

**Reemplazada por una versión más reciente**



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

**UIT-T**

**X.690**

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

(07/94)

**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:  
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**

Reemplazada por una versión más reciente

(Anteriormente «Recomendación del CCITT»)

---

# Reemplazada por una versión más reciente

## 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. En el UIT-T, que es la entidad que establece normas mundiales (Recomendaciones) sobre las telecomunicaciones, participan unos 179 países miembros, 84 empresas de explotación de telecomunicaciones, 145 organizaciones científicas e industriales y 38 organizaciones internacionales.

Las Recomendaciones las aprueban los Miembros del UIT-T de acuerdo con el procedimiento establecido en la Resolución N.º 1 de la CMNT (Helsinki, 1993). Adicionalmente, la Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), que se celebra cada cuatro años, aprueba las Recomendaciones que para ello se le sometan y establece el programa de estudios para el periodo siguiente.

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. El texto de la Recomendación UIT-T X.690 se aprobó el 1 de julio de 1994. Su texto se publica también, en forma idéntica, como Norma Internacional ISO/CEI 8825-1.

---

## NOTA

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

© UIT 1996

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

# Reemplazada por una versión más reciente

RECOMENDACIONES UIT-T DE LA SERIE X

REDES DE DATOS Y COMUNICACIÓN ENTRE SISTEMAS ABIERTOS

(Febrero de 1994)

## ORGANIZACIÓN DE LAS RECOMENDACIONES DE LA SERIE X

Dominio	Recomendaciones
<b>REDES PÚBLICAS DE DATOS</b>	
Servicios y facilidades	X.1-X.19
Interfaces	X.20-X.49
Transmisión, señalización y conmutación	X.50-X.89
Aspectos de redes	X.90-X.149
Mantenimiento	X.150-X.179
Disposiciones administrativas	X.180-X.199
<b>INTERCONEXIÓN DE SISTEMAS ABIERTOS</b>	
Modelo y notación	X.200-X.209
Definiciones de los servicios	X.210-X.219
Especificaciones de los protocolos en modo conexión	X.220-X.229
Especificaciones de los protocolos en modo sin conexión	X.230-X.239
Formularios para enunciados de conformidad de implementación de protocolo	X.240-X.259
Identificación de protocolos	X.260-X.269
Protocolos de seguridad	X.270-X.279
Objetos gestionados de capa	X.280-X.289
Pruebas de conformidad	X.290-X.299
<b>INTERFUNCIONAMIENTO ENTRE REDES</b>	
Generalidades	X.300-X.349
Sistemas móviles de transmisión de datos	X.350-X.369
Gestión	X.370-X.399
<b>SISTEMAS DE TRATAMIENTO DE MENSAJES</b>	X.400-X.499
<b>DIRECTORIO</b>	X.500-X.599
<b>GESTIÓN DE REDES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS Y ASPECTOS DE SISTEMAS</b>	
Gestión de redes	X.600-X.649
Denominación, direccionamiento y registro	X.650-X.679
Notación de sintaxis abstracta uno	X.680-X.699
<b>GESTIÓN DE INTERCONEXIÓN DE SISTEMAS ABIERTOS</b>	X.700-X.799
<b>SEGURIDAD</b>	X.800-X.849
<b>APLICACIONES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS</b>	
Cometimiento, concurrencia y recuperación	X.850-X.859
Tratamiento de transacciones	X.860-X.879
Operaciones a distancia	X.880-X.899
<b>TRATAMIENTO ABIERTO DISTRIBUIDO</b>	X.900-X.999



# Reemplazada por una versión más reciente

## ÍNDICE

*Página*

1	Alcance.....	1
2.	Referencias normativas .....	1
	2.1 Recomendaciones   Normas Internacionales idénticas.....	1
	2.2 Referencias adicionales.....	2
3	Definiciones .....	2
4	Abreviaturas .....	3
5	Notación .....	3
6	Convenios.....	3
7	Conformidad .....	3
8	Reglas de codificación básica .....	3
	8.1 Reglas generales de codificación .....	3
	8.2 Codificación de un valor booleano (boolean) .....	7
	8.3 Codificación de un valor entero (integer) .....	8
	8.4 Codificación de un valor enumerado (enumerated).....	8
	8.5 Codificación de un valor real (real) .....	8
	8.6 Codificación de un valor de cadena de bits (bitstring).....	10
	8.7 Codificación de un valor de cadena de octetos (octetstring).....	11
	8.8 Codificación de un valor nulo (null) .....	11
	8.9 Codificación de un valor de secuencia (sequence) .....	12
	8.10 Codificación de un valor de secuencia de (sequence-of) .....	12
	8.11 Codificación de un valor de conjunto (set) .....	12
	8.12 Codificación de un valor de conjunto de (set-of).....	12
	8.13 Codificación de un valor de elección (choice).....	13
	8.14 Codificación de un valor rotulado (tagged) .....	13
	8.15 Codificación de un tipo abierto.....	13
	8.16 Codificación de un valor de ejemplar de.....	14
	8.17 Codificación de un valor del tipo incrustado .....	14
	8.18 Codificación de un valor del tipo externo .....	15
	8.19 Codificación de un valor de identificador de objeto (object identifier) .....	16
	8.20 Codificación de valores del tipo de cadena de caracteres restringida (restricted character string) .....	17
	8.21 Codificación de valores del tipo de cadena de caracteres no restringida (unrestricted character string type) .....	19
9	Reglas de codificación canónica .....	20
	9.1 Formas de longitud .....	20
	9.2 Formas de codificación de cadenas.....	20
	9.3 Componentes de conjunto (set).....	20
10	Reglas de codificación distinguida.....	21
	10.1 Formas de longitud .....	21
	10.2 Formas de codificación de cadenas.....	21
	10.3 Componentes de conjunto.....	21

# Reemplazada por una versión más reciente

*Página*

11	Restricciones de las reglas de codificación básica empleadas por las reglas de codificación canónica y de codificación distinguida .....	21
11.1	Valores booleanos .....	21
11.2	Bit no utilizados .....	21
11.3	Valores reales .....	21
11.4	Valores de cadena general (GeneralString).....	22
11.5	Componentes de conjunto y de secuencia con valor por defecto.....	22
11.6	Componentes de conjunto de .....	22
11.7	Tiempo generalizado (GeneralizedTime).....	22
12	Utilización de BER, CER y DER en la definición de sintaxis abstractas.....	23
Anexo A	– Ejemplos de codificaciones.....	24
A.1	Descripción ASN.1 de la estructura del registro .....	24
A.2	Descripción ASN.1 de un valor del registro .....	24
A.3	Representación de este valor del registro.....	24
Anexo B	– Asignación de valores de identificador de objeto .....	26
Anexo C	– Ilustración de la codificación de valores reales .....	27
Anexo D	– Utilización de las DER y las CER en la autenticación de origen de los datos .....	29
D.1	Problema que ha de resolverse.....	29
D.2	Planteamiento de una solución.....	30
D.3	Optimización de la realización.....	30

# Reemplazada por una versión más reciente

## Resumen

La presente Recomendación | Norma Internacional define un conjunto de reglas de codificación básica que se pueden aplicar a valores de tipos definidos utilizando la notación ASN.1. La aplicación de estas reglas de codificación produce una sintaxis de transferencia para esos valores. En la especificación de estas reglas de codificación está implícito que se utilizan también para la decodificación. Esta Recomendación | Norma Internacional define también un conjunto de reglas de codificación distinguida y un conjunto de reglas de codificación canónica que proporcionan constricciones de las reglas de codificación básica. La diferencia esencial entre ellas es que las reglas de codificación distinguida utilizan la forma de longitud definida de las codificaciones, mientras que las reglas de codificación canónica utilizan la forma de longitud indefinida. Las reglas de codificación distinguida son más adecuadas para los valores codificados pequeños, mientras que las reglas de codificación canónica son más adecuadas para los grandes. En la especificación de estas reglas de codificación está implícito que se utilizan también para la decodificación.

# Reemplazada por una versión más reciente

## Introducción

Las Rec. UIT-T X.680 | ISO/CEI 8824-1, UIT-T X.681 | ISO/CEI 8824-2, UIT-T X.682 | ISO/CEI 8824-3, y UIT-T X.683 | ISO/CEI 8824-4 (notación de sintaxis abstracta uno o ASN.1) especifican una notación para la definición de sintaxis abstractas que permite que las normas de la capa de aplicación definan los tipos de información que necesitan transferir utilizando el servicio de presentación. Contiene también una notación para la especificación de valores de un tipo definido.

La presente Recomendación | Norma Internacional define reglas de codificación que se pueden aplicar a valores de tipos definidos utilizando la notación ASN.1. La aplicación de estas reglas de codificación produce una sintaxis de transferencia para tales valores. En la especificación de estas reglas de codificación está implícito que las mismas se utilizan también para la decodificación.

Puede haber más de un conjunto de reglas de codificación que se puede aplicar a valores de tipos que se definen utilizando la notación ASN.1. La presente Recomendación | Norma Internacional define tres conjuntos de reglas de codificación, denominados **reglas de codificación básica**, **reglas de codificación canónica** y **reglas de codificación distinguida**. Mientras las reglas de codificación básica ofrecen al emisor de una codificación diversas opciones sobre cómo se pueden codificar los valores de datos, las reglas de codificación canónica y distinguida seleccionan solamente una codificación entre las permitidas por las reglas de codificación básica, eliminando así todas las opciones del emisor. Las reglas de codificación canónica y distinguida difieren entre sí en el conjunto de restricciones que imponen a las reglas de codificación básica.

Las reglas de codificación distinguida son más adecuadas que las reglas de codificación canónica si el valor codificado es suficientemente pequeño para que quepa en la memoria disponible y es necesario saltar rápidamente varios valores anidados. Las reglas de codificación canónica son más adecuadas que las reglas de codificación distinguida si es necesario codificar valores tan grandes que no caben fácilmente en la memoria disponible o es necesario codificar y transmitir una parte de un valor antes de que todo el valor esté disponible. Las reglas de codificación básica son más adecuadas que las reglas de codificación canónica o distinguida si la codificación contiene un valor de conjunto o un valor de conjunto de y no se necesitan las restricciones que imponen las reglas de codificación canónica o distinguida. Esto se debe a la memoria y tara de la unidad central de procesamiento que estas últimas reglas de codificación exigen para garantizar que los valores de conjunto y los valores de conjunto de sólo tienen una codificación posible.

El Anexo A proporciona un ejemplo de la aplicación de las reglas de codificación básica. No forma parte de esta Recomendación | Norma Internacional.

El Anexo B resume la asignación de valores de identificadores de objetos hecha en la presente Recomendación | Norma Internacional. No forma parte de esta Recomendación | Norma Internacional.

El Anexo C contiene ejemplos de la aplicación de las reglas de codificación básica para codificar valores reales. No forma parte de esta Recomendación | Norma Internacional.

El Anexo D ofrece una descripción didáctica sobre la utilización de las reglas de codificación distinguida para proporcionar un servicio de integridad para comunicaciones de OSI. No forma parte de esta Recomendación | Norma Internacional.

## NORMA INTERNACIONAL

## RECOMENDACIÓN UIT-T

**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**

**1 Alcance**

La presente Recomendación | Norma Internacional especifica un conjunto de reglas de codificación básica que se puede utilizar para obtener la especificación de una sintaxis de transferencia para valores de tipos definidos utilizando la notación especificada en las Rec. UIT-T X.680 (1994) | ISO/CEI 8824-1:1995, UIT-T X.681 (1994) | ISO/CEI 8824-2:1995, UIT-T X.682 (1994) | ISO/CEI 8824-3:1995 y UIT-T X.683 (1994) | ISO/CEI 8824-4:1995, denominadas colectivamente notación de sintaxis abstracta uno o ASN.1. Estas reglas de codificación básica se deben aplicar también para la decodificación de una sintaxis de transferencia con miras a identificar los valores de datos que se transfieren. Especifica también un conjunto de reglas de codificación canónica y distinguida que restringe la codificación de valores a sólo una de las alternativas proporcionadas por las reglas de codificación básica.

Estas reglas de codificación son utilizadas en el momento de la comunicación (por el proveedor del servicio de presentación cuando lo requiere un contexto de presentación).

**2 Referencias normativas**

Las siguientes Recomendaciones y Normas Internacionales contienen disposiciones que, mediante la referencia hecha en este texto, constituyen disposiciones de la presente Recomendación | Norma Internacional. En el momento de la publicación las ediciones indicadas eran válidas. Todas las Recomendaciones y Normas son objeto de revisiones, por lo que se preconiza que los participantes en acuerdos basados en la presente Recomendación | Norma Internacional investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y Normas enumeradas a continuación. Los miembros de la CEI y de la ISO mantienen registros de las Normas Internacionales actualmente válidas. La Oficina de Normalización de las Telecomunicaciones de la UIT mantiene una lista de las Recomendaciones UIT-T actualmente válidas.

**2.1 Recomendaciones | Normas Internacionales idénticas**

- Recomendación UIT-T X.200 (1994) | ISO/CEI 7498-1:1994, *Tecnología de la información – Interconexión de sistemas abiertos – Modelo de referencia básico: El modelo básico.*
- Recomendación UIT-T X.226 (1994) | ISO/CEI 8823-1:1994, *Tecnología de la información – Interconexión de sistemas abiertos – Protocolo de presentación con conexión: Especificación del protocolo.*
- Recomendación UIT-T X.680 (1994) | ISO/CEI 8824-1:1995, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de la notación básica.*
- Recomendación UIT-T X.681 (1994) | ISO/CEI 8824-2:1995, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de objetos de información.*
- Recomendación UIT-T X.682 (1994) | ISO/CEI 8824-3:1995, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de constricciones.*
- Recomendación UIT-T X.683 (1994) | ISO/CEI 8824-4:1995, *Tecnología de la información – Notación de sintaxis abstracta uno: Parametrización de las especificaciones de la notación de sintaxis abstracta uno.*

## 2.2 Referencias adicionales

- ISO *International Register of Coded Character Sets to be used with Escape Sequence*.
- ISO/CEI 2022:1994, *Information processing – ISO 7-bit and 8-bit coded character sets – Code extension techniques*.
- ISO 6093:1985, *Information processing – Representation of numerical values in character strings for information interchange*.
- ISO 6429:1992, *Information technology – Control functions for coded character sets*.
- Recomendación X.208 del CCITT (1988), *Especificación de la notación de sintaxis abstracta uno (NSA.1)*.
- ISO/CEI 8824-1 a 8824-4:1990, *Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1)*.
- ISO/CEI 10646-1:1993, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*.

## 3 Definiciones

A los efectos de la presente Recomendación | Norma Internacional son aplicables las definiciones de ISO 7498, las de la Rec. UIT-T X.680 | ISO/CEI 8824-1 y las que se indican a continuación.

**3.1 conformidad dinámica:** Declaración de la necesidad de que una realización se amolde al comportamiento prescrito por esta Recomendación | Norma Internacional en una comunicación.

**3.2 conformidad estática:** Declaración de la necesidad de que una realización admita un conjunto válido de características entre las definidas por esta Recomendación | Norma Internacional.

**3.3 valor de datos:** Información especificada como el valor de un tipo; el tipo y el valor se definen utilizando la notación ASN.1.

**3.4 codificación (de un valor de datos):** Secuencia completa de octetos utilizada para representar el valor de datos.

**3.5 octetos de identificador:** Parte de la codificación de un valor de datos que se utiliza para identificar el tipo del valor.

NOTA – En algunas Recomendaciones UIT-T se utiliza el término «elemento de datos» para designar esta secuencia de octetos, pero en esta Recomendación | Norma Internacional no se utiliza, ya que en otras Recomendaciones | Normas Internacionales el mismo significa «valor de datos».

**3.6 octetos de longitud:** Parte de la codificación de un valor de datos que sigue a los octetos de identificador y que se utiliza para determinar el final de la codificación.

**3.7 octetos de contenido:** Parte de la codificación de un valor de datos que representa un valor determinado, para diferenciarlo de otros valores del mismo tipo.

**3.8 octetos de fin de contenido:** Parte de la codificación de un valor de datos que aparece al final de ésta y que se utiliza para determinar el final de la codificación.

NOTA – No todas las codificaciones requieren octetos de fin de contenido.

**3.9 codificación primitiva:** Codificación de un valor de datos en la que los octetos de contenido representan directamente el valor.

**3.10 codificación construida:** Codificación de un valor de datos en la que los octetos de contenido son la codificación completa de otro u otros valores de datos.

**3.11 receptor:** Realización que decodifica los octetos producidos por un emisor, para identificar el valor de datos que fue codificado.

**3.12 emisor:** Realización que codifica un valor de datos para su transferencia.

**3.13 bit 0 de cola:** Un 0 en la última posición de un valor cadena de bits.

NOTA – El 0 de un valor cadena de bits formada por un solo bit 0 es un bit de cola. Su eliminación produce una cadena de bits vacía.

## 4 Abreviaturas

ASN.1	Notación de sintaxis abstracta uno ( <i>abstract syntax notation one</i> )
BER	Reglas de codificación básica ( <i>basic encoding rules</i> ) (para ASN.1)
CER	Reglas de codificación canónica ( <i>canonical encoding rules</i> ) (para ASN.1)
DER	Reglas de codificación distinguida ( <i>distinguished encoding rules</i> ) (para ASN.1)
ULA	Arquitectura de capa superior ( <i>upper layer architecture</i> )

## 5 Notación

Esta Recomendación | Norma Internacional hace referencia a la notación definida en la Rec. UIT-T X.680 | ISO/CEI 8824-1.

## 6 Convenios

**6.1** La presente Recomendación | Norma Internacional especifica el valor de cada octeto en una codificación mediante el uso de los términos «bit más significativo» y «bit menos significativo».

NOTA – Las especificaciones relativas a capas más bajas utilizan la misma notación para definir el orden de la transmisión de los bits por una línea en serie o la asignación de los bits a canales en paralelo.

**6.2** A los efectos de la presente Recomendación | Norma Internacional solamente, los bits de un octeto se enumeran de 8 a 1, siendo el bit 8 el «bit más significativo» y el bit 1 el «bit menos significativo».

**6.3** A los efectos de la presente Recomendación | Norma Internacional, se pueden comparar dos cadenas de octetos. Una cadena de octetos es igual a otra si tienen la misma longitud y son iguales en cada posición de octetos. Una cadena de octetos,  $S_1$ , es mayor que otra,  $S_2$ , solamente si:

- $S_1$  y  $S_2$  tienen octetos idénticos en cada una de las posiciones hasta el octeto final en  $S_2$ , pero  $S_1$  es más larga; o
- $S_1$  y  $S_2$  tienen diferentes octetos en una o más posiciones y en la primera de estas posiciones el octeto de  $S_1$  es mayor que el de  $S_2$ , considerando los octetos como números binarios sin signo cuyo bit  $n$  tiene la ponderación  $2^{n-1}$ .

## 7 Conformidad

**7.1** La conformidad dinámica se especifica en las cláusulas 8 a 12 inclusive.

**7.2** La conformidad estática es determinada por aquellas normas que especifican la aplicación de una o más de estas reglas de codificación.

**7.3** Las reglas de codificación básica permiten codificaciones alternativas como una opción del emisor. Los receptores que alegan conformidad con las reglas de codificación básica admitirán todas las alternativas.

NOTA – En 8.1.3.2 b) y en el Cuadro 3 figuran ejemplos de estas codificaciones alternativas.

**7.4** Las reglas de codificación económica o de codificación distinguida no permiten codificaciones alternativas.

## 8 Reglas de codificación básica

### 8.1 Reglas generales de codificación

#### 8.1.1 Estructura de la codificación

**8.1.1.1** La codificación de un valor de datos estará formada por cuatro componentes, que deben aparecer en el siguiente orden:

- octetos de identificador (véase 8.1.2);

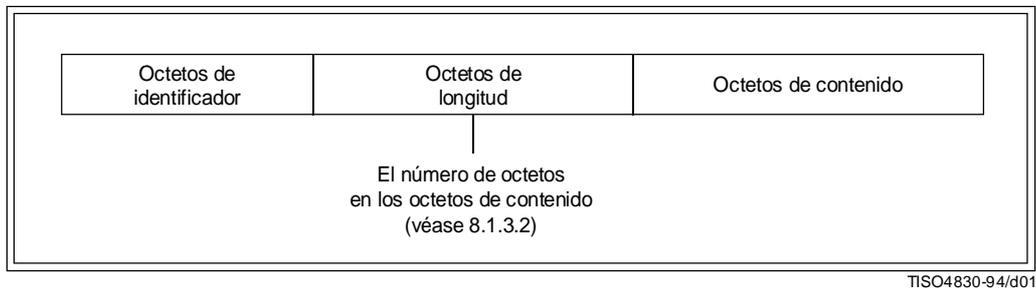
- b) octetos de longitud (véase 8.1.3);
- c) octetos de contenido (véase 8.1.4);
- d) octetos de fin de contenido (véase 8.1.5).

**8.1.1.2** Los octetos de fin de contenido sólo estarán presentes si así lo requiere el valor de los octetos de longitud (véase 8.1.3).

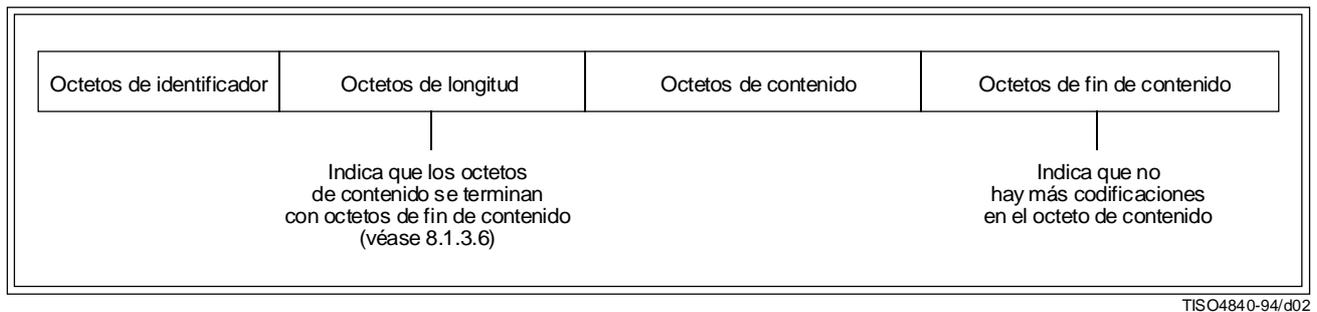
**8.1.1.3** La Figura 1 ilustra la estructura de una codificación (primitiva o construida). La Figura 2 ilustra una codificación construida alternativa.

**8.1.2 Octetos de identificador**

**8.1.2.1** Los octetos de identificador codificarán el rótulo ASN.1 (clase y número) del tipo del valor de datos.



**Figura 1 – Estructura de una codificación**



**Figura 2 – Codificación construida alternativa**

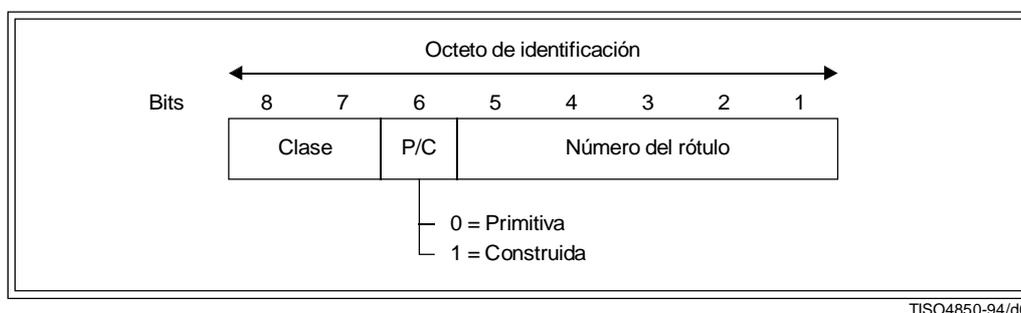
**8.1.2.2** En los rótulos con un número comprendido entre cero y 30 (inclusive), los octetos identificadores incluirán un solo octeto, codificado como sigue:

- a) los bits 8 y 7 estarán codificados para representar la clase del rótulo, tal como se especifica en el Cuadro 1;
- b) el bit 6 será cero o uno, de acuerdo con las normas de 8.1.2.5;
- c) los bits de 5 a 1 codificarán el número del rótulo como un entero binario, siendo el bit 5 el bit más significativo.

**Cuadro 1 – Codificación de la clase del rótulo**

Clase	Bit 8	Bit 7
Universal	0	0
Aplicación	0	1
Específico del contexto	1	0
Privado	1	1

**8.1.2.3** La Figura 3 ilustra la forma de un octeto de identificador para un tipo con un rótulo cuyo número está comprendido entre cero y 30 (inclusive).



**Figura 3 – Octeto de identificador (número del rótulo bajo)**

**8.1.2.4** Para los r tulos con n mero superior o igual a 31, el identificador incluir  un octeto de encabezamiento seguido por uno o m s octetos subsiguientes.

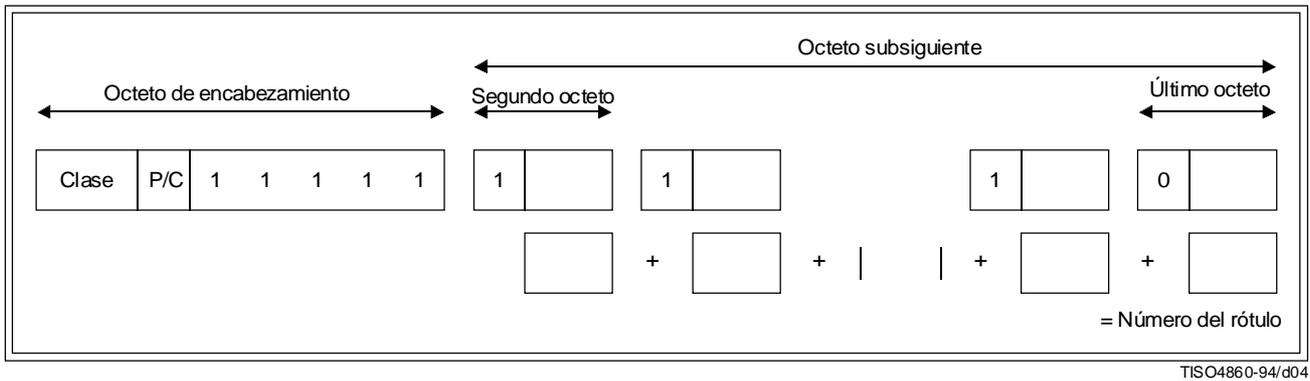
**8.1.2.4.1** El octeto de encabezamiento se codificar  como sigue:

- a) los bits 8 y 7 se codificar n para representar la clase del r tulo, tal como se indica en el Cuadro 1;
- b) el bit 6 ser  cero o uno, de acuerdo con las reglas indicadas en 8.1.2.5;
- c) los bits 5 a 1 se codificar n 11111<sub>2</sub>.

**8.1.2.4.2** Los octetos subsiguientes codificar n el n mero del r tulo como sigue:

- a) el bit 8 de cada octeto se pondr  a uno, a menos que sea el  ltimo octeto de los octetos de identificaci n;
- b) los bits 7 a 1 del primer octeto subsiguiente, seguidos por los bits 7 a 1 del segundo octeto subsiguiente, seguidos a su vez por los bits 7 a 1 de cada octeto adicional, hasta el  ltimo octeto subsiguiente inclusive de los octetos de identificaci n, se codificar n como un entero binario sin signo, igual al n mero del r tulo, siendo el bit 7 del primer octeto subsiguiente el bit m s significativo;
- c) los bits 7 a 1 del primer octeto subsiguiente no ser n todos cero.

**8.1.2.4.3** La Figura 4 ilustra la forma de los octetos de identificaci n para un tipo cuyo r tulo tiene un n mero superior a 30.



TISO4860-94/d04

Figura 4 – Octetos de identificador (número del rótulo alto)

**8.1.2.5** El bit 6 se pondrá a cero si la codificación es primitiva y a uno si la codificación es construida.

NOTA – Las subcláusulas que siguen especifican, para cada tipo, si la codificación es primitiva o construida.

**8.1.2.6** La Rec. UIT-T X.680 | ISO/CEI 8824-1 especifica que el rótulo de un tipo definido por la palabra clave «CHOICE» toma el valor del rótulo del tipo del cual se extrae el valor de datos elegido.

**8.1.2.7** La Rec. UIT-T X.681 | ISO/CEI 8824-2, subcláusulas 14.2 y 14.4, especifica que el rótulo de un tipo definido por "ObjectClassFieldType" es indeterminado, si es un campo de tipo, un campo de valor de tipo variable o un campo de conjunto de valores de tipo variable. Este tipo se define posteriormente como un tipo ASN.1, de modo que la codificación completa resulta idéntica a la de un valor de tipo asignado (incluidos los octetos de identificador).

### 8.1.3 Octetos de longitud

**8.1.3.1** Se especifican dos formas de octetos de longitud, que son:

- a) la forma definida (véase 8.1.3.3); y
- b) la forma indefinida (véase 8.1.3.6).

**8.1.3.2** Un emisor deberá:

- a) utilizar la forma definida (8.1.3.3) si la codificación es primitiva;
- b) utilizar ya sea la forma definida (8.1.3.3) o la forma indefinida (8.1.3.6), a opción del emisor, si la codificación es construida y está disponible en su totalidad inmediatamente;
- c) utilizar la forma indefinida (8.1.3.6) si la codificación es construida pero no está disponible en su totalidad inmediatamente.

**8.1.3.3** En la forma definida, los octetos de longitud constarán de uno o más octetos, y representarán el número de octetos de los octetos de contenido, utilizando ya sea la forma corta (8.1.3.4) o la forma larga (8.1.3.5), a elección del emisor.

NOTA – La forma corta sólo puede ser utilizada si el número de octetos de los octetos de contenido es menor o igual que 127.

**8.1.3.4** En la forma corta, los octetos de longitud estarán formados por un solo octeto, en el que el bit 8 será cero y los bits 7 a 1 codificarán el número de octetos de los octetos de contenido (que puede ser cero) como un entero binario sin signo y con el bit 7 como el bit más significativo.

#### Ejemplo:

L = 38 se puede codificar 00100110<sub>2</sub>.

**8.1.3.5** En la forma larga, los octetos de longitud estarán formados por un octeto inicial y uno o más octetos subsiguientes. El octeto inicial se codificará como sigue:

- a) el bit 8 será uno;

- b) los bits 7 a 1 codificarán el número de octetos subsiguientes de los octetos de longitud como un entero binario sin signo, con el bit 7 como bit más significativo;
- c) no se utilizará el valor 1111111<sub>2</sub>.

NOTA 1 – Esta limitación se establece en previsión de una ampliación futura.

Los bits 8 a 1 del primer octeto subsiguiente, seguidos por los bits 8 a 1 del segundo octeto subsiguiente, seguidos a su vez por los bits 8 a 1 de cada octeto siguiente hasta, e incluido, el último octeto subsiguiente, serán codificados como un entero binario sin signo igual al número de octetos de los octetos de contenido, siendo el bit 8 del primer octeto subsiguiente el bit más significativo.

**Ejemplo:**

L = 201 se puede codificar:

10000001<sub>2</sub>

11001001<sub>2</sub>

NOTA 2 – En la forma larga, el emisor puede, facultativamente, emplear más octetos de longitud que el mínimo necesario.

**8.1.3.6** En la forma indefinida, los octetos de longitud indican que los octetos de contenido se terminan por octetos de fin de contenido (véase 8.1.5), y estarán formados por un solo octeto.

**8.1.3.6.1** El octeto único tendrá el bit 8 puesto a uno y los bits 7 a 1 puestos a cero.

**8.1.3.6.2** Si se utiliza esta forma de longitud, los octetos de fin de contenido (véase 8.1.5) estarán presentes en la codificación después de los octetos de contenido.

**8.1.4 Octetos de contenido**

Los octetos de contenido consistirán en cero, uno o más octetos y codificarán el valor de datos tal como se especifica en las subcláusulas que siguen.

NOTA – Los octetos de contenido dependen del tipo del valor de datos; las siguientes subcláusulas siguen la misma secuencia que la definición de tipos en ASN.1.

**8.1.5 Octetos de fin de contenido**

Los octetos de fin de contenido estarán presentes si la longitud se codifica de acuerdo con lo especificado en 8.1.3.6; de no ser así, no estarán presentes.

Los octetos de fin de contenido estarán formados por dos octetos cero.

NOTA – Los octetos de fin de contenido pueden considerarse como la codificación de un valor cuyo rótulo es de clase universal, su forma es primitiva, el número del rótulo es cero, y carecen de contenido; luego:

Fin de contenido	Longitud	Contenido
00 <sub>16</sub>	00 <sub>16</sub>	Ausente

**8.2 Codificación de un valor booleano (boolean)**

**8.2.1** La codificación de un valor booleano será primitiva. Los octetos de contenido estarán formados por un solo octeto.

**8.2.2** Si el valor booleano es

FALSE (FALSO)

el octeto será cero.

Si el valor booleano es

TRUE (VERDADERO)

el octeto tendrá cualquier valor distinto de cero, a elección del emisor.

**Ejemplo** – Si es del tipo BOOLEAN, el valor TRUE puede codificarse como:

Booleano	Longitud	Contenido
01 <sub>16</sub>	01 <sub>16</sub>	FF <sub>16</sub>

### 8.3 Codificación de un valor entero (integer)

**8.3.1** La codificación de un valor entero será primitiva. Los octetos de contenido estarán formados por uno o más octetos.

**8.3.2** Si los octetos de contenido de la codificación de un valor entero son más de uno, los bits del primer octeto y el bit 8 del segundo octeto:

- a) no serán todos uno; y
- b) no serán todos cero.

NOTA – Estas reglas garantizan que un valor entero se codifique siempre con el menor número de octetos posible.

**8.3.3** Los octetos de contenido serán un número binario de complemento a dos igual al valor entero, y consistirán en los bits 8 a 1 del primer octeto, seguidos por los bits 8 a 1 del segundo octeto, seguidos por los bits 8 a 1 de cada octeto siguiente, hasta, e incluido, el último octeto de los octetos de contenido.

NOTA – El valor de un número binario de complemento a dos se deriva de la numeración de los bits en los octetos de contenido, comenzando con el bit 1 del último octeto como bit cero y terminando la numeración con el bit 8 del primer octeto. Cada bit recibe un valor numérico de  $2^N$ , donde N es su posición en la secuencia de numeración anterior. El valor del número binario de complemento a dos se obtiene sumando los valores numéricos asignados a cada bit para los bits que están puestos a uno, excluido el bit 8 del primer octeto, y restando después a este valor el valor numérico asignado al bit 8 del primer octeto si dicho bit está puesto a uno.

### 8.4 Codificación de un valor enumerado (enumerated)

La codificación de un valor enumerado será la del valor entero al que está asociado.

NOTA – Es primitiva.

### 8.5 Codificación de un valor real (real)

**8.5.1** La codificación de un valor real será primitiva.

**8.5.2** Si el valor real es el valor cero, en la codificación no habrá octetos de contenido.

**8.5.3** Si el valor real es distinto de cero, la base utilizada para la codificación será B', elegida por el emisor. Si B' es 2, 8 ó 16, se utilizará la codificación binaria especificada en 8.5.5. Si B' es 10, se utilizará una codificación de carácter, especificada en 8.5.6.

NOTA – La forma de almacenamiento, generación o procesamiento por los emisores y receptores, así como la forma utilizada en la notación de valor ASN.1, son independientes de la base utilizada para la transferencia.

**8.5.4** El bit 8 del primer octeto de contenido se fijará como sigue:

- a) si el bit 8 = 1, se aplica la codificación binaria especificada en 8.5.5;
- b) si el bit 8 = 0 y el bit 7 = 0, se aplica la codificación decimal especificada en 8.5.6;
- c) si el bit 8 = 0 y el bit 7 = 1, se codifica un "SpecialRealValue" (valor real especial) (véase la Rec. UIT-T X.680 | ISO/CEI 8824-1), tal como se especifica en 8.5.7.

**8.5.5** Cuando se utiliza la codificación binaria (bit 8 = 1), si la mantisa M es distinta de cero, estará representada por un signo S, un valor entero no negativo N y un factor de escala binaria F, de manera que:

$$M = S \times N \times 2^F$$

$$0 \leq F < 4$$

$$S = +1 \text{ ó } -1$$

NOTA – El factor de escala binaria F se necesita en determinadas circunstancias para alinear la coma decimal implícita de la mantisa en la posición requerida por las reglas de codificación de esta subcláusula. Este alineamiento no siempre puede conseguirse modificando el exponente E. Si la base B' utilizada para la codificación es 8 ó 16, la coma decimal implícita sólo puede ser desplazada en pasos de 3 ó 4 bits, respectivamente, cambiando el componente E. Quizá se necesiten, por consiguiente, valores del factor de escala binaria F distintos de cero para desplazar la coma decimal implícita a la posición requerida.

**8.5.5.1** El bit 7 del primer octeto de contenido será 1 si S es -1, y 0 en los demás casos.

**8.5.5.2** Los bits 6 a 5 del primer octeto de contenido codificarán el valor de la base B' como sigue:

<i>Bits 6 a 5</i>	<i>Base</i>
00	base 2
01	base 8
10	base 16
11	Reservado para versiones futuras de esta Recomendación   Norma Internacional.

**8.5.5.3** Los bits 4 a 3 del primer octeto de contenido codificarán el valor del factor de escala binaria F como un entero binario sin signo.

**8.5.5.4** Los bits 2 a 1 del primer octeto de contenido codificarán el formato del exponente como sigue:

- a) si los bits 2 a 1 son 00, el segundo octeto de contenido codifica el valor del exponente como un número binario de complemento a dos;
- b) si los bits 2 a 1 son 01, el segundo y tercer octetos de contenido codifican el valor del exponente como un número binario de complemento a dos;
- c) si los bits 2 a 1 son 10, el segundo, tercero y cuarto octetos de contenido codifican el valor del exponente como un número binario de complemento a dos;
- d) si los bits 2 a 1 son 11, el segundo octeto de contenido codifica el número de octetos, por ejemplo X (como un número binario sin signo) utilizado para codificar el valor del exponente, y los octetos de contenido del tercero al (X más 3)<sup>o</sup> (inclusive) codifican el valor del exponente como un número binario de complemento a dos; el valor de X será al menos uno; los nueve primeros bits del exponente transmitido no serán todos cero o todos uno.

**8.5.5.5** Los restantes octetos de contenido codifican el valor del entero N (véase 8.5.5) como un número binario sin signo.

NOTAS

1 Con BER no canónica no es precisa la normalización de la coma decimal flotante de la mantisa. Esto permite a un realizador transmitir octetos que contienen la mantisa sin efectuar funciones de desplazamiento en la mantisa en memoria. En las reglas de codificación canónica y de codificación distinguida se especifica la normalización, y la mantisa (a menos que sea 0) ha de ser desplazada repetidamente, hasta que el bit menos significativo sea un 1.

2 Esta representación de números reales es muy diferente de los formatos utilizados normalmente en el soporte físico de coma flotante, pero está concebida de manera que pueda ser convertida fácilmente a dichos formatos y a partir de ellos (véase el Anexo C).

**8.5.6** Cuando se utiliza la codificación decimal (bits 8 a 7 = 00), todos los octetos de contenido después del primer octeto de contenido forman un campo, en el sentido que tiene este término en ISO 6093, de una longitud elegida por el emisor, y codificada de acuerdo a ISO 6093. La elección de la representación del número según ISO 6093 se especifica por medio de los bits 6 a 1 del primer octeto de contenido, como sigue:

<i>Bits 6 a 1</i>	<i>Representación del número</i>
00 0001	Forma NR1 ISO 6093
00 0010	Forma NR2 ISO 6093
00 0011	Forma NR3 ISO 6093

Los restantes valores de los bits 6 a 1 están reservados para versiones futuras de esta Recomendación | Norma Internacional.

No se utilizarán los factores de escala especificados en la documentación adjunta (véase ISO 6093).

NOTAS

1 Las recomendaciones de ISO 6093 relativas al uso de cuando menos una cifra a la izquierda de la coma decimal se adoptan también en la presente Recomendación | Norma Internacional, pero las mismas no son obligatorias.

2 El uso de la forma normalizada (véase ISO 6093) es una opción del emisor, y carece de significación.

**8.5.7** Cuando se deban codificar "SpecialRealValues" (valores reales especiales) (bit 8 a 7 = 01), habrá un solo octeto de contenido, con los siguientes valores:

01000000 el valor es PLUS-INFINITY

01000001 el valor es MINUS-INFINITY

Todos los demás valores que tienen los bits 8 y 7 iguales a 0 y 1, respectivamente, están reservados para adiciones a esta Recomendación | Norma Internacional.

## **8.6 Codificación de un valor de cadena de bits (bitstring)**

**8.6.1** La codificación de un valor de cadena de bits puede ser primitiva o construida, a opción del emisor.

NOTA – Cuando es necesario transferir parte de una cadena de bits antes de disponer de la totalidad de la cadena de bits se utiliza la codificación construida.

**8.6.2** Los octetos de contenido para la codificación primitiva contendrán un octeto inicial seguido de cero, uno o más octetos subsiguientes.

**8.6.2.1** Los bits de la cadena de bits, comenzando por el primero y continuando hasta el último bit, estarán colocados en los bits 8 a 1 del primer octeto subsiguiente, seguidos por los bits 8 a 1 del segundo octeto subsiguiente, seguidos por los bits 8 a 1 de cada octeto en turno, seguidos por tantos bits como sea necesario en el octeto subsiguiente final, comenzando con el bit 8.

NOTA – Los términos «primer bit» y «último bit» (bit de cola) se definen en la Rec. UIT-T X.680 | ISO/CEI 8824-1.

**8.6.2.2** El octeto inicial codificará, como un entero binario sin signo con el bit 1 como bit menos significativo, el número de bits no utilizados del octeto subsiguiente final. El número estará comprendido entre cero y siete.

**8.6.2.3** Si la cadena de bits está vacía, no habrá octetos subsiguientes y el octeto inicial será cero.

**8.6.2.4** Cuando se aplica la subcláusula 19.7 de la Rec. UIT-T X.680 | ISO/CEI 8824-1, un codificador/decodificador BER puede añadir al, o eliminar del, valor bits 0 de cola.

NOTA – Si un valor de cadena de bits no tiene bits 1, un codificador puede (como opción del emisor) codificar el valor con longitud 0 y sin octetos de contenido o codificarlo como una cadena de bits con uno o más bits 0.

**8.6.3** Los octetos de contenido para la codificación construida consistirán en cero, una o más codificaciones anidadas.

NOTA – Cada una de dichas codificaciones incluye octetos de identificador, de longitud y de contenido, y puede incluir, si es construida, octetos de fin de contenido.

**8.6.4** Para codificar un valor de cadena de bits de esta manera, se segmenta. Cada segmento estará formado por una serie de bits consecutivos del valor y, con la posible excepción del último, contendrá un número de bits que es un múltiplo de ocho. Cada uno de los bits del valor global estará exactamente en un segmento, pero no se dará significación a las fronteras de segmento.

NOTA – Un segmento puede ser de tamaño cero, es decir, no contener bits.

**8.6.4.1** Cada codificación en los octetos de contenido representará un segmento de la cadena de bits global, resultando la codificación de una aplicación recursiva de esta cláusula. En esta aplicación recursiva, cada segmento se trata como si fuese un valor de cadena de bits. Las codificaciones de los segmentos aparecerán en los octetos de contenido en el orden en que aparecen sus bits en el valor global.

### NOTAS

1 Como consecuencia de esta repetición, cada codificación en los octetos de contenido puede ser primitiva o construida. No obstante, esas codificaciones serán por lo general primitivas.

2 En particular, los rótulos de los octetos de contenido son siempre de clase universal número 3.

**8.6.4.2 Ejemplo** – Si es del tipo BIT STRING, el valor '0A3B5F291CD'H puede codificarse como se muestra más abajo. En este ejemplo, la cadena de bits está representada como una primitiva:

BitString	Longitud	Contenido
03 <sub>16</sub>	07 <sub>16</sub>	040A3B5F291CD0 <sub>16</sub>

El valor anterior puede codificarse también como se muestra a continuación. En este ejemplo, la cadena de bits se representa como un constructor:

BitString	Longitud	Contenido		
23 <sub>16</sub>	80 <sub>16</sub>	Cadena de bits	Longitud	Contenido
		03 <sub>16</sub> 03 <sub>16</sub>	03 <sub>16</sub> 05 <sub>16</sub>	000A3B <sub>16</sub> 045F291CD0 <sub>16</sub>
EOC 00 <sub>16</sub>	Longitud 00 <sub>16</sub>			

## 8.7 Codificación de un valor de cadena de octetos (octetstring)

**8.7.1** La codificación de un valor de cadena de octetos puede ser primitiva o construida, a elección del emisor.

NOTA – Cuando sea necesario transferir parte de una cadena de octetos antes de disponer de la totalidad de la cadena de octetos, se utiliza la codificación construida.

**8.7.2** La codificación primitiva contiene cero, uno o más octetos de contenido, iguales en valor a los octetos del valor de datos, en el orden en que aparecen en el valor de datos y con el bit más significativo de un octeto de valor de datos alineado con el bit más significativo de uno de los octetos de contenido.

**8.7.3** En la codificación construida, los octetos de contenido estarán formados por cero, una o más codificaciones.

NOTA – Cada una de dichas codificaciones incluye octetos de identificación, de longitud y de contenido, y puede incluir octetos de fin de contenido si es construida.

**8.7.3.1** Para codificar un valor de cadena de octetos de esta manera, se segmenta. Cada segmento estará formado por una serie de octetos consecutivos del valor. No se dará significación a las fronteras de segmento.

NOTA – Un segmento puede ser de tamaño cero, es decir, no contener octetos.

**8.7.3.2** Cada codificación en los octetos de contenido representará un segmento de la cadena de octetos global, resultando la codificación de una aplicación recursiva de esta cláusula. En esta aplicación recursiva, cada segmento se trata como si fuese un valor de cadena de octetos. Las codificaciones de los segmentos aparecerán en los octetos de contenido en el orden en que aparecen sus octetos en el valor global.

### NOTAS

1 Como consecuencia de esta repetición, cada codificación en los octetos de contenido puede ser primitiva o construida. No obstante, esas codificaciones serán normalmente primitivas.

2 En particular, los rótulos de los octetos de contenido son siempre de clase universal, número 4.

## 8.8 Codificación de un valor nulo (null)

**8.8.1** La codificación de un valor nulo será primitiva.

**8.8.2** Los octetos de contenido no contendrán ningún octeto.

NOTA – El octeto de longitud es cero.

**Ejemplo** – Si es de tipo NULL, el NULL puede codificarse como:

*Null Longitud*

05<sub>16</sub> 00<sub>16</sub>

## 8.9 Codificación de un valor de secuencia (sequence)

**8.9.1** La codificación de un valor secuencia será construida.

**8.9.2** Los octetos de contenido estarán formados por la codificación completa de un valor de datos de cada uno de los tipos indicados en la definición de tipo secuencia ASN.1, en el orden de su aparición en la definición, a menos que se haya referenciado el tipo con la palabra clave «OPTIONAL» o la palabra clave «DEFAULT».

**8.9.3** La codificación de un valor de datos puede, pero no necesariamente, estar presente para un tipo que fue referenciado con la palabra clave «OPTIONAL» o con la palabra clave «DEFAULT». Si está presente, aparecerá en la codificación en el punto correspondiente a la aparición del tipo en la definición ASN.1.

**Ejemplo** – Si es del tipo

**SEQUENCE {name IA5String, ok BOOLEAN}**

el valor

{name "Smith", ok TRUE}

puede codificarse como:

Sequence	Longitud	Contenido		
30 <sub>16</sub>	0A <sub>16</sub>			
		IA5String	Longitud	Contenido
		16 <sub>16</sub>	05 <sub>16</sub>	"Smith"
		Boolean	Longitud	Contenido
		01 <sub>16</sub>	01 <sub>16</sub>	FF <sub>16</sub>

## 8.10 Codificación de un valor de secuencia de (sequence-of)

**8.10.1** La codificación de un valor secuencia de será construida.

**8.10.2** Los octetos de contenido estarán formados por cero, una o más codificaciones completas de valores de datos del tipo indicado en la definición ASN.1.

**8.10.3** El orden de las codificaciones de los valores de datos será el mismo que el orden de los valores de datos en el valor secuencia de que se ha de codificar.

## 8.11 Codificación de un valor de conjunto (set)

**8.11.1** La codificación de un valor de conjunto será construida.

**8.11.2** Los octetos de contenido estarán formados por la codificación completa de un valor de datos de cada uno de los tipos indicados en la definición ASN.1 de tipo de conjunto, y en el orden elegido por el emisor, a menos que se haya referenciado el tipo con la palabra clave «OPTIONAL» o con la palabra clave «DEFAULT».

**8.11.3** La codificación de un valor de datos puede, pero no necesariamente, estar presente para un tipo que se ha referenciado con la palabra clave «OPTIONAL» o la palabra clave «DEFAULT».

NOTA – El orden de los valores de datos en un valor de conjunto no tiene significación ni impone constricciones al orden durante la transferencia.

## 8.12 Codificación de un valor de conjunto de (set-of)

**8.12.1** La codificación de un valor conjunto de será construida.

**8.12.2** Se aplica el texto de 8.10.2.

**8.12.3** No es necesario conservar el orden de los valores de datos durante la codificación o la decodificación subsiguiente.



## 8.16 Codificación de un valor de ejemplar de

8.16.1 La codificación del tipo ejemplar de será la codificación BER del siguiente tipo de secuencia con el valor especificado en 8.16.2:

```
[UNIVERSAL 8] IMPLICIT SEQUENCE
{
    type-id  <DefinedObjectClass>.&id,
    value    [0] EXPLICIT <DefinedObjectClass>.&Type
}
```

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

NOTA – Cuando el valor es un valor de un tipo ASN.1 y se utiliza para el mismo la codificación BER, la codificación de este tipo es idéntica a una codificación de un valor correspondiente del tipo externo, donde la alternativa de «sintaxis» se utiliza para representar el valor abstracto.

8.16.2 El valor de los componentes del tipo secuencia indicada en 8.16.1 será el mismo que los valores de los componentes correspondientes del tipo asociado en C.7 de la Rec. UIT-T X.681 | ISO/CEI 8824-2.

## 8.17 Codificación de un valor del tipo incrustado

8.17.1 Hay dos subreglas que se utilizan para codificar el tipo incrustado, denominadas EP-A (la regla de fijación de índice) y EP-B (la regla de utilización de índice). Para cualquier valor dado de "identification" (identificación) en el valor abstracto, la primera aparición (dentro de la codificación de un completo) de un valor del tipo incrustado con este valor de «identificación» se codificará utilizando la subregla EP-A.

8.17.2 A reserva de ciertas restricciones indicadas a continuación, los valores siguientes con el mismo valor de «identificación» se codificarán utilizando la subregla EP-B con el «índice» puesto al mismo valor que el "index" (índice) en la codificación EP-A correspondiente.

8.17.3 Las condiciones para la utilización de la subregla EP-B se indican a continuación. Si no se satisface alguna condición, se utilizará la subregla EP-A. Las condiciones para la utilización de la subregla EP-B son:

- el valor de «índice» está comprendido en la gama 0 a 255;
- la codificación de este caso de un valor del tipo incrustado es un múltiplo entero de ocho bits;
- la longitud de la codificación no es mayor que la longitud máxima que puede ser identificada por una codificación de forma larga.

NOTA – No es probable que la última condición sea una restricción en la práctica.

8.17.4 Para la primera ocurrencia de una codificación EP-A, el «índice» tendrá el valor cero y para cada ocurrencia siguiente se incrementará en uno.

8.17.5 INFORMACIÓN – De este modo para cualquier valor de «identificación» dado, hay una codificación EP-A (relativamente ineficaz) con un valor de índice único y el valor de «identificación» completo, seguido de muchas codificaciones EP-B (eficaces) arbitrariamente enlazadas a la codificación EP-A por el valor de índice. Como la codificación EP-B utiliza un solo octeto para el índice, y un cómputo en octetos para la codificación, no se puede emplear si el valor del índice excede de 255 (256 «identificaciones» diferentes están en uso), o si la codificación no es un múltiplo de ocho bits. En estas circunstancias, se emplea la codificación EP-A para todos los casos.

8.17.6 La codificación EP-A será la codificación BER del siguiente tipo de secuencia después de la aplicación de AUTOMATIC TAGS (rótulos automáticos) especificada en la Recomendación UIT-T X.680 | ISO/CEI 8824-1, subcláusulas 22.6 y 26.3:

```
[UNIVERSAL 11] IMPLICIT SEQUENCE {
    index          INTEGER,
    identification CHOICE {
        syntaxes   SEQUENCE {
            abstract OBJECT IDENTIFIER,
            transfer OBJECT IDENTIFIER },
        syntax     OBJECT IDENTIFIER,
        presentation-context-id INTEGER,
        context-negotiation SEQUENCE {
            presentation-context-id INTEGER,
            transfer-syntax     OBJECT IDENTIFIER },
        transfer-syntax OBJECT IDENTIFIER,
        fixed          NULL },
    data-value      BIT STRING }
```

**8.17.7** El valor de "data-value" (valor de datos) será la codificación del valor de datos abstracto que utiliza la sintaxis de transferencia identificada, el valor del "index" (índice) será el determinado anteriormente, y el valor de los otros campos será el mismo que los valores que aparecen en el valor abstracto.

NOTAS

- 1 El componente "index" no está definido en la sintaxis abstracta porque ha de ser suministrado estrictamente a nivel de la regla de codificación (en el mismo sentido que los octetos de fin de contenido se han de aplicar a nivel de la regla de codificación).
- 2 En la sintaxis de transferencia ambas alternativas de "data-value" se codifican idénticamente en la sintaxis abstracta.

**8.17.8** La codificación de EP-B será la codificación BER del tipo ASN.1 siguiente:

**[UNIVERSAL 11] IMPLICIT OCTET STRING**

donde

- a) la codificación es primitiva;
- b) los octetos de longitud tienen la forma definida corta o larga como una opción del transmisor;
- c) los octetos de contenido codifican el "index" en el primer octeto del valor cadena de octetos como un entero cuyo valor va de 0 a 255, seguido de la codificación del "data-value".

NOTA – Un receptor puede distinguir la codificación EP-A de la codificación EP-B fijando el bit primitivo/constructor.

**8.18 Codificación de un valor del tipo externo**

**8.18.1** La codificación del valor de un tipo externo será la codificación BER del siguiente tipo de secuencia, que se supone está definido en un entorno de EXPLICIT TAGS (rótulos explícitos) con un valor especificado en las siguientes subcláusulas:

```
[UNIVERSAL 8] IMPLICIT SEQUENCE {
    direct-reference          OBJECT IDENTIFIER OPTIONAL,
    indirect-reference       INTEGER OPTIONAL,
    data-value-descriptor   ObjectDescriptor OPTIONAL,
    encoding                 CHOICE {
        single-ASN1-type    [0] ABSTRACT-SYNTAX.&Type,
        octet-aligned       [1] IMPLICIT OCTET STRING,
        arbitrary           [2] IMPLICIT BIT STRING } }
```

NOTA – Este tipo de secuencia es igual al especificado en la Rec. X.208 del CCITT (1988) | ISO/CEI 8824 (1990), y la codificación resultante de un valor de un tipo externo no se altera según estas especificaciones.

**8.18.2** El valor de los campos depende de los valores abstractos que se transmiten, que es un valor del tipo especificado en 30.5 de la Rec. UIT-T X.680 | ISO/CEI 8824-1.

**8.18.3** El "data-value-descriptor" (descriptor de valor de datos) anterior está presente solamente si está presente el "data-value-descriptor" en el valor abstracto y tendrán el mismo valor.

**8.18.4** La "direct-reference" (referencia directa) e "indirect-reference" (referencia indirecta) anteriores estarán presentes o ausentes (y si están presentes tendrán los valores del campo abstracto mostrado) de acuerdo con el Cuadro 2, que enumera todas las alternativas para "identification" (identificación) en el valor abstracto (véase 30.5 de la Rec. UIT-T X.680 | ISO/CEI 8824-1).

**Cuadro 2 – Codificaciones alternativas para «identificador»**

identificación	referencia directa	referencia indirecta
sintaxis	*** NO PUEDE OCURRIR ***	*** NO PUEDE OCURRIR ***
sintaxis	sintaxis	AUSENTE
id de contexto de presentación	AUSENTE	id de contexto de presentación
negociación de contexto	sintaxis de transferencia	id de contexto de presentación
sintaxis de transferencia	*** NO PUEDE OCURRIR ***	*** NO PUEDE OCURRIR ***
fija	*** NO PUEDE OCURRIR ***	*** NO PUEDE OCURRIR ***

**8.18.5** El valor de datos se codificará de acuerdo con la sintaxis de transferencia indicada por la codificación y se colocará en una alternativa de la opción de "Encoding" (codificación) especificada a continuación.

**8.18.6** Si el valor de datos es el valor de un tipo de datos ASN.1, y si las reglas de codificación para este valor de datos son iguales a las del tipo de datos "EXTERNAL" completo, la realización emisora utilizará cualquiera de las siguientes opciones de "Encoding" (codificación):

- un tipo ASN.1 (single-ASN1-type)
- alineada en octeto (octet-aligned)
- arbitraria (arbitrary)

como una opción de la realización.

**8.18.7** Si la codificación del valor de datos, utilizando la codificación acordada o negociada, es un número entero de octetos, la realización emisora utilizará cualquiera de las siguientes opciones de "Encoding":

- alineada en octeto
- arbitraria

como una opción de la realización.

NOTA – Un valor de datos que es una serie de tipos ASN.1 para los cuales la sintaxis de transferencia especifica la concatenación simple de las cadenas de octetos producidas por la aplicación de las reglas de codificación básica ASN.1 a cada tipo ASN.1, corresponde a esta categoría, no a la indicada en 8.18.6.

**8.18.8** Si la codificación del valor de datos, utilizando la codificación acordada o negociada, no es un número entero de octetos, la opción de "Encoding" será:

- arbitraria

**8.18.9** Si la opción de "Encoding" se elige como "single-ASN1-type", el tipo ASN.1 sustituirá al tipo abierto, con un valor igual al valor de datos que se ha de codificar.

NOTA – La gama de valores que podría producirse en el tipo abierto es determinada por el registro del valor de identificador de objeto asociado con la "direct-reference", y/o el valor entero asociado con la "indirect-reference".

**8.18.10** Si la opción de "Encoding" elegida es "octet-aligned", el valor de datos se codificará de acuerdo con la sintaxis de transferencia acordada o negociada, y los octetos resultantes formarán el valor de la cadena de octetos.

**8.18.11** Si la opción de "Encoding" elegida es "arbitrary", el valor de datos se codificará de acuerdo con la sintaxis de transferencia acordada o negociada, y el resultado formará el valor de la cadena de bits.

## **8.19 Codificación de un valor de identificador de objeto (object identifier)**

**8.19.1** La codificación de un valor de identificador de objeto será primitiva.

**8.19.2** Los octetos de contenido serán una lista (ordenada) de codificaciones de subidentificadores (véanse 8.19.3 y 8.19.4) concatenadas entre sí.

Cada subidentificador está representado como una serie de (uno o más) octetos. El bit 8 de cada octeto indica si es el último de la serie; el bit 8 del último octeto es cero; el bit 8 de cada octeto precedente es uno. Los bits 7 a 1 de los octetos de la serie codifican colectivamente al subidentificador. Conceptualmente, estos grupos de bits están concatenados para formar un número binario sin signo cuyo bit más significativo es el bit 7 del primer octeto y cuyo bit menos significativo es el bit 1 del último octeto. El subidentificador estará codificado con el menor número posible de octetos, es decir, el octeto de encabezamiento del subidentificador no tendrá el valor 80 (hexadecimal).

**8.19.3** El número de subidentificadores (N) será inferior en una unidad al número de componentes del identificador de objeto en el valor de identificador de objeto que se codifica.

**8.19.4** El valor numérico del primer subidentificador se deduce de los valores de los dos primeros componentes de identificador de objeto del valor de identificador de objeto que se codifica mediante la fórmula

$$(X*40) + Y$$

donde X es el valor del primer componente de identificador de objeto e Y es el valor del segundo componente del identificador de objeto.

NOTA – Esta agrupación de los dos primeros componentes de identificador de objeto refleja el hecho de que sólo asignan tres valores del nodo raíz, y como máximo 39 valores subsiguientes de los nodos alcanzados por  $X = 0$  y  $X = 1$ .

**8.19.5** El valor numérico del  $i$ -ésimo subidentificador ( $2 \leq i \leq N$ ) es el del componente identificador de objeto ( $i + 1$ ).

**Ejemplo** – Un valor de OBJECT IDENTIFIER

{joint-iso-ccitt 100 3}

que es lo mismo que

{2 100 3}

tiene como primer subidentificador 180 y como segundo subidentificador 3. La codificación resultante es:

OBJECT IDENTIFIER	Longitud	Contenido
06 <sub>16</sub>	03 <sub>16</sub>	813403 <sub>16</sub>

## **8.20 Codificación de valores del tipo de cadena de caracteres restringida (restricted character string)**

**8.20.1** El valor de datos consta de una cadena de caracteres del conjunto de caracteres especificado en la definición del tipo ASN.1.

**8.20.2** Cada valor de datos será codificado independientemente de los otros valores de datos del mismo tipo.

**8.20.3** Cada tipo cadena de caracteres se codificará como si hubiera sido declarado,

**[UNIVERSAL x] IMPLICIT OCTET STRING**

donde  $x$  es el número del rótulo de clase universal asignado al tipo cadena de caracteres en la Rec. UIT-T X.680 | ISO/CEI 8824-1. El valor de la cadena de octetos se especifica en 8.20.4 y 8.20.5.

**8.20.4** Cuando en la Rec. UIT-T X.680 | ISO/CEI 8824-1 se especifica un tipo de cadena de caracteres por referencia directa a una tabla enumeradora (NumericString y PrintableString), el valor de la cadena de octetos será el especificado en 8.20.5 para un tipo VisibleString (cadena visible), con el mismo valor de cadena de caracteres.

**8.20.5** En el caso de cadenas de caracteres restringidos, distintas de UniversalString y BMPString, la cadena de octetos contendrá los octetos especificados en ISO 2022 para las codificaciones en un entorno de 8 bits utilizando la secuencia de escape y las codificaciones de caracteres registradas de acuerdo con ISO 2375.

**8.20.5.1** No se utilizará una secuencia de escape a menos que sea una de las especificadas por uno de los números de registro utilizados para definir el tipo de cadena de caracteres en la Rec. UIT-T X.680 | ISO/CEI 8824-1.

**8.20.5.2** Al principio de cada cadena, se supondrá que ciertos números de registro han sido designados como G0 y/o C0 y/o C1, e invocados (utilizando la terminología de ISO 2022). Estos se especifican, para cada tipo, en el Cuadro 3, junto con la secuencia de escape que supuestamente implican.

**8.20.5.3** Algunos tipos de cadena de caracteres no contendrán secuencias de escape explícitas dentro de sus codificaciones; en todos los demás casos, cualquier secuencia de escape permitida por 8.20.5.1 puede aparecer en cualquier momento, incluso al principio de la codificación. El Cuadro 3 indica los tipos para los cuales se permiten secuencias de escape explícitas.

**8.20.5.4** No se utilizarán anunciadores, a menos que estén permitidos explícitamente por el usuario de ASN.1.

NOTA – La elección del tipo ASN.1 proporciona una forma limitada de funcionalidad de anunciador. Los protocolos de aplicación específicos pueden optar por transmitir anunciadores en otros elementos del protocolo, o especificar en detalle la forma de uso de dichos anunciadores.

**Cuadro 3 – Utilización de secuencias de escape**

Tipo	G0 supuesto (Número de registro)	C0 y C1 supuestos (Número de registro)	Secuencia(s) de escape supuesta(s) y cambio con bloqueo (cuando sea aplicable)	¿Se permiten secuencias de escape explícitas?
NumericString (Cadena numérica)	6	Ninguno	ESC 2/8 4/2 LS0	No
PrintableString (Cadena imprimible)	6	Ninguno	ESC 2/8 4/2 LS0	No
TeletexString (Cadena teletex) (Cadena T61)	102	106 (C0) 107 (C1)	ESC 2/8 7/5 LS0 ESC 2/1 4/5 ESC 2/2 4/8	Sí
VideotexString (Cadena videotex)	102	1 (C0) 73 (C1)	ESC 2/8 7/5 LS0 ESC 2/1 4/0 ESC 2/2 4/1	Sí
VisibleString (Cadena visible) (Cadena ISO646)	6	Ninguno	ESC 2/8 4/2 LS0	No
IA5String (Cadena AI5)	6	1 (C0)	ESC 2/8 4/2 LS0 ESC 2/1 4/0	No
GraphicString (Cadena gráfica)	6	Ninguno	ESC 2/8 4/2 LS0	Sí
GeneralString (Cadena general)	6	1 (C0)	ESC 2/8 4/2 LS0 ESC 2/1 4/0	Sí

NOTA – Muchos de los caracteres usados comúnmente (por ejemplo, A a Z) aparecen en varios repertorios de caracteres con secuencias de escape y números de registro individuales. Cuando los tipos ASN.1 permitan las secuencias de escape, son posibles varias codificaciones para una determinada cadena de caracteres (véase también 7.3).

**Ejemplo** – Con la definición de tipo ASN.1

**Name ::= VisibleString**

un valor

"Jones"

puede codificarse (forma primitiva) como

VisibleString	Longitud	Contenido
1A <sub>16</sub>	05 <sub>16</sub>	4A6F6E6573 <sub>16</sub>

o (forma construida, longitud definida) como

VisibleString	Longitud	Contenido
3A <sub>16</sub>	09 <sub>16</sub>	

OctetString	Longitud	Contenido
04 <sub>16</sub>	03 <sub>16</sub>	4A6F6E <sub>16</sub>

OctetString	Longitud	Contenido
04 <sub>16</sub>	02 <sub>16</sub>	6573 <sub>16</sub>

o (forma construida, longitud indefinida) como

VisibleString	Longitud	Contenido
3A <sub>16</sub>	80 <sub>16</sub>	

OctetString	Longitud	Contenido
04 <sub>16</sub>	03 <sub>16</sub>	4A6F6E <sub>16</sub>

OctetString	Longitud	Contenido
04 <sub>16</sub>	02 <sub>16</sub>	6573 <sub>16</sub>
EOC	Longitud	
00 <sub>16</sub>	00 <sub>16</sub>	

**8.20.6** El ejemplo anterior ilustra tres de las (muchas) formas posibles que puede utilizar el emisor. Los receptores tienen que tratar todas las formas permitidas (véase 7.3).

**8.20.7** Para el tipo "UniversalString", la cadena de octetos contendrá los octetos especificados en ISO/CEI 10646-1 utilizando la forma canónica de 4 octetos (véase 14.2 de ISO/CEI 10646-1). No se utilizarán funciones de control ni firmas.

**8.20.8** Para el tipo "BMPString", la cadena de octetos contendrá los octetos especificados en ISO/CEI 10646-1, utilizando la forma BMP de 2 octetos (véase 14.1 de ISO/CEI 10646-1). No se utilizarán funciones de control ni firmas.

## 8.21 Codificación de valores del tipo de cadena de caracteres no restringida (unrestricted character string type)

**8.21.1** Hay dos subreglas que se utilizan para codificar el tipo de cadena de caracteres no restringida, denominadas CH-A (la regla de fijación de índice) y CH-B (la regla de utilización de índice). Para cualquier valor dado de "identification" (identificación) en el valor abstracto, la primera aparición (dentro de la codificación del pdv completo) de un valor del tipo de cadena de caracteres no restringida con este valor de identificación se codificará utilizando la subregla CH-A.

**8.21.2** A reserva de ciertas restricciones enumeradas a continuación, los siguientes valores con el mismo valor de identificación se codificarán utilizando las subreglas CH-B, con el "index" (índice) puesto al mismo valor que el "index" (índice) de la codificación CH-A correspondiente.

**8.21.3** Las condiciones para la utilización de la subregla CH-B se indican a continuación. Si no se satisface alguna condición, se utilizará la subregla CH-A. Las condiciones para la utilización de la subregla CH-B son:

- el valor del "index" (índice) está comprendido en la gama 0 a 255;
- la longitud de la codificación no es mayor que la longitud máxima que puede ser identificada por una codificación de forma larga.

NOTA – No es probable que la última condición sea una restricción en la práctica.

**8.21.4** Para la primera ocurrencia de una codificación CH-A, el "index" (índice) tendrá el valor cero y para cada ocurrencia siguiente se incrementará en uno.

**8.21.5** INFORMACIÓN – De este modo para cualquier valor de «identificación» dado, hay una codificación CH-A (relativamente ineficaz) con un valor de índice único y el valor de «identificación» completo, seguido de muchas codificaciones CH-B (eficaces) arbitrariamente enlazadas a la codificación CH-A por el valor de índice. Como la codificación CH-B utiliza un solo octeto para el índice, y un cómputo en octetos para la codificación, no se puede emplear si el valor del índice excede de 255 (256 «identificaciones» diferentes están en uso), o si la codificación no es un múltiplo de ocho bits. En estas circunstancias, se emplea la codificación CH-A para todos los casos.

**8.21.6** La codificación CH-A será la codificación BER del siguiente tipo de secuencia después de la aplicación de AUTOMATIC TAGS (rótulos automáticos) especificada en 22.6 y 26.3 de la Rec. UIT-T X.680 | ISO/CEI 8824-1:

```
[UNIVERSAL 29] IMPLICIT SEQUENCE {
    index                INTEGER,
    identification       CHOICE {
        syntaxes        SEQUENCE {
            abstract     OBJECT IDENTIFIER,
            transfer     OBJECT IDENTIFIER },
        syntax          OBJECT IDENTIFIER,
        presentation-context-id  INTEGER,
        context-negotiation SEQUENCE {
            presentation-context-id  INTEGER,
            transfer-syntax          OBJECT IDENTIFIER },
        transfer-syntax  OBJECT IDENTIFIER,
        fixed            NULL },
    string-value        OCTET STRING }
```

**8.21.7** El valor de "string-value" (valor de cadena) será la codificación del valor de cadena de caracteres abstracto que utiliza la sintaxis de transferencia de caracteres identificada, el valor del "index" (índice) será el determinado anteriormente, y el valor de los otros campos será el mismo que los valores que aparecen en el valor abstracto.

NOTAS

1 El componente "index" no está definido en la sintaxis abstracta porque ha de ser suministrado estrictamente a nivel de la regla de codificación (en el mismo sentido que los octetos de fin de contenido se han de aplicar a nivel de la regla de codificación).

2 En la sintaxis de transferencia ambas alternativas de "data-value" se codifican idénticamente en la sintaxis abstracta.

**8.21.8** La codificación de CH-B será la codificación BER del tipo:

**[UNIVERSAL 29] IMPLICIT OCTET STRING**

donde

- a) la codificación es primitiva;
- b) los octetos de longitud tienen la forma definida corta o larga como una opción del transmisor;
- c) los octetos de contenido codifican el "index" en el primer octeto del valor cadena de octetos como un entero cuyo valor va de 0 a 255, seguido de la codificación del "data-value".

NOTA – Un receptor puede distinguir la codificación CH-A de la codificación CH-B fijando el bit primitivo/constructor.

## 9 Reglas de codificación canónica

La codificación canónica de un valor de datos será la codificación básica descrita en la cláusula 8 junto con las siguientes restricciones y las enumeradas también en la cláusula 11.

### 9.1 Formas de longitud

Si la codificación es construida, empleará la forma de longitud indefinida. Si la codificación es primitiva, comprenderá el número mínimo de octetos de longitud necesarios. (En contraste con 8.1.3.2 b).)

### 9.2 Formas de codificación de cadenas

Los valores en forma de cadena de bits, cadena de octetos y cadena de caracteres restringidos se codificarán con una codificación primitiva si no requieren más de 1000 octetos de contenido y, en los demás casos, con codificación construida. Los fragmentos de cadena contenidos en la codificación construida se codificarán con una codificación primitiva. La codificación de cada fragmento, excepto quizá el último, tendrá 1000 octetos de contenido. (En contraste con 8.20.6.)

### 9.3 Componentes de conjunto (set)

Las codificaciones de los valores de componentes de un valor en forma de conjunto aparecerán en el orden determinado por sus rótulos según se especifica en 6.4 de la Rec. UIT-T X.680 | ISO/CEI 8824-1. Además, para determinar el orden en el cual se codifican los componentes cuando uno o más componentes es un tipo elección (choice) no rotulado, cada tipo de elección no rotulado se ordena como si tuviese un rótulo igual al rótulo más pequeño del tipo elección o cualesquiera tipos elección no rotulados anidados dentro.

**Ejemplo** – A continuación se supone un entorno de rotulación de IMPLICIT TAGS (rótulos implícitos)

```
A ::= SET
{
  a   [3] INTEGER,
  b   [1] CHOICE
  {
    c   [2] INTEGER,
    d   [4] INTEGER
  },
  e   CHOICE
  {
    f   CHOICE
    {
      g   [5] INTEGER,
      h   [6] INTEGER
    },
    i   CHOICE
    {
      j   [0] INTEGER
    }
  }
}
```

el orden en el cual se codifican los componentes del conjunto será siempre e, b, a, puesto que el rótulo [0] es el más bajo, después [1], después [3].

## 10 Reglas de codificación distinguida

La codificación de valores de datos empleada por las reglas de codificación distinguidas es la codificación básica descrita en la cláusula 8, junto con las siguientes restricciones y con las indicadas en la cláusula 11.

NOTA – La Rec. X.509 | ISO/CEI 9594-8 prohíbe la utilización de valores abstractos de base 10 en las aplicaciones de los directorios.

### 10.1 Formas de longitud

Se utilizará la forma definida de codificación de longitud, codificada en el número mínimo de octetos. (En contraste con 8.1.3.2 b.)

### 10.2 Formas de codificación de cadenas

Para tipos de cadena de bits, cadena de octetos y cadena de caracteres restringidos, no se utilizará la forma de codificación construida. (En contraste con 8.20.6.)

### 10.3 Componentes de conjunto

Las codificaciones de los valores de componentes de un valor de conjunto aparecerán en un orden determinado por sus rótulos, según se especifica en 6.4 de la Rec. UIT-T X.680 | ISO/CEI 8024-1:

NOTA – Cuando un componente del conjunto es un tipo elección no rotulado, la ubicación de ese componente en la ordenación dependerá del rótulo del componente de opción que se codifica.

## 11 Restricciones de las reglas de codificación básica empleadas por las reglas de codificación canónica y de codificación distinguida

Las referencias hechas en la cláusula 8 y en sus subcláusulas a «será la codificación BER» se interpretará como «será la codificación CER o DER, según proceda». (Véanse 8.16.1, 8.17.6, 8.18.1 y 8.21.6.)

### 11.1 Valores booleanos

Si la codificación representa el valor booleano TRUE, su único octeto de contenido tendrá los 8 bits puestos a uno. (En contraste con 8.2.2.)

### 11.2 Bit no utilizados

11.2.1 Cada bit no utilizado en el octeto final de la codificación de un valor de cadena de bit se pondrá a cero.

11.2.2 Cuando se aplique la subcláusula 19.7 de la Rec. UIT-T X.680 | ISO/CEI 8824-1, habrá que eliminar todos los bits 0 de cola de la cadena de bits antes de codificarla.

#### NOTAS

1 En el caso en que se haya aplicado una restricción de tamaño, el valor abstracto entregado por un decodificador a la aplicación será uno que satisfaga la restricción de tamaño y que difiera del valor transmitido solamente en el número de bits 0 de cola.

2 Si un valor de cadena de bits no tiene bits 1, un codificador codificará el valor con longitud 0 y sin octetos de contenido.

### 11.3 Valores reales

11.3.1 Si la codificación representa un valor real cuya base B es 2, se utilizará la codificación binaria que emplea base 2. Antes de codificar, se eligen la mantisa M y el exponente E de modo que M sea 0 o impar.

NOTA – Esto es necesario porque se puede considerar que el mismo valor real es {M, 2, E} y {M', 2, E'} con  $M \neq M'$  si, para algún entero n distinto de cero:

$$M' = M \times 2^{-n}$$

$$E' = E + n$$

Al codificar el valor, el factor de escala binaria F será cero, y M y E se representarán con el número mínimo de octetos necesarios.

**11.3.2** Si la codificación representa un valor real cuya base B es 10, se utilizará la codificación decimal. Al formar la codificación se aplica lo siguiente:

**11.3.2.1** Se utilizará la forma NR3 de ISO 6093 (véase 8.5.6).

**11.3.2.2** No se utilizará SPACE (espacio) dentro de la codificación.

**11.3.2.3** Si el valor real es negativo, comenzará con un signo menos (-), en los demás casos, comenzará con una cifra.

**11.3.2.4** Ni la primera ni la última cifra de la mantisa puede ser un 0.

**11.3.2.5** La última cifra de la mantisa será seguida inmediatamente de un punto (.), seguido de la marca de exponente "E".

**11.3.2.6** Si el componente tiene el valor 0, se escribirá "+0", en los demás casos, la primera cifra del exponente no será cero, y no se utilizará el signo más (+).

## **11.4 Valores de cadena general (GeneralString)**

La codificación de valores del tipo GeneralString (cadena general) (y sus subtipos) generará secuencias de escape para designar e invocar una nueva entrada de registro solamente cuando la entrada de registro del carácter sea diferente del designado actualmente como G0, C0 o C1. Todas las designaciones e invocaciones estarán en el conjunto G0 o en el conjunto C0.

NOTA – Se supone que cada carácter de un valor de cadena de caracteres está asociado con una entrada particular del registro internacional de juegos de caracteres codificados.

## **11.5 Componentes de conjunto y de secuencia con valor por defecto**

La codificación de un valor de conjunto o de un valor de secuencia no incluirá una codificación de ningún valor de componente que sea igual a su valor por defecto.

## **11.6 Componentes de conjunto de**

Las codificaciones de los valores de componentes de un valor de conjunto de aparecerán en orden ascendente, comparándose las codificaciones como cadenas de octetos.

## **11.7 Tiempo generalizado (GeneralizedTime)**

**11.7.1** La codificación terminará con una "Z", como se describe en la cláusula relativa a tiempo generalizado de la Rec. UIT-T X.680 | ISO/CEI 8824-1.

**11.7.2** Los elementos fracciones de segundo, si están presentes, omitirán todos los "0" finales; si los elementos corresponden a 0, se omitirán en su totalidad y se omitirá también el elemento coma decimal.

**Ejemplo** – Un elemento de segundos de "26,000" se representará como "26"; un elemento de segundos de "26,5200" se representará como "26,52".

**11.7.3** El elemento de coma decimal (decimal point), si está presente, será la opción coma ",",.

**11.7.4** La medianoche (GMT) se representará de la forma siguiente:

"YYYYMMDD000000Z"

donde "YYYYMMDD" representa el día siguiente a la medianoche en cuestión.

### **11.7.5 Ejemplos de representaciones válidas**

"19920521000000Z"

"19920622123421Z"

"19920722132100,3Z"

**11.7.6 Ejemplos de representaciones no válidas**

"19920520240000Z" (medianoche representada incorrectamente)

"19920622123421,0Z" (ceros de cola espurios)

"19920722132100,30Z" (ceros de cola espurios)

**12 Utilización de BER, CER y DER en la definición de sintaxis abstractas**

**12.1** Las reglas de codificación especificadas en la presente Recomendación | Norma Internacional pueden ser referenciadas y aplicadas cuando sea necesario especificar una representación de cadena de octetos inequívoca, no dividida y autolimitada para todos los valores de un tipo ASN.1.

NOTA – Todas estas cadenas de octetos son inequívocas dentro del ámbito de cada tipo ASN.1. No serían necesariamente inequívocas si estuvieran mezcladas con codificaciones de un tipo ASN.1 diferente.

**12.2** Se asignan los siguientes valores de identificador de objeto y de descriptor de objeto para identificar y describir las reglas de codificación básica especificadas en la presente Recomendación | Norma Internacional:

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

y

«Codificación básica de un tipo ASN.1»

**12.3** Se asignan los siguientes valores de identificador de objeto y de descriptor de objeto para identificar y describir las reglas de codificación canónica especificadas en la presente Recomendación | Norma Internacional:

{joint-iso-ccitt asn1(1) ber-derived(2) distinguished-encoding(1)}

y

«Codificación canónica de un tipo ASN.1»

**12.4** Se asignan los siguientes valores de identificador de objeto y de descriptor de objeto para identificar y describir las reglas de codificación distinguida especificadas en la presente Recomendación | Norma Internacional:

{joint-iso-ccitt asn1(1) ber-derived(2) distinguished-encoding(1)}

y

«Codificación distinguida de un tipo ASN.1»

**12.5** Cuando una especificación inequívoca define una sintaxis abstracta como un conjunto de valores de datos de presentación, cada uno de los cuales es un valor de algún tipo ASN.1 específicamente denominado, que suele ser (pero no es necesariamente) un tipo elección, se puede utilizar uno de los valores de identificador de objeto especificados en 12.2, 12.3 ó 12.4 con el nombre de la sintaxis abstracta para identificar las reglas de codificación básica, las reglas de codificación canónica o las reglas de codificación distinguida, respectivamente, al tipo ASN.1 específicamente denominado utilizado para definir la sintaxis abstracta.

**12.6** Los nombres especificados en 12.2, 12.3 y 12.4 no se utilizarán con un nombre de sintaxis abstracta para identificar una sintaxis de transferencia a menos que satisfagan las condiciones indicadas en 12.5 para la definición de la sintaxis abstracta (véase D.3 de la Rec. UIT-T X.680 | ISO/CEI 8824-1).

## Anexo A

### Ejemplos de codificaciones

(El presente anexo no es parte integrante de esta Recomendación | Norma Internacional)

Este anexo ilustra las reglas de codificación básica especificadas en esta Recomendación | Norma Internacional, mostrando la representación en octetos de un registro de personal (ficticio) que se define utilizando la ASN.1.

#### A.1 Descripción ASN.1 de la estructura del registro

La estructura del registro de personal ficticio se describe formalmente a continuación utilizando la notación ASN.1 especificada en la Rec. UIT-T X.680 | ISO 8824-1 para la definición de tipos.

```

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
    
```

#### A.2 Descripción ASN.1 de un valor del registro

A continuación se describe formalmente el valor del registro de personal de John Smith utilizando la ASN.1.

```

{ name {givenName "John",initial "P",familyName "Smith"},
  title "Director",
  number 51,
  dateOfHire "19710917",
  nameOfSpouse {givenName "Mary",initial "T",familyName "Smith"},
  children
    {{{givenName "Ralph",initial "T",familyName "Smith"},
      dateOfBirth "19571111"},
     {givenName "Susan",initial "B",familyName "Jones"},
      dateOfBirth "19590717"}}}
    
```

#### A.3 Representación de este valor del registro

A continuación se da la representación en octetos del valor del registro indicado anteriormente (después de aplicar las reglas de codificación básica definidas en esta Recomendación | Norma Internacional). Los valores de los identificadores, las longitudes y los contenidos de los valores enteros se muestran en hexadecimal, con dos cifras hexadecimales por octeto. Los valores de los contenidos de las cadenas de caracteres se muestran como textos, con un carácter por octeto.

Record	Length	Contents
60	8185	
	Name	Length Contents
	61	10
	VisibleString	Length Contents
	1A	04 "John"
	VisibleString	Length Contents
	1A	01 "P"

		VisibleString 1A	Length 05	Contents "Smith"			
Title A0	Length 0A	Contents					
		VisibleString 1A	Length 08	Contents "Director"			
Employee Number 42	Length 01	Contents 33					
Date of Hire A1	Length 0A	Contents					
		Date 43	Length 08	Contents "19710917"			
Name of Spouse A2	Length 12	Contents					
		Name 61	Length 10	Contents			
				VisibleString 1A	Length 04	Contents "Mary"	
				VisibleString 1A	Length 01	Contents "T"	
				VisibleString 1A	Length 05	Contents "Smith"	
[3] A3	Length 42	Contents					
		Set 31	Length 1F	Contents			
				Name 61	Length 11	Contents	
						VisibleString 1A	Length 05
						Contents "Ralph"	
						VisibleString 1A	Length 01
						Contents "T"	
						VisibleString 1A	Length 05
						Contents "Smith"	
				Date of Birth A0	Length 0A	Contents	
						Date 43	Length 08
						Contents "19571111"	
		Set 31	Length 1F	Contents			
				Name 61	Length 11	Contents	
						VisibleString 1A	Length 05
						Contents "Susan"	
						VisibleString 1A	Length 01
						Contents "B"	
						VisibleString 1	Length 05
						Contents "Jones"	
				Date of Birth A0	Length 0A	Contents	
						Date 43	Length 08
						Contents "19590717"	

## Anexo B

### Asignación de valores de identificador de objeto

(El presente anexo no es parte integrante de esta Recomendación | Norma Internacional)

En esta Recomendación | Norma Internacional se asignan los siguientes valores:

**Subcláusula Valor de identificador de objeto**

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

**Valor de descriptor de objeto**

«Codificación básica de un tipo ASN.1»

**Subcláusula Valor de identificador de objeto**

12.3 {joint-iso-ccitt asn1(1) ber-derived(2) canonical-encoding(0)}

**Valor de descriptor de objeto**

«Codificación canónica de un tipo ASN.1»

**Subcláusula Valor de identificador de objeto**

12.4 {joint-iso-ccitt asn1(1) ber-derived(2) distinguished-encoding(1)}

**Valor de descriptor de objeto**

«Codificación distinguida de un tipo ASN.1»

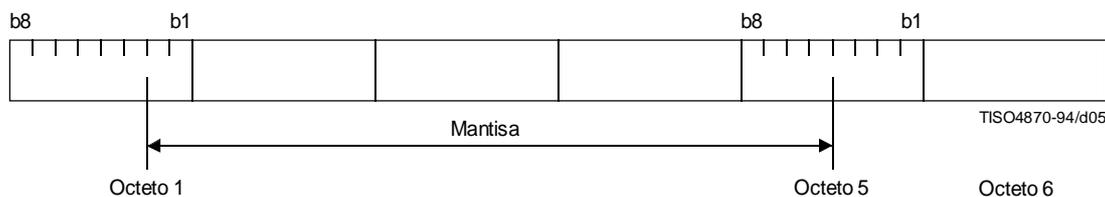
## Anexo C

### Ilustración de la codificación de valores reales

(El presente anexo no es parte integrante de esta Recomendación | Norma Internacional)

**C.1** Normalmente un emisor examinará su propia representación en soporte físico de coma flotante para determinar los algoritmos (independientes del valor) que se utilizarán para transferir valores entre esta representación de coma flotante y los octetos de contenido y longitud de la codificación de un valor ASN.1 real. Este anexo ilustra los pasos que se darían en dicho proceso al utilizar la representación en soporte físico (artificial) de coma flotante de la mantisa que se muestra en la Figura C.1.

Se supone que el exponente puede obtenerse fácilmente a partir del soporte físico de coma flotante, como un valor entero E.



**Figura C.1 – Representación de coma flotante**

**C.2** Los octetos de contenido que se necesita generar para enviar un valor distinto de cero utilizando codificación binaria (como se especifica en el texto de esta Recomendación | Norma Internacional) son:

1 S bb ff ee Octetos para E Octetos para N

donde S (signo de la mantisa) depende del valor que va a convertirse, bb es un valor fijo (por ejemplo 10) para representar la base (en este caso, se supone que la base es 16), ff es el valor F fijo calculado como se describe en C.3 y ee es la longitud fija del valor exponente calculada como se describe en C.4. (Este anexo no trata el caso en que E tiene que exceder de tres octetos.)

**C.3** El algoritmo transmitirá los octetos 1 a 5 de la representación en soporte físico como el valor de N, después de forzar los bits 8 a 3 del octeto 1 y los bits 4 a 1 del octeto 5 a cero. La coma decimal implícita se supone situada entre los bits 2 y 1 del octeto 1 en la representación de soporte físico que entrega el valor de E. Su posición implícita puede desplazarse al punto más cercano después del final del octeto 5, reduciendo el valor de E antes de la transmisión. En este sistema ilustrativo, se puede hacer un desplazamiento de 4 bits por cada decremento del exponente (puesto que se supone una base 16), de manera que un decremento de 9 colocará la coma implícita entre los bits 6 y 5 del octeto 6. Así, el valor de M es N multiplicado por  $2^3$ , para situar la coma correctamente en M. (La posición implícita en N, los octetos transferidos, está después del bit 1 del octeto 5). Se tienen así los parámetros cruciales:

$$F = 3 \quad (\text{con lo que ff vale } 11)$$

$$\text{decremento del exponente} = 9$$

**C.4** La longitud necesaria del exponente se calcula ahora obteniendo el número máximo de octetos necesarios para representar los valores

$$E_{\text{mín}} - \text{exceso} - \text{decremento del exponente}$$

$$E_{\text{máx}} - \text{exceso} - \text{decremento del exponente}$$

donde  $E_{\min}$  y  $E_{\max}$  son los valores enteros mínimo y máximo de la representación del exponente, el exceso es cualquier valor que debe sustraerse para producir el valor real del exponente, y el decremento del exponente es el calculado en C.3. Suponiendo que así se obtiene una longitud de 3 octetos, el valor de  $e$  es entonces de 10. Supóngase también que el exceso es cero.

**C.5** El algoritmo de transmisión consiste ahora en:

- a) transmitir el campo de octetos de identificación de las reglas de codificación básica con un rótulo para tipo ASN.1 real;
- b) comprobar si el valor es cero y, si lo es, transmitir un campo de longitud de las reglas de codificación básica ASN.1 con el valor de cero (sin octetos de contenido) y terminar el algoritmo;
- c) comprobar y recordar el signo de la mantisa, y tomar la opuesta de la mantisa si fuera negativa;
- d) transmitir un campo de longitud de las reglas de codificación básica ASN.1 con el valor de 9 y a continuación:
  - 11101110, si es negativa; o
  - 10101110, si es positiva.
- e) producir y transmitir el exponente de tres octetos con el valor

E – 9

- f) poner a cero los bits 8 a 3 del octeto 1 y los bits 4 a 1 del octeto 5 y transmitir entonces la mantisa de cinco octetos.

**C.6** El algoritmo receptor tiene que estar preparado para tratar cualquier codificación básica ASN.1, pero en este caso se puede utilizar directamente la unidad de coma flotante. Se procede como sigue:

- a) verificar el octeto 1 del contenido; si es 1x101110, se tiene una transmisión compatible, y se puede simplemente invertir el algoritmo emisor;
- b) en los demás casos, para la codificación de caracteres se invoca el soporte lógico normal de conversión de carácter decimal a coma flotante y se trata un "SpecialRealValue" (valor real especial) de acuerdo con la semántica de aplicación (probablemente fijando el número mayor y el número menor que puede tratar el soporte físico de coma flotante);
- c) para una transmisión binaria, poner N en la unidad de coma flotante, perdiendo octetos en el extremo menos significativo si fuera necesario, multiplicar por  $2^F$ , y por  $B^E$ , después volver negativo si fuera necesario. Los realizadores pueden encontrar una optimización posible en casos especiales, pero pueden igualmente encontrar que (aparte de la optimización relacionada con transmisiones desde una máquina compatible), las pruebas necesarias hacen perder más de lo que se gana con dichas optimizaciones.

**C.7** Los algoritmos arriba mencionados sólo sirven de ilustración. Los realizadores determinarán, naturalmente, sus propias estrategias.

## Anexo D

### Utilización de las DER y las CER en la autenticación de origen de los datos

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

#### D.1 Problema que ha de resolverse

**D.1.1** Las DER y las CER se han establecido para facilitar la provisión de mecanismos de seguridad de la integridad utilizando autenticadores para el material que ha de transferirse.

NOTA – Para simplificar, en el resto de este anexo sólo se mencionan las DER. No obstante, el texto se aplica también a las CER.

**D.1.2** El concepto de autenticador es fácilmente comprensible e implica tomar el esquema de bits que ha de transferirse, aplicarle alguna forma de función de elección arbitraria (hash) para reducirlo a unos cuantos octetos, encriptar esos octetos para autenticar el autenticador y después transmitir el autenticador con el material original (que se envía en claro). En la recepción, el autenticador es calculado de nuevo a partir del texto recibido en claro y es comparado con el autenticador recibido. Si son iguales, el texto no ha sido adulterado, y en caso contrario, sí lo habrá sido.

**D.1.3** Este concepto sencillo se hace más difícil cuando se utiliza el modelo de ISO y, en particular, la capa de presentación.

**D.1.4** Se plantean dos problemas. El primero es una cuestión de modelado y de la llamada independencia de capa; el segundo se refiere al empleo de retransmisiones en la capa de aplicación, tal como se utilizan en la Recomendación X.400.

**D.1.5** Por lo que se refiere al asunto del modelado, la función de elección arbitraria y el algoritmo de encriptación forman parte del funcionamiento de la aplicación, pero la aplicación no tiene conocimiento ni control de la codificación real que utilizará la capa de presentación. De manera similar en recepción, la decodificación y, por consiguiente, la destrucción de la cadena de bits en recepción, es un asunto de la capa de presentación. Se han propuesto cuatro soluciones para resolver este problema:

- a) descartar la utilización de los octetos reales producidos por la capa de presentación para el autenticador (el criterio actual adoptado por expertos en presentación y del grupo ULA);
- b) poner los mecanismos de elección arbitraria y de autenticador en la propia capa de presentación (esta solución se rechazó como parte de la cuestión general de apoyar la encriptación en ASN.1; en el momento del rechazo se arguyó, para justificarlo, que el trabajo sobre seguridad todavía no estaba maduro y que no se quería prejuzgar el posible resultado);
- c) modelar una interacción compleja con la capa de presentación según la cual, en transmisión, se presenta un valor para codificación, se produce la codificación y se devuelve a la capa de aplicación, que calcula el autenticador y a continuación se transmite todo; en recepción, además de producir el valor abstracto, se pasan las codificaciones recibidas a la capa de aplicación para comprobar el autenticador (este modelo fue rechazado por el grupo ULA);
- d) efectuar toda la codificación en la capa de aplicación y no utilizar los servicios de presentación para negociar la sintaxis de transferencia (esto es en realidad un rechazo del modelo de referencia de OSI y no sería aceptable como solución generalizada).

**D.1.6** Podría argüirse que la falta de acuerdo sobre un modelo para describir un proceso aparentemente sencillo y realizable (producir la codificación, después el autenticador, transmitir ambos y comprobar con el autenticador en recepción) es algo que no debería aceptarse como una posición a largo plazo. Esta observación sería ciertamente válida si no fuese por el segundo problema, el de retransmisiones de aplicación, y si no hubiera ninguna otra solución viable. (En este anexo se indica una solución alternativa, utilizada en la Rec. X.509 del CCITT | ISO 9594-8, que no plantea los problemas relativos al modelado y al sistema de retransmisión y que es factible.)

**D.1.7** El segundo problema consiste en que, si hay una retransmisión de aplicación, la sintaxis de transferencia utilizada para la segunda transmisión puede diferir de la convenida para la primera (por ejemplo, la utilización de DER en una de ellas y de BER en la otra). Con ello se anularía el autenticador, a menos que éste se abriera y se calculara de nuevo en la retransmisión, lo que implicaría intercambios de seguridad con el relevo, siendo así que lo que se requiere es seguridad de extremo a extremo.

NOTA – Se han hecho sugerencias en el sentido de que se pudiera desear señalar un contexto de presentación como «no decodificar/recodificar en retransmisiones de aplicación», pero esto también causa problemas de modelado y de otro tipo.

**D.1.8** Así pues, nos vemos forzados a intentar trabajar con un modelo en el que la capa de presentación (junto con cualesquiera retransmisiones de aplicación que intervengan) permite la transferencia de la sintaxis abstracta y de la semántica de la información, pero no garantiza que la codificación efectiva del esquema de bits (la sintaxis de transferencia) se mantendrá de extremo a extremo.

**D.1.9** Lo que hay que conseguir, por tanto, es un mecanismo autenticador que pueda funcionar en el tipo de datos abstractos, más bien que en la cadena de bits transmitida.

**D.1.10** El grupo de Directorios fue el primero en tratar de solucionar este problema, por lo que su modelo se describe a continuación.

## **D.2 Planteamiento de una solución**

**D.2.1** En el texto siguiente se describe, en primer lugar, un modelo conceptual de lo que se está haciendo, seguido por una optimización de la realización que elimina la doble codificación/decodificación implicada en el modelo conceptual.

### **D.2.2 El modelo conceptual funciona de la siguiente manera:**

- a) El emisor, en la capa de aplicación, convierte el valor de sintaxis abstracta en una cadena de bits utilizando las DER y genera el autenticador a partir de esa cadena de bits, añadiéndolo al valor de sintaxis abstracta; ambos valores se transmiten utilizando mecanismos normales de capa de presentación y cualquier sintaxis de transferencia. Conceptualmente, el emisor codifica dos veces: una para el autenticador (utilizando las DER) en la capa de aplicación, y otra para la transferencia efectiva (utilizando la sintaxis de transferencia negociada) en la capa de presentación.

NOTA – La cadena de bits producida por las DER tienen la importante propiedad de que está en correspondencia biunívoca con el valor abstracto. Por eso, la transferencia de extremo a extremo sin pérdida de información al nivel de sintaxis abstracta es equivalente a la transferencia de extremo a extremo de la cadena de bits en la que se basa el autenticador.

- b) El receptor decodificará la cadena de bits recibida en la capa de presentación, utilizando la sintaxis de transferencia negociada (que puede diferir de la utilizada por el emisor si hay una retransmisión de aplicación), y pasará el valor abstracto a la aplicación. En la capa aplicación, el valor abstracto se recodifica utilizando las DER para producir la cadena de bits que ha de autenticarse.

**D.2.3** Así pues, conceptualmente, se codifica dos veces en el extremo emisor y se decodifica una vez y a continuación se codifica en el extremo receptor. Es posible que los realizadores decidan hacer esto, efectivamente, si el código que sustenta el funcionamiento de la capa de presentación procede de un suministrador distinto del que produce el código que sustenta la aplicación. En esta etapa no está claro en qué medida constituiría esto una tara importante. Cuando se utiliza una realización integrada, existe, sin embargo, la posibilidad de la optimización descrita más adelante. Hay que señalar además que las DER no son más difíciles de aplicar que las BER, excepto en relación con la utilización de «conjunto de». Si ha de tratarse un gran conjunto de, quizá la realización necesite invocar una rutina de clasificación basada en disco. Los diseñadores de aplicaciones han de ser conscientes de esto y tratar de utilizar «secuencia de» en vez de «conjunto de», cuando se contemple el empleo de las DER.

## **D.3 Optimización de la realización**

**D.3.1** El modelo de OSI y las normas de protocolo especifican el comportamiento requerido; no tratan de limitar, en absoluto, la arquitectura y la estructura del código de la realización real. Un realizador puede, por consiguiente, producir el efecto deseado del modo que él elija.

**D.3.2** En el extremo emisor se puede mantener la cadena de bits producida (conceptualmente en la capa de aplicación) y utilizarla para facilitar la decodificación que se efectúa conceptualmente en la capa de presentación. Esto es adecuado para la emisión si la sintaxis de transferencia negociada es BER o DER de ASN.1. Si no es ninguna de éstas, será necesaria una doble codificación.

**D.3.3** De manera similar, en el extremo receptor, la cadena de bits recibida puede ser retenida (para cualquier sintaxis de transferencia) y la realización puede utilizarla para comprobar el autenticador. Si hay concordancia, se termina el problema; si no la hay, puede tratarse de un problema de sintaxis de transferencia, y es necesaria la recodificación a partir del valor abstracto para determinar si se produjo o no adulteración.

**D.3.4** Para que la probabilidad de no tener que efectuar una doble codificación/decodificación sea máxima, se aconseja que los sistemas que utilizan este mecanismo traten de negociar una sintaxis de transferencia de las DER (empleando el identificador de objeto apropiado) como primera preferencia, recurriendo a continuación a las BER (primera preferencia) o a algunas otras reglas de codificación (segunda preferencia).