



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

**UIT-T**

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

**X.209**

**INTERCONEXIÓN DE SISTEMAS ABIERTOS**

**MODELO Y NOTACIÓN**

---

**ESPECIFICACIÓN DE LAS REGLAS BÁSICAS  
DE CODIFICACIÓN DE LA NOTACIÓN DE  
SINTAXIS ABSTRACTA UNO (NSA.1)**

**Recomendación UIT-T X.209**

(Extracto del *Libro Azul*)

---

## NOTAS

1 La Recomendación UIT-T X.209 se publicó en el fascículo VIII.4 del Libro Azul. Este fichero es un extracto del Libro Azul. Aunque la presentación y disposición del texto son ligeramente diferentes de la versión del Libro Azul, el contenido del fichero es idéntico a la citada versión y los derechos de autor siguen siendo los mismos (Véase a continuación).

2 Por razones de concisión, el término «Administración» se utiliza en la presente Recomendación para designar a una administración de telecomunicaciones y a una empresa de explotación reconocida.

© UIT 1988, 1993

Reservados todos los derechos. No podrá reproducirse o utilizarse la presente Recomendación ni parte de la misma de cualquier forma ni por cualquier procedimiento, electrónico o mecánico, comprendidas la fotocopia y la grabación en micropelícula, sin autorización escrita de la UIT.

**ESPECIFICACIÓN DE LAS REGLAS BÁSICAS DE CODIFICACIÓN  
DE LA NOTACIÓN DE SINTAXIS ABSTRACTA UNO (NSA.1)<sup>1)</sup>**

*(Melbourne, 1988)*

EL CCITT,

*considerando*

- a) la variedad y complejidad de los objetos de información cursados en la capa de aplicación;
- b) la necesidad de una notación de alto nivel para especificar tales objetos de información;
- c) la conveniencia de destacar y normalizar las reglas para la codificación de tales objetos de información,

*recomienda por unanimidad*

que las reglas para la codificación de objetos de información sean las definidas en la presente Recomendación.

**ÍNDICE**

0	<i>Introducción</i>
1	<i>Objeto y campo de aplicación</i>
2	<i>Referencias</i>
3	<i>Definiciones</i>
4	<i>Abreviaturas y notación</i>
4.1	Abreviaturas
4.2	Notación
5	<i>Conformidad</i>
6	<i>Reglas generales de codificación</i>
6.1	Estructura de la codificación
6.2	Octetos de identificación
6.3	Octetos de longitud
6.4	Octetos de contenido
6.5	Octetos de fin de contenido
7	<i>Codificación de un valor boolean (booleano)</i>
8	<i>Codificación de un valor integer (entero)</i>
9	<i>Codificación de un valor enumerated (enumerado)</i>

---

<sup>1)</sup> La Recomendación X.209 y la norma ISO 8825, [Information processing systems – Open systems interconnection – Specification of basic encoding rules for Abstract Syntax Notation One (ASN.1)], ampliada por el Addéndum 1 a ISO 8825, fueron preparadas en estrecha colaboración y están técnicamente armonizadas.

- 10 *Codificación de un valor real*
- 11 *Codificación de un valor bitstring (cadena de bits)*
- 12 *Codificación de un valor octetstring (cadena de octetos)*
- 13 *Codificación de un valor null (nulo)*
- 14 *Codificación de un valor sequence (secuencia)*
- 15 *Codificación de un valor sequence-of (secuencia-de)*
- 16 *Codificación de un valor set (conjunto)*
- 17 *Codificación de un valor set-of (conjunto-de)*
- 18 *Codificación de un valor choice (elección)*
- 19 *Codificación de un valor selection (selección)*
- 20 *Codificación de un valor tagged (rotulado)*
- 21 *Codificación de un valor tipo ANY (CUALQUIERA)*
- 22 *Codificación de un valor object-identifier (identificador de objeto)*
- 23 *Codificación de valores del tipo character string (cadena de caracteres)*
- 24 *Codificación de valores de los tipos útiles en NSA.1*
- 25 *Uso en la definición de la sintaxis de transferencia*

*Apéndice I* - Ejemplos de codificaciones

- I.1 Descripción NSA.1 de la estructura de registro
- I.2 Descripción NSA.1 de un valor de registro
- I.3 Representación de este valor de registro

*Apéndice II* - Asignación de valores identificadores de objeto

*Apéndice II* - Ilustración de la codificación de valores reales

## **0 Introducción**

La Recomendación X.208 (Especificación de la notación de sintaxis abstracta 1) especifica una notación para la definición de sintaxis abstractas, permitiendo que las especificaciones de la capa de aplicación definan los tipos de información que las mismas necesitan transferir utilizando el servicio de presentación. También especifica una notación para la especificación de valores de un tipo definido.

Esta Recomendación define un conjunto de reglas de codificación que pueden aplicarse a los valores de los tipos definidos mediante la notación especificada en la Recomendación X.208. La aplicación de estas reglas de codificación produce una sintaxis de transferencia para dichos valores. En la especificación de estas reglas de codificación está implícito el hecho de que las mismas se utilicen también para la decodificación.

Puede haber más de un conjunto de reglas de codificación que puedan ser aplicadas a valores de los tipos definidos mediante la notación de la Recomendación X.208. Esta Recomendación define un conjunto de reglas de codificación, denominadas reglas básicas de codificación.

Esta Recomendación está técnica y redaccionalmente armonizada con la norma ISO 8825 más el Addendum 1 a la norma ISO 8825.

El apéndice I proporciona ejemplos de la aplicación de las reglas de codificación. No forma parte de esta Recomendación.

El apéndice II resume la asignación de valores identificadores de objeto establecida en esta Recomendación. No forma parte de esta Recomendación.

El apéndice III no forma parte de esta Recomendación y ofrece ejemplos de aplicación de la codificación de valores reales.

## **1 Objeto y campo de aplicación**

Esta Recomendación especifica un conjunto de reglas básicas de codificación que pueden utilizarse para derivar la especificación de una sintaxis de transferencia para valores de los tipos definidos mediante la notación especificada en la Recomendación X.208. Estas reglas básicas de codificación también se aplican a la decodificación de dicha sintaxis de transferencia, para poder identificar los valores de datos que se transfieren.

Estas reglas básicas de codificación se utilizan en el momento de la comunicación (por el proveedor del servicio de presentación cuando así lo requiere el contexto de la presentación).

## **2 Referencias**

- [1] Recomendación X.200, *Modelo de referencia de interconexión de sistemas abiertos para aplicaciones del CCITT* (véase también la norma ISO 7498).
- [2] Recomendación X.208, *Especificación de la notación de sintaxis abstracta uno NSA.1* (véase también la norma ISO 8824).
- [3] Recomendación X.226, *Especificación del protocolo de presentación para la interconexión de sistemas abiertos para aplicaciones del CCITT* (véase también la norma ISO 8823).
- [4] ISO 2022, *Information processing - ISO 7-bit and 8-bit coded character sets - Code extension techniques*.
- [5] ISO 2375, *Data processing - Procedure for registration of escape sequences*.
- [6] ISO 6093, *Information processing - Representation of numerical values in character strings for information interchange*.

## **3 Definiciones**

En esta Recomendación se utilizan las definiciones de la Recomendación X.208.

### **3.1 conformidad dinámica**

Declaración de la necesidad de que una realización se amolde al comportamiento prescrito por esta Recomendación en una instancia de 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.

### **3.3 valor de datos**

Información especificada como el valor de un tipo; el tipo y el valor se definen utilizando la notación NSA.1

### **3.4 codificación (de un valor de datos)**

Secuencia completa de octetos utilizada para representar el valor de datos.

*Nota* - En algunas Recomendaciones del CCITT se utiliza el término “elemento de datos” para designar esta secuencia de octetos, pero en esta Recomendación no se utiliza ya que en las Normas Internacionales ISO el mismo significa “valor de datos”.

### **3.5 octetos de identificación**

Parte de la codificación de un valor de datos que se utiliza para identificar el tipo del valor.

### 3.6 **octetos de longitud**

Parte de la codificación de un valor de datos que sigue a los octetos de identificación y que se utiliza para determinar el final de la codificación.

### 3.7 **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.

### 3.8 **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.

### 3.9 **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 **emisor**

Realización que codifica un valor de datos para su transferencia.

### 3.12 **receptor**

Realización que decodifica los octetos producidos por un emisor, para poder identificar el valor de datos que fue codificado.

## 4 **abreviaturas y notación**

### 4.1 *Abreviaturas*

NSA.1 Notación de sintaxis abstracta uno.

### 4.2 *Notación*

4.2.1 Esta Recomendación hace referencia a la notación definida en la Recomendación X.208.

4.2.2 Esta Recomendación especifica el valor de cada octeto de una codificación mediante los términos “bit más significativo” y “bit menos significativo”.

*Nota* - Las especificaciones de capa inferior utilizan la misma notación para definir el orden de una transmisión de bits por una línea serie, o la asignación de los bits a canales paralelos.

4.2.3 Para los efectos de esta Recomendación, los bits de un octeto están numerados de 8 a 1, donde el bit 8 es el “bit más significativo”, y el bit 1 es el “bit menos significativo”.

## 5 **Conformidad**

5.1 La conformidad dinámica se especifica en los § 6 a 24 inclusive.

5.2 La conformidad se especifica en documentos que especifican la aplicación de estas reglas básicas de codificación.

5.3 Esta Recomendación permite codificaciones alternativas como opción del emisor. Los receptores conformes deberán admitir todas las alternativas.

*Nota* - En el § 6.3.2 b) y en el cuadro 2/X.209 aparecen ejemplos de dichas codificaciones alternativas.

## 6 Reglas generales de codificación

### 6.1 Estructura de la codificación

6.1.1 La codificación de un valor de datos constará de cuatro componentes, que deben aparecer en el siguiente orden:

- a) octetos de identificación (véase el § 6.2);
- b) octetos de longitud (véase el § 6.3);
- c) octetos de contenido (véase el § 6.4);
- d) octetos de fin de contenido (véase el § 6.5).

6.1.2 Los octetos de fin de contenido sólo estarán presentes si así lo requiere el valor de los octetos de longitud (véase el § 6.3).

6.1.3 La figura 1/X.209 ilustra la estructura de una codificación (primitiva o construida). La figura 2/X.209 ilustra una codificación construida alternativa.

### 6.2 Octetos de identificación

6.2.1 Los octetos de identificación codificarán el rótulo NSA.1 (clase y número) del tipo del valor de datos.

6.2.2 En los rótulos con un número comprendido entre cero y 30 (inclusive), los octetos de identificación incluirán un único octeto, codificado como sigue:

a) los bits 8 y 7 estarán codificados para representa la clase de rótulo, tal como se especifica en el cuadro 1/X.209;

b) el bit 6 será cero o uno, de acuerdo con las normas del § 6.2.5;

c) los bits de 5 a 1 codificarán el número del rótulo como un entero binario, siendo el bit 5 el más significativo.

6.2.3 La figura 3/X.209 ilustra la forma de un octeto de identificación para un tipo con un rótulo cuyo número está comprendido entre cero y 30 (inclusive).

6.2.4 Para los rótulos con número mayor o igual que 31, el identificador incluirá un octeto de cabecera seguido por uno o más octetos subsiguientes.

CUADRO 1/X.209

#### 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

6.2.4.1 El octeto de cabecera se codificará como sigue:

- a) los bits 8 y 7 se codificarán para representar la clase de rótulo, tal como se especifica en el cuadro 1/X.209;
- b) el bit 6 será cero o uno, de acuerdo con las normas del § 6.2.5;
- c) los bits 5 a 1 se codificarán 11111<sub>2</sub>.

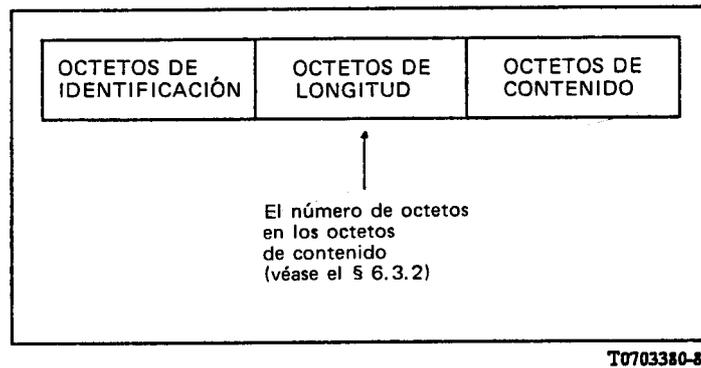


FIGURA 1/X.209  
Estructura de una codificación

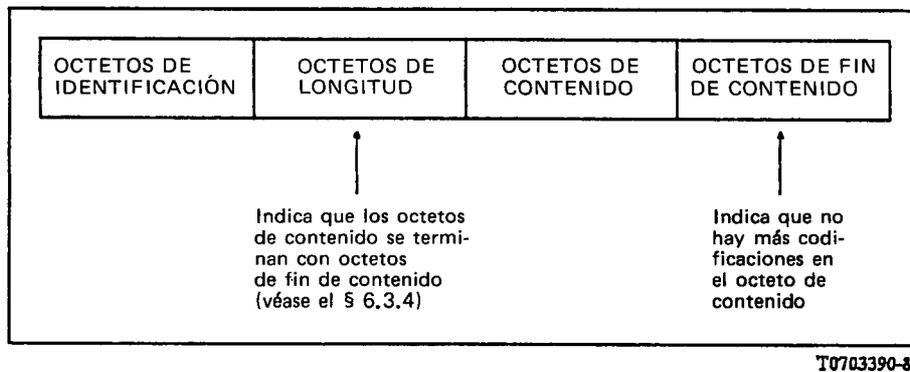


FIGURA 2/X.209  
Codificación construida alternativa

6.2.4.2 Los octetos subsiguientes codificarán el número del rótulo como sigue:

- a) el bit 8 de cada octeto se pondrá a uno, a menos que sea el último octeto de los octetos de identificación;
- 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, e incluido, el último octeto subsiguiente de los octetos de identificación, se codificarán como un entero binario sin signo, igual al número del rótulo, siendo el bit 7 del primer octeto subsiguiente el bit más significativo;
- c) los bits 7 a 1 del primer octeto subsiguiente no serán todos cero.

6.2.4.3 La figura 4/X.209 ilustra la forma de los octetos de identificación para un tipo cuyo rótulo tiene un número superior a 30.

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

*Nota* - Las normas que siguen especifican, para cada tipo si la codificación es primitiva o construida.



6.3.3.1 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 más significativo.

*Ejemplo:*

$L = 38$  puede codificarse  $00100110_2$

6.3.3.2 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 de longitud como un entero binario sin signo, con el bit 7 como bit más significativo;
- c) no se utilizará el valor  $11111111_2$ .

*Nota* - 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:*

$L = 201$  puede codificarse:  $10000001_2$   
 $11001001_2$

*Nota* - En la forma larga, el emisor puede, facultativamente, emplear más octetos de longitud que el mínimo necesario.

6.3.4 En la forma indefinida, los octetos de longitud indican que los octetos de contenido se terminan por octetos de fin de contenido (véase el § 6.5), y estarán formados por un solo octeto.

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

6.3.4.2 Si se utiliza esta forma de longitud, los octetos de fin de contenido (véase el § 6.5) estarán presentes en la codificación después de los octetos de contenido.

#### 6.4 *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 los puntos que siguen.

*Nota* - Los octetos de contenido dependen del tipo del valor de datos; los siguientes puntos siguen la misma secuencia que la definición de tipos en NSA.1.

#### 6.5 *Octetos de fin de contenido*

Los octetos de fin de contenido estarán presentes si la longitud se codifica de acuerdo con lo especificado en el § 6.3.4; 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
$00_{16}$	$00_{16}$	Ausente

### 7 **Codificación de un valor booleano (booleano)**

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

7.2 Si el valor booleano es

FALSE

el octeto será cero.

### 7.2.1 Si el valor booleano es

TRUE

el octeto tendrá cualquier valor no 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 Codificación de un valor integer (entero)

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

8.2 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:

- no serán todos uno; y
- no serán todos cero.

*Nota* - Estas reglas garantizan que un valor entero se codifique siempre con el menor número de octetos posible.

8.3 Los octetos de contenido serán un número binario complementario de 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 complementario de 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 complementario de 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.

## 9 Codificación de un valor enumerated (enumerado)

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

## 10 Codificación de un valor real

10.1 La codificación de un valor real será primitiva.

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

10.3 Si el valor es no 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 el § 10.5. Si B' es 10, se utilizará una codificación de carácter, especificada en el § 10.6.

*Nota* - La forma de almacenamiento, generación o procesamiento por los emisores y receptores, así como la forma utilizada en la notación de valor NSA.1, son independientes de la base utilizada para la transferencia.

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

- si el bit 8 = 1, se aplica la codificación binaria especificada en el § 10.5;
- si el bit 8 = 0 y el bit 7 = 0, se aplica la codificación decimal especificada en el § 10.6;
- si el bit 8 = 0 y el bit 7 = 1, se codifica un "SpecialRealValue" (ValorRealEspecial) (véase la Recomendación X.208), tal como se especifica en el § 10.7.

10.5 Cuando se utiliza la codificación binaria (bit 8 = 1), si la mantisa M es no 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* - Esta libertad para la elección de F tiene por objeto facilitar la generación del formato de transferencia, eliminando la necesidad de alinear la coma decimal implícita de la mantisa con un límite de octeto (véase el apéndice III). La existencia de F no complica en forma apreciable la tarea de los receptores.

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

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

Bits 6 a 5	Base
00	base 2
01	base 8
10	base 16
11	Reservado para versiones futuras de esta Recomendación

10.5.3 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.

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

- si los bits 2 a 1 son 00, el segundo octeto de contenido codifica el valor del exponente como un número binario complementario de dos;
- si los bits 2 a 1 son 01, el segundo y tercer octetos de contenido codifican el valor del exponente como un número binario complementario de dos;
- 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 complementario de dos;
- si los bits 2 a 1 son 11, el segundo octeto de un 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 número binario complementario de dos; el valor de X será al menos uno; los nueve primeros bits del exponente transmitido no serán todos cero o todos uno.

10.5.5 Los restantes octetos de contenido codifican el valor del entero N (véase el § 10.5) como un número binario sin signo.

*Nota 1* - Esta codificación no especifica una representación “normalizada”, ya que hay cierto número de representaciones posibles de cada valor (excepto cero). Esta variación es facultativa del emisor, y puede ser utilizada como una indicación general de precisión.

*Nota 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 apéndice III).

10.6 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 la norma ISO 6093, de una longitud elegida por el emisor, y codificada de acuerdo a la norma 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:

Bits 6 a 1	Representación del número
00 0001	Forma NR1 de ISO 6093
00 0010	Forma NR2 de ISO 6093
00 0011	Forma NR3 de ISO 6093

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

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

*Nota 2* - No se utilizarán los factores de escala especificados en la documentación adjunta (véase ISO 6093).

*Nota 3* - El uso de la forma normalizada (véase ISO 6093) es una opción del emisor, y carece de significación.

10.7 Cuando se deban codificar “SpecialRealValues” (ValoresRealesEspeciales) (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 otros valores que tienen los bits 8 y 7 iguales a 0 y 1 respectivamente, están reservados para versiones futuras de esta Recomendación.

## 11 Codificación de un valor bitstring (cadena de bits)

11.1 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.

11.2 Los octetos de contenido para la codificación primaria contendrán un octeto inicial seguido por cero, uno o más octetos subsiguientes.

11.2.1 Los bits de la cadena de bits, comenzando por el primero y continuando hasta el bit postrero, 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 siguiente, seguidos por tantos bits como sea necesario en el octeto subsiguiente final, comenzando con el bit 8.

*Nota* - La notación “primer bit” y “bit postrero” se especifica en la Recomendación X.208.

11.2.2 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.

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

11.3 Los octetos de contenido para la codificación construida estarán formados por la codificación completa de cero, uno o más valores de datos.

*Nota* - Cada una de dichas codificaciones incluye octetos de identificación, de longitud y de contenido, y puede incluir, si es construida, octetos de fin de contenido.

11.3.1 Cada codificación de valor de datos en los octetos de contenido será la codificación de un valor del tipo BIT STRING.

*Nota* - En especial, los r tulos de los octetos de contenido son siempre de clase universal, n mero 3.

11.3.2 Los bits del valor de cadena de bits que se codifica, comenzando con el primer bit y continuando hasta el bit postrero, deber n aparecer entre el primer bit y el  ltimo bit del primer valor de datos codificado en los octetos de contenido, seguidos por el primer bit y hasta el  ltimo bit del segundo valor de datos codificado en los octetos de contenido, seguidos por el primer bit y hasta el  ltimo bit de cada valor de datos, seguidos a su vez por el primer bit hasta el  ltimo bit del  ltimo valor de datos codificado en los octetos de contenido.

11.3.3 Cada valor de datos codificado en los octetos de contenido, con excepci n del  ltimo, estar  formado por un n mero de bits m ltiplo de 8.

*Nota* - Un valor de datos codificado en los octetos de contenido puede ser una cadena de bits de longitud cero.

11.3.4 Cuando se utiliza codificaci n construida, no se dar  significaci n al l mite entre los valores de datos codificados en los octetos de contenido.

11.3.5 La codificaci n de cada valor de datos en los octetos de contenido puede ser primitiva o construida.

*Nota* - Generalmente, dicha codificaci n es primitiva.

*Ejemplo* - Si es del tipo BIT STRING, el valor '0A3B5F291CD'H puede codificarse como se muestra m s abajo. En este ejemplo, la cadena de bits est  representada como primitiva:

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

El valor anterior puede codificarse tambi n como se muestra a continuaci n. 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>	000A3B <sub>16</sub>
	Cadena de bits	Longitud	Contenido
	03 <sub>16</sub>	05 <sub>16</sub>	045F291CD0 <sub>16</sub>
	EOC	Longitud	
	00 <sub>16</sub>	00 <sub>16</sub>	

## 12 Codificaci n de un valor octetstring (cadena de octetos)

12.1 La codificaci n de un valor de cadena de octetos puede ser primitiva o construida, a elecci n 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 codificaci n construida.

12.2 La codificaci n primitiva contiene cero, uno o m s 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 m s significativo de un octeto de valor de datos alineado con el bit m s significativo de uno de los octetos de contenido.

12.3 En la codificaci n construida, los octetos de contenido estar n formados por la codificaci n completa de cero, uno o m s valores de datos.

*Nota* - Cada una de dichas codificaciones incluye octetos de identificaci n, de longitud y de contenido, y puede incluir octetos de fin de contenido si es construida.

12.3.1 Cada codificaci n de valor de datos en los octetos de contenido ser  la codificaci n de un valor del tipo cadena de octetos.

*Nota* - En especial, los r tulos de los octetos de contenido son siempre de clase universal, n mero 4.

12.3.2 Los octetos del valor de cadena de octetos que se codifica, comenzando con el primer octeto y siguiendo hasta el  ltimo octeto, se colocarn n en el primero y hasta el  ltimo octeto del primer valor de datos codificado en los octetos de contenido, seguidos por el primero y hasta el  ltimo octeto del segundo valor de datos codificado en los octetos de contenido, seguido por el primero y hasta el  ltimo octeto de cada valor de datos adicional, seguidos por el primero y hasta el  ltimo octeto del  ltimo valor de datos codificado en los octetos de contenido.

*Nota* - Un valor de datos codificado en los octetos de contenido puede ser una cadena de octetos de longitud cero.

12.3.3 Cuando se utilice codificación construida, no se dará significación al límite entre los valores de datos codificados en los octetos de contenido.

12.3.4 La codificación de cada valor de datos en los octetos de contenido puede ser primitiva o construida.

*Nota* - De ordinario, dicha codificación es primitiva.

### 13 Codificación de un valor nulo

13.1 La codificación de un valor nulo será primitiva.

13.2 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	<i>Longitud</i>
15 <sub>16</sub>	00 <sub>16</sub>

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

14.1 La codificación de un valor secuencia será construida.

14.2 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 de secuencia NSA.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".

14.3 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 NSA.1.

*Ejemplo* - Si es del tipo

SEQUENCE {name IA5String, ok BOOLEAN}

el valor

{name "Smith", ok TRUE}

puede codificarse como:

Sequence	<i>Longitud</i>	<i>Contenido</i>
30 <sub>16</sub>	0A <sub>16</sub>	
	IA5String	<i>Longitud</i>
	16 <sub>16</sub>	05 <sub>16</sub>
	Boolea	<i>Contenido</i>
	01 <sub>16</sub>	01 <sub>16</sub>
		FF <sub>16</sub>

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

15.1 La codificación de un valor secuencia-de será construida.

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

15.3 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 a codificar.

## 16 Codificación de un valor set (conjunto)

16.1 La codificación de un valor de conjunto será construida.

16.2 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 NSA.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 “DEFAULT”.

16.3 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 limitaciones al orden durante la transferencia.

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

17.1 La codificación de un valor conjunto-de será construida.

17.2 Se aplica el texto del § 15.2.

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

## 18 Codificación de un valor choice (elección)

La codificación de un valor de elección será la misma que la de un valor de tipo elegido.

*Nota 1* - La codificación puede ser primitiva o construida, según el tipo elegido.

*Nota 2* - La etiqueta utilizada en los octetos de identificación es la etiqueta del tipo elegido, tal como se especifica en la definición NSA.1 del tipo elegido.

## 19 Codificación de un valor de selection (selección)

La codificación de un valor de selección será la misma que la de un valor de tipo seleccionado.

*Nota* - La codificación puede ser primitiva o construida, según el tipo seleccionado.

## 20 Codificación de un valor tagged (rotulado)

20.1 La codificación de un valor rotulado se derivará de la codificación completa del valor de datos correspondiente al tipo que aparece en la notación “TipoRotulado” (denominada codificación de base), tal como se especifica en los § 20.2 y 20.3.

20.2 Si no se utilizó la palabra clave “IMPLICIT” en la definición del tipo, la codificación será construida y los octetos de contenido serán la codificación de base completa.

20.3 Si se utilizó la palabra clave “IMPLICIT” en la definición de tipo:

a) la codificación será construida si la codificación de base es construida; de no ser así, será primitiva; y

b) los octetos de contenido serán los mismos que los octetos de contenido de la codificación de base.

*Ejemplo* – Con las definiciones tipo NSA.1 de

```
Type1 ::= VisibleString
Type2 ::= [APPLICATION 3] IMPLICIT Type1
Type3 ::= [2] Type2
Type4 ::= [APPLICATION 7] IMPLICIT Type3
Type5 ::= [2] IMPLICIT Type2
```

un valor de

“Jones”

se codifica como sigue:

Para Tipo1:

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

Para Tipo2:

[Application 3]	<i>Longitud</i>	<i>Contenido</i>
43 <sub>16</sub>	05 <sub>16</sub>	4A6F6E6573 <sub>16</sub>

Para Tipo3:

[2]	<i>Longitud</i>	<i>Contenido</i>
A2 <sub>16</sub>	07 <sub>16</sub>	
[Application 3]	<i>Longitud</i>	<i>Contenido</i>
43 <sub>16</sub>	05 <sub>16</sub>	4A6F6E6573 <sub>16</sub>

Para Tipo4:

[Application 7]	<i>Longitud</i>	<i>Contenido</i>
67 <sub>16</sub>	07 <sub>16</sub>	
[Application 3]	<i>Longitud</i>	<i>Contenido</i>
43 <sub>16</sub>	05 <sub>16</sub>	4A6F6E6573 <sub>16</sub>

Para Tipo5:

[2]	<i>Longitud</i>	<i>Contenido</i>
82 <sub>16</sub>	05 <sub>16</sub>	4A6F6E6573 <sub>16</sub>

## 21 Codificación de un valor de tipo ANY (CUALQUIERA)

La codificación de un tipo ANY será la codificación completa especificada en esta Recomendación para el tipo de valor del tipo ANY.

## 22 Codificación de un valor object identifier (identificador de objeto)

22.1 La codificación de un valor de identificador de objeto será primitiva.

22.2 Los octetos de contenido serán una lista (ordenada) de codificaciones de subidentificadores (véanse los § 22.3 y 22.4) 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).

22.3 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.

22.4 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.

22.5 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>

### 23 Codificación de valores del tipo character string (cadena de caracteres)

23.1 El valor de datos consta de una cadena de caracteres del conjunto de caracteres especificado en la definición NSA.1 del tipo.

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

23.3 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 Recomendación X.208. El valor de la cadena de octetos se especifica en los § 23.4 y 23.5.

23.4 Cuando en la Recomendación X.208, 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 el § 23.5 para un tipo CadenaVisible, con el mismo valor de cadena de caracteres.

23.5 La cadena de octetos contendrá los octetos especificados en la Norma ISO 2022 para las codificaciones en un entorno de 8 bits, utilizando la secuencia de escape y las codificaciones de carácter registradas de acuerdo con la Norma ISO 2375.

23.5.1 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 Recomendación X.208.

CUADRO 2/X.209

**Utilización de secuencias de escape**

Tipo	G0 supuesto (Número de registro)	C0 y C1 (Número de registro)	Secuencia(s) de escape supuesta(s) y bloqueo de cambio (cuando sea aplicable)	¿Se permiten secuencias de escape explícitas?
NumericString (Cadena Numérica)	2	Ninguno	ESC 2/8 4/0 LS0	No
PrintableString (Cadena Imprimible)	2	Ninguno	ESC 2/8 4/0 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
VideotexChain (Cadena Videotex)	102	1(C0) 73(C1)	ESC 2/8 7/5 LS0 ESC 2/1 4/0 ESC 2/2 4/1	Sí
VisibleChain (Cadena Visible) (CadenaISO646)	2	Ninguno	ESC 2/8 4/0 LS0	No
IA5String (Cadena AI5)	2	1(C0)	ESC 2/8 4/0 LS0 ESC 2/1 4/0	No
GraphicString (CadenaGráfica)	2	Ninguno	ESC 2/8 4/0 LS0	Sí
GeneralString (CadenaGeneral)	2	1(C0)	ESC 2/8 4/0 LS0 ESC 2/1	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 NSA.1 permitan las secuencias de escape, son posibles varias codificaciones para una determinada cadena de caracteres (véase también el § 5.3).

23.5.2 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 2/X.209, junto con la secuencia de escape supuesta que implican.

23.5.3 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 el § 23.5.1 puede aparecer en cualquier momento, incluso al principio de la codificación. El cuadro 2/X.209 indica los tipos para los cuales se permiten secuencias de escape explícitas.

23.5.4 No se utilizarán anunciadores, a menos que estén permitidos explícitamente por el usuario de NSA.1.

*Nota* - La elección del tipo NSA.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.

*Ejemplo* - Con la definición tipo NSA.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 determinada) como:

VisibleString 3A <sub>16</sub>	Longitud 09 <sub>16</sub>	Contenido
OctetString 04 <sub>16</sub>	Longitud 03 <sub>16</sub>	Contenido 4A6F6E <sub>16</sub>
OctetString 04 <sub>16</sub>	Longitud 02 <sub>16</sub>	Contenido 6573 <sub>16</sub>

o (forma construida, longitud indeterminada) como:

VisibleString 3A <sub>16</sub>	Longitud 80 <sub>16</sub>	Contenido
OctetString 04 <sub>16</sub>	Longitud 03 <sub>16</sub>	Contenido 4A6F6E <sub>16</sub>
OctetString 04 <sub>16</sub>	Longitud 02 <sub>16</sub>	Contenido 6573 <sub>16</sub>
EOC 00 <sub>16</sub>	Longitud 00 <sub>16</sub>	

Este ejemplo ilustra tres de las (muchas) formas posibles que puede utilizar el emisor. Es necesario que los receptores puedan tratar las formas permitidas (véase el § 5.3).

## 24 Codificación de valores de los tipos NSA.1 útiles

En la Recomendación X.208 se da una definición de estos tipos utilizando NSA.1. La codificación será la obtenida al aplicar las reglas especificadas en esta Recomendación a dicha definición de tipo.

## 25 Utilización en la definición de sintaxis de transferencia

25.1 Las reglas de codificación especificadas en esta Recomendación pueden ser referenciadas y aplicadas siempre que sea necesario especificar una representación de cadena de octetos inequívoca, no dividida y autolimitada para todos los valores de un solo tipo NSA.1.

*Nota* - Todas estas cadenas de octetos son inequívocas dentro del alcance del tipo NSA.1 único. No serían necesariamente inequívocas si estuvieran mezcladas con codificaciones de un tipo NSA.1 diferente.

25.2 Se asignan los valores del descriptor de objeto e identificación de objeto:

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

y

“Codificación Básica de un solo tipo NSA.1”

para identificar y describir las reglas de codificación especificadas en esta Recomendación.

25.3 Cuando una especificación de aplicación define una sintaxis abstracta como un conjunto de valores de datos de presentación, cada uno de los cuales es un valor de algún tipo específicamente denominado en NSA.1, normalmente (pero no necesariamente) un tipo elegido, puede utilizarse el valor del identificador de objeto especificado en el § 25.2 con el nombre de la sintaxis abstracta para identificar la sintaxis de transferencia resultante de la aplicación de las reglas de codificación especificadas en esta Recomendación al tipo específicamente denominado en NSA.1 utilizado en la definición de la sintaxis abstracta.

*Nota* - En especial, esta identificación de las reglas de codificación puede aparecer en el campo de "nombre de la sintaxis de transferencia" del protocolo de presentación (Recomendación X.226).

25.4 El nombre especificado en el § 25.2 no se utilizará con un nombre de sintaxis abstracta para identificar una sintaxis de transferencia si no se satisfacen las condiciones del § 25.3 para la definición de la sintaxis abstracta.

## APÉNDICE I

(a la Recomendación X.209)

### Ejemplos de codificaciones

Este apéndice ilustra las reglas de codificación básicas especificadas en esta Recomendación, mostrando la representación en octetos de un registro de personal (ficticio) que se define mediante NSA.1.

#### I.1 Descripción NSA.1 de la estructura del registro

La estructura del registro de personal ficticio se describe formalmente a continuación utilizando la notación NSA.1 especificada en la Recomendación X.208 para la definición de tipos.

```
PersonnelRecord ::= [APPLICATION 0] IMPLICIT SET
{
    Name,
    title           [0] VisibleString,
    number          EmployeeNumber,
    dataOfHire      [1] Date,
    nameOfSpouse    [2] Name,
    children        [3] IMPLICIT
                   SEQUENCE OF ChildInformation
                   DEFAULT {} }
```

```
ChildInformation ::= SET
{
    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
```

#### I.2 Descripción NSA.1 de un valor del registro

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

```
{
    { 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" }}} }
```

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

A continuación se facilita la representación en octetos del valor del registro indicado anteriormente (después de aplicar las reglas de codificación básica definidas en esta Recomendación). 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.

Personnel  
Record  
60

Length  
8185

Contents

Name  
61      Length  
10      Contents

Visible-String  
1A      Length      Contents  
04      "John"  
Visible-String  
1A      Length      Contents  
01      "P"  
Visible-String  
1A      Length      Contents  
05      "Smith"

Title  
A0      Length  
0A      Contents

Visible-String  
1A      Length      Contents  
08      "Director"

Employee  
Number  
42      Length  
01      Contents  
33

Date of  
Hire  
A1      Length  
0A      Contents

Date  
43      Length      Contents  
08      "19710917"

Name of  
Spouse  
A2      Length  
12      Contents

Name  
61      Length  
10      Contents

Visible-String  
1A      Length      Contents  
04      "Mary"

Visible-String  
1A      Length      Contents  
01      "T"

Visible-String  
1A      Length      Contents  
05      "Smith"

[3]  
A3      Length  
42      Contents

Set  
31      Length  
1F      Contents

Name  
61      Length  
11      Contents

Visible-String  
1A      Length      Contents  
05      "Ralph"

Visible-String  
1A      Length      Contents  
01      "T"

Visible-String  
1A      Length      Contents  
05      "Smith"

Date of  
Birth  
A0      Length  
0A      Contents

Date  
43      Length      Contents  
08      "19571111"

Set  
31      Length  
1F      Contents

Name  
61      Length  
11      Contents

Visible-String  
16      Length      Contents  
05      "Susan"

Visible-String  
16      Length      Contents  
01      "B"

Visible-String  
1      Length      Contents  
05      "Jones"

Date of  
Birth  
A0      Length  
0A      Contents

Date  
43      Length      Contents  
08      "19590717"

## APÉNDICE II

(a al Recomendación X.209)

### Asignación de valores del identificador de objeto

En esta Recomendación se asignan los siguientes valores:

Punto	Valor del identificador de objeto Valor del descriptor de objeto
25.2	{joint-iso-ccitt asn1 (1) basic-encoding (1)} "Codificación Básica de un solo NSA.1"

## APÉNDICE III

(a la Recomendación X.209)

### Ilustración de la codificación de un valor real

III.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 NSA.1 real. Este apéndice 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 III-1/X.209.

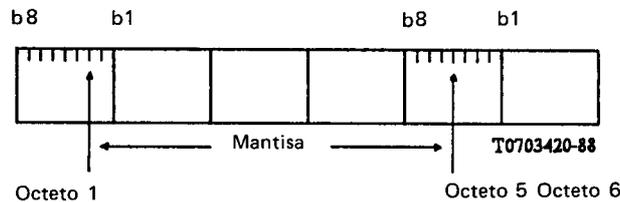


FIGURA III-1/X.209

Representación ficticia en soporte físico

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

III.2 Los octetos de contenido que necesitan generarse para enviar un valor distinto de cero (como se especifica en el texto de esta Recomendación) 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, supongamos que la base es 16), ff es el valor F fijo calculado como se describe en el § III.3, ee es la longitud fija del valor exponente calculada como se describe en el § III.4. (Este apéndice no trata el caso en que E tiene que sobrepasar de tres octetos.)

III.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 único 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, podemos desplazar en 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 tiene así los parámetros cruciales.

$$F = 3 \text{ (de modo que ff es 11)}$$

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

III.4 Se calcula ahora la longitud necesaria del exponente obteniendo el número máximo de octetos necesarios para representar los valores

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

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

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 el § III.3. Supóngase que esto da una longitud de 3 octetos. Entonces ee es 10. Supóngase también que el exceso es cero.

III.5 El algoritmo de transmisión es ahora:

- a) examinar si el valor es cero, y si es así, transmitir una longitud NSA.1 cero (no hay octetos de contenido), y terminar el algoritmo;
- b) examinar y recordar el signo de la mantisa, y tomar la opuesta de mantisa si fuera negativa;
- c) transmitir una longitud NSA.1 de 9, entonces

11101110 si es negativa

o

10101110 si es positiva

- d) producir y transmitir el exponente de tres octetos con el valor

$$E - 9$$

- e) hacer 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.

III.6 El algoritmo receptor tiene que estar preparado para tratar cualquier formato NSA.1, pero aquí puede utilizarse 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 "ValorRealEspecial" de acuerdo con la semántica de aplicación (probablemente fijando el número mayor y el número menos 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 aplicadores pueden encontrar optimizaciones 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.

III.7 Los algoritmos arriba mencionados sólo sirven de ilustración. Por supuesto, los aplicadores determinarán sus propias estrategias.