de instrucciones de codificación en un módulo ASN.1, utilizando un prefijo tipo o en una sección de control de codificación, a fin de especificar variaciones de las codificaciones BASIC-XER. Estas instrucciones de codificación están diseñadas para soportar el establecimiento de correspondencias entre una especificación XSD y una especificación ASN.1. Ello ha significado un cambio de terminología, por la que un tipo con "[...]" delante de ella es un tipo con prefijo, y la notación "[...]" puede o no ser un rótulo. Este cambio de terminología da lugar a cambios en el texto (pero no en lo esencial) de las especificaciones de BER y PER, por lo que también existe una enmienda 1 a dichas especificaciones.
- La adición de NaN (*Not-a-Number*, no un número) y de un cero menos como nuevos valores de **REAL** (en la enmienda 1 a la Rec. UIT X.690 | ISO/CEI 8825-1 y a la Rec. UIT-T X.691 | ISO/CEI 8825-2, así como en la enmienda 1 a la Rec. UIT-T X.693 | ISO/CEI 8825-4 se soporta la codificación de estos nuevos valores).
- La adición de nuevas notaciones de valor XML para **REAL**, **BOOLEAN**, **ENUMERATED**, e **INTEGER** que utilizan texto en lugar de rótulos de elementos vacíos, y que están disponibles en la notación de valor XML y en EXTENDED-XER, pero no en BASIC-XER (por razones de retrocompatibilidad).
- Cambios en la notación de valor XML de secuencia-de (*sequence-of*) (y codificaciones XER) a fin de permitir la delimitación de valores que no sean elementos XML (esto ocurre con las notaciones de valor XML adicionales, y solamente afecta a dichas notaciones de valor XML adicionales). Este cambio solo afecta a la utilización de notaciones de valor XML que se hayan añadido por medio de esta enmienda, y no se permiten en BASIC-XER, que no se ve afectada.

Todo ello proporciona el soporte básico necesario para EXTENDED-XER.

Esta enmienda es función de lo indicado en la Rec. UIT-T X.680 (2002)/enm.1 (2003) | ISO/CEI 8824-1:2002/Amd.1:2003 y en la Rec. UIT-T X.681 (2002)/enm.1 (2003) | ISO/CEI 8824-2:2002/Amd.1:2003 en lo relativo a la provisión de codificaciones alternativas para algunos tipos y en lo relativo a la sintaxis de inserción de instrucciones de codificación XER en una especificación ASN.1. Muchas referencias incluidas en esta enmienda a cláusulas de las mencionadas Recomendaciones | Normas Internacionales lo son a cláusulas introducidas por dichas enmiendas.

El grueso de esta enmienda versa a cerca de la especificación de la sintaxis y la semántica de (nuevas) instrucciones de codificación XER, que pueden utilizarse para requerir que los codificadores EXTENDED-XER proporcionen codificaciones especializadas para tipos ASN.1. Dichas codificaciones especializadas están diseñadas para soportar lo indicado en la Rec. UIT-T X.694 | ISO/CEI 8825-5 (correspondencia de XSD en ASN.1).

Orígenes

La enmienda 1 a la Rec. UIT-T X.693 (2001) fue aprobada el 29 de octubre de 2003 por la Comisión de Estudio 17 (2001-2004) del UIT-T por el procedimiento de la Recomendación UIT-T A.8. Se publica también un texto idéntico como Norma Internacional ISO/CEI 8825-4, enmienda 1.

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 Asamblea Mundial de Normalización de las Telecomunicaciones (AMNT), 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 1 de la AMNT.

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 "Administración" se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

La observancia de esta Recomendación es voluntaria. Ahora bien, la Recomendación puede contener ciertas disposiciones obligatorias (para asegurar, por ejemplo, la aplicabilidad o la interoperabilidad), por lo que la observancia se consigue con el cumplimiento exacto y puntual de todas las disposiciones obligatorias. La obligatoriedad de un elemento preceptivo o requisito se expresa mediante las frases "tener que, haber de, hay que + infinitivo" o el verbo principal en tiempo futuro simple de mandato, en modo afirmativo o negativo. El hecho de que se utilice esta formulación no entraña que la observancia se imponga a ninguna de las partes.

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 2004

Reservados todos los derechos. Ninguna parte de esta publicación puede reproducirse por ningún procedimiento sin previa autorización escrita por parte de la UIT.

ÍNDICE

	<i>Página</i>
Introducción.....	1
1 Alcance.....	2
2.1 Recomendaciones Normas Internacionales idénticas.....	3
2.2 Referencias adicionales.....	3
3 Definiciones.....	4
3.1 Reglas de codificación básica ASN.1 (BER, <i>basic encoding rules</i>).....	4
3.2 Definiciones adicionales.....	4
4 Abreviaturas.....	6
5 Esta cláusula ha sido eliminada por la enmienda 1.....	6
6 <i>bis</i> Instrucciones de codificación especificadas por esta Recomendación Norma Internacional.....	7
8.1 Producción de una codificación BASIC-XER completa.....	8
8.5 Codificación del tipo abierto.....	9
8.6 Decodificación de tipos con marcadores de extensión.....	9
9.1 Reglas generales para XER canónica.....	9
9.4 Valor octetstring (cadena de octetos).....	9
9.12 Valor de tipo abierto.....	10
10 Reglas de codificación XML extendida.....	10
10.1 Generalidades.....	10
10.2 Conformidad EXTENDED-XER.....	11
10.3 Estructura de una codificación EXTENDED-XER.....	13
11 Notación, conjunto de caracteres y elementos de léxico utilizados en instrucciones de codificación XER...	14
12 Palabras clave.....	14
13 Asignación de una instrucción de codificación XER a un tipo ASN.1 utilizando un prefijo de tipo.....	15
14 Asignación de una instrucción de codificación XER mediante una sección de control de codificación.....	18
14.1 Lista de asignación de una instrucción de codificación.....	18
14.2 Identificación de los objetivos de una instrucción de codificación XER mediante una lista de objetivos.....	18
14.2.1 Reglas generales.....	18
14.2.2 Identificación del objetivo utilizando una referencia de tipo e identificadores ASN.1.....	21
14.2.3 Identificación del objetivo utilizando un nombre de tipo integrado.....	22
14.2.4 Utilización de identificadores en contexto.....	23
14.2.5 Utilización de identificación de tipos importados.....	24
15 Asignación múltiple de instrucciones de codificación XER.....	24
15.1 Orden en el que consideran las asignaciones múltiples.....	24
15.2 Efecto de la asignación de una instrucción de codificación de negación.....	24
15.3 Asignación múltiple de instrucciones de codificación con varias categorías.....	25
15.4 Asignación múltiple de instrucciones de codificación XER de la misma categoría.....	25
15.5 Combinaciones permitidas de instrucciones de codificación finales.....	26
16 Soporte de instrucciones de codificación XER para espacios de nombres y nombres cualificados XML.....	27
17 Especificación de codificaciones EXTENDED-XER.....	29
17.1 Elemento documento XML.....	29
17.2 Producción "TypeNameOrModifiedTypeName".....	29
17.3 Producción "AttributeList".....	30
17.4 Producción "ExtendedXMLValue".....	30
17.5 Producción "ExtendedXMLChoiceValue".....	31
17.6 Producciones "ExtendedXMLSequenceValue" y "ExtendedXMLSetValue".....	32
17.7 Producciones "ExtendedXMLSequenceOfValue" y "ExtendedXMLSetOfValue".....	32
17.8 Producción "ModifiedXMLIntegerValue".....	34
17.9 Producción "ModifiedXMLRealValue".....	34

18	Instrucción de codificación ANY - ATTRIBUTES	35
	18.1 Generalidades.....	35
	18.2 Restricciones	35
	18.3 Efecto sobre las codificaciones	36
19	Instrucción de codificación ANY - ELEMENT	37
	19.1 Generalidades.....	37
	19.2 Restricciones	37
	19.3 Efecto sobre las codificaciones	38
20	Instrucción de codificación ATTRIBUTE	38
	20.1 Generalidades.....	38
	20.2 Restricciones	38
	20.3 Efecto sobre las codificaciones	39
21	Instrucción de codificación BASE64	40
	21.1 Generalidades.....	40
	21.2 Restricciones	41
	21.3 Efecto sobre las codificaciones	41
22	Instrucción de codificación DECIMAL.....	42
	22.1 Generalidades.....	42
	22.2 Restricciones	42
	22.3 Efecto sobre las codificaciones	42
23	Instrucción de codificación DEFAULT - FOR - EMPTY.....	42
	23.1 Generalidades.....	42
	23.2 Restricciones	43
	23.3 Efecto sobre las codificaciones	44
24	Instrucción de codificación ELEMENT.....	45
	24.1 Generalidades.....	45
	24.2 Restricciones	45
	24.3 Efecto sobre las codificaciones	45
25	Instrucción de codificación EMBED - VALUES.....	45
	25.1 Generalidades.....	45
	25.2 Restricciones	45
	25.3 Efecto sobre las codificaciones	46
26	Instrucción de codificación GLOBAL - DEFAULTS	47
	26.1 Generalidades.....	47
	26.2 Restricciones	47
	26.3 Efecto en las codificaciones	47
27	Instrucción de codificación LIST.....	47
	27.1 Generalidades.....	47
	27.2 Restricciones	48
	27.3 Efecto en las codificaciones	48
28	Instrucción de codificación NAME	49
	28.1 Generalidades.....	49
	28.2 Restricciones	50
	28.3 Efecto en las codificaciones	50
29	Instrucción de codificación NAMESPACE	51
	29.1 Generalidades.....	51
	29.2 Restricciones	51
	29.3 Efecto en las codificaciones	51

30	Instrucción de codificación PI-OR-COMMENT	52
	30.1 Generalidades	52
	30.2 Restricciones	53
	30.3 Efecto en las codificaciones	53
31	Instrucción de codificación TEXT	53
	31.1 Generalidades	53
	31.2 Restricciones	54
	31.3 Efecto en las codificaciones	54
32	Instrucción de codificación UNTAGGED	55
	32.1 Generalidades	55
	32.2 Restricciones	56
	32.3 Efecto en las codificaciones	56
33	Instrucción de codificación USE-NIL	57
	33.1 Generalidades	57
	33.2 Restricciones	57
	33.3 Efecto en las codificaciones	58
34	Instrucción de codificación USE-NUMBER	58
	34.1 Generalidades	58
	34.2 Restricciones	58
	34.3 Efecto en las codificaciones	58
35	Instrucción de codificación USE-ORDER	59
	35.1 Generalidades	59
	35.2 Restricciones	59
	35.3 Efecto en las codificaciones	60
36	Instrucción de codificación USE-QNAME	60
	36.1 Generalidades	60
	36.2 Restricciones	61
	36.3 Efecto en las codificaciones	61
37	Instrucción de codificación USE-TYPE	61
	37.1 Generalidades	61
	37.2 Restricciones	61
	37.3 Efecto en las codificaciones	62
38	Instrucción de codificación USE-UNION	62
	38.1 Generalidades	62
	38.2 Restricciones	63
	38.3 Efecto en las codificaciones	63
39	Instrucción de codificación WHITESPACE	64
	39.1 Generalidades	64
	39.2 Restricciones	64
	39.3 Efecto en las codificaciones	65
40	Valores de identificador de objeto que hacen referencia a las reglas de codificación	65
Anexo A – Ejemplos de codificaciones BASIC-XER y CXER		66
	A.1 Descripción ASN.1 de la estructura del registro	66
	A.2 Descripción ASN.1 de un valor de registro	66
	A.3 Representación XML básica de este valor de registro	66
	A.4 Representación XML canónica de este valor de registro	67

	<i>Página</i>
Anexo B – Codificaciones de contenido XML parcial y determinística.....	68
B.1 Contenido XML parcial	68
B.2 Restricciones recomendadas de codificaciones que producen un contenido de elemento XML parcial.....	68
Anexo C – Ejemplos de codificaciones EXTENDED-XER utilizando instrucciones de codificación XER.....	70
C.1 Introducción	70
C.2 Ejemplos sencillos.....	71
C.2.1 Una tarjeta de béisbol.....	71
C.2.2 Un empleado	71
C.3 Ejemplos más complejos.....	72
C.3.1 Utilización de la unión de dos tipos sencillos	72
C.3.2 Utilización de un atributo de identificación de tipo	72
C.3.3 Utilización de valores de enumeración	72
C.3.4 Utilización de una codificación vacía para un valor por defecto.....	73
C.3.5 Utilización de valores integrados para la notificación un pago pendiente	73

**NORMA INTERNACIONAL
RECOMENDACIÓN UIT-T**

**Tecnología de la información – Reglas de codificación de notación de sintaxis
abstracta uno: Reglas de codificación del lenguaje de marcaje extensible**

Enmienda 1

Instrucciones de codificación XER y EXTENDED-XER

NOTA – En esta enmienda todo nuevo texto o texto modificado está subrayado en las cláusulas reemplazadas (pero no en las nuevas cláusulas). Cuando dicho texto se incorpora en la Recomendación de base el subrayado se anula.

Sustitúyase la introducción por lo siguiente:

Introducción

Las publicaciones Rec. UIT-T X.680 | ISO/CEI 8824-1, Rec. UIT-T X.681 | ISO/CEI 8824-2, Rec. UIT-T X.682 | ISO/CEI 8824-3, Rec. UIT-T X.683 | ISO/CEI 8824-4 describen conjuntamente la notación de sintaxis abstracta uno (ASN.1), que define los mensajes que se intercambian las aplicaciones pares.

La presente Recomendación | Norma Internacional define reglas de codificación que se pueden aplicar a valores de tipos ASN.1 definidos utilizando la notación que se especifica en las Recomendaciones | Normas Internacionales anteriormente enumeradas. La aplicación de estas reglas de codificación produce una sintaxis de transferencia para dichos valores. En la especificación de estas reglas de codificación está implícito que también pueden utilizarse para la decodificación.

Existe más de un conjunto de reglas de codificación que pueden aplicarse a valores de tipos ASN.1. La presente Recomendación | Norma Internacional define tres conjuntos de reglas de codificación que utilizan el lenguaje de marcaje extensible (XML, *extensible markup language*). Todas estas reglas de codificación generan un documento XML que es conforme con W3C XML 1.0. El primer conjunto se denomina reglas de codificación XML básica (BASIC-XER, *basic XML encoding rules*). El segundo conjunto se denomina reglas de codificación XML canónica (CANONICAL-XER, o CXER) debido a que sólo existe una forma de codificar un valor ASN.1 utilizando dichas reglas de codificación. (Las reglas de codificación canónica se usan generalmente para aplicaciones que utilizan características de seguridad, tales como la firma digital.) El tercer conjunto se denomina reglas de codificación XML extendida (EXTENDED-XER). Las reglas de codificación XML extendida permiten opciones adicionales del codificador, y tiene en cuenta las instrucciones de codificación que especifican variantes de la codificación BASIC-XER a fin de soportar estilos específicos de documentos XML (véase más adelante). Las reglas de codificación XML extendida no son canónicas, no existiendo en esta Recomendación | Norma Internacional ninguna forma canónica para dichas reglas.

Existen numerosos aspectos de una representación de datos XML (tales como la utilización de atributos XML en lugar de elementos hijo, o la utilización de listas delimitadas por espacios blancos) cuya utilización es sólo función del estilo y de la elección que haga el diseñador XML. Si un tipo definido en una especificación ASN.1 se codifica mediante BASIC-XER o mediante CXER, sólo se utiliza un único estilo fijo para la representación XML, sin control por parte del usuario de las características estilísticas. Esta Recomendación | Norma Internacional especifica la sintaxis y la semántica de instrucciones de codificación XER que especifican características estilísticas de XML en una codificación EXTENDED-XER. Las instrucciones de codificación XER pueden también utilizarse para determinar la posible inserción de instrucciones de procesamiento XML en una codificación EXTENDED-XER. Las instrucciones de codificación XER son ignoradas por BASIC-XER y por CXER, pero son utilizadas por EXTENDED-XER.

NOTA – Las "características estilísticas", tales como la utilización de atributos o de listas delimitadas por espacios blancos, pueden también afectar al tamaño de una codificación y a la facilidad de procesamiento de la misma, por lo que la utilización de dichas características no es solamente una cuestión de estilo. Cuando dichos aspectos resultan importantes, puede ser preferible utilizar EXTENDED-XER con instrucciones de codificación que utilizar BASIC-XER o CXER.

En la cláusula 8 se especifica la codificación BASIC-XER de tipos ASN.1.

En la cláusula 9 se especifica la codificación CXER de tipos ASN.1.

En la cláusula 10 se especifica la codificación EXTENDED-XER de tipos ASN.1, haciendo referencia a cláusulas posteriores que definen las instrucciones de codificación XER.

En las cláusulas 11 a 14 se enumeran y categorizan las instrucciones de codificación XER y se especifica la sintaxis para su asignación a un tipo o componente ASN.1, utilizando un prefijo de tipo XER (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 30.3) o una sección de control de codificación XER (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, cláusula 50).

En la cláusula 15 se define el orden de precedencia en caso de que existan instrucciones de codificación XER tanto en un prefijo de tipo XER como en una sección de control de codificación XER.

En la cláusula 16 se especifica el soporte de la instrucción de codificación para espacios de nombres XML cuando se utiliza EXTENDED-XER.

En la cláusula 17 se especifican codificaciones EXTENDED-XER.

En las cláusulas 18 a 39 se especifica:

- a) la sintaxis de cada instrucción de codificación XER utilizada en un prefijo de tipo o en una sección de control de codificación;
- b) restricciones de las instrucciones de codificación XER que pueden asociarse con un tipo ASN.1 específico (resultado de la herencia y de asignaciones múltiples);
- c) modificaciones a las reglas de codificación XER que son necesarias en una codificación EXTENDED-XER cuando se aplica una instrucción de codificación XER.

El anexo A tiene carácter informativo y contiene ejemplos de codificaciones BASIC-XER y CXER.

El anexo B tiene carácter informativo y contiene una descripción del contenido XML parcial que se produce cuando se eliminan los rótulos que enmarcan construcciones tales como secuencia (*sequence*) y secuencia-de (*sequence-of*), junto con restricciones aplicables a las especificaciones EXTENDED-XER que permiten determinar fácilmente el componente ASN.1 con el que está asociado un elemento XML.

El anexo C tiene carácter informativo y contiene ejemplos de instrucciones de codificación XER y de las correspondientes codificaciones EXTENDED-XER.

Sustitúyase la cláusula 1 por la siguiente:

1 Alcance

La presente Recomendación | Norma Internacional especifica un conjunto de reglas de codificación XML básica (BASIC-XER, BASIC-XML encoding rules), que pueden utilizarse para obtener una sintaxis de transferencia para valores de tipos definidos en la Rec. UIT-T X.680 | ISO/CEI 8824-1 y en la Rec. UIT-T X.681 | ISO/CEI 8824-2. La presente Recomendación | Norma Internacional también especifica un conjunto de reglas de codificación XML canónica que permite restringir las reglas de codificación XML básica y que produce una codificación única para cualquier valor ASN.1 dado. La presente Recomendación | Norma Internacional especifica además un conjunto de reglas de codificación XML extendida (EXTENDED-XER) que añade opciones adicionales del codificador y que también permite que el especificador ASN.1 modifique la codificación que se produciría mediante BASIC-XER. En la especificación de estas reglas de codificación está implícito que éstas también se utilizan para decodificar.

Las reglas de codificación especificadas en la presente Recomendación | Norma Internacional:

- se utilizan en el momento de la comunicación;
- están previstas para usarse cuando el interés principal en la elección de reglas de codificación sea la presentación de valores y/o su procesamiento utilizando herramientas XML comúnmente disponibles (tales como navegadores);
- permiten la extensión de una sintaxis abstracta por adición de valores suplementarios para todas las formas de extensibilidad descritas en la Rec. UIT-T X.680 | ISO/CEI 8824-1.

La presente Recomendación | Norma Internacional también especifica la sintaxis y la semántica de instrucciones de codificación XER, y las reglas para su asignación y combinación. Las instrucciones de codificación XER pueden ser utilizadas para controlar la codificación EXTENDED-XER para tipos ASN.1 específicos.

Sustitúyase la cláusula 2.1 por la siguiente:

2.1 Recomendaciones | Normas Internacionales idénticas

- Recomendación UIT-T X.680 (2002) | ISO/CEI 8824-1:2002, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de la notación básica.*
- Recomendación UIT-T X.680 (2002)/enm.1 (2003) | ISO/CEI 8824-1:2002/Amd.1:2003, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de la notación básica – Enmienda 1: Soporte de EXTENDED-XER.*
- Recomendación UIT-T X.681 (2002) | ISO/CEI 8824-2:2002, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de objetos de información.*
- Recomendación UIT-T X.681 (2002)/enm.1 (2003) | ISO/CEI 8824-2:2002/Amd.1:2003, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de objetos de información – Enmienda 1: Soporte de EXTENDED-XER.*
- Recomendación UIT-T X.682 (2002) | ISO/CEI 8824-3:2002, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de constricciones.*
- Recomendación UIT-T X.683 (2002) | ISO/CEI 8824-4:2002, *Tecnología de la información – Notación de sintaxis abstracta uno: Parametrización de especificaciones de notación de sintaxis abstracta uno.*
- Recomendación UIT-T X.690 (2002) | ISO/CEI 8825-1:2002, *Tecnología de la información – Reglas de codificación de notación de sintaxis abstracta uno: Especificación de las reglas de codificación básica, de las reglas de codificación canónica y de las reglas de codificación distinguida.*
- Recomendación UIT-T X.690 (2002)/enm.1 (2003) | ISO/CEI 8825-1:2002/Amd.1:2003, *Tecnología de la información – Reglas de codificación de notación de sintaxis abstracta uno: Especificación de las reglas de codificación básica, de las reglas de codificación canónica y de las reglas de codificación distinguida – Enmienda 1: Soporte de EXTENDED-XER.*
- Recomendación UIT-T X.691 (2002) | ISO/CEI 8825-2:2002, *Tecnología de la información – Reglas de codificación de notación de sintaxis abstracta uno: Especificación de las reglas de codificación compactada.*
- Recomendación UIT-T X.691 (2002)/enm.1 (2003) | ISO/CEI 8825-2:2002/Amd.1:2003, *Tecnología de la información – Reglas de codificación de notación de sintaxis abstracta uno: Especificación de las reglas de codificación compactada – Enmienda 1: Soporte de EXTENDED-XER.*
- Recomendación UIT-T X.692 (2002) | ISO/CEI 8825-3:2002, *Tecnología de la información – Reglas de codificación de notación de sintaxis abstracta uno: Especificación de la notación de control de codificación.*

Sustitúyase la cláusula 2.2 por la siguiente:

2.2 Referencias adicionales

- IETF RFC 2045 (1996), *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies.*
- IETF RFC 2141 (1997), *URN Syntax.*
- IETF RFC 2396 (1998), *Uniform Resource Identifiers (URI): Generic Syntax.*
- IETF RFC 3061(2001), *A URN Namespace of Object Identifiers.*
- ISO/CEI 10646-1:2000, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane.*
- The Unicode Standard, Version 3.2.0, The Unicode Consortium. (Reading, MA, Addison-Wesley)
NOTA – Los caracteres gráficos (y sus codificaciones) definidas por la referencia anterior son idénticas a las definidas en ISO/CEI 10646-1, pero dicha referencia se incluye porque también especifica los nombres de los caracteres de control.
- W3C XML 1.0:2000, *Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, Copyright © [6 October 2000] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/2000/REC-xml-20001006>.*

- W3C XML Namespaces:1999, Namespaces in XML, W3C Recommendation, Copyright © [14 January 1999] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/1999/REC-xml-names-19990114>.

NOTA – La referencia a un documento en esta Recomendación | Norma Internacional no le otorga, como tal, la condición de Recomendación o Norma Internacional.

Sustitúyase la cláusula 3 por la siguiente:

3 Definiciones

Para los fines de esta Recomendación | Norma Internacional, son de aplicación las definiciones de la Rec. UIT-T X.680 | ISO/CEI 8824-1 y las definiciones siguientes.

3.1 Reglas de codificación básica ASN.1 (BER, *basic encoding rules*)

Esta Recomendación | Norma Internacional utiliza los siguientes términos definidos en la Rec. UIT-T X.690 | ISO/CEI 8825-1:

- a) valor de datos;
- b) conformidad dinámica;
- c) codificación (de un valor de datos);
- d) receptor;
- e) emisor;
- f) conformidad estática.

3.2 Definiciones adicionales

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

3.2.1 esquema ASN.1: Definición del contenido y estructura de datos realizada utilizando una definición de tipo ASN.1.

NOTA – Esto habilita que las reglas de codificación puedan producir codificaciones binarias de valores de un tipo ASN.1, o codificaciones que utilicen XML.

Insértense cuatro nuevas subcláusulas 3.2.1 bis, 3.2.1 ter, 3.2.1 quat, y 3.2.1 quin como se indica a continuación:

3.2.1 bis rótulo de elemento vacío asociado: Rótulo de elemento vacío XML que puede sustituir a un rótulo precedente y subsiguiente asociado, cuando existe.

NOTA – Algunas instrucciones de codificación suprimen los rótulos asociados de un "XMLValue".

3.2.1 ter instrucciones de codificación asociadas (para un tipo): Un conjunto de instrucciones de codificación XER asociadas con un tipo.

3.2.1 quat rótulo subsiguiente asociado: Rótulo de terminación XML que sigue al "XMLValue" de un tipo en ausencia de instrucciones de codificación que eliminen los rótulos asociados.

3.2.1 quin rótulo precedente asociado: Rótulo inicial XML que precede al "XMLValue" de un tipo en ausencia de instrucciones de codificación que eliminen los rótulos asociados.

Insértense las 16 nuevas subcláusulas siguientes, desde 3.2.2 bis hasta, 3.2.2 septdec:

3.2.2 bis documento válido XML canónico (para un esquema ASN.1): Documento XML correctamente constituido (véase W3C XML 1.0) cuyo contenido es conforme a la especificación CXER para la codificación de un tipo ASN.1 especificado mediante un esquema ASN.1.

3.2.2 ter tipo codificable mediante caracteres: Un tipo ASN.1 al que puede aplicarse una instrucción de codificación **ATTRIBUTE** (véase 20.2.1).

3.2.2 quat espacio de nombres de control: Espacio de nombres (*namespace*) utilizado para identificar atributos que realizan funciones o transportan valores que controlan una codificación EXTENDED-XER.

NOTA 1 – Un ejemplo de ello sería un atributo de identificación de tipo. El espacio de nombres de control es el valor por defecto del espacio de nombres ASN.1 especificado en 16.9, pero que puede cambiarse mediante la instrucción de codificación **GLOBAL-DEFAULTS**.

NOTA 2 – El espacio de nombres de control puede también contener nombres de atributos que pueden estar presentes, pero que son ignorados por los decodificadores **EXTENDED-XER** (véase 10.2.10). Un ejemplo de dicho nombre de atributo sería **schemaLocation**.

3.2.2 quin tipo (ASN.1) incluido: Un tipo ASN.1 cuyo "XMLValue" en una codificación BASIC-XER está directamente incluido en el "XMLValue" de un tipo ASN.1 (el tipo envolvente).

NOTA – Todos los tipos de una codificación BASIC-XER o EXTENDED-XER son tipos incluidos salvo que se utilicen como tipo raíz (véase 10.3.1 b) en una codificación.

3.2.2 sex elemento envolvente (de un tipo ASN.1): Un "ExtendedXMLTypedValue", "ExtendedXMLChoiceValue", "ExtendedXMLNamedValue" o "ExtendedXMLDelimitedItem" que tiene como "ExtendedXMLValue" la codificación "ExtendedXMLValue" del tipo (véanse 17.1, 17.5, 17.6 y 17.7).

3.2.2 sept tipo envolvente (de un tipo ASN.1): Un tipo ASN.1 cuyo "XMLValue" de una codificación BASIC-XER incluye directamente el "XMLValue" de un tipo ASN.1 (un tipo incluido).

NOTA – El tipo envolvente puede ser un tipo secuencia (*sequence*), un tipo conjunto (*set*), un tipo elección (*choice*), un tipo secuencia-de (*sequence-of*), un tipo conjunto-de (*set-of*), un tipo abierto (*open*), o un tipo cadena de octetos (*octetstring*) o cadena de bits (*bitstring*) (con un **CONTAINING** y sin un **ENCODED BY**).

3.2.2 oct instrucciones de codificación finales (para un tipo): Conjunto de instrucciones de codificación XER asociadas con un tipo como resultado de la especificación ASN.1 completa, y que se utilizan para producir codificaciones de dicho tipo.

3.2.2 non instrucciones de codificación heredadas: Instrucciones de codificación XER asociadas con el tipo identificado mediante una referencia de tipo.

3.2.2 dec nombre de espacio de nombres cualificado: Nombre en un documento XML que tiene un prefijo de espacio de nombres XML o que es un nombre de elemento XML en el ámbito de una declaración de espacio de nombres por defecto XML.

NOTA – Las declaraciones de espacio de nombres por defecto XML sólo afectan a nombres de elementos XML, no a nombres de atributos. A ambos puede aplicarse un prefijo de espacio de nombres.

3.2.2 undec atributo de identificación nulo: Atributo XML que puede aparecer en cualquier elemento para identificar si el contenido tiene un valor nil (nulo) (véase la cláusula 33).

3.2.2 duodec contenido de elemento XML parcial: Elementos hijos XML definidos mediante un tipo ASN.1 que no tiene rótulo (**UNTAGGED**), y que proporciona parte del contenido de elemento XML generado mediante el tipo envolvente.

NOTA – Si el tipo envolvente es **UNTAGGED**, éste sólo puede generar un contenido de elemento XML parcial.

3.2.2 terdec instrucciones de codificación con prefijo: Instrucciones de codificación XER que se asignan utilizando un prefijo de tipo.

NOTA – Las instrucciones de codificación con prefijo pueden suprimir, sustituir o añadir las instrucciones de codificación asociadas de un tipo.

3.2.2 quatdec información de cualificación: Información suministrada como parte de la especificación de un objetivo para la asignación de una instrucción de codificación que identifique valores específicos del tipo objetivo.

3.2.2 quindec instrucciones de codificación objetivo: Instrucciones de codificación XER que se asignan utilizando una lista objetivo en una sección de control de codificación XER.

NOTA – Las instrucciones de codificación objetivo pueden producir la supresión, sustitución o adición de instrucciones de codificación asociadas de un tipo.

3.2.2 sexdec atributo de identificación de tipo: Un atributo XML que puede aparecer en cualquier elemento para identificar el tipo de dicho elemento (véase la cláusula 37).

3.2.2 septdec identificador de recurso uniforme (URI, *uniform resource identifier*): Identificador globalmente inequívoco, que se asigna de conformidad con cualquiera de los posibles esquemas de URI utilizados para identificar espacios de nombres en codificaciones EXTENDED-XER.

NOTA – El esquema de URI utilizado por defecto para ASN.1 permite utilizar un valor de identificador de objeto ASN.1 para identificar espacios de nombres (véanse 16.9 y 29.1.5).

Sustitúyase la subcláusula 3.2.3 por la siguiente:

3.2.3 documento XML válido (para un esquema ASN.1): Documento XML correctamente constituido (véase W3C XML 1.0) y cuyo contenido es conforme con la especificación BASIC-XER, CXER o EXTENDED-XER para la codificación de un tipo ASN.1 especificado mediante un esquema ASN.1, posiblemente incluyendo instrucciones de codificación XER.

Insértense cuatro nuevas subcláusulas 3.2.3 bis a 3.2.3 quin tal como se indica a continuación:

3.2.3 bis instrucciones de codificación XER: Instrucciones de codificación asociadas con un tipo ASN.1 (o con un componente de un tipo ASN.1) mediante la asignación a dicho tipo (o componente) en un prefijo de tipo XER (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 30.3) o en una sección de control de codificación XER (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, cláusula 50).

3.2.3 ter atributo XML: Parte de una codificación EXTENDED-XER que consta de un "XMLValue" incluido entre comillas o apóstrofes, precedido por un nombre (atributo) y un signo igual.

3.2.3 quat elemento XML: Parte de un documento XML especificado en W3C XML 1.0.

NOTA – Un elemento XML es un rótulo de elemento vacío o comienza con un rótulo de inicio y finaliza con un rótulo de terminación. Tanto el rótulo de inicio como el rótulo de elemento vacío pueden contener codificaciones de atributo.

3.2.3 quin nombre de elemento XML: Elemento de léxico que sigue a un elemento de léxico "<" o "</" en los rótulos asociados.

Sustitúyase la subcláusula 3.2.4 por la siguiente:

3.2.4 documento XML: Secuencia de caracteres que son conformes con la definición de documento W3C XML 1.0.

Insértense dos nuevas subcláusulas 3.2.5 y 3.2.6 tal como se indica a continuación:

3.2.5 instrucción de procesamiento XML: Parte de un documento XML que transporta información relativa al procesamiento de parte o de todo el documento (véase W3C XML 1.0).

NOTA – La instrucción de procesamiento identifica el tipo de procesamiento al que es aplicable, siendo ignorada en cualquier otro procesamiento. Puede utilizarse para identificar la hoja de estilo que debe aplicarse si el documento está destinado a ser visualizado por personas.

3.2.6 prólogo XML: Parte inicial de un documento XML (sin información sobre el valor del tipo ASN.1 que ha sido codificado).

Sustitúyase la cláusula 4 por la siguiente:

4 Abreviaturas

A los efectos de esta Recomendación | Norma Internacional, se utilizan las siguientes siglas:

ASN.1	Notación de sintaxis abstracta uno (<i>abstract syntax notation one</i>)
<u>CXER</u>	<u>Reglas de codificación XML canónica (<i>canonical XML encoding rules</i>)</u>
PDU	Unidad de datos de protocolo (<i>protocol data unit</i>)
UCS	Conjunto universal de caracteres codificados en múltiples octetos (<i>universal multiple-octet coded character set</i>)
<u>URI</u>	<u>Identificador de recurso uniforme (<i>uniform resource identifier</i>)</u>
UTC	Tiempo universal coordinado (<i>coordinated universal time</i>)
UTF-8	Formato de transformación UCS, forma de 8 bits (<i>UCS transformation format, 8-bit form</i>)
XER	Reglas de codificación XML (<i>XML encoding rules</i>)
XML	Lenguaje de marcaje extensible (<i>extensible markup language</i>)

Sustitúyase la cláusula 5 por la siguiente:

5 Esta cláusula ha sido eliminada por la enmienda 1

Sustitúyanse las subcláusulas 6.1, 6.2 y 6.3 por las siguientes:

- 6.1** Esta Recomendación | Norma Internacional especifica tres conjuntos de reglas de codificación:
- Reglas de codificación XML básica (BASIC-XER, *basic XML encoding rules*).
 - Reglas de codificación XML canónica (CXER, *canonical XML encoding rules*).
 - Reglas de codificación XML extendida (EXTENDED-XER, *extended XML encoding rules*).

6.2 El conjunto básico de reglas de codificación especificado en esta Recomendación | Norma Internacional es BASIC-XER, que, por lo general, no produce una codificación canónica, y que no proporciona control alguno del usuario sobre el estilo de XML que se produce.

6.3 Un segundo conjunto de reglas de codificación especificado en esta Recomendación | Norma Internacional es CXER, que produce codificaciones canónicas. Se define como una restricción de opciones dependientes de la implementación en la codificación BASIC-XER.

NOTA 1 – Toda implementación conforme a CXER para codificación es también conforme a BASIC-XER para codificación. Toda implementación conforme a BASIC-XER para decodificación es conforme a CXER para decodificación. Por lo tanto, las codificaciones hechas de acuerdo con CXER son codificaciones que están permitidas por BASIC-XER.

NOTA 2 – CXER produce codificaciones que tienen aplicación cuando se deban aplicar autenticadores a valores abstractos.

Insértese una nueva subcláusulas 6.3 bis:

6.3 bis El tercer conjunto de reglas de codificación especificado en esta Recomendación | Norma Internacional es EXTENDED-XER. Se define como variaciones de las codificaciones BASIC-XER especificadas mediante instrucciones de codificación XER (véase 6 bis) asociadas con un tipo ASN.1. En ausencia de instrucciones de codificación XER, una codificación EXTENDED-XER difiere de una codificación BASIC-XER sólo en que proporciona más opciones de codificación.

Sustitúyase la subcláusula 6.4 por la siguiente:

6.4 Si un tipo codificado con CXER contiene tipos **EMBEDDED PDV**, **EXTERNAL** o **CHARACTER STRING**, la codificación exterior deja de ser canónica a menos que la codificación utilizada para todos los tipos **EMBEDDED PDV**, **EXTERNAL** y **CHARACTER STRING** sea canónica.

Insértese una nueva cláusula 6 bis como la siguiente:

6 bis Instrucciones de codificación especificadas por esta Recomendación | Norma Internacional

6 bis.1 Esta Recomendación | Norma Internacional especifica la sintaxis y la semántica de las instrucciones de codificación XER (véanse las cláusulas 11 a 39). Las instrucciones de codificación XER sólo afectan a codificaciones EXTENDED-XER.

6 bis.2 ASN.1 conforma una notación de esquema XML básico. El esquema ASN.1 se utiliza para definir el contenido y la estructura de datos utilizando ASN.1 y las reglas de codificación BASIC-XER (y CXER). Puede utilizarse sin instrucciones de codificación XER.

6 bis.3 Las instrucciones de codificación XER proporcionan una mayor flexibilidad en los documentos XML que pueden ser especificados.

6 bis.4 Las instrucciones de codificación XER se asignan a definiciones de tipo ASN.1 o a referencias de tipo utilizando uno cualquiera o ambos prefijos de tipo XER (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 30.3) y una sección de control de codificación XER (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, cláusula 50). Si existen instrucciones de codificación asociadas a una definición de tipo, se transportan con el tipo ASN.1 (a través de su referencia de tipo) a otras definiciones de tipo y a otros módulos ASN.1. Cuando EXTENDED-XER codifica un tipo que tiene instrucciones de codificación XER asociadas con alguna de sus partes o con todas ellas, dichas instrucciones de codificación finales se aplican y modifican las codificaciones EXTENDED-XER que se producen.

NOTA – Las instrucciones de codificación finales también se utilizan cuando se realiza la validación de la calidad de funcionamiento y/o la decodificación de una codificación EXTENDED-XER.

Sustitúyase la subcláusula 7.1 por la siguiente:

7.1 La conformidad dinámica para las reglas de codificación XML básica se especifica en la cláusula 8, y la conformidad dinámica para las reglas de codificación XML canónica se especifica en la cláusula 9 y la conformidad dinámica para las reglas de codificación XML extendida se especifica en la cláusula 10.

Sustitúyase la subcláusula 7.3 por la siguiente:

7.3 Las reglas de codificación XML básica y las reglas de codificación XML extendida permiten codificaciones alternativas como una opción del codificador. Los decodificadores que declaran la conformidad con BASIC-XER soportarán todas las alternativas de codificación BASIC-XER. Los decodificadores que declaran la conformidad con EXTENDED-XER soportarán todas las alternativas de codificación EXTENDED-XER.

NOTA – Esta cláusula se aplica con independencia de que haya o no instrucciones de codificación finales.

Sustitúyase la subcláusula 8.1 por la siguiente:

8.1 Producción de una codificación BASIC-XER completa

8.1.1 Una codificación BASIC-XER conforme es un documento XML válido que consta (en este orden) de:

- a) un prólogo XML (que puede estar vacío), como se especifica en 8.2;
- b) un elemento de documento XML, que es la codificación completa de un valor de un tipo ASN.1 simple, como se especifica en 8.3.

8.1.2 La especificación incluida en 8.2 a 8.6 define completamente la codificación BASIC-XER.

NOTA – Otras construcciones de W3C XML 1.0, tales como instrucciones de procesamiento XML, no están permitidas por esas subcláusulas, y nunca son producto de un codificador BASIC-XER conforme.

8.1.3 El documento XML se codificará utilizando UTF-8 para producir una cadena de octetos que forma la codificación especificada en esta Recomendación | Norma Internacional. El identificador de objeto ASN.1 para estas reglas de codificación se especifica en la cláusula 40.

8.1.4 Cuando en esta Recomendación | Norma Internacional se utiliza el término "espacio blanco" ("*white-space*"), significa uno o más de los siguientes caracteres de la Norma Unicode: HORIZONTAL TABULATION (tabulación horizontal) (9), LINE FEED (cambio de renglón) (10), CARRIAGE RETURN (retroceso del carro) (13), SPACE (espacio) (32). Los números entre paréntesis son el valor decimal de los caracteres de la Norma Unicode. Cuántos y cuáles caracteres constituyan un "espacio blanco" es una opción del codificador.

Insértese una nueva subcláusula 8.1.5:

8.1.5 Cuando esta Recomendación | Norma Internacional utiliza el término "espacio blanco con escapes", significa uno o más de los caracteres enumerados en 8.1.4, con una opción de decodificador para representar cualquiera de dichos caracteres mediante una secuencia de escape de la forma "&#n;" o "&#xn;" (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.15.8).

Sustitúyase la subcláusula 8.3.1 por la siguiente:

8.3.1 El elemento de documento XML será un "XMLTypedValue", tal como se especifica en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 15.2, con los cambios y restricciones especificados en las siguientes subcláusulas de esta cláusula 8.3.

Insértese una nueva subcláusula 8.3.1bis:

8.3.1 bis Todas las ocurrencias de "ExternalTypeReference" en el "XMLTypedValue" serán sustituidas por la "typereference" de dicho "ExternalTypeReference".

Sustitúyase la subcláusula 8.3.3 por la siguiente:

8.3.3 Cuando la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.1.4, 11.11 y 11.13, permite la utilización de espacio blanco ASN.1 entre elementos léxicos o en "xmlbstring" o en "xmlhstring", los caracteres utilizados estarán restringidos a los que constituyen el "espacio blanco" especificado en 8.1.4.

Insértese una nueva subcláusula 8.3.3 bis:

8.3.3 bis El "XMLBooleanValue" especificado en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 17.3, será solamente "EmptyElementBoolean", y el "XMLSequenceOfValue" y "XMLSetOfValue" con un componente de tipo booleano será "ValueList".

Sustitúyase la subcláusula 8.3.4 por la siguiente:

8.3.4 El "XMLIntegerValue" especificado en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 18.9, será solamente "XMLSignedNumber".

Insértense dos nuevas subcláusulas 8.3.4 bis y 8.3.4 ter:

8.3.4 bis El "XMLEnumeratedValue" especificado en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 19.8, será solamente "EmptyElementEnumerated" y el "XMLSequenceOfValue" y "XMLSetOfValue" con un componente de tipo enumerado será "ValueList".

8.3.4 ter El "XMLSpecialRealValue" especificado en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 20.6, será solamente "EmptyElementReal".

Insértese una nueva subcláusula 8.5:

8.5 Codificación del tipo abierto

Pueden utilizarse las dos alternativas de "XMLOpenTypeFieldVal" (véase la Rec. UIT-T X.681 | ISO/CEI 8824-2, 14.6).

NOTA – En general, no se recomienda la utilización de la alternativa "xmlhstring" de "XMLOpenTypeFieldVal", ya que no existen mecanismos para identificar las reglas de codificación utilizadas para producir la "xmlhstring" en un ejemplar de una codificación. Esta alternativa puede ser conveniente cuando el mensaje codificado en XER (por ejemplo, con fines de visualización) es el resultado de un codificación binaria previa y no ha sido decodificado completamente, o cuando existen acuerdos bilaterales.

Insértese una nueva subcláusula 8.6:

8.6 Decodificación de tipos con marcadores de extensión

8.6.1 Un decodificador BASIC-XER aceptará como documento XML válido codificaciones BASIC-XER de tipos con marcadores de extensión en los que existan extensiones desconocidas.

8.6.2 La presencia de extensiones desconocidas en un tipo secuencia (*sequence*) o conjunto (*set*) da lugar a elementos XML inesperados con nombres diferentes de los nombres del siguiente elemento XML esperado.

NOTA – Pueden existir múltiples nombres para un elemento XML subsiguiente conocido cuando exista la opción para ello, pero las adiciones de extensión tendrán siempre nombres diferentes a todos ellos.

8.6.3 La presencia de extensiones desconocidas en un tipo elección (*choice*) da lugar a un único elemento XML inesperado en lugar de a un elemento correspondiente a una de las elecciones conocidas. Siempre tendrá un nombre de elemento XML diferente tomado de cualquiera de los elementos XML que codifiquen una alternativa conocida del tipo elección.

8.6.4 La presencia de extensiones desconocidas en un tipo enumerado (*enumerated*) da lugar a un elemento XML con un contenido inesperado, pero sin elementos XML inesperados.

8.6.5 Las extensiones desconocidas que hayan aparecido como consecuencia de la relajación de una limitación de subtipo, producen una codificación que puede ser una codificación válida de cualquier valor del tipo no restringido. Dichas codificaciones pueden producir un contenido inesperado, pero no elementos XML inesperados.

Sustitúyase la subcláusula 9.1 por la siguiente:

9.1 Reglas generales para XER canónica

Sustitúyase la subcláusula 9.1.2 por la siguiente (manteniendo la NOTA):

9.1.2 Los elementos léxicos que forman el "XMLTypedValue" no tendrán "espacios blanco" entre ellos (véase 8.3.3).

Sustitúyase la subcláusula 9.3.1 por la siguiente:

9.3.1 Si la alternativa "XMLTypedValue" de "XMLBitStringValue" (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 21.9) puede utilizarse (como se especifica en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 21.10), será utilizada. De lo contrario se utilizará la alternativa "xmlbstring" con todos los "espacios blancos" eliminados (véase 8.3.3).

Sustitúyase la subcláusula 9.4 por la siguiente:

9.4 Valor octetstring (cadena de octetos)

Si la alternativa "XMLTypedValue" de "XMLOctetStringValue" (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 22.3) puede utilizarse (como se especifica en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 22.4), será utilizada. De lo contrario, se utilizará la alternativa "xmlhstring" con todos los "espacios blancos" eliminados (véase 8.3.3) y todas las letras en mayúsculas.

Sustitúyase la subcláusula 9.6.1 por la siguiente:

9.6.1 El tipo conjunto (*set*) tendrá los elementos de su "RootComponentTypeList" clasificados en el orden canónico especificado en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 8.6, y además, a fin de determinar el orden en el cual se codifican los componentes cuando uno o más de ellos son del tipo elección sin rótulo ASN.1, cada tipo elección se ordena como si tuviese un rótulo igual al rótulo más pequeño de la "RootAlternativeTypeList" de dicho tipo elección a cualquiera de dichos tipos elección anidados en su interior.

Sustitúyase la subcláusula 9.7.1 por la siguiente:

9.7.1 El orden de los elementos de un "XMLSetOfValue" (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 27.3) será determinado clasificando las cadenas de caracteres que representan la codificación CXER para cada elemento, como se especifica en 9.7.2 y 9.7.3.

Insértese una nueva subcláusula 9.12:

9.12 Valor de tipo abierto

La "xmlhstring" alternativa de "XMLOpenTypeFieldVal" no se utilizará (véase 8.5).

Insértense las siguientes 30 nuevas cláusulas, de 10 a 39, antes de la cláusula 10 existente (que debido a esta enmienda se renumera como cláusula 40):

10 Reglas de codificación XML extendida

10.1 Generalidades

10.1.1 Las reglas de codificación XML extendida (EXTENDED-XER) amplían y modifican BASIC-XER. Permiten que ASN.1 defina la forma y contenido de una gama mucho más amplia de documentos XML.

10.1.2 EXTENDED-XER amplía BASIC-XER de tres formas principales.

- a) Proporciona opciones adicionales del codificador (por ejemplo, para la inserción de instrucciones de procesamiento XML o de comentarios XML, y para la utilización de identificadores para bits denominados incluidos en un valor cadena de bits (*bitstring*);
- b) Especifica un conjunto de instrucciones de codificación que pueden utilizarse para especificar la modificación de la codificación BASIC-XER de un tipo ASN.1, incluyendo una instrucción de codificación para utilizar un texto sencillo en lugar de rótulos de elementos vacíos para tipos booleanos, enteros (con números denominados), enumerados, de valores reales especiales y cadena de bits (con bits denominados);
- c) Exige que los decodificadores ignoren (en ausencia de instrucciones de codificación) atributos del espacio de nombres de control que son desconocidos (por ejemplo, un atributo **schemaLocation**), y algunos atributos conocidos que pueden insertar otras herramientas XML con valores que pueden ser distintos a los que un codificador conforme inserta (por ejemplo, utilizar un atributo de identificación de tipo). (Véase 10.2.10.)

10.1.3 Si una especificación ASN.1 no contiene instrucciones de codificación XER, cada codificación BASIC-XER de cualquier valor abstracto de un tipo ASN.1 también es una codificación EXTENDED-XER del mismo valor abstracto que dicho tipo.

NOTA – Lo contrario no es cierto. Incluso en ausencia de instrucciones de codificación XER, existen codificaciones EXTENDED-XER que no son conformes a codificaciones BASIC-XER (véase 10.1.2 a y 10.1.2 c).

10.1.4 Todas las ocurrencias de la notación "Type" ASN.1 tienen un conjunto asociado (posiblemente vacío) de instrucciones de codificación XER (las instrucciones de codificación finales asociadas). Las instrucciones de codificación están asociadas con un "Type" por medio de:

- a) (instrucciones de codificación heredadas) La presencia de instrucciones de codificación asociadas en el "Type" utilizado en la definición de una referencia de tipo ("typereference") utilizada como "Type"; y
- b) (instrucción de codificación objetivo) La asignación de una o más instrucciones de codificación XER a una ocurrencia de "Type" utilizando una sección de control de codificación XER (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, cláusula 50); y
NOTA – Un módulo ASN.1 sólo puede contener una sección de control de codificación XER, y por tanto sólo un "EncodingInstructionAssignmentList" XER (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 50.2).
- c) (instrucciones de codificación con prefijo) La asignación de una o más instrucciones de codificación XER a una ocurrencia de "Type" utilizando prefijos de tipo XER (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 30.3); y
- d) (instrucciones de codificación de lista de importación) La asignación de una o más instrucciones de codificación a todas las referencias de tipo importadas de un módulo ASN.1 identificado.

10.1.5 El efecto de asignar una instrucción de codificación XER es la adición, supresión, o sustitución de instrucciones de codificación asociadas (véase en la cláusula 15 las reglas que se aplican en asignaciones múltiples de instrucciones de codificación XER).

10.1.6 El orden (o manera) en la que las instrucciones de codificación llegan a formar parte (o son suprimidas) del conjunto de instrucciones de codificación asociadas no es significativo en la aplicación de las instrucciones de codificación finales.

10.1.7 Las instrucciones de codificación finales afectan a la codificación EXTENDED-XER de tipos. No tienen ningún otro impacto y, en particular, no están asociadas con ninguna referencia de valor definida utilizando el tipo, ni afectan a correspondencias de valores, ni a otras reglas de codificación.

NOTA – Sin embargo, existen requisitos sobre el carácter inequívoco de los nombres que se ven afectados por la presencia de una instrucción de codificación final **NAME**, **NAMESPACE** o **UNTAGGED**. Estos requisitos pueden interpretarse como restricciones sobre la forma en la que los tipos de dichas instrucciones de codificación finales pueden utilizarse, o como restricciones sobre la utilización de dichas instrucciones de codificación.

10.2 Conformidad EXTENDED-XER

10.2.1 Si una especificación ASN.1 asigna instrucciones de codificación XER según se indica en las cláusulas 11 a 17, de forma que un tipo o componente ASN.1 tenga instrucciones de codificación finales que violen las restricciones especificadas en la cláusula 18 y siguientes, dicha especificación ASN.1 no será conforme con esta Recomendación | Norma Internacional, incluso aunque (sin las instrucciones de codificación XER) satisficiera todos los requisitos de la Rec. UIT-T X.680 | ISO/CEI 8824-1.

NOTA – Sólo es ocasionalmente ilegal asignar una instrucción de codificación a un "Type", ya que ésta puede ser negada (suprimida del conjunto de instrucciones de codificación asociadas) por una asignación ulterior. Normalmente son las instrucciones de codificación finales las que determinan la conformidad de la especificación. En algunos casos (pero no en todos), se ignora una instrucción de codificación final que no sea aplicable al tipo al que se desea aplicar. Si las cláusulas que especifican la sintaxis de la aplicación de las instrucciones de codificación identifican circunstancias en las que se ignora una instrucción de codificación en la aplicación de las instrucciones de codificación finales, las cláusulas que especifican las codificaciones no mencionan normalmente la posible presencia de dicha instrucción de codificación final.

10.2.2 Una codificación EXTENDED-XER conforme de un tipo ASN.1 sin instrucciones de codificación finales, será la codificación producida por las reglas de codificación XML básica (BASIC-XER) especificadas en la cláusula 8, con las opciones de codificador adicionales especificadas en 10.2.5 y 10.2.6.

NOTA – Los decodificadores EXTENDED-XER deben, de acuerdo con 10.2.4, aceptar y procesar declaraciones de tipos de documentos W3C XML, pero éstas no son por lo general generadas por codificadores conformes, y no forman parte de codificaciones EXTENDED-XER.

10.2.3 La codificación EXTENDED-XER de un tipo ASN.1 con instrucciones de codificación finales, o con componentes (a cualquier profundidad y después de resolver todas las referencias de tipo) que tienen instrucciones de codificación asociadas, constituyen la codificación especificada en la cláusula 17.

NOTA – Las instrucciones de codificación finales se aplican en una codificación EXTENDED-XER, y también son utilizadas por decodificadores y validadores de codificaciones EXTENDED-XER.

10.2.4 Los decodificadores EXTENDED-XER (se utilice o no **MODIFIED-ENCODINGS** – véase la cláusula 26) procesarán cualquier declaración de tipo de documento (véase W3C XML 1.0, 2.8) que exista, de conformidad con los requisitos aplicables a procesadores XML de no validación (véase W3C XML 1.0, 5.1). Este procesamiento se realizará (conceptualmente) antes de aplicar todos los demás requisitos de decodificación de esta Recomendación | Norma Internacional. Los codificadores EXTENDED-XER no incluirán una declaración de tipo de documento.

10.2.5 Un codificador EXTENDED-XER puede (como opción del codificador) insertar instrucciones de procesamiento XML o un comentario XML (además de cualquier otro que pueda ser exigido de conformidad con la cláusula 30) en el elemento de documento XML o en el prólogo XML en cualquier posición permitida por W3C XML 1.0. La forma sintáctica y la semántica de las instrucciones de procesamiento XML se especifican en W3C XML 1.0, 2.6. La forma sintáctica y la semántica del comentario XML se especifica en W3C XML 1.0, 2.5.

10.2.6 Si no existe una instrucción de codificación **GLOBAL-DEFAULTS** con una palabra clave **MODIFIED-ENCODINGS** (véase la cláusula 26) en la sección de control de codificación XER, entonces:

- a) el "XMLIntegerValue" (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 18.9) puede ser "XMLSignedNumber" o "EmptyElementInteger", como opción del codificador; y
- b) el "XMLBitStringValue" (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 21.9) puede ser cualquiera de las alternativas de esta producción, como opción del codificador. Si se utiliza la "XMLIdentifierList", ésta será la "EmptyElementList".

10.2.7 Si existe una instrucción de codificación **GLOBAL-DEFAULTS** con una palabra clave **MODIFIED-ENCODINGS** (véase la cláusula 26) en la sección de control de codificación XER, entonces:

- a) el "XMLBooleanValue" (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 17.3) será "TextBoolean"; y
- b) el "ExtendedXMLIntegerValue" (véase 17.4) será el "ModifiedXMLIntegerValue" alternativo, definido en 17.8; y
NOTA – Esto permite la utilización de un valor de texto para los "NamedNumber" de tipo entero, como opción del codificador, pero también modifica la sintaxis de las codificaciones numéricas de un valor entero.
- c) el "ExtendedXMLEnumeratedValue" (véase 34.3) no será "EmptyElementEnumerated"; y
NOTA – En ausencia de un **GLOBAL-DEFAULTS** de **MODIFIED-ENCODINGS**, no puede ser "TextEnumerated" (véase 8.3.4 *bis* y 34.3).
- d) el "ExtendedXMLRealValue" (véase 17.4) será el "ModifiedXMLRealValue" alternativo, definido en 17.9; y
- e) el "XMLSpecialRealValue" (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 20.6) será el "TextReal" alternativo; y
- f) la alternativa de "XMLIdentifierList" en "XMLBitStringValue" (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 21.9) será "TextList" (véase 10.2.8 b); y
- g) el "XMLSequenceOfValue" y "XMLSetOfValue" (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 25.3 y 27.3) será "XMLDelimitedItem" para todos los tipos componentes, ignorándose el cuadro 5 (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 25.5); y
- h) el "xmlhstring" (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.13) no contendrá "espacio blanco" (véase 8.1.4); y
- i) todas las ocurrencias de "espacio blanco " que estén fuera de los rótulos XML o dentro de los valores de atributos XML pueden ser "espacios blanco con escapes" (véase 8.1.5), como opción del codificador.

NOTA – Existen varias instrucciones de codificación (tales como **UNTAGGED**) que no pueden ser utilizadas salvo que exista un **GLOBAL-DEFAULTS** de **MODIFIED-ENCODINGS**.

10.2.8 Si en la sección de control de codificación XER existe una instrucción de codificación **GLOBAL-DEFAULTS** con una palabra clave **MODIFIED-ENCODINGS** (véase la cláusula 26), un codificador **EXTENDED-XER** puede (como opción del codificador):

- a) utilizar el "TextInteger" alternativo de "ModifiedXMLIntegerValue" (véase 17.8), siempre que haya un "NamedNumber" para el valor entero en la definición de tipo (véase también 10.2.7 b).
NOTA – La utilización de esta codificación con valores denominados que han sido añadidos en una versión ulterior puede hacer que el valor abstracto representado sea ilegible para una implementación basada en una versión anterior de la especificación.
- b) utilizar "XMLIdentifierList" (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 21.9) para un "XMLBitStringValue", siempre que el valor cadena de bits (*bitstring*) que debe codificarse no contenga bits "uno" que no sean bits denominados (véase también 10.2.7 f);
NOTA – La utilización de esta codificación con valores denominados que han sido añadidos en una versión ulterior puede hacer que el valor abstracto representado sea ilegible para una implementación de una versión anterior de la especificación.

10.2.9 Cuando las opciones del codificador estén permitidas en una codificación **EXTENDED-XER**, los codificadores y validadores conformes, aceptarán todas las opciones.

10.2.10 Los codificadores y validadores conformes aceptarán, pero pueden ignorar, la presencia de un atributo del espacio de nombres de control en cualquier elemento XML de un codificación, salvo que su presencia y utilización sea tal como se especifica en las cláusulas 37 y 38. Los codificadores no generarán dichos atributos con la excepción de lo indicado en las cláusulas 37 y 38.

NOTA – Otras herramientas XML pueden insertar dichos atributos. En general, un decodificador **EXTENDED-XER** no puede determinar fácilmente el valor permitido y el significado de algunos atributos del espacio de nombres de control. Su presencia y valor pueden ser utilizados por una aplicación si (por ejemplo) están presentes elementos hijos XML inesperados que se pasan (como opción del decodificador) a la aplicación – en lugar de ser ignorados o producir un error total de decodificación.

10.2.11 Una especificación ASN.1 es ilegal salvo que, para todos los valores abstractos, el decodificador pueda determinar de forma inequívoca (utilizando solamente el nombre del rótulo XML y los contenidos de cualquier elemento XML previo) el componente ASN.1 (o marcador de extensión) al que está asociado un elemento XML.

NOTA 1 – La asociación no puede depender del contenido del elemento XML, o de sus atributos, o de cualquier elemento XML posterior.

NOTA 2 – Esta condición siempre se satisface cuando no existen instrucciones de codificación XER, pero la utilización inadecuada de **UNTAGGED** para suprimir rótulos asociados de contorno (por ejemplo), repeticiones (*sequence-of* o *set-of*) y alternativas (*choice*) y el uso inadecuado de **NAME**, pueden producir especificaciones ilegales.

NOTA 3 – La subcláusula 10.2.11 es una condición necesaria para codificaciones válidas, pero se reconoce que, en general, una herramienta ASN.1 (o un autor humano) no puede verificar la legalidad exclusivamente en base a esta declaración de alto nivel. En el anexo B se incluye un modelo del efecto de la utilización de **UNTAGGED**, y de reglas que, si se siguen, pueden asegurar la legalidad de la especificación, tal como se define en 10.2.11.

10.2.12 Si una especificación ASN.1 contiene varios "ObjectClassFieldType" que son tipos abiertos (véase la Rec. UIT-T X.681 | ISO/CEI 8824-2, 14.2), con limitaciones de cuadro o de tipo, dichas limitaciones pueden ignorarse en aplicación de 10.2.11.

10.3 Estructura de una codificación EXTENDED-XER

10.3.1 Una codificación EXTENDED-XER completa es un documento XML correctamente elaborado que consta de (en orden):

- a) un prólogo XML (que, como opción del codificador, puede estar vacío) tal como se especifica en 8.2; y
- b) un elemento documento XML que es la codificación completa de un valor de un único tipo ASN.1, denominado tipo raíz (*root*), tal como se especifica en la cláusula 17.

10.3.2 Las codificaciones "XMLValue" utilizadas para codificaciones BASIC-XER se modifican para las codificaciones EXTENDED-XER mediante las instrucciones de codificación finales para los "Type" que codifican, y mediante las instrucciones de codificación finales para sus componentes (hasta cualquier profundidad), junto con cualquier instrucción de codificación **GLOBAL-DEFAULTS**.

NOTA – En un caso extremo, todo el contenido del elemento documento XML para una estructura ASN.1 muy anidada puede (utilizando la instrucción de codificación **UNTAGGED**) constar únicamente de una secuencia lineal de elementos XML, en la que sólo el elemento raíz tiene elementos hijos. La utilización de **UNTAGGED** se limita a asegurar que todas las secuencias lineales resultantes de elementos XML pueden hacerse corresponder inequívocamente con los componentes de un valor abstracto de un tipo raíz ASN.1 (véase 10.2.11).

10.3.3 El elemento documento XML de una codificación EXTENDED-XER consta de un único elemento XML que será un "ExtendedXMLTypedValue" para el tipo que se codifique (el tipo raíz). Puede incluir atributos en su rótulo de inicio o rótulo de elemento vacío, y su contenido puede incluir tanto elementos hijos (véase W3C XML) como texto no rotulado (*untagged*). Los elementos hijos pueden tener atributos y un contenido que incluya tanto elementos hijos como texto no rotulado.

10.3.4 Los valores abstractos de los componentes de un tipo envolvente se codifican como "ExtendedXMLValue" (véase 17.4), posiblemente modificados por instrucciones de codificación que se aplican a ellos o a sus propios componentes. Dichos "ExtendedXMLValue":

- a) pueden estar precedidos por un rótulo de inicio XML e ir seguidos de un rótulo de terminación XML (denominados rótulos asociados) para formar un elemento dentro del "ExtendedXMLValue" del tipo envolvente; o
- b) pueden (utilizando una instrucción de codificación **UNTAGGED** sobre un tipo no codificable mediante caracteres) formar un contenido de elementos XML parcial para el "ExtendedXMLValue" del tipo envolvente; o

NOTA – En el anexo B se describe el resultado de aplicar **UNTAGGED** como la producción de un contenido de elemento XML parcial que puede combinarse con otras codificaciones a fin de formar el contenido de elemento XML para algún elemento envolvente cuyo tipo no sea **UNTAGGED**.

- c) pueden (utilizando una instrucción de codificación **UNTAGGED** sobre un tipo codificable mediante caracteres) formar el "ExtendedXMLValue" completo del componente; o
- d) pueden (utilizando una instrucción de codificación **ATTRIBUTE** sobre un tipo codificable mediante caracteres) formar el "CharacterEncodableValue" en el "QuotedValue" de un "Attribute" (véase 20.3.3).

10.3.5 Si un "ExtendedXMLValue" está vacío, y sus rótulos asociados no han sido suprimidos utilizando una instrucción de codificación **UNTAGGED**, los rótulos precedente y siguiente asociados pueden (como opción del codificador) ser sustituidos por un rótulo de elemento vacío XML (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 16.8). A ello se denomina rótulo de elemento vacío asociado.

10.3.6 La transformación que se especifica en 10.3.5 se realiza conceptualmente después de haber completado todo el proceso de codificación, y puede evitarse mediante una instrucción de codificación **PI-OR-COMMENT** (véase la cláusula 30) que produzca una o más instrucciones de procesamiento XML o elementos de comentarios XML entre el rótulo de inicio y el de terminación.

10.3.7 Al rótulo precedente asociado, rótulo siguiente asociado y rótulo de elemento vacío asociado se les conoce conjuntamente como rótulos asociados. Los nombres de elementos XML en los rótulos asociados se denominan nombres de rótulo asociado, y son (en ausencia de instrucciones de codificación finales **NAME** y **NAMESPACE**) identificadores, nombres de referencia de tipo o "xml:ns1:typename" (véase la Rec. UIT-TX.680 | ISO/CEI 8824-1, 11.25).

11 Notación, conjunto de caracteres y elementos de léxico utilizados en instrucciones de codificación XER

11.1 La notación utilizada para especificar la sintaxis de una "EncodingInstruction" en un prefijo de tipo XER (véase la cláusula 13), y en una "EncodingInstructionAssignmentList" de una sección de control de codificación XER (véase la cláusula 14) es la definida en la Rec. UIT-T X.680 | ISO/CEI 8824-1, cláusula 5.

11.2 La Rec. UIT-T X.680 | ISO/CEI 8824-1, cláusula 10, se aplica a una "EncodingInstruction" XER y a una "EncodingInstructionAssignmentList" XER.

NOTA – En particular, pueden aparecer caracteres espacio blanco ASN.1 arbitrarios entre elementos de léxico en ambas construcciones sintácticas, salvo que se utilice la notación "&" (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 5.4).

11.3 Las reglas generales especificadas en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.1, también se aplican a una "EncodingInstruction" XER y a una "EncodingInstructionAssignmentList" XER.

NOTA – En particular, siempre que se permita utilizar espacio blanco ASN.1 puede incluirse un comentario ASN.1, y en la Rec. UIT-T X.680 | ISO/CEI 8824-1, se especifican los requisitos para la inserción de un espacio blanco o un comentario entre elementos de léxico que, de no ser así, podrían resultar confusos.

11.4 En esta Recomendación | Norma Internacional se utilizan los elementos de léxico siguientes:

comment	(véase Rec. UIT-T X.680 ISO/CEI 8824-1, 11.6)
cstring	(véase Rec. UIT-T X.680 ISO/CEI 8824-1, 11.14)
identifier	(véase Rec. UIT-T X.680 ISO/CEI 8824-1, 11.3)
modulereference	(véase Rec. UIT-T X.680 ISO/CEI 8824-1, 11.5)
number	(véase Rec. UIT-T X.680 ISO/CEI 8824-1, 11.8)
typereference	(véase Rec. UIT-T X.680 ISO/CEI 8824-1, 11.2)
"{"	(véase Rec. UIT-T X.680 ISO/CEI 8824-1, 11.26)
"}"	(véase Rec. UIT-T X.680 ISO/CEI 8824-1, 11.26)
"."	(véase Rec. UIT-T X.680 ISO/CEI 8824-1, 11.26)
":"	(véase Rec. UIT-T X.680 ISO/CEI 8824-1, 11.26)
","	(véase Rec. UIT-T X.680 ISO/CEI 8824-1, 11.26)
;"	(véase Rec. UIT-T X.680 ISO/CEI 8824-1, 11.26)
"#"	(véase Rec. UIT-T X.680 ISO/CEI 8824-1, 11.26)
"*"	(véase Rec. UIT-T X.680 ISO/CEI 8824-1, 11.26)
": : ="	(véase Rec. UIT-T X.680 ISO/CEI 8824-1, 11.16)
"<"	(véase Rec. UIT-T X.680 ISO/CEI 8824-1, 11.26)
">"	(véase Rec. UIT-T X.680 ISO/CEI 8824-1, 11.26)
"</"	(véase Rec. UIT-T X.680 ISO/CEI 8824-1, 11.21)
"/>"	(véase Rec. UIT-T X.680 ISO/CEI 8824-1, 11.22)

En 17.8.3 y en 17.9 se definen y utilizan elementos de léxico adicionales ("modifiedXMLNumber" y "modifiedXMLRealNumber").

12 Palabras clave

12.1 Las palabras especificadas en 12.3 y 12.4 siguientes se pueden utilizar tanto en las "EncodingInstruction" XER como en las "EncodingInstructionAssignmentList" XER (además de algunas palabras reservadas ASN.1), y pueden aparecer en dichas construcciones sintácticas solamente con el significado asignado a ellas en las cláusulas siguientes de esta Recomendación | Norma Internacional, salvo lo especificado en 12.2.

12.2 Las palabras clave no son palabras reservadas, pero en una "EncodingInstruction" XER o una "EncodingInstructionAssignmentList" XER es necesaria una "typereference" ASN.1 que fuera igual a una palabra clave enumerada en 12.3, se utilizará la producción "ModuleAndTypeReference" (véase 14.2.2).

12.3 Las palabras clave son las siguientes:

AFTER-TAG	DEFAULT-FOR-EMPTY	REPLACE
AFTER-VALUE	ELEMENT	TEXT
ANY-ATTRIBUTES	EMBED-VALUES	UNCAPITALIZED
ANY-ELEMENT	GLOBAL-DEFAULTS	UNTAGGED
AS	IN	UPPERCASED
ATTRIBUTE	LIST	USE-NIL
BASE64	LOWERCASED	USE-NUMBER
BEFORE-TAG	MODIFIED-ENCODINGS	USE-ORDER
BEFORE-VALUE	NAME	USE-QNAME
CAPITALIZED	NAMESPACE	USE-TYPE
COLLAPSE	NOT	USE-UNION
CONTROL-NAMESPACE	PI-OR-COMMENT	WHITESPACE
DECIMAL	PREFIX	

12.4 Se utilizan palabras clave adicionales en la producción de "BuiltInTypeName" (véase 14.2.3), pero todas ellas son palabras reservadas ASN.1 (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.27) y nunca pueden ser utilizadas en ASN.1 como un "typereference".

13 Asignación de una instrucción de codificación XER a un tipo ASN.1 utilizando un prefijo de tipo

13.1 Las instrucciones de codificación finales para un tipo pueden:

- requerir la utilización de alternativas de un "ExtendedXMLValue" que no sean alternativas "XMLValue" para dicho tipo; o

NOTA – Las alternativas de la producción "ExtendedXMLValue" incluyen tanto las alternativas (inalteradas) de la producción "XMLValue" utilizadas en BASIC-XER, como las producciones alternativas seleccionadas mediante las instrucciones de codificación XER.
- cambiar el nombre de rótulo asociado, el "AttributeName", o el valor del atributo identificación de tipo para la codificación de dicho tipo; o
- hacer que el "ExtendedXMLValue" de un componente de un tipo ASN.1 se inserte como "CharacterEncodableValue" en el "QuotedValue" de un "Attribute" (véase 20.3.3); o
- especificar el nombre del espacio de nombres XML para los nombres e identificadores de referencia de tipo definidos en un módulo ASN.1 y recomendar el prefijo del espacio de nombres a utilizar con dicho espacio de nombres; o
- especificar cuándo debe utilizarse un nombre de espacio de nombres cualificado (en lugar de un nombre no cualificado) en un elemento XML, o como nombre de un atributo XML; o
- especificar la supresión de los rótulos asociados, dando, en general, lugar a texto no rotulado o a un contenido de elemento XML parcial (que puede estar precedido o seguido de otro contenido de elemento XML parcial – véase anexo B); o
- especificar la inserción de una o más instrucciones de procesamiento XML o comentarios XML (véase la cláusula 30):
 - antes del rótulo precedente asociado o del rótulo de elemento vacío asociado; o
 - entre el rótulo precedente asociado y el "ExtendedXMLValue"; o

NOTA 1 – Ello prohíbe la utilización de un rótulo de elemento vacío asociado.
 - entre el "ExtendedXMLValue" y el rótulo siguiente asociado; o

NOTA 2 – Ello prohíbe la utilización de un rótulo de elemento vacío asociado.
 - después del rótulo siguiente asociado.

NOTA 3 – Todo lo anterior prohíbe la utilización de **UNTAGGED** para suprimir los rótulos asociados (véase 30.2.2).

13.2 Las instrucciones de codificación XER pueden asignarse a tipos ASN.1 utilizando la producción "EncodingInstruction" en un prefijo de tipo XER o la producción "EncodingInstructionAssignmentList" en una sección de control de codificación XER. La asignación mediante un prefijo de tipo se especifica en esta cláusula. La asignación mediante una sección de control de codificación XER se especifica en la cláusula 14.

NOTA – El efecto de asignaciones múltiples de instrucciones de codificación de la misma categoría se especifica en la cláusula 15.

13.3 La producción XER "EncodingInstruction" se define como:

EncodingInstruction ::=

PositiveInstruction
| **NegatingInstruction**

PositiveInstruction ::=

AnyAttributeInstruction
| **AnyElementInstruction**
| **AttributeInstruction**
| **Base64Instruction**
| **DecimalInstruction**
| **DefaultForEmptyInstruction**
| **EmbedValuesInstruction**
| **GlobalDefaultsInstruction**
| **ListInstruction**
| **NameInstruction**
| **NamespaceInstruction**
| **PIOrCommentInstruction**
| **TextInstruction**
| **UntaggedInstruction**
| **UseNilInstruction**
| **UseNumberInstruction**
| **UseOrderInstruction**
| **UseQNameInstruction**
| **UseTypeInstruction**
| **UseUnionInstruction**
| **WhitespaceInstruction**

NegatingInstruction ::=

NOT PositiveInstruction
| **ElementInstruction**

13.4 "ElementInstruction" (véase la cláusula 24) es un sinónimo estricto de **NOT UNTAGGED**, y no es objeto de análisis en esta cláusula.

NOTA 1 – El sinónimo **ELEMENT** permite evitar la doble negación, así como la legibilidad humana de las especificaciones. Normalmente será utilizado (en contraposición a la instrucción de codificación **ATTRIBUTE**) para identificar la naturaleza de los tipos de alto nivel en el módulo ASN.1. Los tipos de alto nivel que no tienen instrucciones de codificación finales **ELEMENT** ni **ATTRIBUTE** soportarán tipos que no se corresponden directamente a atributos o elementos XML, y normalmente serán **UNTAGGED**.

NOTA 2 – No existe instrucción de codificación de negación para **ELEMENT**. Una instrucción de codificación **ELEMENT** puede ser cancelada por otra instrucción de codificación **UNTAGGED** ulterior, pero no se recomienda dicha utilización.

13.5 Cada utilización de "PositiveInstruction" en un prefijo de tipo XER o en una sección de control de codificación, asigna dicha instrucción de codificación XER al correspondiente "Type".

13.6 Si el "Type" de un "TypeAssignment" (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 15.1) tiene instrucciones de codificación finales, todas las utilizations de las correspondientes "typereference" (en el módulo que contiene la "TypeAssignment" o en cualquier otro módulo) heredan sus instrucciones de codificación finales asociadas, excepto las instrucciones de codificación finales **NAME** y **NAMESPACE**, que no se heredan.

NOTA – Estas dos instrucciones de codificación afectan al nombre XML utilizado en lugar del nombre de referencia de tipo. Cuando el nombre de referencia de tipo se utiliza para definir el tipo en una asignación de tipo o el tipo de un componente, no es conveniente heredar dichas instrucciones de codificación finales de su definición.

13.7 Una instrucción de codificación en un prefijo de tipo o en una sección de control de codificación puede ser una instrucción positiva, utilizada para añadir o sustituir una instrucción de codificación (utilización de "PositiveInstruction"), o una instrucción de negación utilizada para cancelar (utilización de "NegatingInstruction") una o más instrucciones de codificación asociadas.

- 13.8** Las instrucciones de codificación XER constan de cuatro partes (algunas de las cuales pueden estar vacías):
- NOT**, para indicar la negación o supresión de instrucciones de codificación de una categoría dada; y
NOTA 1 – Se utiliza en instrucciones de negación (excepto "ElementInstruction"), pero no en instrucciones positivas.
 - una palabra clave que identifique la categoría de la instrucción de codificación; y
NOTA 2 – Siempre está presente.
 - la identificación de una lista objetivo para la asignación de la instrucción de codificación (posiblemente con información de cualificación que restringe su utilización a un subconjunto de valores del tipo); y
NOTA 3 – Cuando se utiliza en un prefijo de tipo, la lista objetivo siempre es la producción "vacía", ya que el objetivo para la asignación siempre es el tipo asociado con el prefijo de tipo (véase 13.12). La lista objetivo está siempre "vacía" para la instrucción de codificación **GLOBAL-DEFAULTS**.
 - la sintaxis, específica de cada categoría de instrucción de codificación, que proporciona información detallada sobre la instrucción de codificación de dicha categoría.
NOTA 4 – Cuando se utilice en una instrucción de negación, siempre es la producción "vacía". Tampoco está presente en algunas instrucciones de codificación XER para las que la palabra clave es una definición suficiente.

13.9 Algunas instrucciones de codificación XER requieren la especificación del valor abstracto de un tipo. Esta especificación utiliza la producción "Value" (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 16.7). Si se utiliza una "valuereference" como "Value", dicha "valuereference" se definirá en (o será importado en) el módulo ASN.1 que contiene la instrucción de codificación XER.

NOTA – Ello significa que el valor puede ser especificado directamente utilizando una notación de valor ASN.1 básica, o mediante una referencia de valor que hubiera sido identificada utilizando una notación de valor ASN.1 básica o una notación de valor XML.

13.10 En la columna 1 del cuadro 1 se enumeran las alternativas que existen en las producciones "PositiveInstruction". En la columna 2 se incluyen las cláusulas en las que se especifican los requisitos para la utilización de estas instrucciones de codificación y las codificaciones modificadas que ellas producen. La columna 3 incluye la categoría de la instrucción de codificación.

NOTA – Estas categorías permiten disponer de declaraciones claras sobre el resultado de asignaciones múltiples de instrucciones de codificación de la misma categoría.

Cuadro 1 – Instrucciones de codificación y sus cláusulas definatorias y categorías

Instrucción de codificación	Cláusula definatoria	Categoría
AnyAttributesInstruction	Cláusula 18	Instrucción de cualquier atributo
AnyElementInstruction	Cláusula 19	Instrucción de cualquier elemento
AttributeInstruction	Cláusula 20	Instrucción de atributo
Base64Instruction	Cláusula 21	Instrucción Base64
DecimalInstruction	Cláusula 22	Instrucción decimal
DefaultForEmptyInstruction	Cláusula 23	Instrucción de valor por defecto para vacío
ElementInstruction	Cláusula 24	Instrucción de elemento
EmbedValuesInstruction	Cláusula 25	Instrucción de valores integrados
GlobalDefaultsInstruction	Cláusula 26	Instrucción de valores por defecto globales (pero véase 15.3)
ListInstruction	Cláusula 27	Instrucción de lista
NameInstruction	Cláusula 28	Instrucción de nombre (pero véase 15.2)
NamespaceInstruction	Cláusula 29	Instrucción de espacio de nombre
PiOrCommentInstruction	Cláusula 30	Instrucción Pi-o-comentario (pero véase 15.3)
TextInstruction	Cláusula 31	Instrucción de texto (pero véase 15.3)
UntaggedInstruction	Cláusula 32	Instrucción no rotulada
UseNilInstruction	Cláusula 33	Instrucción de utilización de nulo
UseNumberInstruction	Cláusula 34	Instrucción de utilización de número
UseOrderInstruction	Cláusula 35	Instrucción de utilización de orden
UseQNameInstruction	Cláusula 36	Instrucción de utilización de qname
UseTypeInstruction	Cláusula 37	Instrucción de utilización de tipo
UseUnionInstruction	Cláusula 38	Instrucción de utilización de unión
WhitespaceInstruction	Cláusula 39	Instrucción de espacio blanco

13.11 Cada una de las alternativas de la producción "PositiveInstruction" está en una categoría definida de instrucción de codificación (o en algunos casos abarca categorías múltiples). La categoría de cada instrucción de codificación se especifica en la columna 3 del cuadro 1 (no obstante, véase también 15.3 para las instrucciones de codificación que abarcan múltiples categorías).

NOTA – Las categorías de instrucciones de codificación se utilizan en 15.4 para determinar el efecto de la asignación múltiple de instrucciones de codificación.

13.12 El "TargetList" de todas las construcciones "EncodingInstruction" que aparecen en un prefijo de tipo estarán "vacías" y el objetivo será el "Type" asociado con el prefijo de tipo.

13.13 Una instrucción de negación está en la misma categoría que la correspondiente instrucción positiva.

13.14 Un tipo ASN.1 nunca puede tener asociado más de una instrucción de codificación XER de una categoría dada (véanse 15.3 y 15.4), con independencia de cómo se hayan asignado. El resultado de asignaciones múltiples de una instrucción de codificación XER de una categoría dada se especifica en la cláusula 15.

14 Asignación de una instrucción de codificación XER mediante una sección de control de codificación

14.1 Lista de asignación de una instrucción de codificación

14.1.1 Las instrucciones de codificación XER pueden asignarse a tipos ASN.1 utilizando la producción "EncodingInstruction" en un prefijo de tipo XER o la producción "EncodingInstructionAssignmentList" en una sección de control de codificación XER. La asignación utilizando un prefijo de tipo se especifica en la cláusula 13. La asignación utilizando una sección de control de codificación XER se especifica esta cláusula.

14.1.2 La producción XER "EncodingInstructionAssignmentList" se define como:

EncodingInstructionAssignmentList ::=
EncodingInstruction
EncodingInstructionAssignmentList ?

14.1.3 La producción "EncodingInstruction" se define en 13.3.

14.1.4 Cada utilización de una "EncodingInstruction" en una sección de control de codificación asigna dicha instrucción de codificación XER a las ocurrencias de "Type" que se identifican en la "TargetList" de la instrucción de codificación, o a las referencias de tipo en una lista de importaciones. La producción "TargetList" y los objetivos que identifica se especifican en 14.2.

14.1.5 Las subcláusulas 13.4 a 13.14 también se aplican a instrucciones de codificación de una sección de control de codificación. Las cláusulas que definen la sintaxis detallada de cada categoría de instrucción de codificación se enumeran en el cuadro 1. En el cuadro 1 también se enumeran las categorías de instrucciones de codificación XER.

14.2 Identificación de los objetivos de una instrucción de codificación XER mediante una lista de objetivos

14.2.1 Reglas generales

14.2.1.1 Las alternativas de "EncodingInstruction" especifican la instrucción de codificación XER asignada y el objetivo u objetivos para dicha asignación en el módulo ASN.1. Todos los objetivos son ocurrencias de la producción "Type" en el módulo ASN.1.

NOTA – Pueden especificarse objetivos múltiples en la misma o en diferentes asignaciones de tipo ASN.1. También puede especificarse un objetivo constituido por todo el módulo, o todas las ocurrencias en el módulo de un tipo o constructor integrado. Por lo tanto, (usando una sección de control de codificación XER) puede utilizarse una única "EncodingInstruction" para asignar una instrucción de codificación XER específica a todos los tipos de un módulo ASN.1 que requieran que se les asigne dicha instrucción de codificación.

14.2.1.2 La producción "TargetList" se utiliza para identificar el objetivo u objetivos para la asignación de una instrucción de codificación XER. Esto se define en las subcláusulas siguientes.

NOTA 1 – En la cláusula 18 y siguientes se hace referencia a la producción "TargetList".

NOTA 2 – La producción "TargetList" tiene una alternativa "vacía". Es la única alternativa permitida si en un prefijo de tipo se utiliza la "EncodingInstruction" (véase 13.12). En esta subcláusula 14.2 sólo se considera la utilización de "TargetList" en una sección de control de codificación.

14.2.1.3 La producción "TargetList" se define como:

```

TargetList ::=
    Targets " , " +
    | empty

Targets ::=
    TypeIdentification
    | BuiltInTypeIdentification
    | IdentifiersInContext
    | ImportedTypesIdentification

```

14.2.1.4 Si la "TargetList" es una lista de una o más producciones "Targets", cada uno de los "Targets" identifica uno o más objetivos (*targets*) (es decir, los "Type" a los que se asignan la instrucción de codificación), pero también pueden proporcionar información de cualificación para la instrucción de codificación, restringiendo su aplicación a codificaciones que utilicen un identificador particular en la definición de tipo objetivo, o a la utilización de rótulos de elemento vacío para caracteres de control especificados en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.15.5.

NOTA – La información de cualificación sólo existe si el objetivo es una definición de tipo booleano, cadena de bits (*bitstring*), enumerado, entero o cadena de caracteres restringida (véase 14.2.2.9).

14.2.1.5 Una "TargetList" compuesta de "vacío" ("*empty*") sólo está permitida en un prefijo de tipo (cuando es la única alternativa permitida) y en la instrucción de codificación **GLOBAL-DEFAULTS**. En un prefijo de tipo, identifica el tipo asociado con el prefijo. En la instrucción de codificación **GLOBAL-DEFAULTS**, identifica todos los "Types" del módulo.

14.2.1.6 La instrucción de codificación XER (posiblemente con información de cualificación asociada) se asigna a todos los tipos identificados en la "TargetList", tal como se especifica en 14.2.1.10 a 14.2.1.16.

NOTA – Sería algo inusual, pero no ilegal, que un "Type" dado fuera identificado más de una vez en la lista objetivo. En tales casos, se aplica la cláusula 15.

14.2.1.7 La identificación (tutorial) del objetivo u objetivos (y la posible información de cualificación) mediante la producción "Targets" emplea una de las cinco formas básicas siguientes:

- a) Utilización de una "typereference" (véase 14.2.2), posiblemente seguida de una lista de identificadores separados por puntos, para identificar:
 - 1) el "Type" de una asignación de tipo (sin identificadores); o
 - 2) el "Type" de una componente de una definición de tipo (que puede incluir componentes de alto nivel introducidos mediante la construcción **COMPONENTS OF** – véase 14.2.1.12); o
 - 3) una de las anteriores 1) o 2), más un identificador final (precedido de dos puntos, no de un punto) para un identificador utilizado en la definición de tipo objetivo, que proporciona la información de cualificación.
- b) Utilización de **ALL** como el último identificador en la forma a), para identificar todos los "Type" textualmente presentes en la definición de tipo (que se identifica mediante la referencia de tipo precedente y la lista de identificadores separados por puntos), o información de cualificación (precedida de dos puntos, no de un punto) para identificar todos los identificadores utilizados para valores de una definición de tipo booleano, cadena de caracteres (*bitstring*), enumerado o entero (que se identifica mediante la referencia de tipo precedente y la lista de identificadores separados por puntos) o para identificar todas las utilizaciones de rótulos de elementos vacíos XML empleados para representar algunos caracteres de control (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.15.5).
- c) Utilización de un "BuiltInTypeName" (véase 14.2.3), para identificar todos los "Type" del módulo definidos mediante el correspondiente nombre o constructor integrado, posiblemente seguido (en el caso de **BOOLEAN**, **BIT STRING**, **ENUMERATED**, **INTEGER** y tipos de cadena de caracteres restringidos solamente) de información de cualificación.
- d) Utilización de una lista de "identifier" seguida de **IN** (o de **ALL** seguido de **IN**, o bien **COMPONENTS** seguido de **IN**) y la forma a) anterior (véase también 14.2.4), para identificar:
 - 1) el "Type" de los componentes identificados de la forma a); o
 - 2) (utilización de **ALL**) todos los "Type" que ocurren textualmente en el "Type" identificado por la forma a); o
 - 3) (utilización de **COMPONENTS**) todos los "Type" que son componentes de alto nivel del "Type" identificado por la forma a).
- e) Utilización de "ImportedTypesIdentification" (véase 14.2.5), lo cual permite identificar todos los "typereference" de la lista **IMPORTS** que se importan de un módulo especificado.

NOTA 1 – El término "definición de tipo" utilizado en a) y b) anteriores, enfatiza que sólo pueden utilizarse identificadores presentes textualmente. Los identificadores no pueden utilizarse si el "Type" es una referencia de tipo.

NOTA 2 – En general, puede hacerse referencia a un componente mediante la utilización de a) o d) anteriores. Si es necesario hacer referencia a más de un componente, el tipo d) sería la opción preferida puesto que es la opción menos prolija, en cualquier otro caso, sería preferible a). Se trata de una cuestión de estilo.

14.2.1.8 Un tipo cadena de bits o cadena de octetos (*octetstring*) con una limitación de contenidos, que sólo contenga un tipo, se tratará como un tipo con una única componente, utilizando "*" como el identificador de componente a fin de asignar una instrucción objetivo al "Type" en la limitación de contenidos.

14.2.1.9 Una definición de tipo secuencia-de o conjunto-de se tratará como un tipo con un único componente, utilizando "*" como identificador de componente, a fin de asignar una instrucción objetivo al "Type" que sea el componente de secuencia-de o de conjunto-de.

NOTA – También es posible identificar este único componente utilizando el identificador de componente (si existe).

14.2.1.10 Si el objetivo es la utilización de un parámetro de relleno de tipo parametrizado, el objetivo hereda las instrucciones de codificación finales del parámetro real antes de que se asignen las instrucciones de codificación que tienen por objetivo el parámetro de relleno. La especificación sólo es legal si son legales las instrucciones de codificación finales resultantes para todas las instancias del tipo parametrizado.

NOTA 1 – Si se exporta el tipo parametrizado, las instrucciones de codificación finales para sus parámetros de relleno se transportan con él.

NOTA 2 – No se proporcionan mecanismos para asignar instrucciones de codificación directamente al "Type" de un parámetro real en una instancia de un tipo parametrizado.

14.2.1.11 Si el objetivo es un "SelectionType", éste hereda las instrucciones de codificación finales de la alternativa seleccionada del tipo elección, al que hace referencia el tipo selección (*selection*) después de lo cual se asignan las instrucciones de codificación asignadas al "SelectionType".

14.2.1.12 Si el objetivo es un componente producido como resultado de la transformación **COMPONENTS OF**, éste hereda las instrucciones de codificación finales del componente del tipo referenciado mediante **COMPONENTS OF**, después de lo cual, se asignan las instrucciones de codificación asignadas a los componentes producidos mediante **COMPONENTS OF**. Las instrucciones de codificación para el "Type" del que se extraen las componentes se ignoran.

14.2.1.13 Si la producción "Targets" es "TypeIdentification", los objetivos que identifica se especifican en 14.2.2.

14.2.1.14 Si la producción "Targets" es "BuiltInTypeIdentification", los objetivos que identifica se especifican en 14.2.3.

14.2.1.15 Si la producción "Targets" es "IdentifiersInContext", los objetivos que identifica se especifican en 14.2.4.

14.2.1.16 Si la producción "Targets" es "ImportedTypesIdentification", los objetivos que identifica se especifican en 14.2.5.

14.2.1.17 EJEMPLO: el ejemplo siguiente muestra una definición de tipo ASN.1 seguida de dos formas diferentes de asignar instrucciones de codificación XER en una sección de control de codificación y, finalmente, la misma definición de tipo ASN.1 con las instrucciones de codificación XER asignadas utilizando prefijos de tipo. Los tres enfoques dan lugar a la misma codificación EXTENDED-XER.

La definición de tipo es:

```
My-Type ::= SEQUENCE {
    field1 INTEGER,
    field2 CHOICE {
        first SEQUENCE OF INTEGER,
        second SEQUENCE OF OBJECT IDENTIFIER } }
```

Las instrucciones de codificación XER en la sección de control de codificación pueden ser:

```
ATTRIBUTE field1 IN My-Type
LIST first IN My-Type.field2
```

Alternativamente, podrían ser:

```
ATTRIBUTE My-Type.field1
LIST My-Type.field2.first
```

La definición de tipo con prefijos de tipo es :

```
My-Type ::= SEQUENCE {
    field1 [ATTRIBUTE] INTEGER,
    field2 CHOICE {
        first [LIST] SEQUENCE OF INTEGER,
        second SEQUENCE OF OBJECT IDENTIFIER } }
```

14.2.2 Identificación del objetivo utilizando una referencia de tipo e identificadores ASN.1

14.2.2.1 La producción de "TypeIdentification" se define como:

```

TypeIdentification ::=
  ALL
  | ModuleAndTypeReference
  ComponentReference ?
  QualifyingInformationPart ?

ModuleAndTypeReference ::=
  typereference
  | modulereference "." typereference

ComponentReference ::=
  "."
  ComponentIdList

ComponentIdList ::=
  ComponentId "." +

ComponentId ::=
  identifier
  | "*"
  | ALL

QualifyingInformationPart ::=
  ":"
  QualifyingInformation

QualifyingInformation
  identifier
  | ALL
  
```

14.2.2.2 Un "TypeIdentification" de ALL identifica todos los "Type" que existen en los "TypeAssignment" del módulo.

14.2.2.3 La producción "ModuleAndTypeReference" identifica el "Type" que se asigna a la "typereference". La "modulereference" en la "ModuleAndTypeReference" es la referencia de módulo para el módulo que contiene la "EncodingInstructionAssignmentList", y la "typereference" es una referencia de tipo que se define en el módulo. Se utiliza si y sólo si la "typereference" consta de los mismos caracteres que una de las palabras clave especificadas en 12.3, en cualquier otro caso, sólo se utilizará la "typereference".

14.2.2.4 Un símbolo "*" identifica el "Type" (el único) componente de un tipo secuencia-de (*sequence-of*) o conjunto-de (*set-of*), o el tipo en una limitación de contenidos que incluya un "Type".

NOTA – Esta forma puede utilizarse incluso cuando el componente secuencia-de (*sequence-of*) o conjunto-de (*set-of*) tiene un identificador, pero es preferible utilizar el identificador.

14.2.2.5 Si se utiliza ALL como un "ComponentId", será el último "ComponentId" de la "ComponentIdList" y no estará seguido de "QualifyingInformation".

14.2.2.6 Si el primer "ComponentId" de la "ComponentIdList" (existe y) es un identificador que esté textualmente presente (o que sea el resultado de la utilización de COMPONENTS OF) como identificador de componente en el "Type" identificado por la "ModuleAndTypeReference", entonces identifica el "Type" de dicho componente. Si no es un identificador que esté textualmente presente (o que sea consecuencia de la utilización de COMPONENTS OF) como identificador de componente en el "Type" identificado por la "ModuleAndTypeReference", dicha ocurrencia de "TypeIdentification" no es ilegal, pero no identifica ningún objetivo.

NOTA – Esto requiere que el tipo referenciado por la "ModuleAndTypeReference" sea una definición de tipo secuencia, conjunto, elección, secuencia-de o conjunto-de, o una definición de tipo cadena de bits o cadena de octetos con una limitación de contenidos que incluya un "Type".

14.2.2.7 Si un "ComponentId" posterior (excepto el último) de la "ComponentIdList" (si existe) es un identificador que esté textualmente presente como identificador de componente en el "Type" identificado por el "ComponentId" previo, entonces identifica el "Type" de dicho componente. Si no es un identificador de componente textualmente presente en el "Type" identificado por el "ComponentId" previo, entonces esta ocurrencia de "TypeIdentification" no es ilegal, pero no identifica ningún objetivo.

NOTA – La primera utilización de "ComponentId" puede hacer referencia a componentes introducidos por un COMPONENTS OF. Los componentes de dichos componentes no pueden ser identificados mediante un "ComponentId" posterior.

14.2.2.8 Si el último "ComponentId" de la "ComponentIdList" (existe y):

- a) es un identificador textualmente presente como identificador de componente en el "Type" identificado por el "ComponentId" previo, entonces identifica el "Type" de dicho componente; la instrucción de codificación será asignada a dicho "Type"; o
- b) es la palabra clave **ALL**, entonces la instrucción de codificación se asignará a todos los "Type" que estén textualmente presentes en la definición de tipo identificada por el "ComponentId" previo, que será un tipo con uno o más componentes.

14.2.2.9 La "QualifyingInformationPart" no estará presente salvo que la "ModuleAndTypeReference" con la "ComponentReference" (si existe) identifique objetivo(s) que sean:

- a) tipos booleanos; o
- b) tipos cadena de bits con bits denominados; o
- c) tipos enumerados; o
- d) tipos entero con números denominados; o
- e) tipos cadena de caracteres restringida.

14.2.2.10 El "identifier" alternativo de "QualifyingInformation" no se utilizará salvo que la "ModuleAndTypeReference" de la "ComponentReference" (si existe) identifique un único objetivo que no sea un tipo de cadena de caracteres restringida, o identifique una lista objetivo cuyos tipos sean booleanos. El "identifier" será un identificador en la definición de tipo objetivo si dicho objetivo no es de tipo booleano, o será **true** o **false**. El "identifier" es una información cualificadora que identifica que la instrucción de codificación sólo se aplica a codificaciones que utilizan dicho identificador.

14.2.2.11 Las alternativas **true** y **false** de "QualifyingInformation" para un tipo booleano especifican la información de cualificación que identifica que la instrucción de codificación sólo se aplica a la codificación de los valores abstractos **TRUE** o **FALSE**, respectivamente.

14.2.2.12 El **ALL** alternativo de "QualifyingInformation" no se utilizará salvo que el objetivo identifique (solamente) una o más definiciones de tipo para los tipos enumerados en 14.2.2.9. No se utilizará si el objetivo identifica uno o más objetivos de cadena de caracteres restringida, salvo que la instrucción de codificación que se aplique sea **NAMESPACE**. Especifica información de cualificación que identifica que la instrucción de codificación se aplica a todos los identificadores en las definiciones de tipo, o en el caso de una tipo cadena de caracteres restringida, a todos los usos de los rótulos de elemento vacío XML utilizados para representar los caracteres de control enumerados en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.15.5.

NOTA – No es posible utilizar información de cualificación con un "identifier" que afecte selectivamente a la representación de caracteres de control. En este caso sólo es posible **ALL**.

14.2.3 Identificación del objetivo utilizando un nombre de tipo integrado

14.2.3.1 La producción "BuiltInTypeIdentification" se define como:

BuiltInTypeIdentification ::=
BuiltInTypeName
BuiltInTypeQualifyingInformationPart ?

BuiltInTypeName ::=
BIT STRING
| BOOLEAN
| CHARACTER STRING
| CHOICE
| EMBEDDED PDV
| ENUMERATED
| EXTERNAL
| GeneralizedTime
| INSTANCE OF
| INTEGER
| NULL
| ObjectDescriptor
| OBJECT IDENTIFIER
| OCTET STRING
| REAL
| RELATIVE-OID

```

| SEQUENCE
| SEQUENCE OF
| SET
| SET OF
| UTCTime
| RestrictedCharacterStringType

```

BuiltInTypeQualifyingInformationPart ::=
": "

BuiltInTypeQualifyingInformation

BuiltInTypeQualifyingInformation

identifier

```

| ALL

```

14.2.3.2 La producción "BuiltInTypeIdentification" especifica que la instrucción de codificación se debe aplicar a todas las ocurrencias textuales en el módulo del correspondiente tipo integrado o de un tipo definido utilizando el constructor correspondiente.

14.2.3.3 "RestrictedCharacterStringType" se define en la Rec. UIT-T X.680 | ISO/CEI 8824-1, cláusula 37.

14.2.3.4 "BuiltInTypeQualifyingInformationPart" no existe salvo que el "BuiltInTypeName" sea **BOOLEAN**, **BIT STRING**, **ENUMERATED**, **INTEGER**, o un tipo cadena de octetos restringida.

NOTA – Para un tipo cadena de octetos restringida solo está permitida la forma **ALL** de "BuiltInTypeQualifyingInformation" (véase 14.2.2.10 y la subcláusula siguiente).

14.2.3.5 El "identifier" alternativo de "BuiltInTypeQualifyingInformation" no se utilizará salvo que el "BuiltInTypeName" sea **BOOLEAN**, y por tanto, sea **true** o **false**. Especifica información de cualificación que identifica que la instrucción de codificación sólo se aplica a la codificación de los valores abstractos **TRUE** o **FALSE**, respectivamente.

14.2.3.6 El **ALL** alternativo de "BuiltInQualifyingInformation" especifica información de cualificación que identifica que la instrucción de codificación se aplica a todos los identificadores utilizados en cualquier instancia de utilización del "BuiltInTypeName" dentro del módulo (o a todos los valores de la definición de tipo **BOOLEAN**, o a todos los rótulos de elemento vacío utilizados en valores del tipo cadena de caracteres restringida especificados – véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.15.5).

14.2.4 Utilización de identificadores en contexto

14.2.4.1 La producción "IdentifiersInContext" se define como:

```

IdentifiersInContext ::=
    IdentifierList
    IN
    TypeIdentification

```

```

IdentifierList ::=
    identifier " , " +
    | ALL
    | COMPONENTS

```

14.2.4.2 "TypeIdentification" se define en 14.2.2, e identifica un tipo definido en una declaración de asignación de tipo en el módulo, o un componente o subcomponente de un tipo definido en el módulo. La "QualifyingInformationPart" no estará presente.

14.2.4.3 El "Type" identificado por la "TypeIdentification" será un tipo secuencia, conjunto o elección, y a los efectos de esta cláusula se denomina "Type" identificado.

NOTA – La "TypeIdentification" en "IdentifiersInContext" no puede utilizarse para un tipo secuencia-de o conjunto-de. Dicha utilización está prohibida en aras de la claridad, pues no resulta menos prolija que la utilización directa de "TypeIdentification" en "Targets".

14.2.4.4 Cada "identifier" en "IdentifierList" será el "identifier" de un componente del "Type" identificado. La instrucción de codificación XER se asigna al "Type" de todos los componentes del "Type" identificado que tiene un componente "identifier" en la "IdentifierList".

14.2.4.5 La utilización de **ALL** para "IdentifierList" especifica que todos los componentes textualmente presentes (y todos los componentes textualmente presentes de dichos componentes, hasta cualquier nivel) en el "Type" identificado, son objetivos a los que se asigna la instrucción de codificación XER.

14.2.4.6 La utilización de **COMPONENTS** para la "IdentifierList" especifica que todos los componentes (al primer nivel) del "Type" identificado son objetivos a los que se asignan la instrucción de codificación XER.

14.2.5 Utilización de identificación de tipos importados

14.2.5.1 La producción "ImportedTypesIdentification" se define como:

ImportedTypesIdentification ::=
ALL IMPORTS FROM modulereference

14.2.5.2 La "modulereference" será una de las "modulereference" utilizadas en una de las "GlobalModuleReferences" de la cláusula importaciones ("imports") del módulo.

14.2.5.3 La instrucción de codificación XER se asigna a cada una de las "typereference" en la correspondiente "SymbolList", después de que se hayan asignado las instrucciones de codificación finales producidas por asignación en el módulo de exportación.

14.2.5.4 Si una "typereference" importada se exporta desde este módulo, las instrucciones de codificación finales heredadas por dicha "typereference" en un módulo que la importe, son las que han sido heredadas en dicho módulo de importación, y no se ven afectadas por la asignación de instrucciones de codificación utilizando una "ImportedTypesIdentification". Esta asignación sólo afecta a la utilización de la referencia de tipo en este módulo.

15 Asignación múltiple de instrucciones de codificación XER

15.1 Orden en el que consideran las asignaciones múltiples

15.1.1 Un "Type" que no sea una "typereference" tiene inicialmente un conjunto vacío de instrucciones de codificación asociadas.

15.1.2 Un "Type" que sea una "typereference" (que puede ser importado) tiene inicialmente el conjunto de instrucciones de codificación finales del "Type" que había sido asignado al mismo cuando se definió (posiblemente modificado por las instrucciones de codificación que le fueron asignadas en la lista de importaciones de un módulo de importación – véase 14.2.5).

15.1.3 A continuación se asignan instrucciones de codificación objetivo para un "Type" (utilizando una sección de control de codificación) en el mismo orden en que las instrucciones de codificación aparecen en la sección de control de codificación. Si el "Type" es identificado por más de un elemento de una "TargetList" (véase 14.2), se tratará como un caso de asignaciones múltiples de la misma instrucción de codificación a dicho "Type", con el mismo orden en que los elementos aparecen en la "TargetList".

NOTA – El efecto de 15.1.2 y 15.1.3 es que la asignación objetivo a un "Type" en una "TypeAssignment" siempre se sustituye por una asignación objetivo al "Type" definida utilizando la correspondiente "typereference", con independencia de la asignación objetivo que aparezca en primer lugar en la sección de control de codificación. Sin embargo, si realiza la misma asignación objetivo a todos los componentes de un tipo, así como a un componente individual de dicho tipo, el efecto dependerá del orden de las instrucciones de codificación en la sección de control de codificación.

15.1.4 A continuación se consideran las instrucciones de codificación con prefijo (utilizando un prefijo de tipo) asignadas a un tipo, considerando en primer lugar la instrucción de codificación situada más a la derecha (la más interna), y en último lugar la instrucción de codificación con prefijo situada más a la izquierda (la más externa).

15.1.5 Tal como se especifica en 14.2.1.10, las instrucciones de codificación se asignan a un parámetro de relleno solo después de que se hayan determinado las instrucciones de codificación finales para el parámetro real.

15.1.6 Tal como se especifica en 14.2.1.11 y 14.2.1.12, un "SelectionType" y los componentes producidos por una transformación **COMPONENTS OF** heredan en primer lugar las instrucciones de codificación finales del tipo original, y a continuación presentan las instrucciones de codificación destinadas a ellos.

15.1.7 Cada asignación de una instrucción de codificación produce un nuevo conjunto de instrucciones de codificación asociadas, tal como se especifica en 15.2 a 15.4.

15.2 Efecto de la asignación de una instrucción de codificación de negación

15.2.1 Todas las asignaciones de una instrucción de codificación de negación dan lugar a la supresión (de entre el conjunto de instrucciones de codificación asociadas) de cualquier instrucción de codificación de la misma categoría. Si no existen instrucciones de codificación asociadas de una categoría distinta, el conjunto queda vacío.

15.2.2 La instrucción de codificación **NOT GLOBAL-DEFAULTS** nunca se asignará.

15.2.3 En el caso de instrucciones de codificación con varias categorías (véase 15.3), una instrucción de codificación de negación suprime todas las instrucciones de codificación en cualquiera de dichas categorías.

NOTA – Una instrucción de codificación de negación nunca se convierte en parte del conjunto de instrucciones de codificación asociadas.

15.3 Asignación múltiple de instrucciones de codificación con varias categorías

15.3.1 Las instrucciones de codificación **NAME** y **TEXT** (véanse las cláusulas 28 y 31) pueden asignarse a un tipo para:

- a) cambiar el nombre de rótulo asociado (no existe "QualifyingInformation"); o
NOTA – Esto sólo se aplica a la instrucción de codificación **NAME**.
- b) cambiar la codificación "ExtendedXMLValue" suministrando un nuevo nombre para un "identifier" especificado que esté presente en la definición de tipo (existe "QualifyingInformation" que no es **ALL**); o
- c) cambiar la codificación "ExtendedXMLValue" suministrando una modificación que debe aplicarse a todos los "identifier" presentes en la definición de tipo (existe "QualifyingInformation" que tiene el valor **ALL**, con un objetivo que no es un tipo cadena de caracteres restringida).

15.3.2 En el caso 15.3.1 b), la instrucción de codificación para un "identifier" especificado se trata como una categoría diferente a la de una instrucción de codificación correspondiente a cualquier otro "identifier", y de la de una instrucción de codificación correspondiente a 15.3.1 a).

15.3.3 En el caso 15.3.1 c), la instrucción de codificación se amplía en un conjunto de instrucciones de codificación de tipo 15.3.1 b), con una instrucción de codificación para cada "identifier" presente en la definición de tipo.

15.3.4 La instrucción de codificación **PI-OR-COMMENT** (véase la cláusula 30) tiene cuatro categorías, correspondientes a las cuatro alternativas de "Position".

15.3.5 Sujeto a lo indicado en 15.3.3 a 15.3.4, la subcláusula 15.4 especifica las normas aplicables a asignaciones múltiples de instrucciones de codificación XER.

15.3.6 Cada una de las alternativas de la instrucción de codificación **GLOBAL-DEFAULTS** constituye una categoría diferenciada, pero cada categoría de esta instrucción de codificación se asignará como máximo una vez.

15.4 Asignación múltiple de instrucciones de codificación XER de la misma categoría

NOTA – Es previsible que no sea habitual la asignación múltiple de instrucciones de codificación XER de la misma categoría, excepto cuando se asigne globalmente una instrucción de codificación XML, y se asigne una instrucción de codificación sustitutiva (posiblemente de negación) a tipos o componentes específicos. Esta subcláusula especifica las reglas aplicables cuando se produce una asignación múltiple de instrucciones de codificación XER de la misma categoría. A esta cláusula también se hace referencia en 15.3.5 para el tratamiento de asignaciones múltiples de instrucciones de codificación **NAME**, **PI-OR-COMMENT** y **TEXT**.

15.4.1 Las asignaciones de instrucciones de codificación positivas dan lugar a la adición (al conjunto de instrucciones de codificación asociadas) de dicha instrucción de codificación XER si no existen otras instrucciones de codificación asociadas de la misma categoría.

15.4.2 La asignación de una instrucción de codificación **ELEMENT** siempre es equivalente a la asignación de una instrucción de codificación **NOT UNTAGGED**.

15.4.3 Si existe una instrucción de codificación de la misma categoría en el conjunto de instrucciones de codificación asociadas, dicha instrucción de codificación se elimina del conjunto y se añade la instrucción de codificación XER asignada.

NOTA – Si se asignan globalmente instrucciones de codificación en una sección de control de codificación, con la intención de sustituirlas en casos específicos, la sustitución debe hacerse utilizando un prefijo de tipo o una instrucción de codificación posterior en la sección de control de codificación, no una anterior.

15.4.4 Si un tipo que está incluido en una "ContentsConstraint" o en una "TypeConstraint", debe codificarse mediante EXTENDED-XER, y las instrucciones de codificación finales (tal como establecen las reglas anteriores) se utilizan para determinar la codificación de dicho tipo. Si un tipo está incluido en cualquier otra limitación ASN.1, se descartan todas las instrucciones de codificación asociadas.

15.5 Combinaciones permitidas de instrucciones de codificación finales

15.5.1 En el cuadro 2 se especifica las combinaciones permitidas de instrucciones de codificación finales para un "Type" cuando se ha utilizado un GLOBAL-DEFAULTS de MODIFIED-ENCODINGS. En la columna 1 se enumeran todas las instrucciones de codificación. En la columna 2 se enumeran todas las instrucciones de codificación que pueden utilizarse junto con la instrucción de codificación de la columna 1 como instrucción de codificación final, pero en muchos casos se aplican restricciones que se enumeran en las cláusulas aplicables.

NOTA – GLOBAL-DEFAULTS no aparece en el cuadro, ya que no se asigna a un "Type".

Cuadro 2 – Combinaciones permitidas de instrucciones de codificación finales con MODIFIED-ENCODINGS

Instrucción de codificación	Otras instrucciones de codificación permitidas
ANY-ATTRIBUTES (véase cláusula 18)	ELEMENT, NAME, NAMESPACE
ANY-ELEMENT (véase cláusula 19)	ELEMENT, NAME, NAMESPACE
ATTRIBUTE (véase cláusula 20)	BASE64, DECIMAL, ELEMENT, LIST, NAME, NAMESPACE, TEXT, USE-NUMBER, USE-QNAME, USE-UNION, WHITESPACE
BASE64 (véase cláusula 21)	ATTRIBUTE, DEFAULT-FOR-EMPTY, ELEMENT, NAME, NAMESPACE, PI-OR-COMMENT, UNTAGGED
DECIMAL (véase cláusula 22)	ATTRIBUTE, DEFAULT-FOR-EMPTY, ELEMENT, NAME, NAMESPACE, PI-OR-COMMENT, UNTAGGED
DEFAULT-FOR-EMPTY (véase cláusula 23)	BASE64, DECIMAL, ELEMENT, EMBED-VALUES, LIST, NAME, NAMESPACE, PI-OR-COMMENT, TEXT, USE-NIL, USE-NUMBER, USE-ORDER, USE-QNAME, USE-UNION, WHITESPACE
ELEMENT (véase 24)	Equivalente a NOT UNTAGGED
EMBED-VALUES (véase cláusula 25)	DEFAULT-FOR-EMPTY, ELEMENT, NAME, NAMESPACE, PI-OR-COMMENT, USE-NIL, USE-ORDER
LIST (véase cláusula 27)	ATTRIBUTE, DEFAULT-FOR-EMPTY, ELEMENT, NAME, NAMESPACE, PI-OR-COMMENT, UNTAGGED
NAME (véase cláusula 28)	Sin restricciones
NAMESPACE (véase cláusula 29)	Sin restricciones
PI-OR-COMMENT (véase cláusula 30)	BASE64, DECIMAL, DEFAULT-FOR-EMPTY, ELEMENT, EMBED-VALUES, LIST, NAME, NAMESPACE, TEXT, USE-NIL, USE-NUMBER, USE-ORDER, USE-QNAME, USE-TYPE, USE-UNION, WHITESPACE
TEXT (véase cláusula 31)	ATTRIBUTE, DEFAULT-FOR-EMPTY, ELEMENT, NAME, NAMESPACE, PI-OR-COMMENT, UNTAGGED
UNTAGGED (véase cláusula 32)	BASE64, DECIMAL, LIST, NAME, NAMESPACE, TEXT, USE-NUMBER, USE-QNAME, USE-UNION, WHITESPACE
USE-NIL (véase cláusula 33)	DEFAULT-FOR-EMPTY, ELEMENT, EMBED-VALUES, NAME, NAMESPACE, PI-OR-COMMENT, USE-ORDER
USE-NUMBER (véase cláusula 34)	ATTRIBUTE, DEFAULT-FOR-EMPTY, ELEMENT, NAME, NAMESPACE, PI-OR-COMMENT, UNTAGGED
USE-ORDER (véase cláusula 35)	DEFAULT-FOR-EMPTY, ELEMENT, EMBED-VALUES, NAME, NAMESPACE, PI-OR-COMMENT, USE-NIL
USE-QNAME (véase cláusula 36)	ATTRIBUTE, DEFAULT-FOR-EMPTY, ELEMENT, NAME, NAMESPACE, PI-OR-COMMENT, UNTAGGED
USE-TYPE (véase cláusula 37)	ELEMENT, NAME, NAMESPACE, PI-OR-COMMENT
USE-UNION (véase cláusula 38)	ATTRIBUTE, DEFAULT-FOR-EMPTY, ELEMENT, NAME, NAMESPACE, PI-OR-COMMENT, UNTAGGED
WHITESPACE (véase cláusula 39)	ATTRIBUTE, DEFAULT-FOR-EMPTY, ELEMENT, NAME, NAMESPACE, PI-OR-COMMENT, UNTAGGED

15.5.2 En el cuadro 3 se especifican las combinaciones permitidas de instrucciones de codificación finales cuando no se ha utilizado un GLOBAL-DEFAULTS de MODIFIED-ENCODINGS. En la columna 1 se enumeran todas las instrucciones de codificación permitidas como instrucciones de codificación finales si no se ha utilizado un GLOBAL-DEFAULTS de MODIFIED-ENCODINGS. La columna 2 dice "No permitida", o bien, enumera todas las instrucciones de codificación que pueden utilizarse en combinación con la instrucción de codificación de la columna 1, como instrucciones de codificación finales, pero en muchos casos se aplican las restricciones identificadas en las cláusulas aplicables. "No permitida" significa que dicha instrucción de codificación no puede utilizarse como instrucción de codificación final si no se ha utilizado un GLOBAL-DEFAULTS de MODIFIED-ENCODINGS.

NOTA – GLOBAL-DEFAULTS no aparece en el cuadro, ya que no se asigna a un "Type".

Cuadro 3 – Combinaciones permitidas de instrucciones de codificación finales sin MODIFIED-ENCODINGS

Instrucción de codificación	Otras instrucciones de codificación permitidas
ANY-ATTRIBUTES	No permitida
ANY-ELEMENT	No permitida
ATTRIBUTE	BASE64, LIST, NAME, TEXT, USE-NUMBER, WHITESPACE
BASE64	ATTRIBUTE, NAME, PI-OR-COMMENT
DECIMAL	No permitida
DEFAULT-FOR-EMPTY	No permitida
ELEMENT	No permitida
EMBED-VALUES	No permitida
LIST	ATTRIBUTE, NAME, PI-OR-COMMENT
NAME	ATTRIBUTE, BASE64, LIST, PI-OR-COMMENT, TEXT, USE-NUMBER, WHITESPACE
NAMESPACE	No permitida
PI-OR-COMMENT	BASE64, LIST, NAME, TEXT, USE-NUMBER, WHITESPACE
TEXT	ATTRIBUTE, NAME, PI-OR-COMMENT
UNTAGGED	No permitida
USE-NIL	No permitida
USE-NUMBER	ATTRIBUTE, NAME, PI-OR-COMMENT
USE-ORDER	No permitida
USE-QNAME	No permitida
USE-TYPE	No permitida
USE-UNION	No permitida
WHITESPACE	ATTRIBUTE, NAME, PI-OR-COMMENT

16 Soporte de instrucciones de codificación XER para espacios de nombres y nombres cualificados XML

16.1 El espacio de nombres ("Namespaces") W3C XML define conceptos y normas que rigen los cualificadores así como mecanismos necesarios para asegurar que el nombre de elemento o de atributo XML pueda identificarse correctamente con la correspondiente especificación de la semántica asociada.

16.2 El espacio de nombres W3C XML define un espacio de nombres XML como un conjunto de nombres inequívocos, identificados mediante un URI, que se utilizan en documentos XML como tipos de elementos y nombres de atributos. El URI que identifica un espacio de nombres se denomina nombre del espacio de nombres. En esta Recomendación | Norma Internacional, los espacios de nombres también se utilizan para cualificar los valores de un tipo que tiene una instrucción de codificación final de USE-QNAME (véase la cláusula 36) y que representa un QName XML (véase el "W3C XML Schema", parte 2, 3.2.18).

16.3 Los nombres e identificadores de referencia de tipo pueden (aunque no necesariamente deben) recibir la asignación de un espacio de nombres.

NOTA – Esta Recomendación | Norma Internacional utiliza un nombre de espacio de nombres que, por defecto, es una forma de URI basada en identificadores de objeto ASN.1 (véase la cláusula 29). Las restantes formas de URI pueden utilizarse para asignar un nombre de espacio de nombres a los nombres presentes en un módulo ASN.1.

16.4 El hecho de que un tipo forme parte de un espacio de nombres XML (y si así es, su nombre de espacio de nombres) viene determinado por la presencia (o ausencia) de una instrucción de codificación final **NAMESPACE**.

NOTA – Una instrucción de codificación **NAMESPACE** sólo puede estar presente si en la sección de control de codificación también está presente una instrucción de codificación **GLOBAL-DEFAULTS MODIFIED-ENCODINGS** (véase 29.2.1).

16.5 Un espacio de nombres está identificado por la producción "NameSpaceSpecification" que proporciona el identificador de recursos uniforme (URI) para el espacio de nombres y, opcionalmente, un prefijo del espacio de nombres recomendado. La "NameSpaceSpecification" se especifica en la cláusula 29.

16.6 Los nombres de elementos y atributos XML en una codificación EXTENDED-XER se generan a partir de varias fuentes. Las subcláusulas de 16.8 enumeran las fuentes de nombres de elementos y atributos XML, identifican los espacios de nombres de los que forman parte y especifican si deben ser nombres de espacio de nombres cualificado o no.

16.7 Un nombre de elemento XML, un nombre de atributo XML, o un valor de un atributo de identificación de tipo pueden (aunque no necesariamente deben) tener una instrucción de codificación final **NAMESPACE** en el "Type" que genera el nombre. Si así es, el nombre será un nombre de espacio de nombres cualificado en la codificación. (La cualificación del espacio de nombres en una codificación puede hacerse explícitamente utilizando un prefijo de espacio de nombres XML definido, o indirectamente estableciendo un espacio de nombres XML por defecto para un ámbito que incluya la utilización del nombre o del valor.) Si no existe una instrucción de codificación **NAMESPACE** en un "Type" que genera un nombre, éste no es un nombre de espacio de nombres cualificado. Los nombres que no sean nombres de espacio de nombres cualificados se denominan nombres no cualificados, y no existirán en el ámbito de un espacio de nombres XML por defecto establecido.

NOTA – BASIC-XER no soporta espacios de nombres XML, y en codificaciones BASIC-XER nunca existen nombres de espacio de nombres cualificados.

16.8 En las subcláusulas siguientes, el término "espacio de nombres ASN.1" hace referencia al espacio de nombres cuyo nombre y prefijo recomendado se especifican en 16.9. El término "espacio de nombres asignado" ("assigned namespace") hace referencia al espacio de nombres asignado por la instrucción de codificación **NAMESPACE** a un tipo. Si los nombres generados no son del espacio de nombres ASN.1, y no existe dicha asignación de nombre de espacio de nombres, los nombres de elementos XML, los nombres de atributos XML, y los valores de atributos de identificación de tipo son nombres no cualificados.

16.8.1 En todas las subcláusulas de 16.8, los nombres de elementos y de atributos de los rótulos XML (ya sean rótulos de elementos vacíos XML o rótulos de inicio XML) son nombres de espacio de nombres cualificado en una codificación si y solo si el "Type" generador tiene una instrucción de codificación final **NAMESPACE**.

16.8.2 Los nombres de elementos en rótulos de elementos vacíos XML utilizados para caracteres de control (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.15.5) no tienen espacio de nombres salvo que se asigne uno mediante la aplicación de una instrucción de codificación **NAMESPACE** al tipo cadena de caracteres restringida con información de cualificación de **ALL**.

16.8.3 Los nombres de elementos en rótulos de elemento vacío XML utilizados para valores de tipos entero, enumerado y cadena de bits, así como valores especiales de tipos reales (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1, 18.9, 19.8, 20.6 y 21.9) serán siempre nombres no cualificados (véase 16.7) en cualquier codificación de dichos tipos.

16.9 El espacio de nombres del atributo de identificación de tipo (véase la cláusula 37) y del atributo de identificación nula (véase la cláusula 33) es el espacio de nombres de control que, por defecto, constituye el espacio de nombres ASN.1, salvo que mediante una instrucción de codificación **GLOBAL-DEFAULTS** se especifique un espacio de nombres de control diferente (véase la cláusula 26). El espacio de nombres ASN.1 tiene un nombre de "urn:oid:2.1.5.2.0.1" (véase 40.3), y un prefijo de espacio de nombres recomendado de "asn1" (véase también 26.3.2).

16.10 Para un tipo cadena de octetos con una limitación de contenido que especifica una codificación EXTENDED-XER, cualquier valor abstracto del tipo cadena de octetos será una codificación EXTENDED-XER completa de un valor de un tipo ASN.1 (véase la Rec. UIT-T X.682 | ISO/CEI 8824-3, 11.5 y 11.6), y contendrá todas las declaraciones de espacio de nombres necesarias para todos los nombres cualificados con prefijo y sin prefijo que existan en el valor abstracto cadena de octetos.

NOTA – Dicho tipo de cadena de octetos se codifica como un "xmlhstring" o "Base64OctetStringValue". Las declaraciones de espacio de nombres que existan en el documento XML que contiene el "xmlhstring" o el "Base64OctetStringValue" no incluyen en su ámbito los nombres incluidos en la cadena de octetos.

16.11 Cuando un tipo abierto se codifica como "xmlhstring" o "Base64XMLOpenTypeFieldVal", y las reglas de codificación utilizadas para el tipo contenido son EXTENDED-XER, el "xmlhstring" o el "Base64XMLOpenTypeFieldVal" serán representaciones hexadecimales o en base64 (respectivamente) de una cadena de octetos que constituye una codificación EXTENDED-XER completa de un valor del tipo contenido, e incluirá todas las declaraciones de espacios de nombres necesarias para todos los nombres cualificados con prefijo y sin prefijo presentes en el mismo.

NOTA – Las declaraciones de espacio de nombres que existan en el documento XML que contiene el "xmlhstring" o el "Base64XMLOpenTypeFieldVal", no incluyen en su ámbito los nombres incluidos en la cadena de octetos.

17 Especificación de codificaciones EXTENDED-XER

La especificación de codificaciones EXTENDED-XER utiliza las producciones especificadas en las subcláusulas siguientes. Estas producciones permiten usar toda la sintaxis de las producciones correspondientes que utiliza BASIC-XER (del mismo nombre, pero eliminando el término "Extended") y proporcionan una sintaxis adicional permitida en las codificaciones EXTENDED-XER. La utilización de esta sintaxis adicional viene determinada por la aplicación de instrucciones de codificación XER, y se especifica en las cláusulas 18 a 39.

NOTA – Las producciones alternativas disponibles están frecuentemente restringidas por la utilización o no de una instrucción de codificación GLOBAL-DEFAULTS con la palabra clave MODIFIED-ENCODINGS (véanse 10.2.7 y 10.2.8). En particular, ello permite controlar la utilización de elementos vacíos o codificaciones de texto para algunos tipos integrados.

17.1 Elemento documento XML

17.1.1 El elemento documento XML será un "ExtendedXMLTypedValue".

17.1.2 "ExtendedXMLTypedValue" se define como:

```
ExtendedXMLTypedValue ::=
  "<" & TypeNameOrModifiedTypeName AttributeList ">"
  ExtendedXMLValue
  "</" & TypeNameOrModifiedTypeName ">"
  | "<" & TypeNameOrModifiedTypeName "/>"
```

NOTA – Las diferencias con respecto a la producción "XMLTypedValue" son la inclusión de una "AttributeList" probablemente vacía, y la utilización de un "ExtendedXMLValue" en lugar de un "XMLValue" para los contenidos del elemento XML.

17.1.3 "TypeNameOrModifiedTypeName" se define en 17.2.

17.1.4 "AttributeList" se define en 17.3.

17.1.5 "ExtendedXMLValue" se define en 17.4 y será el "ExtendedXMLValue" del tipo identificado por el "TypeNameOrModifiedTypeName".

17.1.6 La segunda alternativa de "XMLTypedValue" (empleo de un rótulo de elemento vacío XML) puede utilizarse sólo si una instancia de la producción "ExtendedXMLValue" está vacía.

NOTA – Si la producción "ExtendedXMLValue" fuera una "xmlcstring" que sólo contuviera "espacios blancos", no estaría vacía y no podría utilizarse la segunda alternativa.

17.2 Producción "TypeNameOrModifiedTypeName"

17.2.1 "TypeNameOrModifiedTypeName" se define como:

```
TypeNameOrModifiedTypeName ::=
  NonParameterizedTypeName
  | QualifiedOrUnqualifiedName
```

17.2.2 "NonParameterizedTypeName" se define en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 13.2, y se utiliza (tal como se especifica en dicha subcláusula y en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 13.4 a 13.7) como un nombre de elemento XML que identifica un tipo ASN.1.

17.2.3 "QualifiedOrUnqualifiedName" se especifica en 29.3.2. El "QualifiedOrUnqualifiedName" alternativo se utilizará si y solo si existe una instrucción de codificación final NAME o NAMESPACE aplicada a dicho tipo (véase la cláusula 28), en caso contrario, se utilizará el "NonParameterizedTypeName".

17.3 Producción "AttributeList"

17.3.1 "AttributeList" se define como:

```
AttributeList ::=
  Attribute AttributeList
  | empty
```

17.3.2 "Attribute" se define en 20.3.3.

17.3.3 "AttributeList" estará vacío salvo que la aplicación de instrucciones de codificación finales requiera su utilización (véanse las cláusulas 20, 33 y 37).

17.3.4 Los "Attribute" de la "AttributeList" estarán precedidos de "espacios blancos" (véase 8.1.4).

17.4 Producción "ExtendedXMLValue"

17.4.1 "ExtendedXMLValue" se define como:

```
ExtendedXMLValue ::=
  ExtendedXMLBuiltinValue
  | ExtendedXMLObjectClassFieldValue
  | empty
```

```
ExtendedXMLBuiltinValue ::=
  XMLBitStringValue
  | XMLBooleanValue
  | ExtendedXMLCharacterStringValue
  | ExtendedXMLChoiceValue
  | XMLEmbeddedPDVValue
  | ExtendedXMLEnumeratedValue
  | XMLExternalValue
  | XMLInstanceOfValue
  | ExtendedXMLIntegerValue
  | XMLNullValue
  | XMLObjectIdentifierValue
  | ExtendedXMLOctetStringValue
  | ExtendedXMLRealValue
  | XMLRelativeOIDValue
  | ExtendedXMLSequenceValue
  | ExtendedXMLSequenceOfValue
  | ExtendedXMLSetValue
  | ExtendedXMLSetOfValue
  | ExtendedXMLPrefixedValue
```

```
ExtendedXMLCharacterStringValue ::=
  ExtendedXMLRestrictedCharacterStringValue
  | XMLUnrestrictedCharacterStringValue
```

```
ExtendedXMLRestrictedCharacterStringValue ::=
  XMLRestrictedCharacterStringValue
  | Base64XMLRestrictedCharacterStringValue
```

```
ExtendedXMLObjectClassFieldValue ::=
  ExtendedXMLOpenTypeFieldVal
  | XMLFixedTypeFieldVal
```

```
ExtendedXMLOpenTypeFieldVal ::=
  ExtendedXMLTypedValue
  | Base64XMLOpenTypeFieldVal
  | xmlhstring
```

```
ExtendedXMLOctetStringValue ::=
  ExtendedXMLTypedValue
  | Base64XMLOctetStringValue
  | xmlhstring
```

ExtendedXMLRealValue ::=
XMLRealValue
 | **ModifiedXMLRealValue**

ExtendedXMLIntegerValue ::=
XMLIntegerValue
 | **ModifiedXMLIntegerValue**

ExtendedXMLPrefixedValue ::=
ExtendedXMLValue

17.4.2 Las alternativas de "ExtendedXMLBuiltinValue" cuyos nombres de producción no comiencen por "Extended", así como su utilización para codificar valores abstractos, se especifican íntegramente en la Rec. UIT-T X.680 | ISO/CEI 8824-1 (véanse 16.10 y 16.2 de dicha Recomendación | Norma Internacional) y en la Rec. UIT-T X.681 | ISO/CEI 8824-2, 14.6 (para "XMLFixedTypeFieldVal" y la tercera alternativa de "ExtendedXMLOpenTypeFieldVal").

17.4.3 "Base64XMLRestrictedCharacterStringValue" se define en 21.3.5 y sólo se utilizará tal como se especifica en dicha subcláusula.

17.4.4 "ExtendedXMLChoiceValue" se define en 17.5 y sólo se utilizará tal como se especifica en dicha subcláusula.

17.4.5 "ExtendedXMLEnumeratedValue" se define en 34.3 y sólo se utilizará tal como se especifica en dicha subcláusula.

17.4.6 "ExtendedXMLSequenceValue" y "ExtendedXMLSetValue" se definen en 17.6 y sólo se utilizarán tal como se especifica en dicha subcláusula.

17.4.7 "ExtendedXMLSequenceOfValue" y "ExtendedXMLSetOfValue" se definen en 17.7 y sólo se utilizarán tal como se especifica en dicha subcláusula.

17.4.8 "Base64XMLOctetStringValue" y "Base64XMLOpenTypeFieldVal" se definen en 21.3.2 y 21.3.4, y sólo se utilizarán tal como se especifica en dichas subcláusulas.

17.4.9 "ModifiedXMLIntegerValue" se define en 17.8 y sólo se utilizará tal como se especifica en dicha subcláusula.

17.4.10 "ModifiedXMLRealValue" se define en 17.9 y sólo se utilizará tal como se especifica en dicha subcláusula.

17.4.11 La alternativa "vacío" ("*empty*") de "ExtendedXMLValue" sólo se utilizará tal como se especifica en la cláusula 23.

NOTA – Las restantes alternativas de "ExtendedXMLValue" también pueden producir un elemento de léxico "vacío". Esta subcláusula no afecta a la utilización de dichas ocurrencias.

17.5 Producción "ExtendedXMLChoiceValue"

17.5.1 "ExtendedXMLChoiceValue" se define como:

ExtendedXMLChoiceValue ::=
 "<" & **TagName** **AttributeList** ">"
ExtendedXMLValue
 "</" & **TagName** ">"
 | **ExtendedXMLValue**

TagName ::=
IdentifierOrModifiedIdentifier

IdentifierOrModifiedIdentifier ::=
identifier
 | **QualifiedOrUnqualifiedName**

17.5.2 "QualifiedOrUnqualifiedName" se define en 29.3.2. "QualifiedOrUnqualifiedName" se utiliza si y solo si existe una instrucción de codificación final **NAME** (véase la cláusula 28), o una instrucción de codificación final **NAMESPACE** aplicada al tipo (véase la cláusula 29), en caso contrario, se utilizará el "identifier".

NOTA – Si se utiliza "identifier", la codificación no puede incluir una declaración de espacio de nombres XML por defecto con un ámbito que incluya la utilización de dicho "identifier" (véase 16.7).

17.5.3 "AttributeList" y su utilización se define en 17.3 y en las cláusulas que allí se referencian.

17.5.4 El "ExtendedXMLValue" de las dos alternativas de "ExtendedXMLChoiceValue" es el "ExtendedXMLValue" de la alternativa seleccionada del tipo elección (*choice*).

17.5.5 La segunda alternativa de "ExtendedXMLChoiceValue" se utiliza si:

- a) la alternativa seleccionada del tipo de elección tiene una instrucción de codificación final **UNTAGGED** (véase la cláusula 32); o
- b) el tipo de elección tiene una instrucción de codificación final **USE-TYPE** o **USE-UNION** (véanse las cláusulas 37 y 38).

NOTA – Esto significa que la presencia de estas instrucciones de codificación finales produce la omisión de rótulos XML como elección determinante, y la determinación de la elección debe producirse por otros medios (véanse las cláusulas 37 y 38 y el anexo B).

17.6 Producciones "ExtendedXMLSequenceValue" y "ExtendedXMLSetValue"

17.6.1 "ExtendedXMLSequenceValue" y "ExtendedXMLSetValue" se definen como:

ExtendedXMLSequenceValue ::=

ExtendedXMLComponentValueList
| **empty**

ExtendedXMLSetValue ::=

ExtendedXMLComponentValueList
| **empty**

ExtendedXMLComponentValueList ::=

ExtendedXMLNamedValue
| **ExtendedXMLComponentValueList ExtendedXMLNamedValue**

ExtendedXMLNamedValue ::=

"<" & TagName AttributeList ">"
ExtendedXMLValue
"</" & TagName ">"
| **ExtendedXMLValue**

17.6.2 Las alternativas "vacías" de "ExtendedXMLSequenceValue" y "ExtendedXMLSetValue" sólo se utilizan si ningún componente del tipo secuencia o conjunto (a cualquier nivel) produce un "ExtendedXMLNamedValue", después de la resolución de todas las referencias de tipo y después de la aplicación de todas las instrucciones de codificación finales.

NOTA – Ello incluye (aunque no está limitado a) los casos en los que todos los componentes están marcados como **DEFAULT** u **OPTIONAL** y en los que se omiten todos los valores, tienen una instrucción de codificación final **UNTAGGED** y sus valores tienen una codificación vacía y cuando tienen una instrucción de codificación final **ATTRIBUTE**. También se incluyen las combinaciones de los casos anteriores, así como cuando la notación tipo sea **SEQUENCE {}** o **SET {}**.

17.6.3 "TagName" se define en 17.5.1. El "QualifiedOrUnqualifiedName" en la forma "IdentifierOrModifiedIdentifier" de "TagName" se utiliza si y solo si existe una instrucción de codificación final **NAME** o **NAMESPACE** aplicada a dicho tipo (véase la cláusula 29), en caso contrario, se utilizará el "identifier".

17.6.4 La "AttributeList" y su utilización se define en 17.3 y en las cláusulas a las que ésta hace referencia.

17.6.5 El "ExtendedXMLValue" en ambas alternativas de "ExtendedXMLNamedValue" será el "ExtendedXMLValue" del componente del tipo secuencia o conjunto.

17.6.6 La segunda alternativa de "ExtendedXMLSequenceValue" y "ExtendedXMLSetValue" se utiliza si y solo si la alternativa tiene una instrucción de codificación final **UNTAGGED** (véase la cláusula 32).

17.7 Producciones "ExtendedXMLSequenceOfValue" y "ExtendedXMLSetOfValue"

17.7.1 "ExtendedXMLSequenceOfValue" y "ExtendedXMLSetOfValue" se definen como:

ExtendedXMLSequenceOfValue ::=

ExtendedXMLValueList
| **ExtendedXMLDelimitedItemList**
| **empty**
| **ExtendedXMLListValue**

ExtendedXMLSetOfValue ::=

ExtendedXMLValueList
| **ExtendedXMLDelimitedItemList**
| **empty**
| **ExtendedXMLListValue**

```

ExtendedXMLValueList ::=
    ExtendedXMLValueOrEmpty
  | ExtendedXMLValueOrEmpty ExtendedXMLValueList

ExtendedXMLValueOrEmpty ::=
    ExtendedXMLValue
  | "<" & TypeNameOrModifiedTypeName ">"

ExtendedXMLDelimitedItemList ::=
    ExtendedXMLDelimitedItem
  | ExtendedXMLDelimitedItem ExtendedXMLDelimitedItemList

ExtendedXMLDelimitedItem ::=
    "<" & TypeNameOrModifiedTypeName AttributeList ">"
  ExtendedXMLValue
    "</" & TypeNameOrModifiedTypeName ">"
  | "<" & IdentifierOrModifiedIdentifier AttributeList ">"
  ExtendedXMLValue
    "</" & IdentifierOrModifiedIdentifier ">"
  | ExtendedXMLValue

```

17.7.2 La utilización de alternativas de "ExtendedXMLSequenceOfValue", "ExtendedXMLSetOfValue" y de "ExtendedXMLValueList" serán conformes con la utilización de las alternativas correspondientes de "XMLSequenceOfValue", "XMLSetOfValue" y "XMLValueList" (respectivamente), tal como se especifica en la Rec. UIT-T X.680 | ISO/CEI 8824-1, cláusulas 25 y 27, salvo que exista una instrucción de codificación **GLOBAL-DEFAULTS** con una palabra clave **MODIFIED-ENCODINGS**, la "ExtendedXMLValueList" no se utilizará nunca (véase también 10.2.7 g).

17.7.3 "ExtendedXMLListValue" se define en 27.3.2. Estas alternativas de "ExtendedXMLSequenceOfValue" y "ExtendedXMLSetOfValue" se utilizan si y solo si existe una instrucción de codificación final **LIST** (véase la cláusula 27) en el tipo secuencia-de o conjunto-de.

17.7.4 La primera alternativa del "ExtendedXMLDelimitedItem" se utiliza si y solo si el tipo secuencia-de o conjunto-de no contiene un "identifier" y el componente no tiene una instrucción de codificación final **UNTAGGED**. Se aplican las subcláusulas siguientes.

17.7.4.1 Si el componente del tipo secuencia-de o conjunto-de es una "typereference" o una "ExternalTypeReference" (posiblemente con uno o más "TypePrefix"), el "TypeNameOrModifiedTypeName" será la "typereference" o la "typereference" presente en la "ExternalTypeReference", respectivamente, posiblemente modificada conforme a las instrucciones de codificación **NAME** y **NAMESPACE** finales aplicadas al componente (véase la cláusula 28).

17.7.4.2 Si el componente del tipo secuencia-de o conjunto-de (después de ignorar cualquier ocurrencia de "TypePrefix") no es una "typereference" o una "ExternalTypeReference", el "TypeNameOrModifiedTypeName" será el "xmlas1typename" especificado en la Rec. UIT-T X.680 | ISO/CEI 8824-1, cuadro 4, correspondiente al tipo integrado del componente, posiblemente modificado conforme a la instrucción de codificación final **NAMESPACE** aplicada al componente (véase la cláusula 29).

17.7.5 La segunda alternativa del "ExtendedXMLDelimitedItem" se utiliza si y solo si el tipo secuencia-de o conjunto-de contiene un "identifier" y el componente no tiene una instrucción de codificación final **UNTAGGED**. El "IdentifierOrModifiedIdentifier" será dicho "identifier", posiblemente modificado conforme a las instrucciones de codificación finales **NAME** y **NAMESPACE** aplicadas al componente (véanse las cláusulas 28 y 29).

17.7.6 La tercera alternativa del "ExtendedXMLDelimitedItem" se utiliza si y solo si el componente del tipo secuencia-de o conjunto-de contiene una instrucción de codificación final **UNTAGGED** (véase la cláusula 32).

17.7.7 El "ExtendedXMLValue" presente en todas las alternativas de "ExtendedXMLDelimitedItem" será el "ExtendedXMLValue" del componente repetido del tipo secuencia-de o conjunto-de.

17.7.8 El "TypeNameOrModifiedTypeName" de "ExtendedXMLValueOrEmpty" será el "xmlas1typename" especificado en la Rec. UIT-T X.680 | ISO/CEI 8824-1, cuadro 4, correspondiente al tipo integrado del componente, posiblemente modificado conforme a cualquier instrucción de codificación final **NAMESPACE** aplicada al componente (véase la cláusula 29).

17.8 Producción "ModifiedXMLIntegerValue"

17.8.1 "ModifiedXMLIntegerValue" se define como:

ModifiedXMLIntegerValue ::=
ModifiedXMLSignedNumber
| TextInteger

ModifiedXMLSignedNumber ::=
modifiedXMLNumber
| "-" & modifiedXMLNumber
| "+" & modifiedXMLNumber

17.8.2 Esta alternativa de "ExtendedXMLIntegerValue" (véase 17.4) sólo se utiliza si se asigna una instrucción de codificación **GLOBAL-DEFAULTS** con una palabra clave **MODIFIED-ENCODINGS**.

17.8.3 El elemento de léxico "modifiedXMLNumber" consta de uno o más dígitos.

NOTA 1 – El elemento de léxico "modifiedXMLnumber" se hace corresponder con un valor entero interpretándolo como notación decimal.

NOTA 2 – Este elemento de léxico sólo difiere del elemento de léxico "number" (véase Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.8) en que permite cualquier número de dígitos "0" iniciales.

17.8.4 Cualquier valor entero positivo puede codificarse utilizando, como opción del codificador, la primera o la tercera alternativa de "ModifiedXMLSignedNumber". Un valor entero negativo se codifica utilizando la segunda alternativa. El valor entero cero puede codificarse utilizando, como opción del codificador, cualquiera de las tres alternativas.

17.8.5 "TextInteger" se define en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 18.9, y proporciona una codificación alternativa (como opción del codificador) para valores enteros que tengan una definición "NamedNumber".

17.9 Producción "ModifiedXMLRealValue"

17.9.1 "ModifiedXMLRealValue" se define como:

ModifiedXMLRealValue ::=
ModifiedXMLNumericRealValue
| XMLSpecialRealValue
| XMLDecimalMinusZeroRealValue

ModifiedXMLNumericRealValue ::=
modifiedXMLRealNumber
| "-" & modifiedXMLRealNumber
| "+" & modifiedXMLRealNumber

17.9.2 Esta alternativa de "ExtendedXMLRealValue" (véase 17.4) sólo se utiliza si se asigna una instrucción de codificación **GLOBAL-DEFAULTS** con una palabra clave **MODIFIED-ENCODINGS**.

17.9.3 El elemento de léxico "modifiedXMLRealNumber" consta de una parte entera formada por una serie de uno o más dígitos y, opcionalmente, un punto decimal (.). El punto decimal puede estar seguido, opcionalmente, de una parte fraccionaria de uno o más dígitos. La parte entera, el punto decimal o la parte fraccionaria (cualquiera que sea la última que esté presente) puede, opcionalmente, estar seguida de una letra "e" o "E" y de un exponente compuesto por uno o más dígitos que, también opcionalmente, puede incluir signo.

NOTA – Este elemento de léxico sólo difiere del elemento de léxico "realnumber" (véase Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.9) porque permite cualquier número de ceros iniciales en el exponente.

17.9.4 Cualquier valor real positivo y el valor real cero positivo pueden codificarse utilizando, como opción del codificador, la primera o tercera alternativa de "ModifiedXMLNumericRealValue". Cualquier valor real negativo se codificará utilizando la segunda alternativa de "ModifiedXMLNumericRealValue". El valor real cero negativo se codifica utilizando la segunda alternativa.

17.9.5 "XMLDecimalMinusZeroRealValue" se define en 22.3.2 y sólo se utiliza tal como se especifica en dicha subcláusula.

NOTA – La instrucción de codificación **DECIMAL** definida en 22.3.2. facilita esta producción como representación alternativa del valor real abstracto cero positivo, pero requiere que el valor abstracto cero negativo se excluya del tipo al que se aplica.

18 Instrucción de codificación ANY-ATTRIBUTES

18.1 Generalidades

18.1.1 "AnyAttributesInstruction" se define como:

```
AnyAttributesInstruction ::=
  ANY-ATTRIBUTES
  TargetList
  NamespaceRestriction ?
```

```
NamespaceRestriction ::=
  FROM URIList
  | EXCEPT URIList
```

```
URIList ::=
  QuotedURIorAbsent
  | URIList QuotedURIorAbsent
```

```
QuotedURIorAbsent ::=
  QuotedURI
  | ABSENT
```

18.1.2 La producción "TargetList" se define en 14.2.

18.1.3 "QuotedURI" se define en 29.1.1.

18.1.4 Esta instrucción de codificación se asigna a un tipo ASN.1 que sea un tipo secuencia-de o conjunto de con un componente **UTF8String** cuyo valor proporcione cero, uno o más nombres y valores de atributos (uno en cada **UTF8String**), cada uno de los cuales está sujeto a la "NamespaceRestriction" que esté presente (véase 18.2).

NOTA – Aunque secuencia-de pueda utilizarse para la especificación de los atributos, esta utilización de secuencia-de no implica que el orden sea semánticamente significativo, pudiendo el proceso de codificación/decodificación generar un orden diferente de los componentes de secuencia-de.

18.1.5 El contenido de cada **UTF8String** se codifica como un "Attribute" del elemento XML envolvente. El nombre del componente secuencia-de o conjunto-de se ignora.

18.1.6 Las cláusulas **FROM** y **EXCEPT** (si existen) identifican listas de nombres de espacio de nombres, o la palabra clave especial **ABSENT**.

18.1.7 **FROM** restringe que los nombres de atributos sean nombres de espacio de nombres cualificados de uno de los espacios de nombres especificados. Si la "URIList" incluye **ABSENT**, también pueden utilizarse nombres no cualificados.

18.1.8 **EXCEPT** admite nombres de espacios de nombres cualificados de cualquier espacio de nombres excepto los enumerados. También admite nombres no cualificados salvo que **ABSENT** esté presente en la "URIList".

18.2 Restricciones

18.2.1 Un tipo ASN.1 no incluirá esta instrucción de codificación final salvo que sea un tipo conjunto-de o secuencia-de con un componente que sea un tipo **UTF8String**.

18.2.2 Un tipo con esta instrucción de codificación final sólo se utilizará como componente de un tipo envolvente secuencia o conjunto, y el componente no se marcará como **OPTIONAL** o **DEFAULT**. En el tipo envolvente sólo incluirá uno de dichos componentes.

18.2.3 A un tipo conjunto-de o secuencia-de con esta instrucción de codificación final se le aplicará la restricción impuesta por el formato y el contenido especificados en 18.2.6 a 18.2.11 en cada ocurrencia del **UTF8String**, mediante la referencia a esta cláusula 18 o de cualquier otra forma.

NOTA – Se recomienda que la limitación relativa al tipo secuencia-de o conjunto-de se exprese como:

```
(CONSTRAINED BY
  { /* Each UTF8String shall conform to the "AnyAttributeFormat" specified in
    ITU-T Rec. X.693 | ISO/IEC 8825-4, clause 18. */ })
```

18.2.4 No podrá existir una instrucción de codificación final **UNTAGGED** en ningún tipo que incluya esta instrucción de codificación final ni en el tipo envolvente.

18.2.5 Cada "URIList" incluirá como máximo una ocurrencia de **ABSENT** y no podrá incluir dos "QuotedURI" idénticos.

18.2.6 El formato de cada **UTF8String** será conforme con la producción "AnyAttributeFormat":

AnyAttributeFormat ::=
URI ?
NCName & "=" & xmlcstring

18.2.7 Véanse 29.1.4 y 29.1.7 para la definición de la producción "URI" y para la definición de la producción "NCName" respectivamente. El elemento de léxico "xmlcstring" se define en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.15.

18.2.8 Si existe un "NamespaceRestriction" de **FROM**, el "URI" en el "AnyAttributeFormat" será el "URI" presente en un "QuotedURI" de la "URIList", y sólo podrá faltar si hay una palabra clave **ABSENT** en la "URIList".

18.2.9 Si existe una "NamespaceRestriction" de **EXCEPT**, el "URI" en el "AnyAttributeFormat" no será el "URI" presente en un "QuotedURI" de la "URIList" y no podrá faltar si hay una palabra clave **ABSENT** en la "URIList".

18.2.10 El "xmlcstring" será un valor de atributo XML sintácticamente correcto (definido en W3C XML, cláusula 3) precedido y seguido por un único carácter APOSTROPHE (39) (apóstrofe) o por un único carácter QUOTATION MARK (34) (comillas).

18.2.11 La aplicación de esta instrucción de codificación y de la instrucción de codificación **ATTRIBUTE** a diferentes componentes del tipo envolvente no violará lo indicado en 20.3.11.

18.2.12 Esta instrucción de codificación no se asignará salvo que exista una instrucción de codificación **GLOBAL-DEFAULT MODIFIED-ENCODINGS** en la sección de control de codificación.

18.2.13 Un tipo con esta instrucción de codificación final tampoco tendrá ninguna de las instrucciones de codificación finales **LIST**, **PI-OR-COMMENT** o **UNTAGGED**.

NOTA – Las instrucciones de codificación finales siguientes no pueden coexistir con esta instrucción de codificación final porque está prohibida su aplicación a dicho tipo: **ANY-ELEMENT**, **ATTRIBUTE**, **BASE64**, **DECIMAL**, **DEFAULT-FOR-EMPTY**, **EMBED-VALUES**, **TEXT**, **USE-NIL**, **USE-NUMBER**, **USE-ORDER**, **USE-QNAME**, **USE-TYPE**, **USE-UNION**, **WHITESPACE**.

18.2.14 "TargetList" no incluirá información de cualificación

18.3 Efecto sobre las codificaciones

18.3.1 Si el tipo está codificado como tipo de nivel superior, se ignorará esta instrucción de codificación.

18.3.2 El "ExtendedXMLNamedValue" de este componente no estará incluido en el "ExtendedXMLSequenceValue" o "ExtendedXMLSetValue" del tipo envolvente secuencia o conjunto. En su lugar, el valor del tipo envolvente se codificará utilizando el valor de cada **UTF8String** como un "Attribute" (véase la cláusula 20) del elemento envolvente, tal como se especifica más adelante.

18.3.3 El codificador:

- a) tratará cada "URI" que exista en un **UTF8String** requiriendo que el siguiente "NCName" (el nombre del atributo) sea un espacio de nombres cualificado con el espacio de nombres que especifica el "URI", tratará la ausencia de "URI" en un **UTF8String** especificando que el siguiente "NCName" no sea un espacio de nombres cualificado y suprimirá el "URI" del **UTF8String**; y
- b) insertará en la codificación las declaraciones de espacio de nombres que sean necesarias con ámbitos que incluyan los atributos insertados, a fin de asegurar que pueda conseguirse la necesaria cualificación de espacio de nombres de los "NCName" identificados en a) anterior; y
- c) insertará cada **UTF8String** (después de eliminar el "URI") como un atributo en el elemento envolvente, insertando los prefijos de espacios de nombres que sean necesarios antes de cada "NCName" a fin de asegurar que se satisfacen los requisitos anteriormente señalados en a).

18.3.4 El orden de todos los atributos del elemento envolvente (resultado de la presencia de uno o más componentes del tipo envolvente con una instrucción de codificación final **ATTRIBUTE** o **ANY-ATTRIBUTES**) es una opción del codificador.

18.3.5 Un decodificador EXTENDED-XER generará un **UTF8String** con el formato indicado en 18.2.6 para cada atributo del elemento envolvente que no sea del espacio de nombres de control y cuyo nombre no sea el del identificador (posiblemente modificado por alguna instrucción de codificación final **NAME** o **NAMESPACE**) de otro componente del tipo envolvente que tenga una instrucción de codificación final **ATTRIBUTE**.

19 Instrucción de codificación ANY-ELEMENT

19.1 Generalidades

19.1.1 "AnyElementInstruction" se define como:

```
AnyElementInstruction ::=
  ANY-ELEMENT
  TargetList
  NamespaceRestriction ?
```

19.1.2 La producción "TargetList" se define en 14.2.

19.1.3 "NamespaceRestriction" se define en 18.1.

19.1.4 Esta instrucción de codificación permite que un tipo ASN.1 que sea un **UTF8String** proporcione la especificación de un único elemento XML.

NOTA – El contenido y los atributos del elemento XML no están restringidos. Puede tener atributos o elementos hijos, cuyos nombres estén o no cualificados y no estar afectados por ninguna "NamespaceRestriction".

19.1.5 Si existe una "NamespaceRestriction", el nombre del elemento debe cumplir la "NamespaceRestriction" (véanse las cláusulas 18.1.6 a 18.1.8); en caso contrario no existe restricción alguna.

19.1.6 La **UTF8String** con dicha instrucción de codificación final puede ser el tipo raíz de la codificación, o puede ser un componente de un tipo elección, secuencia, conjunto, secuencia-de o conjunto-de. Si es un tipo de nivel superior, se ignora el nombre de referencia del tipo. Si es un componente, se ignora el nombre del componente.

19.2 Restricciones

19.2.1 Un tipo ASN.1 no tendrá esta instrucción de codificación final, salvo que sea un tipo **UTF8String**. El componente debe tener la restricción impuesta por el formato y el contenido especificados en 19.2.4 a 19.2.9, mediante la referencia a esta cláusula 19 o de cualquier otra forma.

NOTA – Se recomienda que la limitación de **UTF8String** se exprese de la forma siguiente:

```
(CONSTRAINED BY
  {/* Shall conform to the "AnyElementFormat" specified in
     ITU-T Rec. X.693 | ISO/IEC 8825-4, clause 19. */})
```

19.2.2 El tipo no tendrá una instrucción de codificación final **UNTAGGED**.

19.2.3 Cada "URIList" incluirá como máximo una ocurrencia de **ABSENT** y no podrá incluir dos "QuotedURI" idénticos.

19.2.4 El formato de los valores abstractos de **UTF8String** será conforme con la producción "AnyElementFormat":

```
AnyElementFormat ::=
  xmlcstring
```

19.2.5 La "xmlcstring" será un elemento XML sintácticamente correcto definido en espacios de nombres W3C XML 1.0 y W3C XML.

19.2.6 Sólo utilizará prefijos de espacios de nombres que sean declarados en declaraciones de espacios de nombres incluidos en la "xmlcstring". Si existen nombres cualificados sin prefijo, existirá la correspondiente declaración de espacio de nombres por defecto.

19.2.7 El valor de la **UTF8String** no hará que se incumpla lo indicado en 10.2.11.

19.2.8 Si existe una "NamespaceRestriction" de **FROM**, el nombre del elemento (más externo) en el "AnyElementFormat" será el "URI" de un "QuotedURI" en la "URIList", y sólo podrá faltar si la "URIList" contiene la palabra clave **ABSENT**.

19.2.9 Si existe una "NamespaceRestriction" de **EXCEPT**, el nombre del elemento (más externo) en el "AnyElementFormat" no será el "URI" de un "QuotedURI" en la "URIList", y no podrá faltar si la "URIList" contiene la palabra clave **ABSENT**.

19.2.10 Esta instrucción de codificación no se asignará salvo que exista una instrucción de codificación **GLOBAL-DEFAULTS MODIFIED-ENCODINGS** en la sección de control de codificación.

19.2.11 Un tipo con esta instrucción de codificación final tampoco contendrá ninguna de las instrucciones de codificación finales **ATTRIBUTE**, **BASE64**, **DEFAULT-FOR-EMPTY**, **PI-OR-COMMENT**, **UNTAGGED** o **WHITESPACE**.

NOTA – Las instrucciones de codificación finales siguientes no pueden coexistir con esta instrucción de codificación final porque está prohibida su aplicación a dicho tipo: **ANY-ATTRIBUTES**, **DECIMAL**, **EMBED-VALUES**, **LIST**, **TEXT**, **USE-NIL**, **USE-NUMBER**, **USE-ORDER**, **USE-QNAME**, **USE-TYPE**, **USE-UNION**.

19.2.12 "TargetList" no incluirá información de cualificación.

19.3 Efecto sobre las codificaciones

19.3.1 Un codificador **EXTENDED-XER** incluirá en la codificación el valor abstracto de **UTF8String** como un elemento XML, en lugar del elemento XML que, en cualquier otro caso, sería generado para este componente (ignorando el identificador del componente) o para el tipo raíz. El elemento incluido será idéntico al valor abstracto de **UTF8String**, excepto en lo que se especifica en 19.3.2.

19.3.2 Las declaraciones de espacio de nombres que existan en el primer rótulo de inicio (o rótulo de elemento vacío) del elemento y que sean idénticas a las declaraciones de espacio de nombres que estén en el ámbito de interés en el punto de inserción, pueden ser (aunque no necesariamente deben serlo) eliminadas como opción del codificador.

NOTA – No se permite el cambio, movimiento o supresión de otras declaraciones de espacios de nombre en la **UTF8String**, ya que dichas actuaciones pueden afectar al espacio de nombre y cualificación de QNames XML presentes en los valores de atributos o del contenido de nombres, y porque generalmente no es posible que un codificador determine si dichos valores de atributos o contenido son o no QNames.

19.3.3 Un decodificador **EXTENDED-XER** generará el formato identificado en 19.2.4 a partir del documento XML entrante como el valor abstracto de la **UTF8String**.

19.3.4 El decodificador incluirá, en el primer rótulo de inicio (o rótulo de elemento vacío) del valor abstracto de la **UTF8String**, atributos de declaración de espacio de nombres para todas las declaraciones de espacio de nombre que pertenezcan al ámbito del elemento que se decodifica, pero que no existan en el rótulo de inicio de dicho elemento.

20 Instrucción de codificación **ATTRIBUTE**

20.1 Generalidades

20.1.1 "AttributeInstruction" se define como:

AttributeInstruction ::=
ATTRIBUTE
TargetList

20.1.2 La producción "TargetList" se define en 14.2.

20.1.3 Esta instrucción de codificación especifica que un tipo ASN.1 codificable mediante caracteres debe codificarse como un atributo XML.

NOTA – Un caso particular (pero importante) de tipo codificable mediante caracteres es un tipo elección (cuyas alternativas son todas tipos codificables mediante caracteres) que tenga una instrucción de codificación final **USE-UNION**.

20.2 Restricciones

20.2.1 Un tipo ASN.1 no incluirá esta instrucción de codificación final salvo que, para conseguirlo, tenga al menos una codificación "ExtendedXMLValue" (teniendo en cuenta las opciones del codificador) para cada uno de sus valores abstractos, que no contenga ningún rótulo XML y no se base en la utilización de "xmlhstring" (si el tipo es un tipo abierto o tipo cadena de octetos) o de "xmlbstring" (si es un tipo cadena de bits) o en una instrucción de codificación final **UNTAGGED**, **ATTRIBUTE** o **ANY-ATTRIBUTES** aplicada a sus componentes (si es un tipo secuencia o conjunto).

NOTA 1 – Esto implica que un tipo cadena de caracteres restringida con una instrucción de codificación final **ATTRIBUTE** debe estar limitado de forma que no contenga ninguno de los caracteres de control enumerados en la Rec. UIT-T X.680 | ISO/CEI 8824-1, cuadro 3 (secuencias de escape para caracteres de control en una "xmlcstring"), o debe tener una instrucción de codificación final **BASE64**.

NOTA 2 – Esto no incluye tipos abiertos, o tipos cadena de octetos ni cadena de bits con **CONTAINING** sin **ENCODED BY** porque sus "ExtendedXMLValue" pueden incluir rótulos salvo que estén codificados como una "xmlhstring".

NOTA 3 – Se admite que algunas herramientas ASN.1 pueden no ser capaces de verificar estadísticamente si se cumplen las restricciones anteriores para todos los valores abstractos, pero los codificadores conformes no pueden generar codificaciones en las que el "ExtendedXMLValue" viole esta restricción (véase 20.3.14).

20.2.2 Un tipo con esta instrucción de codificación final sólo se utilizará como componente de un tipo secuencia o conjunto.

NOTA – El componente puede ser **OPTIONAL** o **DEFAULT**.

20.2.3 No podrá existir una instrucción de codificación final **UNTAGGED** en ningún tipo que incluya esta instrucción de codificación final ni en el tipo envolvente que la contenga como componente.

20.2.4 Si las instrucciones de codificación finales sobre otras componentes del tipo envolvente incluyen esta instrucción de codificación o la instrucción de codificación **ANY-ATTRIBUTES**, no se incumplirá lo indicado en 20.3.11.

20.2.5 Un tipo con esta instrucción de codificación final tampoco tendrá ninguna de las instrucciones de codificación finales **ANY-ELEMENT**, **DEFAULT-FOR-EMPTY**, **PI-OR-COMMENT** o **UNTAGGED**.

NOTA – Las instrucciones de codificación finales siguientes no pueden coexistir con esta instrucción de codificación final porque está prohibida su aplicación a dicho tipo: **ANY-ATTRIBUTES**, **EMBED-VALUES**, **USE-NIL**, **USE-ORDER**, **USE-TYPE**.

20.2.6 "TargetList" no incluirá información de cualificación.

20.3 Efecto sobre las codificaciones

20.3.1 Si el tipo está codificado como un tipo de nivel superior, se ignorará esta instrucción de codificación.

20.3.2 El "ExtendedXMLNamedValue" de este componente no estará incluido en el "ExtendedXMLSequenceValue" o en el "ExtendedXMLSetValue" del tipo envolvente secuencia o conjunto. En lugar de ello, el valor del componente (si existe) se codificará como un "Attribute" (véanse las cláusulas 20.3.3 a 20.3.15) del elemento envolvente.

20.3.3 La producción "Attribute" se define como:

Attribute ::=

AttributeName
"_"

QuotedValue

AttributeName ::=

IdentifierOrModifiedIdentifier
| **ControlAttributeName**

QuotedValue ::=

DoubleQuotedValue
| **SingleQuotedValue**

DoubleQuotedValue ::=

"_" & **CharacterEncodableValue** & ""

SingleQuotedValue ::=

"'" & **CharacterEncodableValue** & "'"

ControlAttributeName ::= QualifiedName

CharacterEncodableValue ::= ExtendedXMLValue

20.3.4 La producción "IdentifierOrModifiedIdentifier" se define en 17.5.1, y su utilización en el contexto de esta instrucción de codificación se define en 17.6.3.

20.3.5 La producción "ControlAttributeName" no es utilizada directamente por esta cláusula. Todos los "QualifiedName" de esta producción pertenecen al espacio de nombres de control (véase 16.9). Dichos atributos solamente se generan de conformidad con las cláusulas 33 y 37, pero es necesario que los decodificadores acepten atributos de control imprevistos (véase 10.2.10).

20.3.6 "QualifiedName" se define en 29.3.2.

20.3.7 "ExtendedXMLValue" se define en 17.4.

20.3.8 El "AttributeName" será el "identifier" del componente que tiene esta instrucción de codificación final o, si existen instrucciones de codificación finales **NAME** o **NAMESPACE**, será el "QualifiedOrUnqualifiedName" que determinan dichas instrucciones de codificación, tal como se especifica en las cláusulas 28 y 29.

20.3.9 El "CharacterEncodableValue" presente en el "QuotedValue" del atributo (véase 20.3.3) será el "ExtendedXMLValue" de este tipo, posiblemente modificado tal como se especifica en las cláusulas 20.3.12 a 20.3.15.

20.3.10 El orden en el que aparecen los "Attribute" en una "AttributeList" es una opción del codificador, en función de si éstos son generados por esta instrucción de codificación o por la instrucción de codificación **ANY-ATTRIBUTES**.

NOTA – En una codificación EXTENDED-XER, no puede aplicarse semántica alguna al orden de los atributos. Esta es una restricción de W3C XML 1.0, 3.1.

20.3.11 Cuando una "AttributeList" en un ejemplar de una codificación contiene varios atributos, para cualesquiera dos "Attribute" de la lista se cumple lo siguiente:

- a) si los "AttributeName" de los dos atributos son nombres no cualificados, serán diferentes;
- b) si los "AttributeName" de los dos atributos son nombres de espacio de nombres cualificados, tendrán nombres de espacio de nombres diferentes o serán nombres diferentes en el mismo espacio de nombres.

La utilización de las instrucciones de codificación es ilegal si la aplicación de instrucciones de codificación finales para cualquier valor abstracto del tipo de nivel superior codificado viola esta condición.

20.3.12 Si el "QuotedValue" es un "DoubleQuotedValue", y el "ExtendedXMLValue" del "CharacterEncodableValue" contiene un carácter QUOTATION MARK (34), dicho carácter será sustituido por caracteres:

";

o, como opción del codificador, por una secuencia de escape de la forma **&#n;** o **&#xn;** especificada en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.15.8.

20.3.13 Si el "QuotedValue" es un "SingleQuotedValue" y el "ExtendedXMLValue" del "CharacterEncodableValue" contiene un carácter APOSTROPHE (39), dicho carácter será sustituido por caracteres:

'

o, como opción del codificador, por una secuencia de escape de la forma **&#n;** o **&#xn;**, especificada en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.15.8.

20.3.14 El "ExtendedXMLValue" del "CharacterEncodableValue" será una de las codificaciones del tipo codificable mediante caracteres que no contenga rótulos XML.

20.3.15 Si el "ExtendedXMLValue" contiene caracteres HORIZONTAL TABULATION (9) (tabulación horizontal), LINE FEED (10) (salto de renglón), o CARRIAGE RETURN (13) (retroceso de carro), dichos caracteres serán sustituidos en "ExtendedXMLValue" por secuencias de escape de la forma **&#n;** o **&#xn;**, especificadas en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.15.8.

21 Instrucción de codificación BASE64

21.1 Generalidades

21.1.1 "Base64Instruction" se define como:

Base64Instruction ::=
BASE64
TargetList

21.1.2 La producción "TargetList" se define en 14.2.

21.1.3 Esta instrucción de codificación puede asignarse a una **OCTET STRING**, a un tipo abierto o a cualquier tipo de cadena de caracteres restringida.

21.1.4 La aplicación de esta instrucción de codificación final a un tipo cadena de octetos o a un tipo abierto suprime la opción de codificación hexadecimal, pero permite la opción de codificación Base64 (tal como se especifica en la RFC-2045 del IETF, 6.8). La aplicación de esta instrucción de codificación final a un tipo cadena de caracteres restringida requiere que la codificación del valor del tipo cadena de caracteres restringida se haga mediante codificación Base64.

21.2 Restricciones

21.2.1 Si las instrucciones de codificación finales de un tipo ASN.1 contienen una instrucción de codificación **BASE64** el tipo será:

- a) una **OCTET STRING**; o
- b) un tipo abierto; o
- c) un tipo cadena de caracteres restringida.

21.2.2 Un tipo con esta instrucción de codificación final no tendrá ninguna de las instrucciones de codificación finales **ANY-ELEMENT** o **WHITESPACE**.

NOTA – Las instrucciones de codificación finales siguientes no pueden coexistir con esta instrucción de codificación final porque está prohibida su aplicación a dicho tipo: **ANY-ATTRIBUTES**, **DECIMAL**, **EMBED-VALUES**, **LIST**, **TEXT**, **USE-NIL**, **USE-NUMBER**, **USE-ORDER**, **USE-QNAME**, **USE-TYPE**, **USE-UNION**.

21.2.3 "TargetList" no incluirá información de cualificación.

21.3 Efecto sobre las codificaciones

21.3.1 Esta instrucción de codificación afecta solamente al "ExtendedXMLValue" del tipo al que se aplica. Requiere la utilización de la primera o la segunda alternativa de "ExtendedXMLOctetStringValue" y "ExtendedXMLOpenTypeFieldVal" (como opción del codificador), y prohíbe la tercera alternativa (véase 17.4). Requiere la utilización de la segunda alternativa de "ExtendedXMLRestrictedCharacterStringValue" (véase 17.4).

21.3.2 "Base64XMLOctetStringValue" se define como:

Base64XMLOctetStringValue ::=
XMLBase64String

"XMLBase64String" se define en 21.3.6.

21.3.3 Se aplica la Rec. UIT-T X.680 | ISO/CEI 8824-1, 22.4.

21.3.4 "Base64XMLOpenTypeFieldVal" se define como:

Base64XMLOpenTypeFieldVal ::=
XMLBase64String

21.3.5 "Base64XMLRestrictedCharacterStringValue" se define como:

Base64XMLRestrictedCharacterStringValue ::=
XMLBase64String

21.3.6 "XMLBase64String" se define como:

XMLBase64String ::=
XMLRestrictedCharacterStringValue

El "XMLRestrictedCharacterStringValue" será la codificación de transferencia de contenido ("*Content-Transfer-Encoding*") especificada en la RFC 2045 del IETF, 6.8, excepto en que no se aplica el límite de 76 caracteres, y se permite que existan "espacios blancos con escape" (véase 8.1.5) en cualquier posición de "XMLBase64String".

NOTA – La RFC 2045 del IETF requiere que existan saltos de línea que dividan las líneas de codificación para que éstas tengan 76 caracteres como máximo, cosa que no exige la codificación EXTENDED-XER. También permite que se inserten "espacios blancos" en cualquier posición de la codificación base64.

21.3.7 Si se aplica a un tipo cadena de caracteres restringida, cada carácter de la cadena de caracteres se codificará con UTF-8 (véase ISO 10646, anexo D). Los octetos resultantes de toda la cadena de caracteres se codificarán en caracteres, tal como se especifica en la RFC 2045 del IETF, apartado 6.8, y los caracteres resultantes conformarán el "ExtendedXMLValue".

22 Instrucción de codificación DECIMAL

22.1 Generalidades

22.1.1 "DecimalInstruction" se define como:

```
DecimalInstruction ::=
    DECIMAL
    TargetList
```

22.1.2 La producción "TargetList" se define en 14.2.

22.1.3 El objetivo de esta instrucción de codificación es modificar la codificación de un tipo real de forma que se prohíba la notación exponencial, y un guión seguido de "0" denote el valor cero más en lugar del valor cero menos.

NOTA – El valor cero menos no puede representarse.

22.2 Restricciones

22.2.1 Esta instrucción de codificación sólo se asignará a un tipo real.

22.2.2 El tipo real al que se aplique esta instrucción de codificación estará restringido de forma que los valores cero menos, **MINUS-INFINITY**, **PLUS-INFINITY**, y **NOT-A-NUMBER** no están permitidos y la **base** es 10.

NOTA – Se recomienda que para ello se apliquen las restricciones siguientes:

```
(WITH COMPONENTES { . . . , base(10) })
(ALL EXCEPT (-0 | MINUS-INFINITY | PLUS-INFINITY | NOT-A-NUMBER))
```

22.2.3 Esta instrucción de codificación no se asignará salvo que exista una instrucción de codificación **GLOBAL-DEFAULTS MODIFIED-ENCODINGS** en la sección de control de codificación.

22.2.4 Un tipo con esta instrucción de codificación final puede permitir otras instrucciones de codificación finales para dicho tipo.

NOTA – Las instrucciones de codificación finales siguientes no pueden coexistir con esta instrucción de codificación final porque está prohibida su aplicación a dicho tipo: **ANY-ATTRIBUTES**, **ANY-ELEMENT**, **BASE64**, **EMBED-VALUES**, **LIST**, **TEXT**, **USE-NIL**, **USE-NUMBER**, **USE-ORDER**, **USE-QNAME**, **USE-TYPE**, **USE-UNION**, **WHITESPACE**.

22.2.5 "TargetList" no incluirá información de cualificación.

22.3 Efecto sobre las codificaciones

22.3.1 El "modifiedXMLRealNumber" (véase 17.9.3) no contendrá una "e" o una "E" seguida de un exponente.

NOTA – Todos los valores abstractos, incluidos los que son números reales muy grandes o muy pequeños, se codifican por tanto como una parte entera seguida, opcionalmente, por un punto decimal y una parte fraccionaria.

22.3.2 Es opcional del codificador que el valor real cero más se codifique en la forma "XMLDecimalMinusZeroRealValue" definida como sigue:

```
XMLDecimalMinusZeroRealValue ::=
    "-" & modifiedXMLRealNumber
```

donde "modifiedXMLRealNumber" está restringido según 22.3.1 y no contiene ningún dígito excepto el cero.

NOTA – No debe confundirse lo anterior con el valor real cero menos, ya que dicho valor queda suprimido por la restricción obligatoria aplicable al tipo real (véase 22.2.2).

23 Instrucción de codificación DEFAULT-FOR-EMPTY

23.1 Generalidades

23.1.1 "DefaultForEmptyInstruction" se define como:

```
DefaultForEmptyInstruction ::=
    DEFAULT-FOR-EMPTY
    TargetList
    AS Value
```

23.1.2 La producción "TargetList" se define en 14.2.

23.1.3 Esta instrucción de codificación especifica un valor abstracto que puede codificarse mediante una codificación EXTENDED-XER (opción del codificador), como la alternativa "vacío" del "ExtendedXMLValue" para un tipo (véase 17.4) codificado como el contenido único de un elemento XML.

NOTA – Este mecanismo, que actúa por defecto, soporta la presencia de un elemento XML sin contenido (que típicamente, aunque no necesariamente, se codifica como un rótulo de elemento vacío). Ello difiere de la utilización del **DEFAULT** ASN.1, que está relacionada con la ausencia del "ExtendedXMLNamedValue" de un componente de una secuencia o conjunto.

23.1.4 "TargetList" no utilizará la palabra clave **ALL** e identificará un único objetivo.

23.1.5 Existen cinco casos diferenciados, que se identifican a continuación, en los que puede utilizarse esta instrucción de codificación.

23.1.5.1 El primer caso ocurre cuando la asignación es directa a un tipo codificable mediante caracteres que no sea **UNTAGGED** (véase la cláusula 32). Si el elemento envolvente tiene un contenido vacío, éste representa el "Value" especificado del tipo codificable mediante caracteres (que es lo que determina "Value").

23.1.5.2 El segundo caso ocurre cuando se asigna a un tipo secuencia (**NOT UNTAGGED**, **NOT EMBED-VALUES** y **NOT USE-NIL**) con un componente **UNTAGGED** codificable mediante caracteres cuya codificación constituye el único contenido (para todos los valores abstractos de tipo secuencia) del elemento envolvente del tipo secuencia. Si el elemento envolvente del tipo secuencia tiene un contenido vacío, éste representa el "Value" especificado del componente codificable mediante caracteres (que es lo que determina "Value").

NOTA – El componente codificable mediante caracteres puede ser el único contenido porque es el único componente, o porque todos los demás componentes tienen una instrucción de codificación final **ATTRIBUTE** (véase la cláusula 20) o **ANY-ATTRIBUTES** (véase la cláusula 18).

23.1.5.3 El tercer caso ocurre cuando se asigna a un tipo secuencia (**NOT UNTAGGED** y **NOT USE-NIL**) con una instrucción de codificación final **EMBED-VALUES** (véase 25.3.1.4). Si el elemento envolvente del tipo secuencia tiene un contenido vacío, dicho contenido vacío representa un valor abstracto del tipo secuencia que en cualquier otro caso produciría un contenido que sería exclusivamente el "Value" especificado de un único **UTF8String** en la secuencia-de **EMBED-VALUES** (**UTF8String** determina "Value").

23.1.5.4 El cuarto caso ocurre cuando se asigna a un tipo secuencia (**NOT UNTAGGED**, **NOT EMBED-VALUES**) con una instrucción de codificación final **USE-NIL** (véase la cláusula 33) cuyo componente **OPTIONAL** es un tipo codificable mediante caracteres. Si el elemento envolvente del tipo secuencia tiene un atributo de identificación nulo con valor **true**, el **DEFAULT-FOR-EMPTY** no afecta al significado de la codificación. Si el elemento envolvente del tipo secuencia tiene un atributo de identificación nulo con valor **false** (o no tiene atributo de identificación nulo), y tiene un contenido vacío, dicho contenido vacío representa el "Value" especificado del componente **OPTIONAL** (que es lo que determina "Value").

23.1.5.5 El quinto caso ocurre cuando se asigna a un tipo secuencia (**NOT UNTAGGED**) una instrucción de codificación final **EMBED-VALUES** (véase 25.3.1.4) y una instrucción de codificación final **USE-NIL** (véase la cláusula 33), cuyo componente **OPTIONAL** es un tipo secuencia. Si el elemento envolvente del tipo secuencia tiene un atributo de identificación nulo con valor **true**, el **DEFAULT-FOR-EMPTY** no afecta al significado de la codificación. Si el elemento envolvente del tipo secuencia tiene un atributo de identificación nulo con valor **false** (o no tiene atributo de identificación nulo), y tiene un contenido vacío, éste representa un valor abstracto del tipo secuencia que en cualquier otro caso produciría un contenido que sería exclusivamente el "Value" especificado de un único **UTF8String** en la secuencia-de **EMBED-VALUES** (**UTF8String** determina "Value").

23.1.6 "Value" se define en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 16.7.

NOTA – Esto permite utilizar un valor de referencia definido en el módulo o importando al mismo. La referencia de valor puede definirse utilizando una notación de valor XML, pero dicha notación no puede utilizarse directamente en la "DefaultForEmptyInstruction".

23.2 Restricciones

23.2.1 Si las instrucciones de codificación finales de un tipo ASN.1 que no sea un tipo codificable mediante caracteres **NOT UNTAGGED** contienen una instrucción de codificación **DEFAULT-FOR-EMPTY**, dicho tipo no será un componente (de un **SEQUENCE** o **SET** ASN.1) con un valor **DEFAULT** ASN.1.

NOTA – Esta restricción no es estrictamente necesaria, pero se utiliza para evitar la confusión entre el ASN.1 normal y los mecanismos EXTENDED-XER que actúan por defecto.

23.2.2 Esta instrucción de codificación sólo se asignará a:

- a) un tipo codificable mediante caracteres sin una instrucción de codificación final **UNTAGGED**; o
- b) un tipo secuencia **NOT UNTAGGED**, sin una instrucción de codificación final **EMBED-VALUES**, o **USE-NIL**, uno de cuyos componentes es un tipo codificable mediante caracteres con una instrucción de codificación final **UNTAGGED** y en que los restantes componentes (si existen) tienen una instrucción de codificación final **ATTRIBUTE** o **ANY-ATTRIBUTES**; o
- c) un tipo secuencia **NOT UNTAGGED**, sin una instrucción de codificación final **USE-NIL**, pero con una instrucción de codificación final **EMBED-VALUES** (véase 25.3.1.4); o
- d) un tipo secuencia **NOT UNTAGGED**, sin una instrucción de codificación final **EMBED-VALUES**, pero con una instrucción de codificación final **USE-NIL**, cuyo componente **OPTIONAL** es un tipo codificable mediante caracteres; o
- e) un tipo secuencia **NOT UNTAGGED** con una instrucción de codificación final **EMBED-VALUES** y con una instrucción de codificación final **USE-NIL**, cuyo componente **OPTIONAL** es un tipo secuencia.

23.2.3 Si se aplica 23.2.2 a), y "empty" (vacío) es un "ExtendedXMLValue" válido para uno de los valores abstractos (por ejemplo, V) del tipo (posiblemente limitado), y V es distinto del "Value" de la "DefaultForEmptyInstruction", habrá al menos una codificación alternativa para V.

23.2.4 Si se aplica 23.2.2 b) o d), y "empty" (vacío) es un "ExtendedXMLValue" válido para uno de los valores abstractos (por ejemplo, V) del componente **UNTAGGED** [caso b)] o del componente **OPTIONAL**, [caso d)], y V es distinto del "Value" de la "DefaultForEmptyInstruction", habrá al menos una codificación alternativa para V.

NOTA – Es posible que algunas herramientas ASN.1 no puedan verificar estadísticamente si se cumplen las restricciones anteriores para todos los valores abstractos, pero los codificadores conformes no pueden generar codificaciones en las que el "ExtendedXMLValue" viole esta restricción.

23.2.5 Si se aplica 23.2.2 c), el tipo **SEQUENCE** estará limitado de forma que (sin **DEFAULT-FOR-EMPTY**) no existe un valor abstracto que produzca un contenido vacío para el elemento envolvente.

23.2.6 Si un tipo codificable mediante caracteres [caso 23.2.2 a)] que incluya esta instrucción de codificación final tiene un tipo envolvente secuencia-de o conjunto-de con una instrucción de codificación final **LIST**, o un tipo elección con una instrucción de codificación **USE-UNION**, se ignorará dicha instrucción de codificación final.

23.2.7 Esta instrucción de codificación no se asignará salvo que exista una instrucción de codificación **GLOBAL-DEFAULTS MODIFIED-ENCODINGS** en la sección de control de codificación.

23.2.8 Un tipo con esta instrucción de codificación final no tendrá ninguna de las instrucciones de codificación finales **ANY-ELEMENT**, **ATTRIBUTE** o **UNTAGGED**.

NOTA – Las instrucciones de codificación finales siguientes no pueden coexistir con esta instrucción de codificación final porque está prohibida su aplicación a dicho tipo: **ANY-ATTRIBUTES**, **USE-TYPE**.

23.2.9 "TargetList" no incluirá información de cualificación.

23.3 Efecto sobre las codificaciones

23.3.1 Esta instrucción de codificación sólo afecta al "ExtendedXMLValue" del tipo que determina cuál es el valor de "Value", (véase 23.1.5).

23.3.2 Es opcional del codificador que la codificación "ExtendedXMLValue" del valor abstracto especificado por "Value" sea

- a) la codificación "ExtendedXMLValue" de dicho valor que se produciría si no existiera **DEFAULT-FOR-EMPTY** (codificación normal); o
- b) "empty" ("vacío").

NOTA – Los decodificadores deben aceptar la codificación normal y la codificación "empty", como forma de denominar el valor vacío por defecto.

23.3.3 Si se aplica 23.2.2 a), y "empty" es un "ExtendedXMLValue" válido para uno de los valores abstractos (por ejemplo, V) del tipo, y V es distinto del "Value" especificado en la "DefaultForEmptyInstruction", se podrá utilizar cualquiera de las codificaciones alternativas para V en lugar de "empty" (opción del codificador).

23.3.4 Si se aplica 23.2.2 b) o d), y "empty" es un "ExtendedXMLValue" válido para uno de los valores abstractos (por ejemplo, V) del componente **UNTAGGED** [caso b)] o el componente **OPTIONAL** [caso d)], y V es distinto del "Value" especificado en la "DefaultForEmptyInstruction", se podrá utilizar cualquiera de las codificaciones alternativas para V en lugar de "empty" (opción del codificador).

23.3.5 Si se aplica 23.2.2 c), el efecto de esta instrucción de codificación se especifica en 25.3.1.4 y 25.3.1.5.

23.3.6 Si se aplica 23.2.2 e), el efecto de esta instrucción de codificación se especifica en 25.3.1.6.

24 Instrucción de codificación **ELEMENT**

24.1 Generalidades

24.1.1 "ElementInstruction" se define como:

ElementInstruction ::=
ELEMENT
TargetList

24.1.2 La producción "TargetList" se define en 14.2.

24.1.3 Esta instrucción de codificación es sinónimo de **NOT UNTAGGED**, y no implica una semántica diferente de **NOT UNTAGGED**.

24.2 Restricciones

24.2.1 "TargetList" no incluirá información de cualificación.

24.2.2 Para no confundir al lector, esta instrucción de codificación no debe utilizarse como una instrucción de codificación con prefijo en combinación de cualquiera de las instrucciones de codificación con prefijo **ANY-ATTRIBUTES**, **ANY-ELEMENT** o **ATTRIBUTE**.

24.3 Efecto sobre las codificaciones

Esta instrucción de codificación niega una instrucción de codificación **UNTAGGED**, y no afecta de ninguna forma a las demás codificaciones.

25 Instrucción de codificación **EMBED-VALUES**

25.1 Generalidades

25.1.1 "EmbedValuesInstruction" se define como:

EmbedValuesInstruction ::=
EMBED-VALUES
TargetList

25.1.2 La producción "TargetList" se define en 14.2.

25.1.3 Esta instrucción de codificación permite que el primer componente de un tipo secuencia (**NOT UNTAGGED**) proporcione cadenas de caracteres que se insertan antes del primer elemento XML, después del último elemento XML, y entre los elementos XML que forman la codificación "ExtendedXMLValue" del tipo secuencia.

25.1.4 Si también existe una instrucción de codificación final **USE-NIL**, y el componente **OPTIONAL** que soporta **USE-NIL** no existe en un valor abstracto en concreto, no habrá elementos XML para componentes del tipo secuencia, y no se proporcionan cadenas de caracteres para dicho valor abstracto. En caso contrario, el número de cadenas de caracteres para todos los valores abstractos debe ser uno más que el número de elementos de la codificación del tipo secuencia. Algunas o todas las cadenas de caracteres pueden estar vacías.

25.2 Restricciones

25.2.1 Un tipo ASN.1 no tendrá esta instrucción de codificación final salvo que sea un tipo secuencia. El primer componente de la secuencia será una **SEQUENCE OF UTF8String** y no estará marcada como **OPTIONAL** o **DEFAULT**.

25.2.2 No existirá una instrucción de codificación final **UNTAGGED** (véase la cláusula 32) en el tipo secuencia-de ni en el componente de secuencia-de.

25.2.3 No existirá una instrucción de codificación final **UNTAGGED** en ningún componente del tipo secuencia que sea un tipo codificable mediante caracteres.

25.2.4 Si el tipo secuencia también tiene una instrucción de codificación final **USE-NIL**, el componente **OPTIONAL** que soporta la instrucción de codificación **USE-NIL** no será un tipo codificable mediante caracteres (véase también 33.2.4).

25.2.5 Ninguno de los componentes de la secuencia estará marcado como **DEFAULT** salvo que tenga una instrucción de codificación final **ATTRIBUTE**. Si existen componentes de un tipo **SEQUENCE** o **SET** (a cualquier nivel) que mediante la utilización de **UNTAGGED** puedan producir elementos en el "ExtendedXMLValue" que sean elementos hijos directos del tipo secuencia, no se marcarán como **DEFAULT**.

25.2.6 El tipo secuencia estará restringido de forma que:

- a) si el tipo tiene también una instrucción de codificación final **USE-NIL** y no existe el componente **OPTIONAL** que soporte **USE-NIL**, el número de repeticiones del componente secuencia-de debe ser cero;
- b) en cualquier otro caso, el número de repeticiones del componente secuencia-de de cada valor abstracto debe ser igual a uno más el número de elementos XML en el "ExtendedXMLValue" del tipo secuencia, determinado después de la aplicación de todas las instrucciones de codificación finales a los demás componentes de la secuencia, e ignorando el primer componente.

NOTA – Se recomienda que la limitación sobre el tipo secuencia se exprese como:

(CONSTRAINED BY

{/ Shall conform to ITU-T Rec. X.693 | ISO/IEC 8825-4, clause 25 */}*)

25.2.7 Esta instrucción de codificación no se asignará salvo que exista una instrucción de codificación **GLOBAL-DEFAULTS MODIFIED-ENCODINGS** en la sección de control de codificación.

25.2.8 Un tipo con esta instrucción de codificación final no tendrá una instrucción de codificación final **UNTAGGED**.

NOTA – Las instrucciones de codificación finales siguientes no pueden coexistir con esta instrucción de codificación final porque está prohibida su aplicación a dicho tipo: **ANY-ATTRIBUTES**, **ANY-ELEMENT**, **ATTRIBUTE**, **BASE64**, **DECIMAL**, **LIST**, **TEXT**, **USE-NUMBER**, **USE-QNAME**, **USE-TYPE**, **USE-UNION**, **WHITESPACE**.

25.2.9 "TargetList" no incluirá información de cualificación.

25.3 Efecto sobre las codificaciones

25.3.1 Un codificador producirá, en primer lugar, una codificación parcial "ExtendedXMLValue" del tipo secuencia envolvente, ignorando el primer componente. Modificará esta codificación tal como se especifica en las subcláusulas siguientes.

NOTA – Los valores **UTF8String** que se inserten pueden estar "vacíos".

25.3.1.1 El primer valor **UTF8String** de la secuencia-de se insertará (sujeto a lo indicado en 25.3.1.6) al comienzo de la codificación parcial, antes del rótulo de inicio del primer elemento XML (si existe).

25.3.1.2 Cualquier valor **UTF8String** ulterior (si lo hay), se insertará entre el rótulo de terminación de un elemento XML y el rótulo de inicio del siguiente elemento XML, procediendo para ello desde el primer elemento al último elemento.

NOTA – Lo anterior implica que no se inserta ningún valor **UTF8String** en estos componentes, aunque tengan elementos hijos.

25.3.1.3 El último valor **UTF8String** (si existe uno) se insertará al final de la codificación parcial, después del rótulo de terminación del último elemento XML.

25.3.1.4 Si en la codificación parcial no existen elementos XML, y existe una instrucción de codificación final **DEFAULT-FOR-EMPTY** (véase la cláusula 23) en el tipo secuencia, siendo el valor del primer (y solo del primer) **UTF8String** del secuencia-de idéntico al valor "Value" que se especifica en la instrucción de codificación **DEFAULT-FOR-EMPTY**, un codificador puede codificar opcionalmente **UTF8String** como una cadena vacía (no obstante, véase 25.3.1.6).

25.3.1.5 Si no existen elementos XML en la codificación parcial, y existe una instrucción de codificación final **DEFAULT-FOR-EMPTY** en el tipo secuencia y la codificación está vacía, un decodificador lo interpretará como una codificación para el "Value" especificado en la instrucción de codificación **DEFAULT-FOR-EMPTY** y asignará este valor abstracto al primer (y solo al primer) **UTF8String** presente en el secuencia-de (no obstante, véase 25.3.1.6).

NOTA – Esto significa que un valor sin elementos XML y con un valor único **UTF8String** vacío no puede ser codificado. El tipo secuencia debe estar limitado a fin de prohibir dichos valores (véase 23.2.5).

25.3.1.6 Si el tipo también tiene una instrucción de codificación final **USE-NIL** y no existe el componente **OPTIONAL**, la instrucción de codificación **EMBED-VALUES** no tiene efecto alguno. Si el tipo también tiene una instrucción de codificación final **USE-NIL** y el componente **OPTIONAL** existe, se aplica lo señalado en la cláusula 25.3.1.4. Si un decodificador determina que existe el componente **OPTIONAL**, por la ausencia de un atributo de identificación nulo (o por su presencia con el valor falso), se aplica 25.3.1.5.

26 Instrucción de codificación **GLOBAL-DEFAULTS**

26.1 Generalidades

26.1.1 "GlobalDefaultsInstruction" se define como:

GlobalDefaultsInstruction ::=
GLOBAL-DEFAULTS TargetList DefaultSetting

DefaultSetting ::=
ControlNamespace
 | **MODIFIED-ENCODINGS**

ControlNamespace ::=
CONTROL-NAMESPACE
QuotedURI
Prefix ?

26.1.2 La producción "TargetList" se define en 14.2, y será "empty".

26.1.3 "QuotedURI" y "Prefix" se definen en 29.1.1.

26.1.4 La producción "ControlNamespace" especifica el nombre del espacio de nombres de control (el "URI" del "QuotedURI"), y un prefijo recomendado para dicho espacio de nombres. Si esta instrucción de codificación **GLOBAL-DEFAULTS** no existe, el espacio de nombres de control será el especificado en 16.9.

26.1.5 La utilización de **MODIFIED-ENCODINGS** produce "ExtendedXMLValues" que se modifican de conformidad con 10.2.7 y 10.2.8.

26.2 Restricciones

26.2.1 La instrucción de codificación **GLOBAL-DEFAULTS** sólo se asignará en una sección de control de codificación y no estará precedida de ninguna otra instrucción de codificación excepto por otras instrucciones de codificación **GLOBAL-DEFAULTS**.

26.2.2 Cada una de las alternativas de **GLOBAL-DEFAULTS** se utilizará como máximo una vez en cualquier sección de control de codificación.

26.2.3 El **GLOBAL-DEFAULTS MODIFIED-ENCODINGS**, si existe, será la primera instrucción de codificación de la sección de control de codificación XER de un módulo ASN.1.

26.3 Efecto en las codificaciones

26.3.1 La aplicación de **MODIFIED-ENCODINGS** requiere que las codificaciones se modifiquen tal como se especifica en 10.2.7 y 10.2.8.

26.3.2 El espacio de nombres de control utilizado para todo el documento XML será el espacio de nombres de control asignado al tipo ASN.1 cuya codificación conforma el elemento raíz de dicho documento XML.

27 Instrucción de codificación **LIST**

27.1 Generalidades

27.1.1 "ListInstruction" se define como:

ListInstruction ::=
LIST
TargetList

27.1.2 La producción "TargetList" se define en 14.2.

27.1.3 Esta instrucción de codificación requiere que el "ExtendedXMLSequenceOfValue" o "ExtendedXMLSetOfValue" de un tipo secuencia-de o conjunto-de (véase 17.7), sea el "ExtendedXMLListValue", produciendo una lista con separaciones mediante espacios con los valores del componente de secuencia-de o conjunto-de.

NOTA – Una asignación común de esta instrucción de codificación es a una **SEQUENCE OF INTEGER**, a la que también se asigna una instrucción de codificación **ATTRIBUTE** (véase la cláusula 20).

27.2 Restricciones

27.2.1 El tipo al que se asigna esta instrucción de codificación será un tipo secuencia-de o conjunto-de.

27.2.2 El componente del tipo secuencia-de o conjunto-de será:

- a) un tipo codificable mediante caracteres; y
- b) tal que, para todos sus valores abstractos exista al menos una codificación "ExtendedXMLValue" (teniendo en cuenta todas las opciones del codificador) que no esté "vacía" y que no contenga "espacios blancos con escapes" (véase 8.1.5).

NOTA 1 – Las restricciones anteriores implican que el componente no puede ser un tipo secuencia-de o conjunto-de con una instrucción de codificación **LIST**, o contener un tipo secuencia-de o conjunto-de anidado con una instrucción de codificación **LIST** a cualquier nivel.

NOTA 2 – Las restricciones anteriores se satisfacen si el tipo del componente de la secuencia-de o conjunto-de es del tipo entero, real, identificador de objeto, identificador de objeto relativo, o de los tipos útiles **GeneralizedTime** y **UTCTime**. También se satisfacen si es un tipo cadena de caracteres restringida de tal forma que siempre tenga al menos un carácter en la cadena de caracteres y ninguno de sus valores contenga un carácter "espacio blanco".

NOTA 3 – Es posible que algunas herramientas ASN.1 no puedan verificar estadísticamente el cumplimiento de las reglas anteriores, pero no obstante, un codificador conforme no generará codificaciones que violen lo indicado en b).

27.2.3 Un tipo con esta instrucción de codificación final no tendrá ninguna instrucción de codificación final **ANY-ATTRIBUTES**.

NOTA – Las instrucciones de codificación finales siguientes no pueden coexistir con esta instrucción de codificación final porque está prohibida su aplicación a dicho tipo: **ANY-ELEMENT**, **BASE64**, **DECIMAL**, **EMBED-VALUES**, **TEXT**, **USE-NIL**, **USE-NUMBER**, **USE-ORDER**, **USE-QNAME**, **USE-TYPE**, **USE-UNION**, **WHITESPACE**.

27.2.4 "TargetList" no incluirá información de cualificación.

27.3 Efecto en las codificaciones

27.3.1 Esta instrucción de codificación sólo afecta a la codificación del tipo al que se aplica.

27.3.2 La producción "ExtendedXMLSequenceOfValue" o "ExtendedXMLSetOfValue" (véase 17.7) será la "ExtendedXMLListValue" alternativa. "ExtendedXMLListValue" se define como:

ExtendedXMLListValue ::=
empty
| CharacterEncodableValueExtendedXMLListValue

27.3.3 Entre cada pareja de "CharacterEncodableValue" (valor codificable mediante caracteres) adyacentes en el "ExtendedXMLListValue", habrá "espacios blancos con escape" (véase 8.1.5)

27.3.4 "CharacterEncodableValue" se define en 20.3.3. Cada "CharacterEncodableValue" codificará un valor de un componente de la secuencia-de o conjunto-de.

27.3.5 El orden en que los "CharacterEncodableValue" aparecen en "ExtendedXMLListValue" es el mismo en el que los "ExtendedXMLValue" correspondientes aparecerían en un "ExtendedXMLSequenceOfValue" o "ExtendedXMLSetOfValue" si no existiera una instrucción de codificación final **LIST**.

27.3.6 Los "CharacterEncodableValue" del "ExtendedXMLListValue" no tomarán el valor "vacío" ni contendrán "espacios blancos con escape" (véase 8.1.5).

NOTA – La subcláusula 27.2.2 b) garantiza que ello es posible, pero 27.3.4 puede restringir las opciones del codificador.

28 Instrucción de codificación **NAME**

28.1 Generalidades

28.1.1 "NameInstruction" se define como:

NameInstruction ::=

NAME

TargetList

AS

newNameOrKeyword

newNameOrKeyword ::=

newName

| Keyword

newName ::=

RestrictedCharacterStringValue

Keyword ::=

CAPITALIZED

| UNCAPITALIZED

| UPPERCASED

| LOWERCASED

28.1.2 La producción "TargetList" se define en 14.2.

28.1.3 Esta instrucción de codificación tiene cinco propósitos diferentes:

- a) cambiar el nombre del rótulo asociado, el nombre del atributo, o el valor de un posible atributo de identificación de tipo ("NewName" sin "QualifyingInformation" en la "TargetList") del objetivo; o
- b) cambiar el tipo de letra – mayúscula o minúscula – (o el tipo de la letra inicial) del nombre de rótulo asociado, del nombre de atributo, o del valor de un posible atributo de identificación de tipo ("Keyword" sin "QualifyingInformation" en la "TargetList") del objetivo u objetivos; o
- c) cambiar el nombre del elemento utilizado normalmente en un rótulo de elemento vacío (tal como se especifica en la Rec. UIT-T X.680 | ISO/CEI 8824-1) derivado de un identificador especificado utilizado en la definición de tipo ("NewName" con "QualifyingInformation" en la "TargetList" que no sea **ALL**) del objetivo; o
- d) cambiar el tipo de letra – mayúscula o minúscula – (o el tipo de la letra inicial) del nombre de elemento utilizado normalmente en un rótulo de elemento vacío derivado de un identificador especificado utilizado en la definición de tipo ("Keyword" -palabra clave- con "QualifyingInformation" en la "TargetList" que no sea **ALL**) del objetivo u objetivos; o
- e) cambiar el tipo de letra – mayúscula o minúscula – (o el tipo de la letra inicial) de los nombres de elementos utilizados en la codificación "ExtendedXMLValue" derivada de cualquier identificador utilizado en la definición de tipo ("Keyword" con "QualifyingInformation" en la "TargetList" que no sea **ALL**) del objetivo u objetivos.

NOTA 1 – "NewName" puede utilizarse para cambiar los nombres utilizados en una codificación EXTENDED-XER derivada de identificadores o de referencias de tipo, pero resulta poco útil si el nuevo nombre pudiera haberse utilizado en primer lugar como identificador ASN.1 o referencia de tipo. Por tanto, la utilización normal de la instrucción de codificación **NAME** es para producir el elemento o nombres de atributos XML requeridos que, en cualquier otro caso, no estarían permitidos debido a las reglas ASN.1 sobre el tipo de letra inicial de los identificadores o nombres de referencia de tipos, o cuando las reglas ASN.1 para diferentes identificadores en construcciones secuencia, conjunto y elección impiden la codificación XML deseada.

NOTA 2 – La utilización de **ALL IN ALL AS CAPITALIZED** para que todos los identificadores de un módulo se escriban con mayúsculas puede resultar particularmente útil para proporcionar un estilo común, utilizando letras iniciales mayúsculas.

NOTA 3 – Si una instrucción de codificación **NAME** se asigna utilizando un objetivo identificado mediante un "identifier" o "typereference", ello influye en el nombre utilizado en una codificación EXTENDED-XER, pero no afecta al nombre utilizado para identificar el mismo objetivo en instrucciones de codificación XER ulteriores.

28.1.4 "RestrictedCharacterStringValue" se define en la Rec. UIT-T X.680 | ISO/CEI 8824-1, cláusula 37.

28.2 Restricciones

28.2.1 "NewName" no se utilizará si "QualifyingInformation" es **ALL**.

28.2.2 La instrucción de codificación **NAME** con "QualifyingInformation" sólo se asignará a las definiciones de tipo siguientes:

- a) una definición de tipo booleano; o
- b) una definición de tipo cadena de bits con bits denominados; o
- c) una definición de tipo enumerado; o
- d) una definición del tipo entero con números denominados.

28.2.3 El "RestrictedCharacterStringValue" del "NewName" utilizado en la instrucción de codificación **NAME**, será un "NCName" definido en los espacios de nombres W3C XML, cláusula 2, producción 4, o una cadena de caracteres vacía. No será una cadena de caracteres vacíos salvo que la instrucción de codificación **NAME** se aplique a una alternativa de un tipo elección con una instrucción de codificación final **USE-UNION**.

NOTA 1 – Los espacios de nombres W3C XML exigen que un "NCName" no comience por caracteres que en mayúsculas correspondan a "XML".

NOTA 2 – La producción "NewNameOrKeyword" (y asimismo la producción "NewName") se utiliza también en la cláusula 31. Las restricciones anteriores sobre "RestrictedCharacterStringValue" no son de aplicación al uso descrito en la cláusula 31.

28.2.4 Si existe una instrucción de codificación **GLOBAL-DEFAULTS** con una palabra clave **MODIFIED-ENCODINGS**, "TargetList" no incluirá "QualifyingInformation".

NOTA – Ello se debe a que en este caso no se utilizan los rótulos de elemento vacío. La instrucción de codificación **TEXT** puede ser utilizada en su lugar para modificar la codificación de los valores individuales de un tipo.

28.2.5 Para evitar confundir al lector, esta instrucción de codificación no debe utilizarse como instrucción de codificación con prefijo conjuntamente con cualquiera de las instrucciones de codificación con prefijo siguientes: **ANY-ATTRIBUTES**, **ANY-ELEMENT** o **UNTAGGED**.

28.3 Efecto en las codificaciones

28.3.1 Si el tipo al que se aplica esta instrucción de codificación tiene una instrucción de codificación final **ATTRIBUTE**, el "AttributeName" (que en este caso es un "IdentifierOrModifiedIdentifier") del "Attribute" (véase 20.3.3) será el "QualifiedOrUnqualifiedName" alternativo tal como se especifica de 28.3.3 a 28.3.6.

28.3.2 Si el tipo al que se aplica esta instrucción de codificación no tiene una instrucción de codificación final **ATTRIBUTE**, el nombre del rótulo del elemento envolvente (que es "TagName" – véase 17.5.1) será el "QualifiedOrUnqualifiedName" alternativo, tal como se especifica en 28.3.3 a 28.3.6.

28.3.3 Se utilizarán los "IdentifierOrModifiedIdentifier" y "QualifiedOrUnqualifiedName" alternativos. El "UnprefixedName" del "QualifiedOrUnqualifiedName" será el "identificador" del componente modificado de conformidad con el "NewNameOrKeyword", tal como se especifica más adelante.

28.3.4 Si se utiliza el "NewName" alternativo, el "UnprefixedName" será sustituido por el "NewName".

28.3.5 Si se utiliza la "Keyword" alternativa, el "UnprefixedName" se modificará tal como se especifica en las subcláusulas siguientes (28.3.5).

28.3.5.1 Si la "Keyword" es **CAPITALIZED** y el primer carácter del "UnprefixedName" es una letra minúscula, dicho carácter se sustituye por su equivalente en mayúsculas, en caso contrario el nombre no se ve afectado.

28.3.5.2 Si la "Keyword" es **UNCAPITALIZED** y el primer carácter del "UnprefixedName" es una letra mayúscula, dicho carácter se sustituye por su equivalente en minúsculas, pero en caso contrario el nombre no se ve afectado.

28.3.5.3 Si la "Keyword" es **UPPERCASED**, todos los caracteres de "UnprefixedName" que sean letras minúsculas serán sustituidos por sus equivalentes en mayúsculas. Los restantes caracteres no se modifican.

28.3.5.4 Si la "Keyword" es **LOWERCASED**, todos los caracteres de "UnprefixedName" que sean letras mayúsculas serán sustituidos por sus equivalentes en minúsculas. Los restantes caracteres no se modifican.

28.3.6 El "QualifiedOrUnqualifiedName" será un nombre de espacio de nombres cualificado si y solo si "Type" tiene una instrucción de codificación final **NAMESPACE**.

29 Instrucción de codificación **NAMESPACE**

29.1 Generalidades

29.1.1 "NamespaceInstruction" se define como:

```
NamespaceInstruction ::=
  NAMESPACE
  TargetList
  NamespaceSpecification ?
```

```
NamespaceSpecification ::=
  AS
  QuotedURI
  Prefix ?
```

```
Prefix ::=
  PREFIX
  QuotedNCName
```

```
QuotedURI ::=
  "\"" & URI & "\""
```

```
QuotedNCName ::=
  "\"" & NCName & "\""
```

29.1.2 La producción "TargetList" se define en 14.2.

NOTA – El uso más común de esta instrucción de codificación es **NAMESPACE ALL**.

29.1.3 Esta instrucción de codificación permite asignar al objetivo u objetivos un prefijo de nombre de espacio de nombres y de espacio de nombres recomendado.

29.1.4 La producción "URI" no se define en esta Recomendación | Norma Internacional, pero consta de caracteres que identifican un identificador de recursos uniforme (URI, *uniform resource identifier*). La sintaxis (y la semántica) de un URI se define en la RFC 2396 del IETF, y comienza con el nombre de un esquema URI. Para atribuciones de nombres de espacio de nombres con la instrucción de codificación **NAMESPACE**, puede utilizarse cualquier esquema URI.

29.1.5 Si "NamespaceSpecification" no existe, se asigna un valor por defecto cuyo "prefijo" ("Prefix") recomendado toma el valor "modulereference" y cuyo "URI" toma su valor de acuerdo con lo siguiente:

- el esquema de URI (véase RFC 2396 del IETF) será **urn**;
- el identificador de espacio de nombres URN (véase RFC 2141 del IETF) será **oid**;
- la cadena específica del espacio de nombres URN (véase RFC 2141 del IETF) será el "DefinitiveIdentifier" del módulo, expresado como "XMLObjectIdentifierValue" (véase RFC 3061 del IETF).

29.1.6 EJEMPLO: si el valor de identificador de objeto es {**iso standard 1564 modules(0) basic(1)**}, el "URI" sería la cadena de caracteres "**urn:oid:1.0.1564.0.1**".

29.1.7 La producción "NCName" se define en espacios de nombres W3C XML, cláusula 2, producción 4, y no comenzará con caracteres que en mayúsculas correspondan a "**XML**".

NOTA – Éste es un requisito que imponen los espacios de nombres XML W3C.

29.2 Restricciones

29.2.1 Esta instrucción de codificación no se asignará salvo que exista una instrucción de codificación **GLOBAL-DEFAULTS MODIFIED-ENCODINGS** en la sección de control de codificación.

29.3 Efecto en las codificaciones

29.3.1 Puede ser necesario un nombre de espacio de nombres cualificado para un nombre de rótulo asociado, para un nombre de atributo o para el valor de un atributo de identificación de tipo. Si el tipo que genera el nombre tiene una instrucción de codificación final **NAMESPACE**, es necesario un nombre de espacio de nombres cualificado.

29.3.2 "QualifiedOrUnqualifiedName" se define como:

```
QualifiedOrUnqualifiedName ::=
    QualifiedName |
    UnqualifiedName

QualifiedName ::=
    PrefixedName |
    UnprefixedName

UnqualifiedName ::=
    UnprefixedName

PrefixedName ::=
    DeclaredPrefix & ":" & UnprefixedName

UnprefixedName ::= NCName

DeclaredPrefix ::= NCName
```

29.3.3 La codificación de un nombre de espacio de nombres cualificado requiere:

- la utilización del "PrefixedName" alternativo para "QualifiedName" que añade a los elementos XML atributos adicionales que proporcionan declaraciones de espacio de nombres (tal como se especifica en los espacios de nombres XML W3C); o
- la utilización del "UnprefixedName" alternativo para "QualifiedName" que añade a los elementos XML atributos adicionales que proporcionan declaraciones de espacio de nombres por defecto (tal como se especifica en los espacios de nombres XML W3C).

29.3.4 La elección de estos dos mecanismos y de los elementos XML a los que se añaden los atributos de declaración de espacio de nombres son una opción del codificador.

NOTA 1 – El Namespace (espacio de nombres) XML 3WC especifica que una declaración de espacio de nombres por defecto sólo abarque en su ámbito el nombre del elemento en el que se declara (y los nombres de elementos hijos), pero no de atributos de dicho elemento o elementos hijos.

NOTA 2 – Es recomendable, aunque no necesario, utilizar el prefijo recomendado en la instrucción de codificación **NAMESPACE**.

NOTA 3 – La utilización del prefijo recomendado puede ser inadecuada si en el módulo existen instrucciones de codificación **NAMESPACE** con distintos nombres de espacio de nombres pero el mismo prefijo recomendado.

30 Instrucción de codificación PI-OR-COMMENT

30.1 Generalidades

30.1.1 "PIOrCommentInstruction" se define como:

```
PIOrCommentInstruction ::=
    PI-OR-COMMENT
    TargetList
    AS
    RestrictedCharacterStringValue
    Position

Position ::=
    BEFORE-TAG
    | BEFORE-VALUE
    | AFTER-VALUE
    | AFTER-TAG
```

30.1.2 La producción "TargetList" se define en 14.2.

30.1.3 Esta instrucción de codificación hace que se inserten instrucciones de procesamiento y/o comentarios XML especificadas antes o después del "ExtendedXMLValue", o antes o después de los rótulos asociados.

NOTA – La subcláusula 10.2.5 permite a un codificador (opción del mismo) insertar instrucciones de procesamiento XML adicionales y comentarios XML.

30.1.4 "RestrictedCharacterStringValue" se define en la Rec. UIT-T X.680 | ISO/CEI 8824-1, cláusula 37.

30.2 Restricciones

30.2.1 El valor de "RestrictedCharacterStringValue" será la concatenación de una o más cadenas de caracteres, cada una de las cuales sea conforme con la sintaxis de una instrucción de procesamiento XML especificada en W3C XML 1.0, 2.6, o con la sintaxis de un comentario XML especificado en W3C XML 1.0, 2.5, y define las instrucciones de procesamiento y/o comentarios que deben insertarse en el documento XML.

30.2.2 Un tipo ASN.1 no tendrá una instrucción de codificación final **UNTAGGED** ni una instrucción de codificación final **PI-OR-COMMENT**.

30.2.3 Un tipo con esta instrucción de codificación final no tendrá ninguna de las instrucciones de codificación finales **ANY-ATTRIBUTES**, **ANY-ELEMENT**, **ATTRIBUTE** o **UNTAGGED**.

30.2.4 "TargetList" no incluirá información de cualificación.

30.3 Efecto en las codificaciones

30.3.1 Si "Position" es **BEFORE-TAG**, las instrucciones de procesamiento y/o los comentarios se insertarán antes del rótulo de inicio asociado o rótulo de elemento vacío. Si dicho rótulo de inicio o rótulo de elemento vacío es el comienzo de algún "ExtendedXMLValue" envolvente, las instrucciones de procesamiento y/o comentarios insertados antes de dicho "ExtendedXMLValue" (que utilice **BEFORE-VALUE** en el tipo correspondiente) precederán a las instrucciones de procesamiento y/o comentarios en el documento XML.

30.3.2 Si "Position" es **BEFORE-VALUE**, las instrucciones de procesamiento y/o los comentarios se insertarán al inicio del "ExtendedXMLValue". Si dicho "ExtendedXMLValue" comienza con un rótulo que es el rótulo de inicio asociado de un "ExtendedXMLValue" integrado, las instrucciones de procesamiento y/o comentarios insertados antes de dicho rótulo de inicio asociado (que utilice **BEFORE-TAG** en el tipo correspondiente) irán a continuación de las instrucciones de procesamiento y/o comentarios en el documento XML.

NOTA – En este caso, el contenido de los rútlulos asociados nunca es vacío, no pudiendo utilizarse el rótulo de elemento vacío.

30.3.3 Si "Position" es **AFTER-VALUE**, las instrucciones de procesamiento y/o los comentarios se insertarán al final del "ExtendedXMLValue". Si dicho "ExtendedXMLValue" finaliza con un rótulo que es el rótulo de terminación asociado de un "ExtendedXMLValue" integrado, las instrucciones de procesamiento y/o comentarios insertados antes de dicho rótulo de terminación asociado (que utilice **AFTER-TAG** en el tipo correspondiente) precederán a las instrucciones de procesamiento y/o comentarios en el documento XML.

NOTA – En este caso, el contenido de los rútlulos asociados nunca es vacío, no pudiendo utilizarse el rótulo de elemento vacío.

30.3.4 Si "Position" es **AFTER-TAG**, las instrucciones de procesamiento y/o los comentarios se insertarán después del rótulo de terminación asociado o rótulo de elemento vacío. Si dicho rótulo de terminación o rótulo de elemento vacío es la terminación de un "ExtendedXMLValue" envolvente, las instrucciones de procesamiento y/o comentarios se insertarán después de dicho "ExtendedXMLValue" (que utilice **AFTER-VALUE** en el tipo correspondiente) e irán a continuación de las instrucciones de procesamiento y/o comentarios en el documento XML.

31 Instrucción de codificación **TEXT**

31.1 Generalidades

31.1.1 "TextInstruction" se define como:

```
TextInstruction ::=
    TEXT
    TargetList
    TextToBeUsed ?
```

```
TextToBeUsed ::=
    AS
    NewNameOrKeyword
```

31.1.2 La producción "TargetList" se define en 14.2.

31.1.3 El objetivo de esta instrucción de codificación es:

- a) en ausencia de **GLOBAL-DEFAULTS MODIFIED-ENCODINGS**, permitir que valores de tipo booleano, de tipo enumerado, de tipo cadena de bits con bits denominados y de tipo entero con números denominados, se codifiquen como cadenas de caracteres en lugar de como rótulos de elemento vacío;
- b) en presencia de **GLOBAL-DEFAULTS MODIFIED-ENCODINGS**, permitir que puedan modificarse las cadenas de caracteres utilizadas para los valores de tipo booleano, de tipo enumerado, de tipo cadena de bits con bits denominados y de tipo entero con números denominados.

31.1.4 "NewNameOrKeyword" se define en cláusula 28. El "NewName" de "NewNameOrKeyword" contendrá al menos un carácter.

31.2 Restricciones

31.2.1 Esta instrucción de codificación sólo se asignará a los tipos siguientes, con información de cualificación que identifique a uno o más de los identificadores utilizados en la definición del tipo (**true** o **false** para el tipo booleano):

- a) una definición de tipo booleano; o
- b) una definición de tipo cadena de bits con bits denominados; o
- c) una definición de tipo enumerado; o
- d) una definición de tipo entero con números denominados.

31.2.2 Las cadenas de caracteres finales utilizadas para los valores del tipo al que se asigna esta instrucción de codificación serán diferentes.

31.2.3 "NewName" de "NewNameOrKeyword" no se utilizará si "QualifyingInformation" es **ALL**. La subcláusula 28.2.3 no es aplicable a esta utilización de "NewNameOrKeyword".

31.2.4 En ausencia de una instrucción de codificación **GLOBAL-DEFAULTS MODIFIED-ENCODINGS**, el conjunto de instrucciones de codificación finales **TEXT** para un tipo no producirá codificaciones de texto para algunos valores abstractos y codificaciones de elemento vacío para otros valores abstractos.

NOTA – Si existe una instrucción de codificación **GLOBAL-DEFAULTS MODIFIED-ENCODINGS**, todas las codificaciones son de texto.

31.2.5 Si la instrucción de codificación **TEXT** se aplica a un tipo cadena de bits con bits denominados y se utiliza "NewName", ésta no contendrá "espacios blancos con escapes" (véase 8.1.5) y no comenzará por "0" (DÍGITO CERO) o por "1" (DÍGITO UNO).

31.2.6 Un tipo con esta instrucción de codificación final tampoco tendrá una instrucción de codificación final **USE-NUMBER**.

NOTA – Las instrucciones de codificación finales siguientes no pueden coexistir con esta instrucción de codificación final porque está prohibida su aplicación a dicho tipo: **ANY-ATTRIBUTES**, **ANY-ELEMENT**, **BASE64**, **DECIMAL**, **EMBED-VALUES**, **LIST**, **USE-NIL**, **USE-ORDER**, **USE-QNAME**, **USE-TYPE**, **USE-UNION**, **WHITESPACE**.

31.2.7 "QualifyingInformation" siempre estará presente.

31.3 Efecto en las codificaciones

31.3.1 Se aplica una de las cinco subcláusulas siguientes (31.3.2 a 31.3.6).

31.3.2 Si el tipo no es un tipo cadena de bits con bits denominados y no existe el "TextToBeUsed", la codificación "ExtendedXMLValue" de cada valor referenciado por la información de cualificación para esta instrucción será una cadena de caracteres con los caracteres del identificador (o bien, será **true** o **false** en el caso de tipos booleanos). Para tipos enteros con valores denominados, se utilizarán los identificadores o los correspondientes números (opción del codificador).

31.3.3 Si el tipo es un tipo cadena de bits con bits denominados y no existe el "TextToBeUsed", una cadena de caracteres idéntica al identificador del bit representará al bit cuando éste haya sido fijado. Cada valor abstracto se codificará como la concatenación (posiblemente vacía) de dichas cadenas de caracteres de caracteres para todos los bits fijados, separados por "espacios blancos con escapes" (véase 8.1.5).

31.3.4 Si el tipo no es un tipo cadena de bits con bits denominados y existe el "TextToBeUsed", se aplican las subcláusulas siguientes (no obstante, véase 31.3.5).

31.3.4.1 Si se utiliza el "NewName" alternativo, la cadena de caracteres utilizada para codificar el valor identificado por "QualifyingInformation" es "NewName". Cada ocurrencia de los caracteres "<", ">", y "&" en el "NewName" será sustituida por una de las secuencias de escape "<"; ">"; y "&"; respectivamente, o por una secuencia de escape de la forma "&#n;"; o "&#xn;";, especificada en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 11.15.8.

31.3.4.2 Si se utiliza la "Keyword" (palabra clave) alternativa, la cadena de caracteres utilizada para codificar valores del tipo es el nombre del identificador, modificado como se especifica a continuación.

31.3.4.3 Si la "Keyword" es **CAPITALIZED**, el primer carácter del nombre se sustituye por su equivalente en mayúsculas, en caso contrario, el nombre no se ve afectado.

31.3.4.4 Si la "Keyword" es **UNCAPITALIZED**, el nombre no se ve afectado.

31.3.4.5 Si la "Keyword" es **UPPERCASED**, todos los caracteres del nombre que sean letras minúsculas se sustituyen por sus equivalentes en mayúsculas. Los demás caracteres no se modifican.

31.3.4.6 Si la "Keyword" es **LOWERCASED**, todos los caracteres del nombre que sean letras mayúsculas se sustituyen por sus equivalentes en minúsculas. Los demás caracteres no se modifican.

31.3.5 Si el tipo es un tipo entero con valores denominados, las cadenas de caracteres producidas por la aplicación de las subcláusulas 31.3.4.1 a 31.3.4.6 se utilizarán en lugar de los identificadores. Se utilizarán las cadenas de caracteres o los números correspondientes (opción del codificador).

31.3.6 Si el tipo es un tipo cadena de bits con bits denominados y existe "TextToBeUsed", se aplicarán las subcláusulas 31.3.4.1 a 31.3.4.6 a cada identificador de bit para producir la cadena de caracteres que representa el bit cuando éste se fija. El valor cadena de bits se codificará como la concatenación (posiblemente vacía) de estas cadenas de caracteres para todos los bits que se fijan, separados por "espacios blancos con escapes ".

32 Instrucción de codificación **UNTAGGED**

32.1 Generalidades

32.1.1 "UntaggedInstruction" se define como:

```
UntaggedInstruction ::=
    UNTAGGED
    TargetList
```

32.1.2 La producción "TargetList" se define en 14.2.

32.1.3 (Tutorial) El anexo B incluye una descripción informal del efecto de **UNTAGGED** sobre los constructores ASN.1. Esta cláusula y sus subcláusulas proporcionan una introducción didáctica (tutorial) para ilustrar algunos de los efectos de la utilización de **UNTAGGED**.

32.1.4 Utilizado (posiblemente de forma repetida y anidada) conjuntamente con secuencia, conjunto, elección, secuencia-de y conjunto-de, permite especificar un patrón casi arbitrario de elementos XML. Su efecto es la supresión del rótulo de inicio XML que precede al "ExtendedXMLValue" del "Type" al que se aplica y el rótulo XML de terminación que le sigue, dando lugar a que los elementos XML normalmente contenidos entre dichos rótulos se conviertan en contenido XML parcial.

32.1.5 Aplicado a un tipo elección como componente de una secuencia o conjunto, especifica la inclusión en dicho punto de la secuencia (o conjunto) de exactamente una de las alternativas del tipo elección (o ninguna si el tipo elección es un componente **OPTIONAL**). El identificador del tipo elección no aparece en la codificación. Algunas alternativas del tipo elección pueden ser elementos XML, pero otras puede ser un contenido XML parcial con un patrón casi arbitrario de múltiples elementos, gracias a la utilización de **UNTAGGED** en la definición de dichas alternativas.

32.1.6 Aplicado a un tipo secuencia-de como componente de una secuencia o conjunto, especifica la inclusión en ese punto de la secuencia (o conjunto) de un número especificado o arbitrario de repeticiones del componente secuencia-de (que puede producir un único elemento XML, o un contenido XML parcial si es **UNTAGGED**).

32.1.7 Aplicado a un tipo secuencia (o conjunto) o a un tipo secuencia-de (o conjunto-de) como alternativa a un tipo elección, permite que dicha alternativa se componga de un contenido XML parcial que es el "ExtendedXMLValue" de la secuencia, conjunto, secuencia-de o conjunto-de.

32.1.8 Una función diferente de **UNTAGGED** cuando se aplica a un tipo codificable mediante caracteres es permitir que en la codificación de una secuencia aparezca un contenido formado por caracteres, sin rótulos alrededor de dicho contenido. Esta utilización sólo se permite a un componente de una secuencia que no sea **UNTAGGED**.

NOTA – La restricción tiene por objeto simplificar las reglas necesarias para asegurar una decodificación fácil e inequívoca.

32.2 Restricciones

32.2.1 En todos los ejemplares de utilización, el tipo envolvente será un tipo secuencia, conjunto, elección, secuencia-de o conjunto-de.

32.2.2 Si el tipo es un tipo codificable mediante caracteres, el tipo envolvente será un tipo secuencia sin una instrucción de codificación final **UNTAGGED**. El tipo no se marcará como **OPTIONAL** o **DEFAULT**. Los restantes componentes del tipo secuencia envolvente (si existen) tendrán una instrucción de codificación final **ATTRIBUTE** o **ANY-ATTRIBUTES**.

32.2.3 Si el tipo no es un tipo codificable mediante caracteres, será un tipo secuencia, conjunto, elección, secuencia-de, conjunto-de, cadena de octetos o cadena de bits, incluyendo un "Type" sin **ENCODED BY**, o un tipo abierto.

NOTA – En el anexo B proporciona directrices para garantizar que no existan ambigüedades como consecuencia de la utilización de esta instrucción de codificación.

32.2.4 Esta instrucción de codificación no se aplicará a un tipo que tenga una codificación "ExtendedXMLValue" vacía para uno de sus valores abstractos si el tipo se utiliza como:

- a) un componente de un tipo secuencia o conjunto con **OPTIONAL** o **DEFAULT**; o
- b) el componente de un tipo secuencia-de o conjunto-de; o
- c) una alternativa de un tipo elección, si otra alternativa del mismo tipo elección tiene una codificación "ExtendedXMLValue" vacía para uno de sus valores abstractos y una instrucción de codificación final **UNTAGGED**.

EJEMPLO: Un tipo secuencia cuyos componentes sean todos **OPTIONAL** tiene un valor abstracto con una codificación "ExtendedXMLValue" vacía, tal como ocurre para un tipo secuencia-de que permita cero repeticiones.

32.2.5 Esta instrucción de codificación no se asignará salvo que exista una instrucción de codificación **GLOBAL-DEFAULTS MODIFIED-ENCODINGS** en la sección de control de codificación.

32.2.6 Un tipo con esta instrucción de codificación final no tendrá ninguna de las instrucciones de codificación finales **ANY-ATTRIBUTES**, **ANY-ELEMENT**, **ATTRIBUTE**, **DEFAULT-FOR-EMPTY**, **EMBED-VALUES**, **PI-OR-COMMENT**, **USE-NIL**, **USE-ORDER** o **USE-TYPE**.

32.2.7 "TargetList" no incluirá ninguna información de cualificación.

32.3 Efecto en las codificaciones

32.3.1 Esta instrucción de codificación se ignora cuando el tipo se codifica como un tipo de nivel superior.

32.3.2 Si el tipo envolvente es un tipo elección, el "ExtendedXMLChoiceValue" (véase 17.5.1) para esta alternativa del tipo envuelto será el "ExtendedXMLValue" de la alternativa (la segunda alternativa en la producción "ExtendedXMLChoiceValue").

NOTA – El "ExtendedXMLValue" para la alternativa puede ser un único elemento XML o un contenido XML parcial con múltiples elementos XML.

32.3.3 Si el tipo envolvente es un tipo secuencia o conjunto, el "ExtendedXMLNamedValue" (véase 17.6) para este componente del tipo envuelto será sustituida por el "ExtendedXMLValue" del componente (la segunda alternativa en la producción "ExtendedXMLNamedValue").

NOTA – Este "ExtendedXMLValue" puede ser un elemento XML sencillo o un contenido XML parcial que conste de múltiples elementos XML.

32.3.4 Si el tipo envolvente es un tipo secuencia-de o conjunto-de, el "ExtendedXMLDelimitedItem" (si se utiliza – véase 17.7) de cada repetición será sustituido por el "ExtendedXMLValue" envuelto en el "ExtendedXMLDelimitedItem".

NOTA 1 – No es posible utilizar **UNTAGGED** salvo que se haya incluido un **GLOBAL-DEFAULTS** de **MODIFIED-ENCODINGS** en la sección de control de codificación, en cuyo caso no se permite utilizar "ExtendedXMLValueList" (véase 17.7.2).

NOTA 2 – El "ExtendedXMLValue" puede ser un elemento XML sencillo o un contenido XML parcial con múltiples elementos XML.

32.3.5 Si el tipo es un tipo cadena de octetos o cadena de bits que contenga un "Type" sin **ENCODED BY**, o un tipo abierto, el "ExtendedXMLValue" será un "ExtendedXMLTypedValue" (no un "xmlhstring" ni un "XMLBase64String").

NOTA – Dichos tipos no concuerdan con la definición de tipo codificable mediante caracteres (véase 3.2.2 *ter*). Lo indicado en la subcláusula 32.3.5 implica que cuando tengan una instrucción de codificación final **UNTAGGED**, siempre se codifican como elementos XML.

33 Instrucción de codificación **USE-NIL**

33.1 Generalidades

33.1.1 "UseNilInstruction" se define como:

UseNilInstruction ::=
USE-NIL
TargetList

33.1.2 La producción "TargetList" se define en 14.2.

33.1.3 Esta instrucción de codificación proporciona una codificación optimizada **EXTENDED-XER** para una secuencia con un único componente **OPTIONAL** cuyos restantes componentes (si existen) tengan una instrucción de codificación final **ATTRIBUTE** o **ANY-ATTRIBUTES**, posiblemente precedida de un tipo inicial secuencia-de que soporte **USE-ORDER** (véase la cláusula 35).

33.1.4 En ausencia de esta instrucción de codificación, el componente opcional se codificaría como se indica a continuación:

- a) (caso "no ausente pero vacío") si el componente existe en el valor abstracto, y el valor abstracto tiene una codificación "ExtendedXMLValue" vacía, existe un "ExtendedXMLNamedValue" para el componente en el documento XML, que normalmente es un rótulo de elemento vacío (o con rótulos adyacentes de inicio y terminación);
- b) (caso "ausente") si el componente no existe en el valor abstracto, tampoco existe "ExtendedXMLNamedValue";
- c) (caso "no ausente ni vacío ") si el componente existe en el valor abstracto con un valor abstracto que no tenga una codificación vacía, existe un "ExtendedXMLNamedValue" para el componente con un contenido no vacío.

33.1.5 La utilización de **USE-NIL** requiere que se señalice la ausencia del componente opcional [caso b) anterior] mediante la inclusión de un atributo de identificación nulo de nombre "nil" y un valor que sea "true" o "1".

33.1.6 En los casos a) y c) de 33.1.4, el atributo de identificación nulo puede omitirse (opción del codificador), o puede estar presente con un valor que sea "false" o "0". El componente opcional se codificará omitiendo los rótulos asociados.

33.2 Restricciones

33.2.1 La instrucción de codificación **USE-NIL** sólo se asignará a un tipo secuencia que tenga un componente **OPTIONAL** sin una instrucción de codificación final **ATTRIBUTE**. Los restantes componentes del tipo secuencia, si existen, tendrán una instrucción de codificación final **ATTRIBUTE** o **ANY-ATTRIBUTES**, o serán los componentes de secuencia-de que soporten una instrucción de codificación **USE-ORDER** o **EMBED-VALUES**, que también son instrucciones de codificación finales en el tipo secuencia.

33.2.2 El tipo secuencia no incluirá una instrucción de codificación final **UNTAGGED**.

33.2.3 El componente **OPTIONAL** no tendrá ninguna de las instrucciones de codificación finales **ANY-ELEMENT**, **ANY-ATTRIBUTES**, **DEFAULT-FOR-EMPTY**, **EMBED-VALUES**, **PI-OR-COMMENT**, **UNTAGGED**, **USE-NIL**, **USE-ORDER** o **USE-TYPE**.

NOTA – Además de **UNTAGGED**, las instrucciones de codificación enumeradas son las que no pueden aplicarse a un tipo que tenga una instrucción de codificación final **UNTAGGED**.

33.2.4 Si el componente **OPTIONAL** no es un tipo codificable mediante caracteres, será un tipo secuencia, conjunto, elección, secuencia-de, conjunto-de, un tipo abierto o un tipo cadena de octetos o cadena de bits que incluya un "Type" y sin **ENCODED BY**.

33.2.5 Si el componente **OPTIONAL** es un tipo secuencia, ninguno de sus componentes tendrá una instrucción de codificación final **ATTRIBUTE** o **ANY-ATTRIBUTES**.

33.2.6 Esta instrucción de codificación no se asignará salvo que exista una instrucción de codificación **GLOBAL-DEFAULTS MODIFIED-ENCODINGS** en la sección de control de codificación.

33.2.7 Un tipo con esta instrucción de codificación final tampoco tendrá ninguna de las instrucciones de codificación finales **UNTAGGED** o **USE-QNAME**.

NOTA – Las instrucciones de codificación finales siguientes no pueden coexistir con esta instrucción de codificación final porque está prohibida su aplicación a dicho tipo: **ANY-ATTRIBUTES**, **ANY-ELEMENT**, **ATTRIBUTE**, **BASE64**, **DECIMAL**, **LIST**, **TEXT**, **USE-NUMBER**, **USE-TYPE**, **USE-UNION**, **WHITESPACE**.

33.2.8 "TargetList" no incluirá información de cualificación.

33.3 Efecto en las codificaciones

33.3.1 Si el componente **OPTIONAL** no existe [caso b) de 33.1.4], a la "AttributeList" del elemento envolvente se añadirá un atributo de identificación nulo de nombre "nil" y con un valor "true" o "1".

33.3.2 Si el componente **OPTIONAL** existe [casos a) y c) de 33.1.4], el atributo de identificación nulo puede omitirse (opción del codificador), o puede añadirse a la "AttributeList" del elemento envolvente con un valor que sea "false" o "0". El componente opcional se codificará omitiendo los rótulos asociados.

34 Instrucción de codificación **USE-NUMBER**

34.1 Generalidades

34.1.1 "UseNumberInstruction" se define como:

```
UseNumberInstruction ::=
    USE-NUMBER
    TargetList
```

34.1.2 La producción "TargetList" se define en 14.2.

34.1.3 El objeto de esta instrucción de codificación es modificar la codificación de un tipo enumerado de forma que se utilicen los números de las enumeraciones "NamedNumber" en lugar de los nombres.

34.2 Restricciones

34.2.1 Esta instrucción de codificación se ignorará salvo que se aplique a un tipo enumerado.

34.2.2 Un tipo con esta instrucción de codificación final tampoco tendrá una instrucción de codificación final **TEXT**.

NOTA – Las instrucciones de codificación finales siguientes no pueden coexistir con esta instrucción de codificación final porque está prohibida su aplicación a dicho tipo: **ANY-ATTRIBUTES**, **ANY-ELEMENT**, **BASE64**, **DECIMAL**, **EMBED-VALUES**, **LIST**, **USE-NIL**, **USE-ORDER**, **USE-QNAME**, **USE-TYPE**, **USE-UNION**, **WHITESPACE**.

34.2.3 "TargetList" no incluirá información de cualificación.

34.3 Efecto en las codificaciones

34.3.1 "ExtendedXMLEnumeratedValue" se define como:

```
ExtendedXMLEnumeratedValue ::=
    EmptyElementEnumerated
    | TextEnumerated
    | XMLSignedNumber
```

34.3.2 "EmptyElementEnumerated" y "TextEnumerated" se definen en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 19.8 y 19.9.

34.3.3 "XMLSignedNumber" se define en la Rec. UIT-T X.680 | ISO/CEI 8824-1, 18.9 y 18.12, y será el número en el "NamedNumber" de la enumeración.

34.3.4 La alternativa XMLSignedNumber se utilizará si y sólo si el tipo enumerado tiene esta instrucción de codificación final.

NOTA – Si en la sección de control de codificación XER existe un **GLOBAL-DEFAULTS** de **MODIFIED-ENCODINGS** pero el tipo enumerado no tiene esta instrucción de codificación final, se utiliza la segunda alternativa. Si en la sección de control de codificación XER no existe un **GLOBAL-DEFAULTS** de **MODIFIED-ENCODINGS**, se utiliza la primera alternativa.

35 Instrucción de codificación USE-ORDER

35.1 Generalidades

35.1.1 "UseOrderInstruction" se define como:

UseOrderInstruction ::=
USE-ORDER
TargetList

35.1.2 La producción "TargetList" se define en 14.2.

35.1.3 El objeto de esta instrucción de codificación es permitir la codificación EXTENDED-XER optimizada de un tipo secuencia en la que exista un componente secuencia-de que determina el orden semántico de los valores de los componentes siguientes del tipo secuencia que están codificados como elementos. También puede utilizarse, cuando exista además una instrucción de codificación final **USE-NIL** (véase la cláusula 33) y la única componente **OPTIONAL** requerida para la utilización de **USE-NIL** sea una secuencia, a fin de determinar el orden semántico de los componentes de dicha secuencia **OPTIONAL**.

35.1.4 El componente secuencia-de que determina el orden semántico es el primer componente de la secuencia, salvo que también exista un componente secuencia-de que soporte una instrucción de codificación final **EMBED-VALUES** en el tipo secuencia. En ese caso, el componente secuencia-de que soporta la instrucción de codificación **EMBED-VALUES** precede al componente secuencia-de que soporta la instrucción de codificación **USE-ORDER**.

35.1.5 El componente que determina el orden semántico debe ser un tipo secuencia-de con un componente que sea un tipo enumerado. Dicho tipo secuencia-de y su semántica es función de la presencia o no de una instrucción de codificación **USE-NIL** en el tipo secuencia, tal como se describe en las subcláusulas siguientes.

35.1.5.1 Cuando no exista instrucción de codificación final **USE-NIL**, los nombres de las enumeraciones serán idénticos a los identificadores ASN.1 de los componentes del tipo secuencia. El orden de las enumeraciones en cada valor abstracto determina el orden semántico de los valores de los componentes siguientes del tipo secuencia que existen en la codificación.

35.1.5.2 Cuando también exista una instrucción de codificación final **USE-NIL**, el componente **OPTIONAL** necesario por la utilización de **USE-NIL** debe ser un tipo secuencia (por ejemplo, de tipo B), y los nombres de las enumeraciones serán idénticas a los identificadores ASN.1 de los componentes del tipo secuencia B. El orden de las enumeraciones en cada valor abstracto determina el orden semántico de los valores de los componentes del tipo secuencia B que existen en la codificación.

35.2 Restricciones

35.2.1 Esta instrucción de codificación sólo se asignará a un tipo secuencia. El tipo secuencia contendrá un componente que sea un tipo secuencia-de (por ejemplo, de tipo A) con un componente que sea un tipo enumerado. Si el tipo secuencia no tiene una instrucción de codificación final **EMBED-VALUES**, el tipo A será el primer componente, y en cualquier otro caso, será el segundo componente. Si no existe una instrucción de codificación final **USE-NIL**, el tipo secuencia tendrá al menos otro componente sin instrucción de codificación final **ATTRIBUTE** o **ANY-ATTRIBUTES** (un componente sin atributo). Si existe una instrucción de codificación final **USE-NIL**, el componente **OPTIONAL** que soporte **USE-NIL** será un tipo secuencia, y al menos tendrá un componente.

35.2.2 El tipo enumerado tendrá identificadores que dependan de la presencia o ausencia de una instrucción de codificación final **USE-NIL** en el tipo secuencia con la instrucción de codificación **USE-ORDER**, tal como se especifica en las subcláusulas siguientes.

35.2.2.1 Si no existe instrucción de codificación final **USE-NIL**, el tipo enumerado tendrá identificadores para las enumeraciones que se correspondan biunívocamente (y en el mismo orden textual) con los identificadores de los siguientes componentes sin atributo (véase 35.2.1) de la secuencia. El tipo secuencia-de estará limitado de forma que cada valor abstracto contenga exactamente un identificador para cada componente sin atributo de la secuencia presente en el valor abstracto.

35.2.2.2 Si existe una instrucción de codificación final **USE-NIL**, el tipo enumerado tendrá identificadores para las enumeraciones que se correspondan biunívocamente (y en el mismo orden textual) con los identificadores de los componentes del componente **OPTIONAL** en el tipo secuencia. El tipo secuencia-de estará limitado de forma que cada valor abstracto contenga exactamente un identificador para cada componente de la secuencia **OPTIONAL** presente en el valor abstracto.

NOTA – Se recomienda que la limitación sobre el tipo secuencia se exprese de la forma siguiente:
(**CONSTRAINED BY** { /* Shall conform to ITU-T Rec. X.693 | ISO/IEC 8825-4, clause 35 */ })

35.2.2.3 Todos los "EnumerationItem" de las enumeraciones serán "identificadores" o "NamedNumber" con el valor 0 en el primer "EnumerationItem", 1 en el segundo, y así sucesivamente, hasta el último "EnumerationItem".

35.2.3 El tipo secuencia-de no estará marcado como **OPTIONAL** o **DEFAULT**.

35.2.4 Los siguientes componentes de la secuencia (si no existe instrucción de codificación final **USE-NIL**), y los componentes de la secuencia **OPTIONAL** (si existe una instrucción de codificación final **USE-NIL**) no se marcarán como **DEFAULT**, salvo que tengan una instrucción de codificación final **ATTRIBUTE**.

35.2.5 Ningún componente de la secuencia con esta instrucción de codificación final ni de la secuencia **OPTIONAL** (cuando exista una instrucción de codificación final **USE-NIL**) tendrá una instrucción de codificación final **UNTAGGED**, con independencia de que el tipo de dicho componente sea o no un tipo codificable mediante caracteres.

35.2.6 Ningún componente de la secuencia con esta instrucción de codificación final tendrá una instrucción de codificación final **ANY-ELEMENT**.

35.2.7 Esta instrucción de codificación no se asignará salvo que exista una instrucción de codificación **GLOBAL-DEFAULTS MODIFIED-ENCODINGS** en la sección de control de codificación.

35.2.8 Un tipo con esta instrucción de codificación final tampoco tendrá una instrucción de codificación final **UNTAGGED**.

NOTA – Las instrucciones de codificación finales siguientes no pueden coexistir con esta instrucción de codificación final porque está prohibida su aplicación a dicho tipo: **ANY-ATTRIBUTES**, **ANY-ELEMENT**, **ATTRIBUTE**, **BASE64**, **DECIMAL**, **LIST**, **TEXT**, **USE-NUMBER**, **USE-QNAME**, **USE-TYPE**, **USE-UNION**, **WHITESPACE**.

35.2.9 "TargetList" no incluirá información de cualificación.

35.3 Efecto en las codificaciones

35.3.1 El tipo secuencia-de con componente enumerado no se codificará directamente.

35.3.2 Un codificador codificará la semántica de este tipo (el orden semántico de los componentes secuencia o de los componentes secuencia **OPTIONAL**) codificando los componentes que están codificados como elementos en el orden especificado por el tipo secuencia-de con el componente enumerado. Un decodificador recuperará el valor del componente secuencia-de utilizando para ello el orden de los elementos codificados.

36 Instrucción de codificación USE-QNAME

36.1 Generalidades

36.1.1 "UseQNameInstruction" se define como:

UseQNameInstruction ::=
USE-QNAME
TargetList

36.1.2 La producción "TargetList" se define en 14.2.

36.1.3 El propósito de esta instrucción de codificación es modificar la codificación de un tipo secuencia, cada uno de cuyos valores especifica un nombre de espacio de nombres (un URI) opcional y un nombre sin prefijo, de forma que realiza la codificación como un nombre de espacio de nombres XML cualificado o no cualificado.

NOTA – Se facilita porque está disponible en otras notaciones de esquema. Un ejemplo de un tipo secuencia al que se aplicaría es el tipo **QName** definido en la Rec. UIT-T X.694 | ISO/CEI 8825-5.

36.1.4 Si el componente opcional existe en un valor abstracto del tipo secuencia, dicho valor abstracto representa un nombre de espacio de nombres cualificado. Si el componente opcional no existe, el tipo secuencia representa un nombre no cualificado.

36.2 Restricciones

36.2.1 Esta instrucción de codificación sólo se asignará a una secuencia que tenga exactamente dos componentes, ambos del tipo **UTF8String**. El primer componente será **OPTIONAL**.

36.2.2 El primer componente estará restringido a representar un URI (véase la RFC 2396 del IETF). El segundo componente estará restringido a contener un "NCName", tal como se especifica en W3C XML Namespaces, cláusula 2, producción 4, y no comenzará por caracteres que en mayúsculas coincidan con "**XML**".

36.2.3 Esta instrucción de codificación no se asignará salvo que exista una instrucción de codificación **GLOBAL-DEFAULTS MODIFIED-ENCODINGS** en la sección de control de codificación.

36.2.4 Un tipo con esta instrucción de codificación final no tendrá una instrucción de codificación final **USE-NIL**.

NOTA – Las instrucciones de codificación finales siguientes no pueden coexistir con esta instrucción de codificación final porque está prohibida su aplicación a dicho tipo: **ANY-ATTRIBUTES**, **ANY-ELEMENT**, **BASE64**, **DECIMAL**, **EMBED-VALUES**, **LIST**, **TEXT**, **USE-NUMBER**, **USE-ORDER**, **USE-TYPE**, **USE-UNION**, **WHITESPACE**.

36.2.5 "TargetList" no incluirá información de cualificación.

36.3 Efecto en las codificaciones

36.3.1 La presencia de esta instrucción de codificación en un tipo, cuando existe el componente opcional, requiere que haya una declaración de espacio de nombres (o una declaración de espacio de nombres por defecto) que pueda corresponder con el valor de atributo o contenido de elemento que codifique el valor de este tipo, de conformidad con la cláusula 29. El valor del atributo o contenido de elemento se codifica entonces como se especifica en la cláusula 29 para un nombre de espacio de nombres cualificado.

36.3.2 Si no existe el componente opcional, no existirá una declaración de espacio de nombres por defecto que pueda corresponder con el valor del atributo o contenido de elemento que codifique el valor de este tipo.

37 Instrucción de codificación **USE-TYPE**

37.1 Generalidades

37.1.1 "UseTypeInstruction" se define como:

```
UseTypeInstruction ::=
    USE-TYPE
    TargetList
```

37.1.2 La producción "TargetList" se define en 14.2.

37.1.3 Esta instrucción de codificación optimiza la codificación **EXTENDED-XER** de un tipo elección. Requiere que se codifique un atributo de identificación de tipo en el elemento envolvente a fin de identificar la alternativa que ha sido codificada (salvo que sea la primera alternativa) y la supresión del rótulo de inicio y del rótulo de terminación que enmarcan la codificación de las alternativas.

37.1.4 El atributo de identificación de tipo identifica el tipo de un elemento XML. Es necesario que el nombre del atributo sea el nombre "**type**" del espacio de nombres de control (véase 16.9) y su valor identifica una alternativa del tipo elección a la que se aplica esta instrucción de codificación (permite una determinación alternativa para el tipo elección).

37.2 Restricciones

37.2.1 El tipo al que se asigne **USE-TYPE** será un tipo elección sin una instrucción de codificación final **UNTAGGED**.

37.2.2 Ninguna de las alternativas del tipo elección tendrá una instrucción de codificación final **UNTAGGED**.

37.2.3 Ninguna de las alternativas del tipo elección será un tipo elección con una instrucción de codificación final **USE-TYPE**.

NOTA – Una o más alternativas del tipo elección pueden ser de tipo elección con una instrucción de codificación final **USE-UNION**.

37.2.4 Esta instrucción de codificación no se asignará salvo que exista una instrucción de codificación **GLOBAL-DEFAULTS MODIFIED-ENCODINGS** en la sección de control de codificación.

37.2.5 Un tipo con esta instrucción de codificación final no tendrá ninguna de las instrucciones de codificación finales **UNTAGGED** o **USE-UNION**.

NOTA – Las instrucciones de codificación finales siguientes no pueden coexistir con esta instrucción de codificación final porque está prohibida su aplicación a dicho tipo: **ANY-ATTRIBUTES**, **ANY-ELEMENT**, **ATTRIBUTE**, **BASE64**, **DECIMAL**, **DEFAULT-FOR-EMPTY**, **EMBED-VALUES**, **LIST**, **TEXT**, **USE-NIL**, **USE-NUMBER**, **USE-ORDER**, **USE-QNAME**, **WHITESPACE**.

37.2.6 "TargetList" no incluirá información de cualificación.

37.3 Efecto en las codificaciones

37.3.1 Si la alternativa del tipo elección que se codifica no es la primera alternativa de dicha elección, se añadirá un atributo de identificación de tipo (véanse 37.3.3 y 37.3.4) a la "AttributeList" del elemento envolvente, salvo que se aplique 37.3.8.

37.3.2 Si la alternativa del tipo elección que se codifica es la primera alternativa, es una opción del codificador añadir u omitir el atributo de identificación de tipo, salvo que se aplique 37.3.8.

37.3.3 El atributo identificación de tipo será una instancia de la producción "Attribute" (véase 20.3.3) con un espacio de nombres cualificado "ControlAttributeName" (véase 20.3.5) de "**type**" del espacio de nombres de control (véase 16.9).

37.3.4 El valor del atributo de identificación de tipo será el identificador de la alternativa elegida, posiblemente modificada de conformidad con instrucciones de codificación finales **NAME** y **NAMESPACE**.

37.3.5 Si en la codificación de un tipo con esta instrucción de codificación final no existe atributo de identificación de tipo, el decodificador asumirá que existe la primera alternativa de la elección.

37.3.6 La presencia de un atributo de identificación de tipo con un valor inesperado no dará lugar a un error de decodificación. Cuando en una codificación se detecte este atributo, el decodificador asumirá que se ha identificado la primera alternativa de la elección, y puede ignorar el atributo de identificación de tipo (o pasarlo a la aplicación). Además, en tales casos, el decodificador puede ignorar (o pasar a la aplicación) los atributos inesperados y los elementos hijos inesperados que se encuentren después de todos los elementos hijo esperados existentes en el "ExtendedXMLValue" de la alternativa.

37.3.7 Todos los "Attribute" que en otro caso estarían en la "AttributeList" del "ExtendedXMLChoiceValue" se añadirán a la "AttributeList" del elemento envolvente y el "ExtendedXMLChoiceValue" del tipo elección será sustituido por el "ExtendedXMLValue" en el "ExtendedXMLChoiceValue".

37.3.8 Si una o más alternativas del tipo elección con la instrucción de codificación final **USE-TYPE** son de tipo elección con una instrucción de codificación final **USE-UNION**, es opcional del codificador que el atributo de identificación de tipo pueda identificar una de las alternativas del tipo elección con la instrucción de codificación final **USE-UNION** en lugar de una alternativa del tipo elección con la instrucción de codificación final **USE-TYPE**.

38 Instrucción de codificación USE-UNION

38.1 Generalidades

38.1.1 "UseUnionInstruction" se define como:

```
UseUnionInstruction ::=  
  USE-UNION  
  TargetList
```

38.1.2 La producción "TargetList" se define en 14.2.

38.1.3 Esta instrucción de codificación optimiza la codificación de un tipo elección cuando la codificación de los valores abstractos de cada alternativa es lo suficientemente distinta de la codificación de valores abstractos de otras alternativas, como para que un decodificador determine el valor abstracto representado analizando la codificación.

38.1.4 Si el tipo elección con una instrucción de codificación final **USE-UNION** no tiene una instrucción de codificación final **ATTRIBUTE** o una instrucción de codificación final **UNTAGGED**, esta instrucción de codificación puede dar lugar a la inserción de un atributo de identificación de tipo en el elemento envolvente para identificar la alternativa codificada. Si el tipo elección tiene una instrucción de codificación final **ATTRIBUTE** o **UNTAGGED**, o es el componente de un tipo secuencia-de o conjunto-de con una instrucción de codificación **LIST**, no es posible insertar el atributo de identificación de tipo.

38.1.5 Esta instrucción de codificación produce la supresión del rótulo de inicio y del rótulo de terminación que enmarcan la codificación de la alternativa.

38.2 Restricciones

38.2.1 Un tipo con una instrucción de codificación final de **USE-UNION** será un tipo elección.

38.2.2 Todas las alternativas del tipo elección serán tipos codificables mediante caracteres, pero no serán tipos elección con una instrucción de codificación final **USE-UNION**.

38.2.3 Si el tipo elección tiene una instrucción de codificación final **ATTRIBUTE** o **UNTAGGED** o se utiliza en una definición de tipo como componente de un tipo secuencia-de o conjunto-de con una instrucción de codificación final **LIST**, las alternativas del tipo elección estarán limitadas de forma que, para cualquier alternativa, todos sus valores abstractos tendrán al menos una codificación (su "ExtendedXMLValue") diferente a las codificaciones permitidas de todas las alternativas textualmente precedentes.

NOTA – La razón de este requisito se debe a que resulta imposible insertar un atributo de identificación de tipo que determine la alternativa que había sido seleccionada. Sin este requisito, la codificación sería ambigua.

38.2.4 En las dos subcláusulas siguientes, el término "identificador" significa identificador (posiblemente modificado de conformidad con cualesquiera instrucciones de codificación finales **NAME** y **NAMESPACE**) de una alternativa (del tipo elección).

38.2.5 Si el tipo elección (por ejemplo, un tipo U1) se codifica como una alternativa de un tipo elección envolvente (por ejemplo, un tipo E) que tenga una instrucción de codificación final **USE-TYPE**, y el identificador de una de las alternativas de E es idéntico a una de las alternativas de U, cada valor abstracto de dicha alternativa de U tendrá al menos una codificación diferente de todas las codificaciones de las alternativas de U textualmente precedentes.

NOTA – La razón de este requisito se debe a que en este caso resulta imposible identificar la alternativa de U, ya que el identificador en un atributo de identificación de tipo para E, solamente identificaría a U en su conjunto.

38.2.6 Si el tipo elección (por ejemplo, un tipo U1) se codifica como una alternativa de un tipo elección envolvente (por ejemplo, un tipo E) con una instrucción de codificación final **USE-TYPE**, y E contiene otro tipo elección (por ejemplo, un tipo U2) con una instrucción de codificación **USE-UNION** que textualmente sea posterior a U1 en E, y el identificador de cualquiera de las alternativas de U2 es idéntico a uno de los identificadores que existen en U1, cada valor abstracto de dicha alternativa de U2 tendrá al menos una codificación diferente a las codificaciones de todas las alternativas de U1.

NOTA – La razón de este requisito se debe a que en este caso resulta imposible identificar la alternativa de U2, ya que el identificador en un atributo de identificación de tipo para E identificaría la alternativa en U1.

38.2.7 Esta instrucción de codificación no será asignada salvo que exista una instrucción de codificación **GLOBAL-DEFAULTS MODIFIED-ENCODINGS** en la sección de control de codificación.

38.2.8 Un tipo con esta instrucción de codificación final no tendrá una instrucción de codificación final **USE-TYPE**.

NOTA – Las instrucciones de codificación finales siguientes no pueden coexistir con esta instrucción de codificación final porque está prohibida su aplicación a dicho tipo: **ANY-ATTRIBUTES**, **ANY-ELEMENT**, **BASE64**, **DECIMAL**, **EMBED-VALUES**, **LIST**, **TEXT**, **USE-NIL**, **USE-NUMBER**, **USE-ORDER**, **USE-QNAME**, **WHITESPACE**.

38.2.9 "TargetList" no incluirá información de cualificación.

38.3 Efecto en las codificaciones

38.3.1 Si el tipo elección no tiene una instrucción de codificación final **ATTRIBUTE** o **UNTAGGED** y no está codificada como el componente de un tipo secuencia-de o conjunto-de con una instrucción de codificación final **LIST**, es opcional del codificador añadir un atributo de identificación de tipo a la "AttributeList" del elemento envolvente (no obstante, véase 38.3.8).

NOTA – Si el tipo elección está codificado como una alternativa de una elección con una instrucción de codificación **USE-TYPE**, el atributo de identificación de tipo especificado por la instrucción de codificación **USE-UNION** puede utilizarse en lugar del atributo de identificación de tipo especificado por la instrucción de codificación **USE-TYPE** (véase 37.3.8).

38.3.2 Si cada posible codificación del valor abstracto codificado es idéntica a una de las codificaciones de un valor abstracto de una alternativa textualmente precedente, se añadirá un atributo de identificación de tipo.

NOTA – Esta subcláusula 38.3.2 suprime la opción del codificador de la subcláusula 38.3.1 y hace obligatorio el atributo de identificación de tipo. Las restricciones especificadas en 38.2.4 a 38.2.6 aseguran que esto sólo ocurra cuando el tipo elección se codifique como un elemento y no exista ambigüedad alguna debida a la posible existencia de identificadores idénticos.

38.3.3 Si el tipo elección tiene una instrucción de codificación final **ATTRIBUTE** o **UNTAGGED** o su tipo envolvente es un tipo secuencia-de o conjunto-de con una instrucción de codificación final **LIST**, no puede insertarse ningún atributo de identificación de tipo en ningún elemento. En el caso de los escenarios descritos en 38.2.4 a 38.2.6, no puede insertarse un atributo de identificación de tipo para identificar con precisión algunas de las alternativas de U o U2. Por lo tanto, los decodificadores se basan en las condiciones expuestas en 38.2.4 a 38.2.6 para determinar el valor abstracto que ha sido codificado.

NOTA – Estas reglas implican que, en ausencia de una identificación de tipo (o si la que existe es ambigua), es necesario que el decodificador intente realizar la decodificación a partir de la primera alternativa textual, después a partir de la siguiente, y así sucesivamente, aceptando la primera decodificación exitosa que se produzca (o diagnosticando un error si no se consigue ninguna decodificación exitosa).

38.3.4 El atributo de identificación de tipo será un ejemplar de la producción "Attribute" (véase 20.3.3) con un espacio de nombres cualificado "ControlAttributeName" (véase 20.3.5) cuyo "type" es del espacio de nombres de control (véase 16.9).

38.3.5 El valor del atributo de identificación de tipo será el identificador de la alternativa elegida, posiblemente modificada de conformidad con instrucciones de codificación finales **NAME** y **NAMESPACE**.

38.3.6 Todos los "Attribute", que en otro caso, estarían en la "AttributeList" del "ExtendedXMLChoiceValue", se añadirán a la "AttributeList" del elemento envolvente y el "ExtendedXMLChoiceValue" del tipo elección será sustituido por el "ExtendedXMLValue" del "ExtendedXMLChoiceValue".

38.3.7 El "ExtendedXMLValue" del tipo codificable mediante caracteres será una de las codificaciones que no contenga ningún rótulo XML.

NOTA – Esto puede limitar las opciones del codificador.

38.3.8 Si una alternativa del tipo elección tiene una instrucción de codificación final **NAME AS ""**, no se añadirá ningún atributo de identificación de tipo para dicha alternativa.

39 Instrucción de codificación **WHITESPACE**

39.1 Generalidades

39.1.1 "WhiteSpaceInstruction" se define como:

```
WhiteSpaceInstruction ::=  
    WHITESPACE  
    TargetList  
    WhiteSpaceAction
```

```
WhiteSpaceAction ::=  
    REPLACE  
    | COLLAPSE
```

39.1.2 La producción "TargetList" se define en 14.2.

39.1.3 Esta instrucción de codificación requiere que los decodificadores acepten opciones adicionales en la codificación del carácter SPACE (32) y en la utilización de "espacios blancos con escapes" anteriores y posteriores (véase 8.1.5) para codificaciones de cadena de caracteres.

39.2 Restricciones

39.2.1 Esta instrucción de codificación sólo puede asignarse a un tipo cadena de caracteres restringida que no contenga, o esté limitada a no contener, los caracteres siguientes:

- HORIZONTAL TABULATION (9) (tabulación horizontal);
- LINE FEED (10) (cambio de renglón);
- CARRIAGE RETURN (13) (retroceso de carro).

39.2.2 Si esta instrucción de codificación tiene la opción **COLLAPSE**, no se aplicará a un tipo cadena de caracteres restringida salvo que dicho tipo esté limitado a no tener espacios anteriores o posteriores o múltiples espacios adyacentes para cualquier valor abstracto.

NOTA – Es posible que algunas herramientas ASN.1 no puedan verificar estadísticamente si se cumple la restricción anterior para todos los valores abstractos, pero los codificadores conformes no pueden generar codificaciones en las que el "ExtendedXMLValue" viole esta restricción.

39.2.3 Un tipo con esta instrucción de codificación final tampoco tendrá ninguna de las instrucciones de codificación finales **ANY-ELEMENT** o **BASE64**.

NOTA – Las instrucciones de codificación finales siguientes no pueden existir conjuntamente con esta instrucción de codificación final porque su aplicación a dicho tipo está prohibida: **ANY-ATTRIBUTES**, **DECIMAL**, **EMBED-VALUES**, **LIST**, **TEXT**, **USE-NIL**, **USE-NUMBER**, **USE-ORDER**, **USE-QNAME**, **USE-TYPE**, **USE-UNION**.

39.2.4 "TargetList" no incluirá información de cualificación.

39.3 Efecto en las codificaciones

39.3.1 Si se utiliza la palabra clave **REPLACE**, es opcional del codificador sustituir cada SPACE (32) que exista en el valor abstracto por un único carácter que sea "espacio blanco con escapes" (véase 8.1.5).

39.3.2 Si se utiliza la palabra clave **COLLAPSE**, es opcional del codificador sustituir cada SPACE (32) por cualquier número de caracteres "espacio blanco con escapes". Además, es opcional del codificador añadir uno o más de dichos caracteres al comienzo y/o al final de la codificación "ExtendedXMLValue".

Sustitúyase la cláusula 10 actual y sus subcláusulas por lo siguiente (nótese que se han reenumerado como cláusula 40):

40 Valores de identificador de objeto que hacen referencia a las reglas de codificación

40.1 Las reglas de codificación especificadas en esta Recomendación | Norma Internacional pueden ser referenciadas y aplicadas cuando sea necesario especificar una representación inequívoca de una cadena de caracteres para los valores de un solo tipo ASN.1 identificado.

40.2 Los siguientes valores de identificador de objeto y descriptor de objeto se asignan para identificar las reglas de codificación especificadas en esta Recomendación | Norma Internacional:

For **BASIC-XER**:

```
{joint-iso-itu-t asn1 (1) xml-encoding (5) basic (0)}
"Basic XML encoding of a single ASN.1 type"
```

For **CXER**:

```
{joint-iso-itu-t asn1 (1) xml-encoding (5) canonical (1)}
"Canonical XML encoding of a single ASN.1 type"
```

For **EXTENDED-XER**:

```
{joint-iso-itu-t asn1 (1) xml-encoding (5) extended (2)}
"Extended XML encoding of a single ASN.1 type"
```

Insértese una nueva subcláusula 40.3 como se indica a continuación:

40.3 Los siguientes valores de identificador de objeto y descriptor de objeto se asignan a fin de identificar el espacio de nombres ASN.1 (véase 16.9):

```
asn1Namespace OBJECT IDENTIFIER ::=
    {joint-iso-itu-t asn1 (1) xml-encoding (5) extended (2)
     modules (0) support (1) }
"ASN.1 namespace for EXTENDED-XER support"
```

Sustitúyase el anexo A por el siguiente:

Anexo A

Ejemplos de codificaciones BASIC-XER y CXER

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

El presente anexo ilustra la utilización de las reglas de codificación XML básica y canónica especificadas en esta Recomendación | Norma Internacional, mostrando representaciones de marcaje XML de un registro de personal (hipotético) que se define mediante el ASN.1.

A.1 Descripción ASN.1 de la estructura del registro

La estructura del registro de personal hipotético se describe formalmente a continuación mediante la notación ASN.1 especificada en la Rec. UIT-T X.680 | ISO/CEI 8824-1. Este ejemplo es idéntico al definido en el anexo A de la Rec. UIT-T X.690 | ISO/CEI 8825-1.

```

PersonnelRecord ::= [APPLICATION 0] IMPLICIT SET {
    name                Name,
    title               [0] VisibleString,
    number              EmployeeNumber,
    dateOfHire          [1] Date,
    nameOfSpouse        [2] Name,
    children             [3] IMPLICIT
        SEQUENCE OF ChildInformation DEFAULT {} }
ChildInformation ::= SET
    { name                Name,
      dateOfBirth        [0] Date}
Name ::= [APPLICATION 1] IMPLICIT SEQUENCE
    {givenName          VisibleString,
     initial            VisibleString,
     familyName         VisibleString}
EmployeeNumber ::= [APPLICATION 2] IMPLICIT INTEGER
Date ::= [APPLICATION 3] IMPLICIT VisibleString -- YYYYMMDD

```

NOTA – En este ejemplo se utilizan rótulos sólo porque se consideró apropiado utilizar un ejemplo idéntico al que aparecía en una versión anterior de la Rec. UIT-T X.680 | ISO/CEI 8824-1. Dichos rótulos no tienen ningún efecto en las codificaciones BASIC-XER, CXER y EXTENDED-XER.

A.2 Descripción ASN.1 de un valor de registro

El valor del registro de personal correspondiente a John Smith se describe formalmente a continuación mediante la NOTación de valor ASN.1 básica:

```

{ name                {givenName "John", initial "P", familyName "Smith"},
  title               "Director",
  number              51,
  dateOfHire          "19710917",
  nameOfSpouse        {givenName "Mary", initial "T", familyName "Smith"},
  children             { {name {givenName "Ralph", initial "T", familyName "Smith"},
                        dateOfBirth "19571111"},
                       {name {givenName "Susan", initial "B", familyName "Jones"},
                        dateOfBirth "19590717"} } }

```

A.3 Representación XML básica de este valor de registro

La anterior representación del valor de registro (después de aplicar las reglas de codificación XML básica definidas en esta Recomendación | Norma Internacional) se muestra a continuación, suponiendo un prólogo vacío.

La longitud de esta codificación en BASIC-XER es de 653 octetos si se ignoran todos los "espacio blanco" ("white-space"). A efectos de comparación, el mismo valor de registro de personal (PersonnelRecord) codificado con la variante UNALIGNED de PER (véase la Rec. UIT-T X.690 | ISO/CEI 8825-1) es de 84 octetos, con la variante ALIGNED de PER es de 94 octetos, con BER (véase la Rec. UIT-T X.691 | ISO/CEI 8825-2) y la forma de longitud definida es de al menos 136 octetos, y con la BER y la forma de longitud indefinida es de al menos 161 octetos.

```

<PersonnelRecord>
  <name>
    <givenName>John</givenName>
    <initial>P</initial>
    <familyName>Smith</familyName>
  </name>
  <title>Director</title>
  <number>51</number>
  <dateOfHire>19710917</dateOfHire>
  <nameOfSpouse>
    <givenName>Mary</givenName>
    <initial>T</initial>
    <familyName>Smith</familyName>
  </nameOfSpouse>
  <children>
    <ChildInformation>
      <name>
        <givenName>Ralph</givenName>
        <initial>T</initial>
        <familyName>Smith</familyName>
      </name>
      <dateOfBirth>19571111</dateOfBirth>
    </ChildInformation>
    <ChildInformation>
      <name>
        <givenName>Susan</givenName>
        <initial>B</initial>
        <familyName>Jones</familyName>
      </name>
      <dateOfBirth>19590717</dateOfBirth>
    </ChildInformation>
  </children>
</PersonnelRecord>

```

A.4 Representación XML canónica de este valor de registro

La representación del anterior valor de registro (después de aplicar las reglas de codificación XML canónica definidas en esta Recomendación | Norma Internacional) se muestra a continuación:

```

<PersonnelRecord><name><givenName>John</givenName><initial>P</initial><familyName>Smith</familyName></name><number>51</number><title>Director</title><dateOfHire>19710917</dateOfHire><nameOfSpouse><givenName>Mary</givenName><initial>T</initial><familyName>Smith</familyName></nameOfSpouse><children><ChildInformation><name><givenName>Ralph</givenName><initial>T</initial><familyName>Smith</familyName></name><dateOfBirth>19571111</dateOfBirth></ChildInformation><ChildInformation><name><givenName>Susan</givenName><initial>B</initial><familyName>Jones</familyName></name><dateOfBirth>19590717</dateOfBirth></ChildInformation></children></PersonnelRecord>

```

Insértese un nuevo anexo B tal como se indica a continuación:

Anexo B

Codificaciones de contenido XML parcial y determinística

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

B.1 Contenido XML parcial

NOTA – Este anexo describe la validez cuando se utiliza **MODIFIED-ENCODINGS**.

B.1.1 Las subcláusulas siguientes describen la construcción del contenido de elemento XML parcial. Esta cláusula describe qué contenido de elemento XML parcial se produce como parte de codificaciones, y en B.2 se especifican restricciones que afectan al contenido del elemento XML parcial y que son necesarias para cumplir los requisitos de 10.2.11. Si una especificación ASN.1 con instrucciones de codificación XER no viola estas restricciones, es una especificación legal, pudiendo las herramientas verificar fácilmente su legalidad. Aunque se violen las restricciones, es posible que la especificación no viole los requisitos normativos señalados en 10.2.11, pero puede resultar difícil para las herramientas verificar que ese es el caso.

NOTA – Las restricciones se diseñan para asegurar que un decodificador puede recuperar fácil e inequívocamente los valores abstractos que ha utilizado un codificador en la codificación.

B.1.2 Un contenido de elemento XML parcial se compone de una combinación de elementos XML proporcionados por un **[ELEMENT] SEQUENCE, SET, SEQUENCE OF, SET OF** o **CHOICE**, y de otro contenido de elemento XML parcial proporcionado por y **[UNTAGGED] SEQUENCE, SET, SEQUENCE OF, SET OF** o **CHOICE**.

NOTA – La delimitación del contenido de elemento XML parcial situado en un contenido de elemento XML parcial mayor no es visible en la codificación, pero puede determinarse a partir del esquema ASN.1 y de las restricciones aplicables a los nombres de elementos.

B.1.3 Un contenido de elemento XML parcial puede constar de:

- a) un único elemento XML; o
- b) un grupo de concatenación, compuesto por la concatenación ordenada de cero, uno o más contenidos de elemento XML parcial en los que algún contenido de elemento XML parcial puede no existir en una instancia de codificación (que represente la ausencia de un valor abstracto opcional); o

NOTA – Una codificación de un tipo **[UNTAGGED] SEQUENCE** o **SET** producirá, en general, un grupo de concatenación.

- c) un grupo de repetición, compuesto por la repetición (de forma ilimitada o restringida) del contenido de elemento XML parcial (que se denomina componente repetido) producido a partir del componente de una **SEQUENCE OF** o **SET OF**; o

NOTA – Una codificación de un tipo **[UNTAGGED] SEQUENCE OF** o **SET OF** producirá, en general, un grupo de repetición.

- d) un grupo de alternativas, compuesto por la presencia de un contenido de un único elemento XML parcial, elegido de entre un conjunto de contenidos de elementos XML parcial alternativos (de los que sólo uno está presente en una codificación).

NOTA – Una codificación de tipo **CHOICE** genera un grupo de alternativas. Cada alternativa del tipo **CHOICE** produce uno de los contenidos de elemento XML parcial alternativos para dicho tipo **CHOICE**.

B.2 Restricciones recomendadas de codificaciones que producen un contenido de elemento XML parcial

B.2.1 Solamente a los efectos de esta cláusula, cualquier grupo de repetición se trata como si fuera opcional, es decir, puede tener cero repeticiones.

NOTA – La restricción de que un grupo de repetición se trate como si fuera opcional no es estrictamente necesaria si existen restricciones que requieran al menos una repetición del correspondiente tipo ASN.1, pero se utiliza por simplicidad.

B.2.2 Solamente a los efectos de esta cláusula, el requisito de que los nombres de los elementos sean diferentes debe interpretarse como se indica a continuación:

- a) todas las comparaciones se realizan después de la aplicación de cualquier instrucción de codificación **NAME** y **NAMESPACE** sobre el tipo que generó el nombre;
- b) los nombres que son nombres de espacio de nombres cualificados son diferentes de los nombres no cualificados;

- c) los nombres de espacio de nombres cualificados son diferentes si y solo si difieren en su nombre sin prefijo, en su nombre de espacio de nombres, o en ambos.

B.2.3 Para cualquier contenido de elemento XML parcial resultante, puede existir alguna ambigüedad (y por tanto, una posible violación de lo indicado en 10.2.11) si las condiciones especificadas en esta subcláusula B.2 no se cumplen para todas las posibles elecciones de alternativas de un grupo de alternativas, para todas las posibles opciones que puedan ejercerse sobre un grupo de concatenación, para todas las posibles repeticiones de un grupo repetido y para todas las posibles ordenaciones de las codificaciones de los componentes de un conjunto.

NOTE – En la lectura e implementación de las cláusulas siguientes, es muy importante el texto anterior que dice "para todas las posibles". Los implementadores de herramientas que determinen lo que es y lo que no es una especificación inequívoca deberán analizar todas las combinaciones posibles de elecciones, opciones, repeticiones y ordenaciones.

B.2.4 (Requisito de delimitación) No deben existir dos contenidos de elementos XML parciales adyacentes con el mismo nombre de elemento para el primer elemento del segundo contenido de elemento XML parcial y para el último elemento del primer contenido de elemento XML parcial, salvo que el primer contenido del elemento XML parcial esté autodelimitado.

EJEMPLO 1: El contenido de elemento XML parcial producido por una [UNTAGGED] SEQUENCE está autodelimitado si no termina con un elemento OPTIONAL.

EJEMPLO 2: El contenido de elemento XML parcial producido por una [UNTAGGED] SEQUENCE OF está autodelimitado si tiene un número fijo de iteraciones, que también estén autodelimitadas. Esto significa, entre otras cosas, que SEQUENCE OF [UNTAGGED] SEQUENCE OF INTEGER es ambigua y viola lo indicado en 10.2.11, salvo que el número de repeticiones de la segunda SEQUENCE OF sea fijo.

EJEMPLO 3: El contenido de elemento XML parcial producido por un [UNTAGGED] SET nunca está autodelimitado si tiene elementos opcionales.

B.2.5 (Requisito de determinación alternativa) Todos los primeros elementos XML del contenido del elemento XML parcial alternativo en un grupo de alternativas deben tener nombres de elementos diferentes.

NOTA – El texto anterior ignora la posible utilización de USE-TYPE y USE-UNION, que están fuera del ámbito de este anexo.

EJEMPLO 4: Una codificación:

```
BadExample1 ::= CHOICE {
  -- First alternative partial XML element content
  alt1 [UNTAGGED] SEQUENCE {
    name      UTF8String,
    zip-code  UTF8String },
  alt2 [UNTAGGED] SEQUENCE {
    name      UTF8String,
    post-code UTF8String } }
```

no es, de hecho, una codificación EXTENDED-XER ambigua (para un decodificador humano), pero viola el requisito anterior así como lo indicado en 10.2.11. Constituye una utilización ilegal de instrucciones de codificación.

B.2.6 (Requisito de determinación de opciones). Los nombres de elementos XML del primer elemento XML de todos los contenidos de elemento XML parcial opcionales consecutivos y el del siguiente contenido de elemento XML parcial obligatorio, deben ser diferentes.

NOTA – Esto significa, entre otras cosas, que cualquier contenido de elemento XML parcial opcional al final de un grupo que se repita y cualquier contenido de elemento XML parcial opcional en su inicio, deben tener nombres de elementos XML diferentes, salvo que el número de repeticiones esté limitado a un máximo de 1. Si todo el contenido de elemento XML parcial del grupo que se repite es opcional, los nombres de sus elementos XML deben ser diferentes.

EJEMPLO 5: Una codificación de:

```
BadExample2 ::= SEQUENCE OF {
  [UNTAGGED] SEQUENCE {
    first  [UNTAGGED] CommonInitialParms,
    second MainInformation,
    third  [UNTAGGED] CommonEndParms } }

where
CommonInitialParms ::= SEQUENCE { date GeneralizedTime OPTIONAL,
                                  married BOOLEAN }
CommonEndParms ::= SEQUENCE { name UTF8String,
                               date GeneralizedTime OPTIONAL }
```

viola el requisito de determinación de opciones así como lo indicado en 10.2.11. Constituye una utilización ilegal de instrucciones de codificación.

B.2.7 (Requisito de determinación del cómputo de repetición) Todos los grupos de repetición con un número no fijo de repeticiones que deben ir seguidos de un contenido de elemento XML parcial cuyo primer elemento XML tiene un nombre diferente del nombre del primer elemento XML del contenido de elemento XML parcial que se repite.

EJEMPLO 6: Una codificación de:

```
BadExample3 ::= SEQUENCE {
    required-items [UNTAGGED] SEQUENCE OF Book,
    optional-items [UNTAGGED] SEQUENCE OF Book }
```

viola el requisito de determinación del cómputo de repetición así como lo indicado en 10.2.11. Constituye una utilización ilegal de instrucciones de codificación. Alternativamente:

```
GoodExample1 ::= SEQUENCE {
    required-items [UNTAGGED] SEQUENCE OF required-books Book ,
    optional-items [UNTAGGED] SEQUENCE OF optional-books Book }
```

sería una utilización legal de instrucciones de codificación.

B.2.8 (Requisito de determinación de componentes del conjunto) El primer elemento XML del contenido XML parcial de los componentes de un grupo de concatenación que sea una codificación de un tipo conjunto (*set*) debe tener un nombre de elemento XML diferente del nombre del primer elemento XML del contenido XML parcial de todos los restantes componentes.

EJEMPLO 7: Una codificación de:

```
BadExample4 ::= SET {
    uk-mailing [UNTAGGED] SEQUENCE {name UTF8String, post-code UTF8String}
    us-mailing [UNTAGGED] SEQUENCE {name UTF8String, zip-code UTF8String}}
```

viola el requisito de determinación de componentes así como lo indicado en 10.2.11. Constituye una utilización ilegal de instrucciones de codificación. De forma alternativa:

```
GoodExample2 ::= SET {
    uk-mailing [UNTAGGED] SEQUENCE {uk-name UTF8String, post-code UTF8String}
    us-mailing [UNTAGGED] SEQUENCE {us-name UTF8String, zip-code UTF8String}}
```

sería una utilización legal de instrucciones de codificación.

Insértese un nuevo anexo C tal como se indica a continuación:

Anexo C

Ejemplos de codificaciones EXTENDED-XER utilizando instrucciones de codificación XER

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

C.1 Introducción

C.1.1 Este anexo presenta información didáctica (tutorial) y ejemplos de la aplicación de instrucciones de codificación XER.

NOTA – En todos los ejemplos ASN.1 de este anexo se asume un entorno de **AUTOMATIC TAGS**.

C.1.2 Normalmente sólo es necesario asignar las instrucciones de codificación a una especificación ASN.1 si el diseñador tiene el requisito de concordancia entre la forma real de la codificación XML y la definida mediante otras especificaciones de esquema, o la que es previsible mediante otras herramientas XML. En cualquier otro caso, puede utilizarse sólo ASN.1 (con codificación BASIC-XER o CXER).

C.1.3 Si se utiliza ASN.1 como la notación de definición de esquema, la utilización adicional de instrucciones de codificación proporcionará en general codificaciones XML más compactas que utilizando exclusivamente ASN.1, pero las codificaciones resultan aún mucho más prolijas que utilizando ASN.1 con PER.

NOTA – Los ejemplos (y los identificadores de nombres y de tipos empleados) están diseñados para ilustrar las características de EXTENDED-XER, y en general, no representan especificaciones reales.

C.1.4 Las instrucciones de codificación XER se clasifican, en general, en dos categorías.

C.1.5 La primera categoría está formada por las instrucciones de codificación que es probable que sean útiles en el diseño de la forma de un documento XML. Éstas están generalmente permitidas, incluso cuando no existe **GLOBAL-DEFAULTS MODIFIED-ENCODINGS**. La dos más útiles son **ATTRIBUTE** y **LIST**; en C2 se incluyen ejemplos sencillos de su utilización.

C.1.6 La segunda categoría está formada por las instrucciones de codificación diseñadas para permitir la correspondencia con los esquemas W3C XML especificados en la Rec. UIT-T X.694 | ISO/CEI 8825-5. Éstas requieren en general la presencia de **GLOBAL-DEFAULTS MODIFIED-ENCODINGS** en una sección de control de codificación, aunque ese caso no se muestra en los ejemplos. En éstos se asume que cualquier referencia de tipo que comience por "XSD." es importada del anexo A de la Rec. UIT-T X.694 | ISO/CEI 8825-5. En C.3 se incluyen ejemplos sobre su utilización. **Dichos ejemplos no son módulos ASN.1 completos, ni documentos XML completos**: generalmente se omiten los encabezamientos de los módulos; y se supone que cualquier atributo XML que comience por "asn1:" es un atributo de control que utiliza el espacio de nombres ASN.1 para el atributo de control, asumiéndose que ya se ha declarado el prefijo "asn1". (En la práctica, si la codificación se deriva del esquema W3C XML, es más probable que se utilice el prefijo "xsi", con el espacio de nombres XSI.)

C.1.7 En casi todos los casos, y para una mayor claridad, se utilizan instrucciones de codificación con prefijo, aunque en una especificación real se conseguirá más brevedad (y una separación más clara entre la definición de la sintaxis abstracta y los elementos de codificación) utilizando una sección de control de codificación.

C.2 Ejemplos sencillos

C.2.1 Una tarjeta de béisbol

```

BBCard ::= SEQUENCE {
    name      [ATTRIBUTE] IA5String,
    team      [ATTRIBUTE] IA5String,
    age       INTEGER,
    position  IA5String,
    handedness ENUMERATED {
        left-handed,
        right-handed,
        ambidextrous },
    batting-average REAL }

```

Ignorando las instrucciones de codificación (BASIC-XER) se tendría:

```

<BBCard>
  <name>Jorge Posada</name>
  <team>New York Yankees</team>
  <age>29</age>
  <position>C</position>
  <handedness><right-handed/></handedness>
  <batting-average>0.277</batting-average>
</BBCard>

```

La codificación EXTENDED-XER (con MODIFIED-ENCODINGS) del mismo valor es:

```

<BBCard name = "Jorge Posada" team = "New York Yankees" >
  <age>29</age>
  <position>C</position>
  <handedness>right-handed</handedness>
  <batting-average>0.277</batting-average>
</BBCard>

```

C.2.2 Un empleado

```

Employee ::= [NAME AS UNCAPITALIZED] SEQUENCE {
    id          [ATTRIBUTE] INTEGER(0..MAX),
    recruited   XSD.Date,
    salaries    [LIST] SEQUENCE
                OF salary REAL }

```

Ignorando las instrucciones de codificación (BASIC-XER), se tendría:

```
<Employee>
  <id>239</id>
  <recruited>27-11-2002</recruited>
  <salaries>
    <salary>29876</salary>
    <salary>54375</salary>
    <salary>98435</salary>
  </salaries>
</Employee>
```

La codificación EXTENDED-XER del mismo valor es:

```
<employee id = "239">
  <recruited>27-11-2002</recruited>
  <salaries>29876 54375 98435</salaries>
</employee>
```

Utilizando una sección de control de codificación XER, se tendría:

```
Employee ::= SEQUENCE {
  id          INTEGER(0..MAX),
  recruited   Date,
  salaries    SEQUENCE
              OF salary REAL }

ENCODING-CONTROL XER
  NAME Employee AS UNCAPITALIZED
  ATTRIBUTE Employee.id
  LIST Employee.salaries
```

C.3 Ejemplos más complejos

C.3.1 Utilización de la unión de dos tipos sencillos

```
Int-or-boolean ::= [USE-UNION] CHOICE {
  int          INTEGER,
  boolean      BOOLEAN }
```

Las codificaciones podrían ser:

```
<Int-or-boolean><int>39</int></Int-or-boolean>          -- BASIC-XER
<Int-or-boolean><boolean><true/></boolean></Int-or-boolean> -- BASIC-XER
<Int-or-boolean>39</Int-or-boolean>                   -- EXTENDED-XER
<Int-or-boolean>true</Int-or-boolean>                  -- EXTENDED-XER
```

C.3.2 Utilización de un atributo de identificación de tipo

```
Int-or-boolean ::= [USE-TYPE] CHOICE {
  int          INTEGER,
  boolean      BOOLEAN }
```

Las codificaciones podrían ser:

```
<Int-or-boolean><int>39</int></Int-or-boolean>          -- BASIC-XER
<Int-or-boolean><boolean><true/></boolean></Int-or-boolean> -- BASIC-XER
<Int-or-boolean asn1:type="int">39</Int-or-boolean>     -- EXTENDED-XER
<Int-or-boolean asn1:type="boolean">true</Int-or-boolean> -- EXTENDED-XER
```

C.3.3 Utilización de valores de enumeración

```
PrimesUnder30 ::= [USE-NUMBER] ENUMERATED {
  int2(2), int3(3), int5(5), int7(7), int11(11), int13(13),
  int17(17), int19(19), int23(23), int29(29)}
InputValues ::= [ATTRIBUTE] [LIST] SEQUENCE OF PrimesUnder30
PrimeProducts ::= SEQUENCE {
  input  InputValues,
  output [ATTRIBUTE] [DECIMAL] REAL}
```

Las codificaciones podrían ser:

```
<PrimeProducts>
  <input><int2/><int7/><int17/><int23/><int29/><int3/></input>
  <output>476338.00</output>
</PrimeProducts>                                -- BASIC-XER
<PrimeProducts input="2 7 17 23 29 3" output="476338.00"/>
                                                    -- EXTENDED-XER
```

C.3.4 Utilización de una codificación vacía para un valor por defecto

```
Responses ::= ENUMERATED {ringing, engaged, number-not-known }
CallDetails ::= [DEFAULT-FOR-EMPTY number-not-known] SEQUENCE {
  number      [ATTRIBUTE] NumericString,
  response    Response }
```

Las codificaciones podrían ser:

```
<CallDetails>
  <number>0164593746</number>
  <response><number-not-known/></response>
</CallDetails>                                -- BASIC-XER
<CallDetails number="0164593746"/>           -- EXTENDED-XER
```

C.3.5 Utilización de valores integrados para la notificación un pago pendiente

```
Notification ::= SEQUENCE {
  text [EMBED-VALUES] SEQUENCE OF UTF8String,
  account INTEGER,
  amount-due INTEGER,
  payable-by XSD:Date } (CONSTRAINED BY { /* Shall conform to ITU-T Rec. X.693 |
ISO/IEC 8825-4, 25.2 */)
```

Un valor en NOTación de valor ASN.1 básica podría ser:

```
firstNotification Notification ::= {
  text {"Please note the following details:",
    "(your business account)",
    "This is in excess of your normal monthly allowance",
    "or earlier"},
  account 568903,
  amount-due 536,
  payable-by "27-08-2003" }
```

La codificación EXTENDED-XER sería:

```
<Notification>
  Please note the following details:
  <account>568903</account>
  (your business account)
  <amount-due>536</amount-due>
  This is in excess of your normal monthly allowance
  <payable-by>27-08-2003</payable-by>
  or earlier
</Notification>
```


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	Redes de cable y transmisión de programas radiofónicos y televisivos, 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
Serie X	Redes de datos y comunicación entre sistemas abiertos
Serie Y	Infraestructura mundial de la información, aspectos del protocolo Internet y Redes de la próxima generación
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación